



Environment  
Canada

Environnement  
Canada

C. C. I. W.  
LIBRARY

# **PROGRAM FRONT**

## **Two-Dimensional Simulation of a Moving Intrusion Front in a Thin Horizontal Confined Aquifer**

A. Vandenberg\*

\* The author has formerly published  
under the name Vanden Berg.

**INLAND WATERS DIRECTORATE,  
WATER RESOURCES BRANCH,  
OTTAWA, CANADA, 1975**

**(Résumé en français)**

Contract No. KL 398-5-8027

THORN PRESS LIMITED

# Contents

	Page
ABSTRACT . . . . .	v
RÉSUMÉ . . . . .	vii
A. GENERAL . . . . .	1
1. Title of program . . . . .	1
2. Acronym . . . . .	1
3. Programmer . . . . .	1
4. Use of this writeup . . . . .	1
B. STATEMENT OF THE PHYSICAL PROBLEM . . . . .	3
C. METHOD OF COMPUTER SOLUTION . . . . .	5
1. Numerical analysis . . . . .	5
2. Flow diagram . . . . .	6
D. COMPUTER PROGRAM . . . . .	7
1. Language . . . . .	7
2. Listing . . . . .	7
3. Program options . . . . .	7
4. Breakdown of program . . . . .	7
5. Input data on cards . . . . .	7
(a) Necessary card input . . . . .	7
(b) Data deck instructions . . . . .	9
(c) Sample data deck . . . . .	10
6. Input data on mass storage device . . . . .	11
7. Printed output . . . . .	15
8. Technical information . . . . .	15
9. Preparation of program deck for submission . . . . .	15
E. LIMITATIONS . . . . .	19
F. REFERENCES . . . . .	20
APPENDIX . . . . .	33

## Tables

	Page
1. Card deck of input data . . . . .	12
2. Printed output . . . . .	16

## Illustrations

Figure 1. Typical physical model . . . . .	23
Figure 2. Nonpumping, steady-state piezometric head in the model aquifer . . . . .	24
Figure 3. Piezometric head in the model aquifer showing cones of depression caused by pumping wells . . . . .	25
Figure 4. Regular grid superimposed on the model aquifer . . . . .	26
Figure 5. Grid element and values of transmissivity used in the calculation of the transmissivity at a point in the interior of the grid element . . . . .	27
Figure 6. Flow diagram of the procedure . . . . .	28
Figure 7. Interface positions for the example at various times . . . . .	29

## Abstract

This report contains the necessary documentation for the application of computer program FRONT (Two-Dimensional Simulation of a Moving Intrusion Front in a Thin Horizontal Confined Aquifer). FRONT uses the time-variant piezometric head, the transmissivity, the thickness and the porosity of the aquifer to calculate the seepage velocity at any point of the aquifer at any time. By following the motion of a number of points on the intrusion front, the growth of the zone of encroachment can be followed through time. The piezometric head and the transmissivity are defined at the nodes of a rectangular grid superimposed on the aquifer; the thickness and porosity are assumed constant over the aquifer. Furthermore, the piezometric head is assumed to be input to the program as a time series in the form in which it is output on mass storage by program SOPH.

The preparation of the input card deck is described in detail and further clarified by an example. A listing of the FORTRAN program is given in the Appendix.

## Résumé

Le présent rapport renferme tous les renseignements permettant d'utiliser le programme d'informatique FRONT (pour la simulation bi-dimensionnelle d'un front d'intrusion se déplaçant dans une nappe aquifère mince, horizontale et captive). Dans ce programme, la charge piézométrique, la conductibilité, l'épaisseur et la porosité de la nappe servent à calculer la vitesse d'infiltration, en tout point et à tout moment. On peut connaître dans le temps la croissance de la zone d'envahissement en suivant le mouvement d'un certain nombre de points situés sur le front d'intrusion. La charge piézométrique et la conductibilité sont définies aux points d'intersection d'une grille rectangulaire superposée à la nappe; on suppose que l'épaisseur et la porosité sont constantes dans l'ensemble de cette dernière. On suppose, de plus, que la charge piézométrique est introduite dans le programme sous forme de série chronologique, sous laquelle elle est extraite d'une mémoire de masse par le programme SOPH.

On explique en détail comment préparer les cartes d'entrée avec exemple à l'appui. En annexe, on trouve l'énoncé du programme en FORTRAN.

# PROGRAM FRONT—Two-Dimensional Simulation of a Moving Intrusion Front in a Thin Horizontal Confined Aquifer

A. Vandenberg

## A. GENERAL

1. Title of Program: Moving intrusion front in a thin horizontal confined aquifer.
2. Acronym: FRONT
3. Programmer: A. Vandenberg  
Water Resources Branch  
Environment Canada  
Ottawa, Ontario
4. Use of This Writeup

This program writeup is designed primarily for the non-programming scientist, who is familiar with the scientific aspects of the problem, but not with computer methods. The following sections should provide sufficient information for the successful application of the program with minimum help from programming staff.

- B. Statement of the physical problem.
- C. 2. Flow diagram presentation of the method of solution.
- D. 3. Program options.
- D. 5. Input data on cards.
- D. 6. Input data on mass storage device.
- D. 7. Printed output.
- D. 9. Preparation of program deck for submission.
- E. Limitations of program.
- Appendix. Explanation of FORMAT terminology.

For those who are not completely familiar with the physical problem, suitable references are provided at the conclusion of the report.

In addition, a description of the numerical analysis (where applicable), a program listing, and a section containing technical information are provided for the use of programmers or programming scientists.

## B. STATEMENT OF THE PHYSICAL PROBLEM

The objective of this program is to determine the position of a moving interface in a thin horizontal aquifer, as for example the salt-water fresh-water interface in a coastal aquifer. The term "thin aquifer" is used here, since vertical variations within the aquifer and in the chemical composition of the groundwater are ignored. Furthermore, no mixing between the waters on opposite sides of the interface is allowed, that is, the movement is treated as horizontal piston flow. The aquifer is confined and isotropic, but may be nonhomogeneous and of arbitrary configuration and may be subjected to pumping and/or recharging through a number of wells. Figure 1 shows a typical physical model of a peninsular aquifer; the constant head boundary,  $H=0$ , represents the coastline, and the constant head boundary,  $H=2$  ft (0.61 m), represents a fresh-water lake which is hydraulically connected to the aquifer. Recharge from the lake keeps the piezometric head along its circumference at a constant elevation of 2 ft (0.61 m) above sea level, and if no water is being withdrawn from or recharged to the aquifer by wells, there is a steady flow of fresh groundwater radially outward from the lake toward the coast. Figure 2 shows the steady piezometric head in this case.

The pumping wells, however, create cones of depression in the piezometric surface which may reverse the direction of the gradient of the head and cause sea water to seep into the aquifer (Fig. 3).

Program FRONT is designed to use the output on a mass storage device (disk or tape) from program SOPH (Vanden Berg, 1974a) as its main input, and only the initial location of the interface, the porosity and thickness of the aquifer, and a small number of operational parameters need to be read in from cards.

For the application of program FRONT, a rectangular grid is superimposed on the aquifer (Fig. 4), and local values of transmissivity and piezometric head are specified at the nodes of the grid. Similarly, storativity values for the area inside the grid boundaries occupied by the aquifer are specified by the S-matrix. Elements of this matrix are assigned the value of one if the corresponding node is outside or on the boundary of the aquifer, and may have any other value inside the aquifer.

The use of program FRONT in conjunction with program SOPH means that the parameters specifying the grid, the transmissivity matrix and the S-matrix, as well as the parameters associated with the recharge/discharge points, and the time series of piezometric-head matrices that have all been specified previously in program SOPH, can be automatically passed through the mass storage file; if the program is not used in conjunction with program SOPH, these data must be prepared in the required format by some other program.

The movement of the interface is simulated by following the movement of a finite number of points on the interface, for which the initial locations are given on input. The displacement vectors  $\vec{\Delta s}$  over a small time interval  $\partial t$  are calculated from:

$$\vec{\Delta s} = \vec{V}_s \partial t = -(T/nd) [(\partial H/\partial x)\vec{i} + (\partial H/\partial y)\vec{j}] \partial t$$

where  $\vec{V}_s$  = seepage velocity vector  
 $T$  = transmissivity  
 $n$  = porosity  
 $d$  = thickness of the aquifer  
 $\vec{i}$  = unit vector in the x-direction  
 $\vec{j}$  = unit vector in the y-direction  
 $H$  = piezometric head.

## C. METHOD OF COMPUTER SOLUTION

### 1. Numerical Analysis

Three numerical techniques for the calculation of the head gradient

$$\frac{\partial H}{\partial x} \vec{i} + \frac{\partial H}{\partial y} \vec{j}$$

are given by Vanden Berg (1974b); the highest-order approximation uses 13 nodes in the neighbourhood of the point and should not be used near points of discontinuity of the piezometric head; the lowest-order approximation uses only the 4 nodes at the corners of the elementary rectangle in which the point is located and can be used for any location inside the aquifer.

Although in general the higher-order approximations are more accurate, they may under certain circumstances introduce spurious maxima or minima into the piezometric surface. Since the negative gradient in the neighbourhood of a minimum will be directed toward it and become zero at the minimum itself, moving points tend to become trapped in such locations; this problem has been experienced in applications where the aquifer was very inhomogeneous or near irregular aquifer boundaries. In such cases, the lowest-order approximation should be used exclusively; an input parameter is made available for the selection of this option.

The transmissivity at a point inside a grid element is calculated by fitting a polynomial of the form:

$$T = a_{00} + a_{10}\alpha + a_{01}\beta + a_{11}\alpha\beta$$

to the values of T at the 4 corner nodes of the grid element; the definition of  $\alpha$  and  $\beta$  is shown in Figure 5. The coefficients  $a_{ij}$  then become:

$$a_{00} = T_{i,j}$$

$$a_{10} = T_{i+1,j} - T_{i,j}$$

$$a_{01} = T_{i,j+1} - T_{i,j}$$

$$a_{11} = T_{i,j} + T_{i+1,j+1} - T_{i,j+1} - T_{i+1,j}$$

where the subscripts of the T's correspond to the grid coordinates shown in Figure 5.

Vanden Berg (1974b) describes how the time step is adjusted to keep acceptable accuracy, yet not waste computer time, as well as the special measures that must be taken whenever a point moves close to a sink or to a boundary of the aquifer.

## 2. Flow Diagram

Figure 6 shows the simplified flow diagram for the computer calculations. A detailed flow diagram is on file in the Hydrology Research Division of the Inland Waters Directorate.

#### D. COMPUTER PROGRAM

1. Language: FORTRAN IV extended (McCracken, 1965; Control Data Corporation, 1973).
2. Listing: A listing of the main program is included near the end of this report.
3. Program Options: The gradient calculation may be performed by
  - (a) the lowest-order approximation exclusively or
  - (b) any one of the three routines provided; in this case the selection of the highest-order routine that is permissible for the location is entirely automatic.
4. Breakdown of Program: Main Program - FRONT
  - reads all data,
  - carries out all the tests and calculations except the calculation of the space derivatives of piezometric head, and
  - prints all results.

Subroutines DG, DH and DF

- calculate the space derivatives of piezometric head at a point inside an elementary rectangle from the values of piezometric head at nearby nodes; DG, DH and DF use 4, 9 and 13 nodes, respectively.

5. Input Data on Cards
  - (a) Necessary Card Input

The necessary input data, to be punched on cards, are listed here in the order in which they must appear in the input stream.

- DT2        -    Initial value of the time step (unit must be consistent with the unit for the transmissivity read from the mass storage file (FILE 1) as discussed in Section D.6). DT2 provides

an estimate of the initial time step only, on which the calculation of the first set of displacements will be based. These displacements, however, will immediately be compared with the preset limits (see discussion of DI and BI later) and a new value for DT2 may be calculated. Each time there is a disturbance of the piezometric surface, i. e., a pump starting or stopping, the original value of DT2 is again taken as the first estimate of the optimum time step.

- |      |   |
|------|---|
| DI   | <p>- (1) Dimensionless parameter imposing maximum value on the displacement per time step. If for any of the moving points the displacement in the x-direction is greater than <math>DI \times DX</math>, or the displacement in the y-direction is greater than <math>DI \times DY</math>, the calculated displacements for that time step are rejected and a smaller value for the time step is calculated; DX and DY are the sides of the elementary rectangle and are input from FILE 1 (Section D.6).</p> <p>(2) Determines when a moving point is so close to a discharging node that it can be taken out of circulation.</p> |
| AMIN | <p>- Dimensionless parameter determining how frequently a set of locations is printed; for printing to take place, the displacement of at least one of the points, in the x-direction or y-direction, accumulated since the last printout must be at least <math>AMIN \times DX</math> (in the x-direction) or <math>AMIN \times DY</math> (in the y-direction).</p>  |
| TMAX | <p>- Maximum time of the simulation (same unit as DT2).</p>   |
| RHO  | <p>- Porosity of the aquifer (dimensionless).</p>   |

- THICK - Thickness of the aquifer [ unit must be consistent with the unit for the nodal spacings DX and DY read from FILE 1 (Section D.6)].
- BI - Dimensionless parameter specifying minimum value of the displacement per time step. If for all the moving points the displacement in the x-direction is less than  $BI \times DX$ , and the displacement in the y-direction is less than  $BI \times DY$ , the duration of the next time step will be increased accordingly.
- TSTEP - Prevents output if the time elapsed since last printout of the locations of the moving points is less than TSTEP; TSTEP takes precedence over AMIN in suppressing output (same unit as DT2).
- NSW - Number of moving points ( $NSW \leq 100$ ) (dimensionless, integer).
- LET - If  $LET = 0$ , the higher-order derivative subroutines will be bypassed in the calculation of the gradient.
- SWX (100) - Initial x-coordinate of a moving point (dimensionless, relating to the grid-node numbering).
- SWY (100) - Initial y-coordinate of a moving point (dimensionless, relating to the grid-node numbering).

(b) Data Deck Instructions

The following table specifies the setup of data cards for use with this program. In the Appendix, the FORMAT terminology used in the table is explained.

GROUP	CARD	FORMAT	DATA
A	1	(3F10.3, F10.0, 3F10.3, F5.0, I3, I2)	DT2, DI, AMIN, TMAX, RHO, THICK, BI, TSTEP, NSW, LET
B	1	(12F6.0)	SWX, SWY, SWX, SWY...,
	2	(12F6.0)	6 pairs of values per
	'	'	card
	'	'	
	'	'	

(c) Sample Data Deck

As an example of the required card input, the example given by Vanden Berg (1974a) for the application of program SOPH is expanded with an application of program FRONT. The example is a simulation of the motion of the salt-water fresh-water interface in the aquifer of Figure 1; Figures 2 and 3 show the piezometric head in the aquifer before and toward the end of pumping, respectively, as calculated in phase 1 and phase 2 of program SOPH; Figure 4 shows the regular grid. The pumping schedule of the three wells follows:

Well number	Location		Pumping rate lgpm (m <sup>3</sup> /min)	Start of operation (min)	End of operation (min)
	x-index	y-index			
1	19	27	-300 (-1.36)	0.	800 000.
2	22	25	-500 (-2.27)	0.	1 000 000.
3	12	22	-800 (-3.64)	0.	1 000 000.

Other pertinent data are:

DT2	-	100 min	THICK	-	100 ft (33 m)
DI	-	0.06	BI	-	0.02
AMIN	-	0.1	TSTEP	-	40 min
TMAX	-	$6 \times 10^6$ min	NSW	-	8
RHO	-	0.1	LET	-	0

Table 1 shows the sample data deck for the example that is worked out. Note that

- (1) TMAX is not restricted by the maximum time specified in the associated run of program SOPH, nor by the maximum time F(I, 5), indicating the longest period of pumping. After the last piezometric head matrix is read from FILE 1, the program assumes that a steady state has been reached, and also that any discharge/recharge point that is still active at that time will remain active for the duration of the simulation.
- (2) The initial positions of the moving points do not have to be on the boundary. These locations have been chosen more in accord with the true location of the physical boundary (Fig. 1) than with the location of the grid boundary.

#### 6. Input Data on Mass Storage Device

Although the data on FILE 1 are normally written by program SOPH and do not require any action from the user, program FRONT can be used in conjunction with any other program that is designed to write the same type of records as program SOPH.

Records read from FILE 1 are unformatted, i.e., in machine code, and are organized in the following manner.

Table 1. Card Deck of Input Data

[illegible]

(1) The first record, containing the data:

- DX - Nodal spacing in x-direction (length).
- DY - Nodal spacing in y-direction (length, same units as DX).
- MX - Number of nodes in the x-direction (dimensionless, integer,  $MX \leq 50$ ).
- MY - Number of nodes in the y-direction (dimensionless, integer  $MY \leq 50$ ).
- NF - Number of source/sink nodes (dimensionless, integer,  $NF \leq 10$ ).
- T(50,50)- Transmissivity matrix, a value for each of the  $MX \times MY$  nodes ( $\text{length}^2/\text{time}$ ; units must be consistent with the length unit for DX and with the time unit for DT2 as described in Section D.5(a) on necessary card input. Note that if FILE 1 is written by program SOPH and the English system option has been selected for SOPH, all of the values written on FILE 1 will be in the foot-minute system.
- S(50,50)- Storativity matrix, must have a value for each of the  $MX \times MY$  nodes; must have the value 1.0 on the boundary of the aquifer and everywhere outside the aquifer; may have any value other than 1.0 inside the aquifer (dimensionless).
- Q(50,50)- Matrix of initial piezometric head, a value for each of the  $MX \times MY$  nodes (length, same unit as DX).
- F(10,5) - Matrix of source/sink parameters,  $NF \times 5$  values. For each source/sink node ( $N=1, 2 \dots NF$ ), the following parameters must be given:

F(N, 1) - x-coordinate of the Nth source/sink node (dimensionless).

F(N, 2) - y-coordinate of the Nth source/sink node (dimensionless).

Since sources/sinks must be located at grid points, only the integral parts of F(N, 1) and F(N, 2) are used, and are interpreted as the node indices of the source or sink.

F(N, 3) - Pumping rate (negative) or recharge rate (positive) (length<sup>3</sup>/time). If the English system option of SOPH is used, F(N, 3) will be converted to ft<sup>3</sup>/min.

F(N, 4) - Time at which source/sink starts.

F(N, 5) - Time at which source/sink stops.

Note that the term F(N, 6), used in program SOPH to store the well radius, is not used in program FRONT and not written on FILE 1.

The READ statement causing read-in of this record appears on lines 46 and 47 of the program listing (Appendix).

(2) Several identical records each of which contains the data

TIME2 - The total elapsed time since the beginning of the simulation for which the piezometric head matrix is given on the record.

DT - The time elapsed since the time recorded on the previous record.

QQ  
(50, 50) - The MXxMY matrix of the piezometric head at time TIME2 (length).

The READ statement causing repeated read-in of this type of record appears on line 69 of the program listing (Appendix).

## 7. Printed Output

Table 2 shows part of the printed output. Lines 1 and 2 echo some of the data read from FILE 1; Lines 3 and 4 echo the data read from Card 1, Group A [Section D.5(b)]. Then follow the boundary locations as pairs of coordinates, headed each time by the number of the time step and the total elapsed time. Note that values are not printed for all the time steps, as can be seen from the sequence of time-step numbers.

When the end of file on FILE 1 is sensed, the message "STEADY STATE HAS BEEN ASSUMED" is printed and calculations continue until the total elapsed time equals TMAX.

Some of the interface positions are plotted in Figure 7.

## 8. Technical Information

- (1) Machine and Computer Centre: The program listing shows the FORTRAN IV extended language in use at the Department of Energy, Mines and Resources, Ottawa. The program has been run successfully there on a CDC 6400.
- (2) Storage requirements: FRONT occupies 75 000 (octal) or 31 232 words on the CDC 6400.
- (3) Computer time: The central processor time for compilation and execution of the sample program was 15.6 seconds.

## 9. Preparation of Program Deck for Submission

There are three components to a complete deck which is ready for submission for execution: control cards, program deck and data deck.

This writeup has described the preparation of the data deck for the problem at hand. The data deck will of course vary with each run. The program listing, which can be punched on cards to provide

Table 2. Printed Output

```

DX, DY, MX, MY, NF, AS READ FROM TAPE1
.660000E+03      .660000E+03      34      44      7
012, 014, MIN, TMAX, QHO, THICK, BI, ISTER, NSW, LET
.10000E+03      .60000E-01      .10000E+00      .60000E+07      .10000E+00      .10000E+03      .20000E-01      .40000E+02      8      0
1. BOUNDARY LOCATIONS AT TIME = 0.
5.000 24.700 6.090 25.000 6.500 26.000 7.200 26.600 9.000 26.600 10.000 27.000 11.300 27.300 12.000 27.900
1. BOUNDARY LOCATIONS AT TIME = .52958E+03
5.000 24.000 6.000 25.000 6.500 26.000 7.200 26.600 9.000 26.600 10.000 27.000 11.300 27.300 12.000 27.900
2. BOUNDARY LOCATIONS AT TIME = .53810E+03
5.000 24.700 6.000 25.000 6.500 26.000 7.200 26.600 9.000 26.600 10.000 27.000 11.300 27.300 12.000 27.900
4. BOUNDARY LOCATIONS AT TIME = .88686E+05
5.004 24.000 6.043 24.994 6.523 25.990 7.210 26.572 9.017 26.521 10.081 26.883 11.325 27.208 12.165 27.788
5. BOUNDARY LOCATIONS AT TIME = .17982E+06
5.008 24.000 6.088 24.967 6.548 25.979 7.221 26.542 9.038 26.439 10.162 26.763 11.354 27.113 12.208 27.677
6. BOUNDARY LOCATIONS AT TIME = .27174E+06
5.013 24.000 6.132 24.949 6.573 25.967 7.233 26.512 9.062 26.356 10.243 26.643 11.386 27.015 12.305 27.570
6. BOUNDARY LOCATIONS AT TIME = .36446E+06
5.017 24.000 6.176 24.932 6.598 25.954 7.246 26.481 9.090 26.270 10.323 26.523 11.422 26.904 12.397 27.468
7. BOUNDARY LOCATIONS AT TIME = .45799E+06
5.022 24.000 6.220 24.914 6.624 25.941 7.259 26.450 9.122 26.193 10.402 26.404 11.459 26.789 12.485 27.369
7. BOUNDARY LOCATIONS AT TIME = .55234E+06
5.026 24.000 6.264 24.896 6.651 25.928 7.274 26.418 9.157 26.093 10.480 26.284 11.497 26.673 12.569 27.275
8. BOUNDARY LOCATIONS AT TIME = .64750E+06
5.030 24.000 6.307 24.878 6.677 25.913 7.289 26.386 9.197 26.002 10.558 26.164 11.537 26.556 12.648 27.184
9. BOUNDARY LOCATIONS AT TIME = .74348E+06
5.035 23.999 6.351 24.860 6.705 25.898 7.306 26.353 9.241 25.918 10.634 26.044 11.577 26.437 12.723 27.096
STEADY STATE HAS BEEN ASSUMED
129. BOUNDARY LOCATIONS AT TIME = .82465E+06
5.039 23.999 6.395 24.845 6.727 25.886 7.320 26.325 9.279 25.850 10.691 25.944 11.608 26.341 12.768 27.031
155. BOUNDARY LOCATIONS AT TIME = .90785E+06
5.043 23.999 6.446 24.830 6.747 25.874 7.334 26.300 9.317 25.784 10.735 25.842 11.630 26.257 12.780 26.973
163. BOUNDARY LOCATIONS AT TIME = .99105E+06
5.047 23.999 6.447 24.815 6.708 25.862 7.349 26.275 9.356 25.717 10.781 25.740 11.651 26.172 12.790 26.907

```

Table 2. Printed Output (cont.)

5.189	23.981	7.410	24.331	7.749	25.154	8.276	25.201	12.000	22.000	12.000	22.000	12.007	22.641
1330 .BOUNDARY LOCATIONS AT TIME = .38945E+07													
5.189	23.981	7.411	24.331	7.751	25.153	8.278	25.199	12.000	22.000	12.000	22.000	12.009	22.534
1332 .BOUNDARY LOCATIONS AT TIME = .38981E+07													
5.189	23.981	7.412	24.330	7.752	25.152	8.280	25.197	12.000	22.000	12.000	22.000	12.010	22.427
1334 .BOUNDARY LOCATIONS AT TIME = .39019E+07													
5.189	23.981	7.413	24.329	7.754	25.150	8.282	25.195	12.000	22.000	12.000	22.000	12.011	22.320
1336 .BOUNDARY LOCATIONS AT TIME = .39054E+07													
5.189	23.981	7.414	24.329	7.755	25.149	8.284	25.194	12.000	22.000	12.000	22.000	12.013	22.214
1338 .BOUNDARY LOCATIONS AT TIME = .39092E+07													
5.189	23.981	7.415	24.328	7.757	25.148	8.286	25.192	12.000	22.000	12.000	22.000	12.014	22.108
1340 .BOUNDARY LOCATIONS AT TIME = .39126E+07													
5.190	23.981	7.417	24.328	7.759	25.146	8.287	25.190	12.000	22.000	12.000	22.000	12.000	22.000
1441 .BOUNDARY LOCATIONS AT TIME = .40953E+07													
5.199	23.979	7.471	24.298	7.839	25.040	8.388	25.098	12.000	22.000	12.000	22.000	12.000	22.000
1538 .BOUNDARY LOCATIONS AT TIME = .42708E+07													
5.207	23.977	7.571	24.271	7.919	25.013	8.499	25.004	12.000	22.000	12.000	22.000	12.000	22.000
1624 .BOUNDARY LOCATIONS AT TIME = .44264E+07													
5.215	23.976	7.564	24.246	7.990	24.947	8.581	24.903	12.000	22.000	12.000	22.000	12.000	22.000
1711 .BOUNDARY LOCATIONS AT TIME = .45839E+07													
5.222	23.974	7.606	24.222	8.079	24.878	8.572	24.803	12.000	22.000	12.000	22.000	12.000	22.000
1801 .BOUNDARY LOCATIONS AT TIME = .47457E+07													
5.230	23.972	7.648	24.198	8.173	24.866	8.764	24.703	12.000	22.000	12.000	22.000	12.000	22.000
1895 .BOUNDARY LOCATIONS AT TIME = .49186E+07													
5.239	23.970	7.690	24.172	8.271	24.729	8.858	24.602	12.000	22.000	12.000	22.000	12.000	22.000
1995 .BOUNDARY LOCATIONS AT TIME = .50935E+07													
5.248	23.968	7.733	24.146	8.371	24.648	8.952	24.502	12.000	22.000	12.000	22.000	12.000	22.000
2094 .BOUNDARY LOCATIONS AT TIME = .52788E+07													
5.257	23.965	7.774	24.122	8.466	24.569	9.053	24.410	12.000	22.000	12.000	22.000	12.000	22.000
2181 .BOUNDARY LOCATIONS AT TIME = .54362E+07													
5.264	23.963	7.809	24.100	8.547	24.501	9.153	24.330	12.000	22.000	12.000	22.000	12.000	22.000
2275 .BOUNDARY LOCATIONS AT TIME = .56043E+07													
5.273	23.961	7.843	24.078	8.630	24.429	9.254	24.249	12.000	22.000	12.000	22.000	12.000	22.000

the program deck, is also included. The control cards vary with the computer centre, and it is recommended that the non-programming scientist obtain the help of the computer centre staff in the arrangement of his deck and the insertion of the correct control cards. The staff programmers can also explain how to store an object program on disk if the program is to be used on a production basis.

#### E. LIMITATIONS

- (1) The program, as listed here, is limited to a 50 x 50 nodal array and 10 source/sink terms.
- (2) The mathematical development on which this program is based is limited to thin, horizontal, isotropic and confined aquifers.

## F. REFERENCES

Control Data Corporation. 1973. FORTRAN Extended Reference Manual,  
Publ. No. 60305600.

McCracken, D.D. 1965. A Guide to FORTRAN IV Programming. New  
York: John Wiley and Sons Inc.

Vanden Berg, A. 1974a. PROGRAM SOPH - Simulation of Time-Variant  
Piezometric Surface in a Confined Aquifer Subjected to Pumping.  
Inland Waters Directorate, Department of the Environment.

Vanden Berg, A. 1974b. A Digital Simulation of Horizontal Salt Water  
Encroachment Induced by Fresh-Water Pumping. Inland Waters  
Directorate, Scientific Series No. 41.

**Figures 1 to 7**

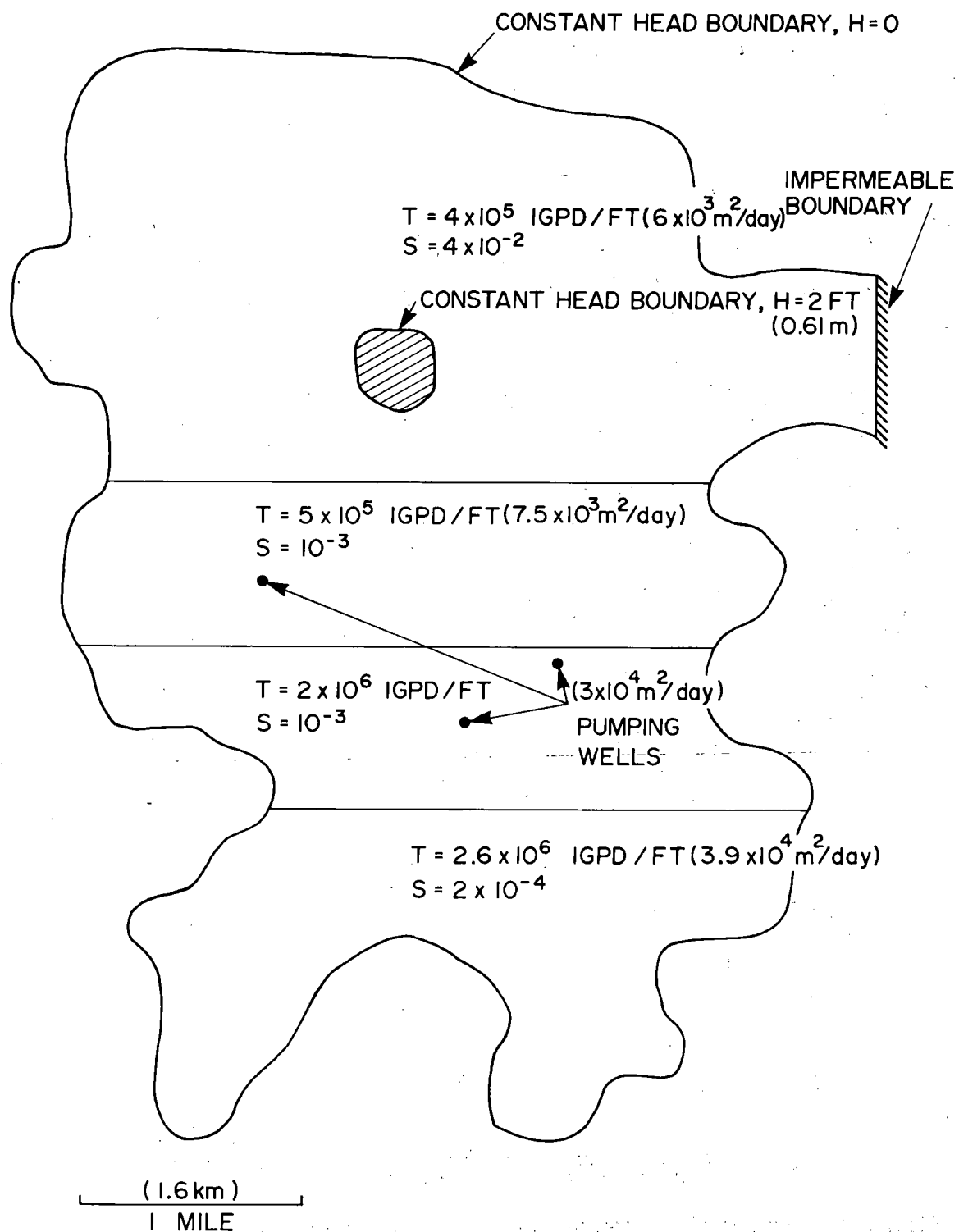


Figure 1. Typical physical model.

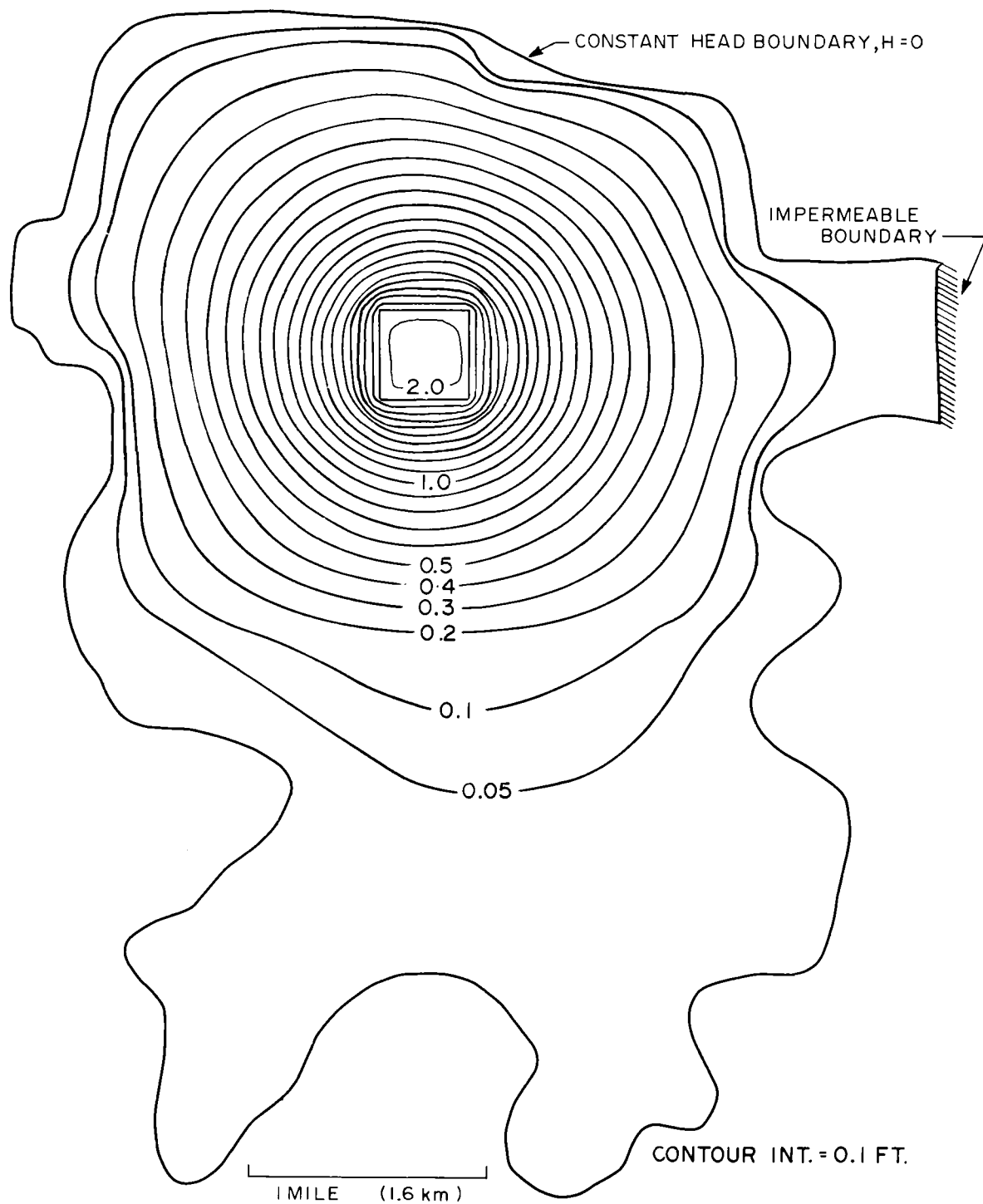


Figure 2. Nonpumping, steady-state piezometric head in the model aquifer.

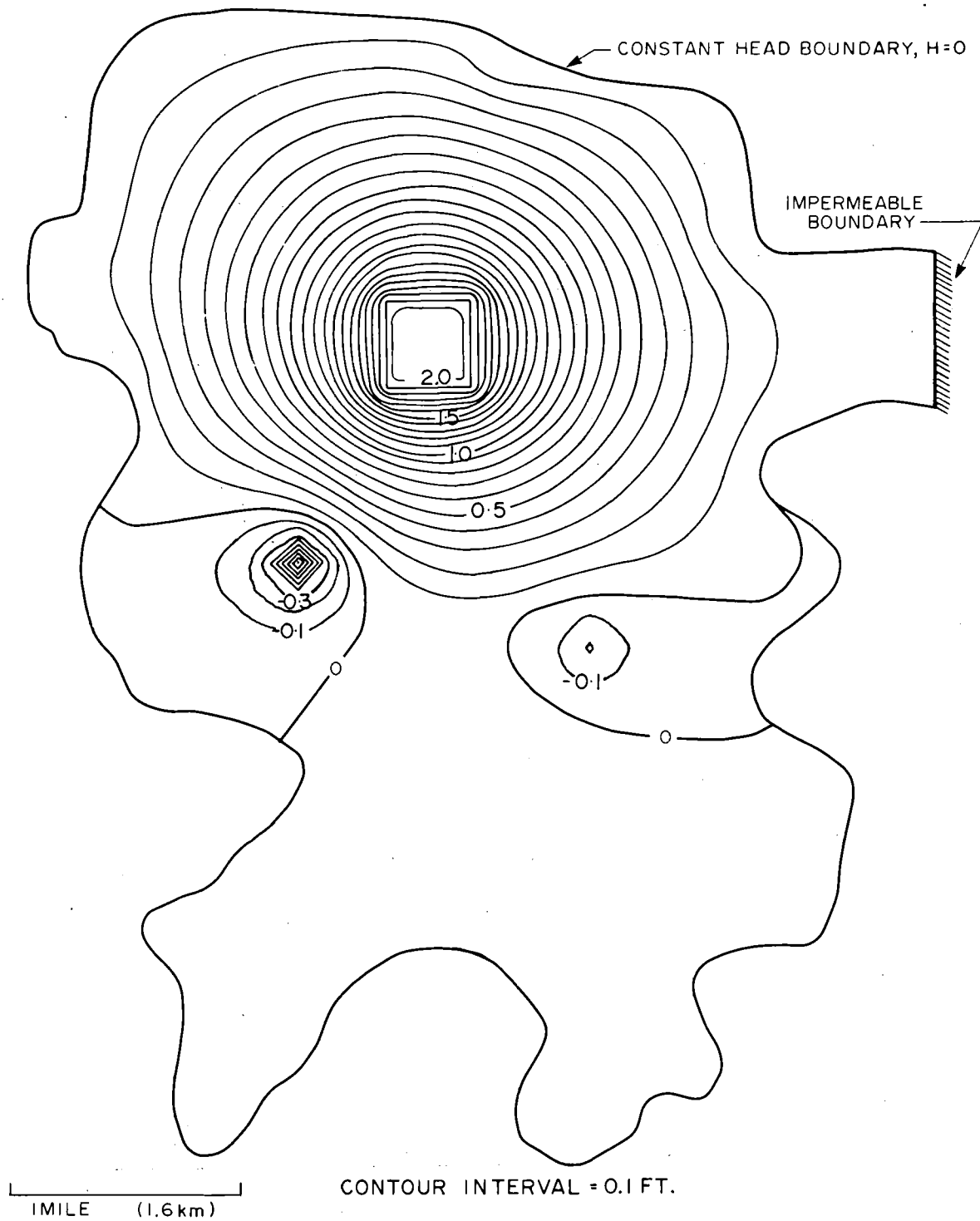


Figure 3. Piezometric head in the model aquifer showing cones of depression caused by pumping wells.

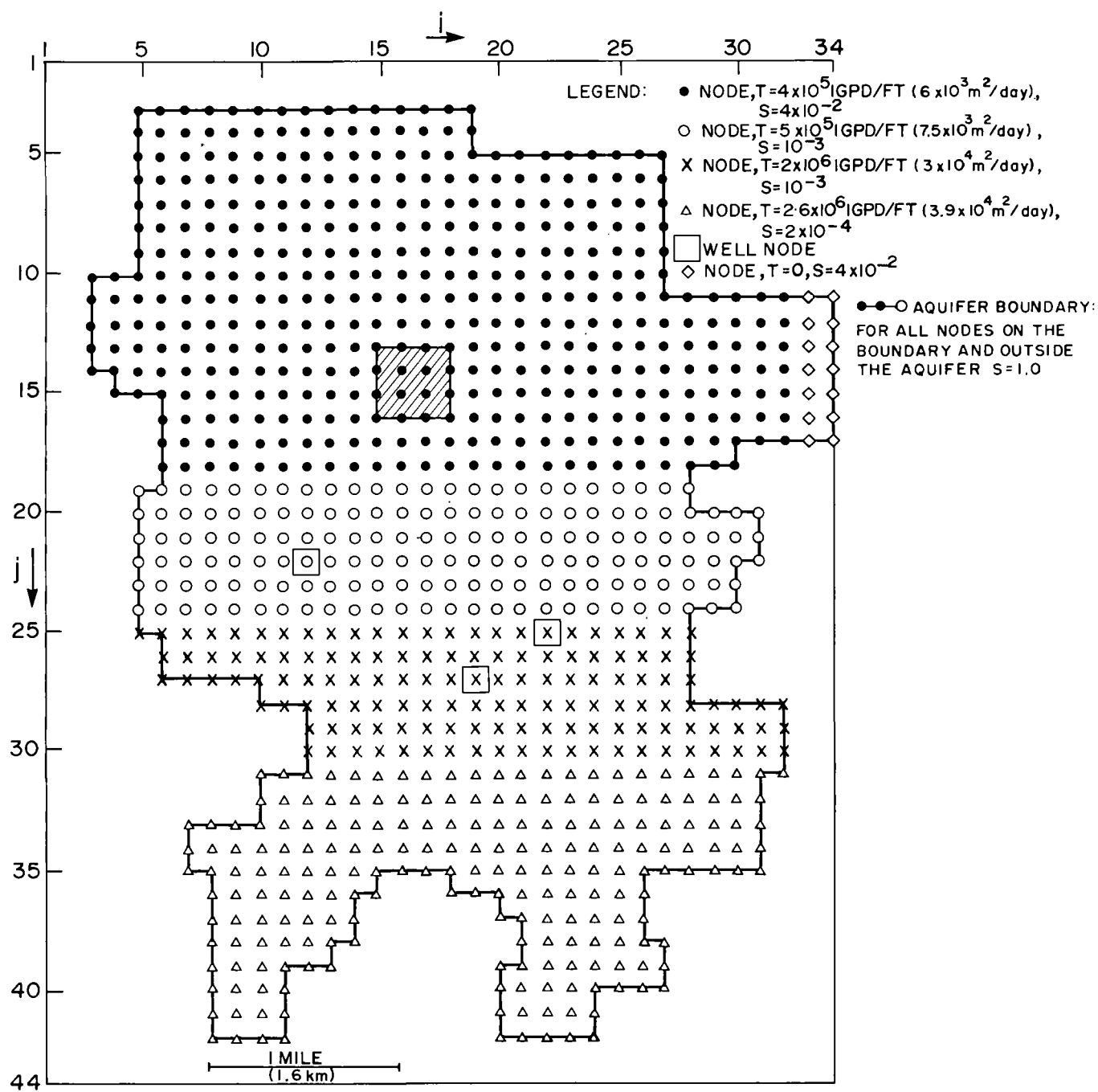


Figure 4. Regular grid superimposed on the model aquifer.

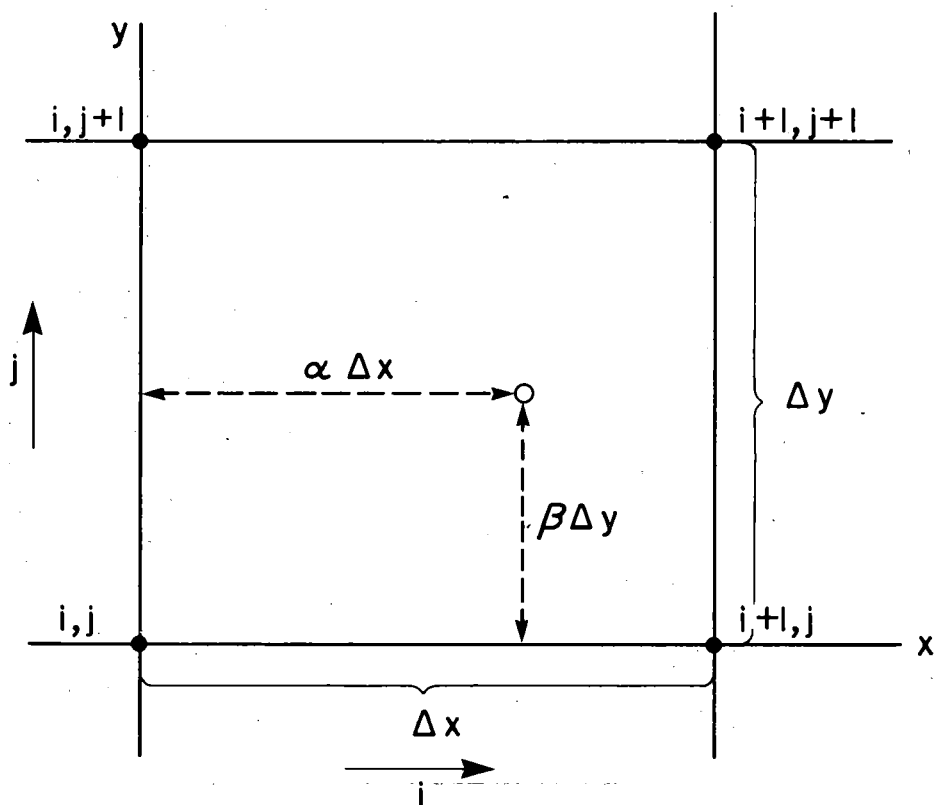


Figure 5. Grid element and values of transmissivity used in the calculation of the transmissivity at a point in the interior of the grid element.

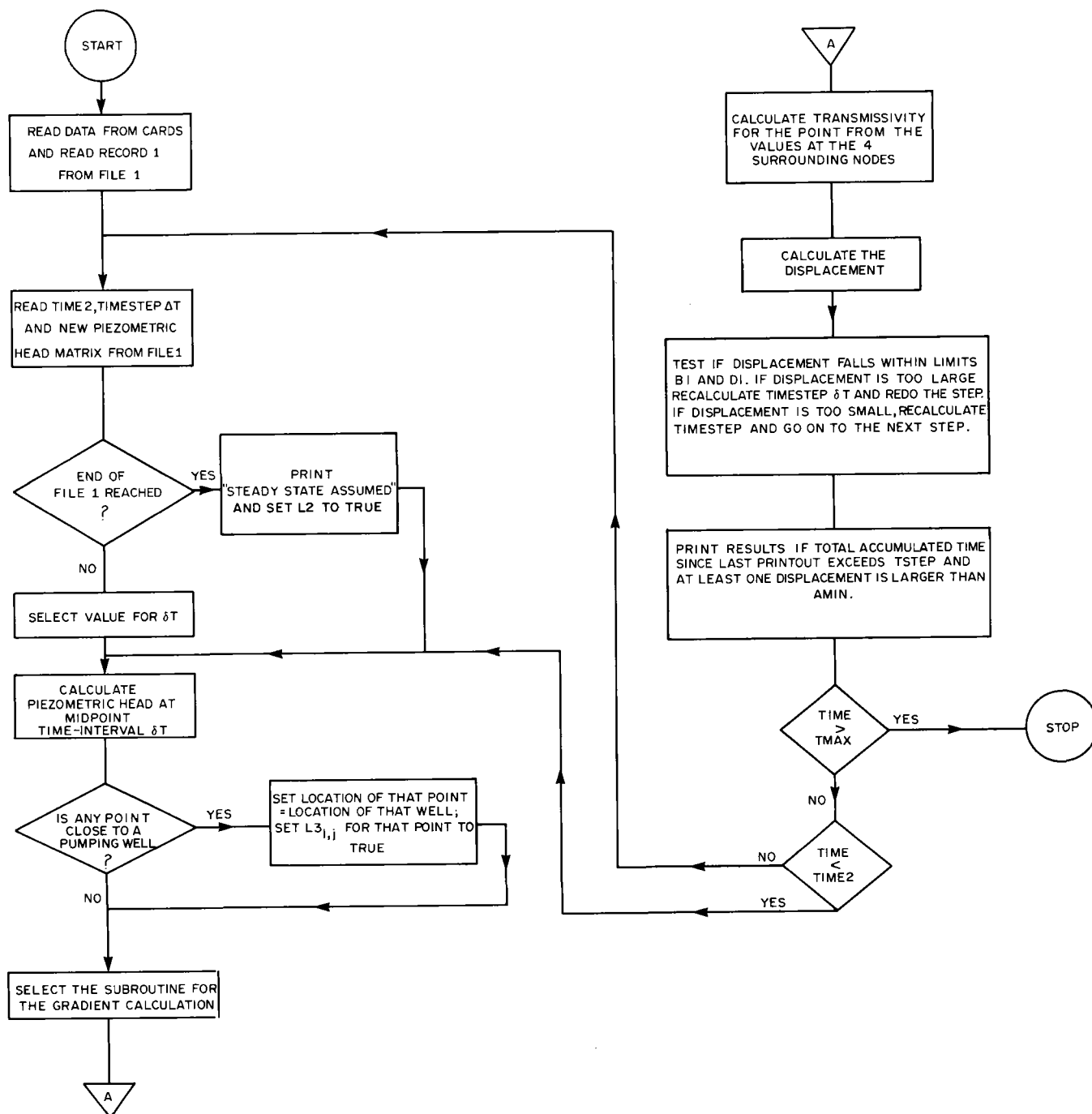


Figure 6. Flow diagram of the procedure.

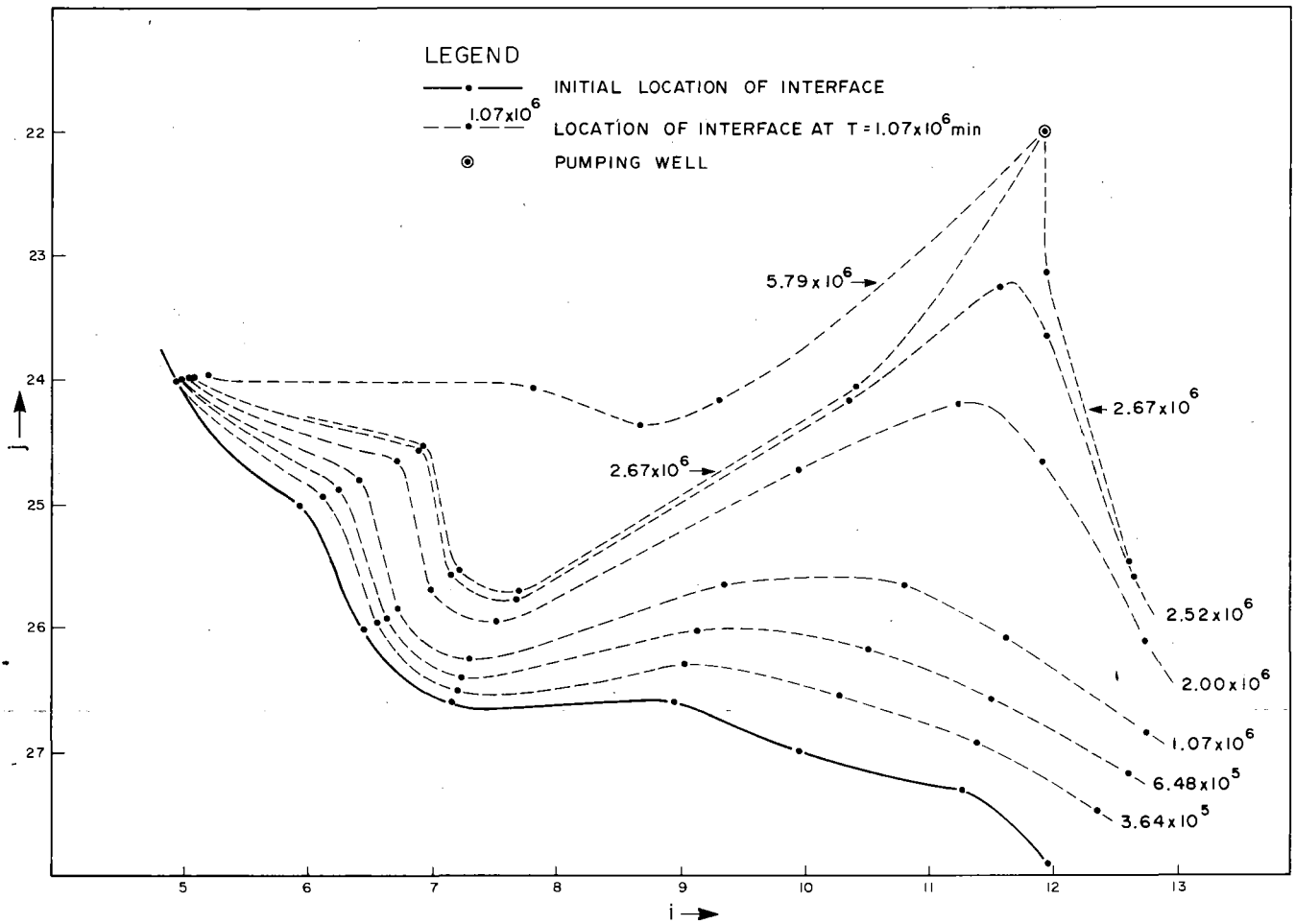


Figure 7. Interface positions for the example at various times.

## Appendix

## EXPLANATION OF FORMAT TERMINOLOGY

- I Format        for integer values, no decimal point;
- F Format        for real values, with decimal point.
- I3, I2        means that two integer values are to be placed, the first integer occupying a field of three columns, the second integer occupying a field of two columns. An integer value can be placed in each field using the extreme right-hand columns. For example, since NSW=8, an 8 is placed in column 78 of Card 1, Group A [table, Section D.5(c)].
- 3F10.3        means that three real values can be placed in three fields of ten columns each. A decimal point may be placed in any of the ten columns, but if no decimal point is supplied, one will be assumed before the third digit from the end, i.e., between columns 7 and 8 of that field. In general, an F-format specification is of the form  $n\text{F}m.d$ , where
- $n$  = number of fields with the same format,
  - $m$  = number of columns in each field, and
  - $d$  = number of digits to the right of the decimal point within the field. If a decimal point is provided in the field, it overrides  $d$ . The complete FORMAT list for Card 1, Group A is (3F10.3, F10.0, 3F10.3, F5.0, I3, I2) and indicates the following position of the data.

Column	1-10	DT2	}	If not punched, decimal point is assumed between the seventh and eighth columns of each field.
	11-20	DI		
	21-30	AMIN		
	31-40	TMAX		
				If not punched, decimal point is assumed after the tenth column of the field.
	41-50	RHO	}	If not punched, decimal point is assumed between the seventh and eighth column of each field.
	51-60	THICK		
	61-70	BI		
	71-75	TSTEP		If not punched, decimal point is assumed after the fifth column of the field.
	76-78	NSW		Integer.
	79-80	LET		Integer.

```

PROGRAM FRONT(INPUT,OUTPUT,TAPE1,TAPE5=INPUT,TAPE6=OUTPUT)
***
*** THIS PROGRAM WAS WRITTEN BY ***
*** A.VANDENBERG ***
5 *** INLAND WATERS DIRECTORATE ***
*** ENVIRONMENT CANADA ***
*** PROGRAM TO CALCULATE MOVEMENT OF AN INTRUSION FRONT IN A ***
*** 2-DIMENSIONAL AQUIFER.THE FRONT IS REPRESENTED BY A SET OF ***
*** MOVING POINTS WHOSE INITIAL POSITION IS SPECIFIED.THE POINTS ***
10 *** MOVE UNDER INFLUENCE OF THE PIEZOMETRIC HEAD GRADIENT,WHICH IS ***
*** ASSUMED TO BE KNOWN AT ALL TIME AT THE NODES OF A REGULAR GRID.A **
*** TIME SERIES OF PIEZOMETRIC HEAD VALUES AT THE GRID NODES IS READ **
*** FROM TAPE1,FOR EXAMPLE AS WRITTEN BY PROGRAM SOPH.SOME CONTROL ***
*** DATA ARE ALSO WRITTEN BY SOPH,WHILE SOME OTHER PARAMETERS MUST ***
15 *** BE SUPPLIED ON PUNCHED CARDS.FOR FURTHER INFO SEE THE WRITEUP. ***
*** THIS PROGRAM WAS ORIGINALLY WRITTEN IN MARCH 1971. ****
*** LAST UPDATE--DECEM,1974. ***
*** COMMENTS ADDED MARCH,1975.
100 FORMAT(3F10.3,F10.0,3F10.3,F5.0,I3,I2)
20 101 FORMAT("0",I8," .BOUNDARY LOCATIONS AT TIME =",E12.5,/(16F8.3))
102 FORMAT("0 STEADY STATE HAS BEEN ASSUMED"/)
103 FORMAT(12F6.0)
104 FORMAT(2E15.6,3I5)
888 FORMAT(8E14.5,2I4)
25 2000 FORMAT(*10X,DY,MX,MY,NF,AS READ FROM TAPE1*)
2001 FORMAT(*3 DT2,DI,AMIN,TMAX,RHO,THICK,BI,TSTEP,NSW,LET*)
DIMENSION T(50,50),S(50,50),QQ(50,50),Q(50,50),F(10,5),SWX(100),
1SWY(100),DHX(100),DHY(100),SPX(100),SPY(100),SWPX(100),SWPY(100)
DIMENSION QRES(50,50)
30 COMMON DX,DY,QQQ(50,50),JX,JY
LOGICAL L2,L3(100)
DATA (L3(I),I=1,100)/100*(.FALSE.)/,L2/.FALSE./
C
C LOGICAL SWITCH L2 IS SET TO TRUE WHEN STEADY STATE IS REACHED,-END
35 C OF DATASET 1- IN ORDER TO BYPASS CALCULATION OF THE AVERAGE
C PRESSURE MATRIX.
C LOGICAL SWITCHES L3(I),ONE FOR EACH MOVING POINT,ARE SET TO TRUE
C WHEN A POINT STARTS OSCILLATING.IT IS THEN TAKEN OUT OF CIRCULATION
C TILL THE NEXT PIEZOMETRIC HEAD MATRIX IS READ FROM TAPE1.
40 C
TIME=0.
NTEL=0
C
C READ RECORD 1 FROM FILE 1
45 C
READ(1)DX,DY,MX,MY,NF,((T(I,J),S(I,J),Q(I,J),I=1,MX),J=1,MY),
1((F(N,I),I=1,5),N=1,NF)
PRINT 2000
WRITE(6,104)DX,DY,MX,MY,NF
50 C
C READ DATA FROM CARDS
C
PEAD(5,100)DT2,DI,AMIN,TMAX,RHO,THICK,BI,TSTEP,NSW,LET
DT5=DT2
55 THRHO=THICK*RHO
PRINT 2001
WRITE(6,888)DT2,DI,AMIN,TMAX,RHO,THICK,BI,TSTEP,NSW,LET

```

```

C
C READ STARTING LOCATIONS FROM CARDS.
50 C
      READ(5,103) (SWX(I),SWY(I),I=1,NSW)
      WRITE(6,101) NTEL,TIME,(SWX(I),SWY(I),I=1,NSW)
      DO 11 I=1,NSW
        SPX(I)=SWX(I)
65      11 SPY(I)=SWY(I)
C
C READ NEW PIEZOMETRIC-HEAD MATRIX FROM FILE 1.
C
      1 READ (1)      TIME2,DT,((QQ(I,J),I=1,MX),J=1,MY)
70 C
C END OF FILE 1 REACHED(QUESTION MARK).
C
      IF (EOF(1)) 7,111
C
75 C SELECT VALUE FOR TIMESTEP.
C
      111 DT2=DT5
      DO 500 I=1,NSW
600 L3(I)=.FALSE.
80      DT4=DT
C
C CALCULATE AVERAGE PRESSURE MATRIX AT TIME MIDWAY BETWEEN T AND T+DT
C
      5 IF(DT.EQ.0.)GOTO 1
85 514 IF(DT.LT.DT2)GOTO 2
      DT3=DT2
      FAC=DT3/DT
      6 DO 3 I=1,MX
        DO 3 J=1,MY
90      QQ=FAC*(QQ(I,J)-Q(I,J))
      QQQ(I,J)=Q(I,J)+QQ/2.
      QRES(I,J)=Q(I,J)
      3 Q(I,J)=Q(I,J)+QQ
      DTRES=DT
95      DT=DT-DT3
      9 TIME=TIME+DT3
      IF (TIME.GT.TMAX) GO TO 55
C
C CALCULATE DISPLACEMENTS FOR BOUNDARY LOCATIONS
100 C
      DO 703 I=1,NSW
        TEMP=DHX(I)
        TEMQ=DHY(I)
        DHX(I)=0.
105      DHY(I)=0.
        IF(L3(I))GOTO 703
C
C TEST BOUNDARY POINTS FOR PROXIMITY TO WELLS
C
110      DO 734 J=1,NF
        IF(F(J,3).GE.0..OR.F(J,4).GT.TIME..OR.F(J,5).LT.TIME)GOTO 734
        IF(SPX(I).EQ.F(J,1).AND.SPY(I).EQ.F(J,2))GOTO 703
        XWELL=ABS(SPX(I)-F(J,1))
        IF(XWELL.GT.D1)GOTO 734

```

```

115      YWELL=ABS(SPY(I)-F(J,2))
        IF(YWELL.GT.DI)GOTO 734
C
C      IF A POINT IS TOO CLOSE TO A PUMPING WELL
C      SET LOCATION OF THAT POINT EQUAL TO LOCATION OF THAT WELL.
120 C      SET L3(I) FOR THAT POINT TO TRUE,INDICATING THAT NO DISPLACEMENT
C      HAS TO BE CALCULATED FOR THAT POINT UNTILL THE WELL STOPS PUMPING.
C
        SPY(I)=F(J,2)
        SPX(I)=F(J,1)
125      GOTO 706
734 CONTINUE
C
C      SELECT THE SUBROUTINE FOR THE GRADIENT CALCULATION.
C
130      JX=SPX(I)
        JY=SPY(I)
        IF(JX.GT.(MX-2).OR.JX.LT.3.OR.JY.LT.3.OR.JY.GT.(MY-2))GOTO 707
        AJX=JX
        AJY=JY
135      IF(S(JX,JY).EQ.1.)GOTO 601
        IF(LET.EQ.0)GOTO 601
        IF(SPX(I)-AJX.GT.0.500)JX=JX+1
        AJX=JX
        IF(SPY(I)-AJY.GT.0.500)JY=JY+1
140      AJY=JY
601      ALPHA=SPX(I)-AJX
        BETA=SPY(I)-AJY
        IF(LET.EQ.0)GOTO 510
        IF(S(JX,JY+1).EQ.1.)GOTO 509
145      IF(S(JX,JY-1).EQ.1.)GOTO 509
        IF(S(JX+1,JY).EQ.1.)GOTO 509
        IF(S(JX-1,JY).EQ.1.)GOTO 509
        JAB=0
        AAJX=AJX
150      AAJY=AJY
507      DO 508 J=1,NF
        IF(AAJX.NE.F(J,1).OR.AAJY.NE.F(J,2))GOTO 508
        IF(F(J,3).EQ.0.)GOTO 508
        IF(F(J,4).GT.TIME.OR.F(J,5).LT.TIME)GOTO 508
155      GOTO 509
508 CONTINUE
        IF(JAB.EQ.0)GOTO 506
        IF(JAB.EQ.1)GOTO 505
        IF(JAB.EQ.2)GOTO 504
160      IF(JAB.EQ.3)GOTO 503
        CALL DF(ALPHA,BETA,DHX(I),DHY(I))
        GOTO 511
506      JAB=1
        AAJX=AJX+1.
165      GOTO 507
505      JAB=2
        AAJX=AJX-1.
        GOTO 507
504      JAB=3
170      AAJX=AJX
        AAJY=AJY-1.

```

```

        GOTO 507
503 JAB=4
    AAJY=AJY+1.
175 GOTO 507
509 IF (S(JX,JY).EQ.1.)GOTO 510
    IF (LET.EQ.0)GOTO 510
    CALL DH(ALPHA,BETA,DHX(I),DHY(I))
    GOTO 511
180 510 CALL DG(ALPHA,BETA,DHX(I),DHY(I))
511 IF (NTEL.LT.8)GOTO 708
    IF (DHX(I)*TEMP.LT.0..AND.DHY(I)*TEMP.LT.0.)GOTO 709
C
C   CALCULATE T FROM THE VALUES AT THE 4 SURROUNDING NODES.
185 C
    708 ALBE=ALPHA*BETA
        TAB=T(JX,JY)*(ALBE-ALPHA-BETA)+T(JX+1,JY)*ALPHA*(1.-BETA)+
        1T(JX,JY+1)*BETA*(1.-ALPHA)+T(JX+1,JY+1)*ALBE+T(JX,JY)
C
190 C   CALCULATE THE DISPLACEMENT.
C
        PROD=TAB*DT3/THRHO
        DHX(I)=PROD*DHX(I)/DX
        DHY(I)=DHY(I)*PROD/DY
195        TEMPX=SPX(I)-DHX(I)
        TEMPY=SPY(I)-DHY(I)
        DEL=DT2/DT3
        DHX(I)=DHX(I)*DEL
        DHY(I)=DHY(I)*DEL
200 C
C   TEST FOR NEW BOUNDARY LOCATIONS TO BE IN AREA COVERED BY AQUIFER,
C   OTHERWISE RETAIN OLD BOUNDARY LOCATION
C
        JX=TEMPX
        JY=TEMPY
205        IF (S(JX ,JY ).NE.1.)GOTO 704
        IF (S(JX+1,JY ).NE.1.)GOTO 704
        IF (S(JX+1,JY+1).NE.1.)GOTO 704
        IF (S(JX ,JY+1).NE.1.)GOTO 704
210        GOTO 706
    704 SPX(I)=TEMPX
        SPY(I)=TEMPY
        GOTO 703
    707 SPX(I)=0.
        SPY(I)=0.
215    709 L3(I)=.TRUE.
        GOTO 703
    706 DHX(I)=0.
        DHY(I)=0.
220    L3(I)=.TRUE.
    703 CONTINUE
C
C   TEST IF DISPLACEMENT FALLS WITHIN LIMITS BI AND DI. IF DISPLACEMENT
C   IS TOO LARGE RECALCULATE TIMESTEP AND REDO THE STEP.
225 C   IF DISPLACEMENT IS TOO SMALL,RECALCULATE TIMESTEP AND GO ON TO THE
C   NEXT STEP.
C
        IF (L2)GOTO 600

```

```

230      IF (DT4.LT.DT2) GOTO 20
        600 AMAXI=0.
          DO 16 I=1,NSW
            IF (L3(I)) GOTO 16
            TEST=ABS(DHX(I))
            IF (TEST.LE.AMAXI) GOTO 17
235      AMAXI=TEST
          17 TEST=ABS(DHY(I))
            IF (TEST.LE.AMAXI) GOTO 16
            AMAXI=TEST
          16 CONTINUE
240      IF (AMAXI.EQ.0.0.AND.L2) STOP
          IF (AMAXI.EQ.0.) DT2=DT4
          IF (AMAXI.EQ.0.) GOTO 110
          IF (AMAXI.GE.BI.AND.AMAXI.LE.DI) GOTO 13
          IF (AMAXI.GT.DI) GOTO 512
245      IF (AMAXI.LT.BI) DT2=BI/AMAXI*DT2
          IF (L2) GOTO 110
          IF (DT2.GT.DT4) DT2=DT4
        110 CONTINUE
          IF (DT2.LT.0.0001) GO TO 55
250      GOTO 13
        512 DT2=.90*DI*DT2/AMAXI
          TIME =TIME-DT3
          DT=DTRES
          DO 513 I=1,MX
255      DO 513 J=1,MY
        513 Q(I,J)=QRES(I,J)
          GOTO 514
C
C      PRINT RESULTS IF TOTAL ACCUMULATED TIME SINCE LAST PRINTOUT
260 C      EXCEEDS TSTEP AND AT LEAST ONE DISPLACEMENT IS LARGER THAN AMIN.
C
        20 DO 12 I=1,NSW
          IF (ABS(SWX(I)-SPX(I)).GT.AMIN) GOTO 13
          IF (ABS(SWY(I)-SPY(I)).GT.AMIN) GOTO 13
265      12 CONTINUE
          GOTO 14
        13 DO 15 I=1,NSW
          SWX(I)=SPX(I)
          15 SWY(I)=SPY(I)
270      NTEL=NTEL+1
          IF (NTEL.LT.3) GOTO 1000
          IF ((TIME-TLAST).LT.TSTEP) GOTO 14
          DO 1002 I=1,NSW
            XTST=ABS(SWX(I)-SWPX(I))
275      YTST=ABS(SWY(I)-SWPY(I))
            IF (XTST.GT.AMIN) GOTO 1000
            IF (YTST.GT.AMIN) GOTO 1000
        1002 CONTINUE
          GOTO 14
280      1000 WRITE(6,101) NTEL,TIME,(SWX(I),SWY(I),I=1,NSW)
          DO 1003 I=1,NSW
            SWPX(I)=SWX(I)
            1003 SWPY(I)=SWY(I)
          TLAST=TIME
285      14 IF (L2) GOTO 9

```

```

      GOTO 5
2  DT3=DT
   FAC=1.
   GOTO 6
290  C
   C PRINT ,, STEADY STATE REACHED,, AND SET L2 TO TRUE.
   C
      7 WRITE(5,102)
      DT3=DT2
295  L2=.TRUE.
      DO 2002 I=1,NF
2002 IF (F(I,5).GT.TIME2) F(I,5)=TMAX*2.
      GOTO 9
55  STOP
300  END

      SUBROUTINE DF(A,B,DHX,DHY)
      COMMON DX,DY,Q(50,50),JX,JY
      AB=(A*A+B*B-1.)/2.
      R1=(1.+A+A*B)/4.
5     R2=(1.-A-A*B)/4.
      R3=(1.-A-A*B)/4.
      R4=(1.+A+A*B)/4.
      R5=A*A/4.
      DHX=(A-A*B)*Q(JX+1,JY)+(A+A*B)*Q(JX-1,JY)-2.*A*Q(JX,JY)+B*(R1*Q(JX+
10  11,JY+1)-R2*Q(JX-1,JY+1))+B*(R3*Q(JX-1,JY-1)-R4*Q(JX+1,JY-1))
      +R5*(Q(JX+2,JY)-Q(JX-2,JY))+A*B*(Q(JX,JY-1)-Q(JX,JY+1))
      R1=(1.+B+B*A)*A/4.
      R2=(1.+B+B*A)*A/4.
      R3=(1.-B-B*A)*A/4.
15     R4=(1.-B-B*A)*A/4.
      R5=B*B/4.
      DHY=(B-A*B)*Q(JX,JY+1)+(B+A*B)*Q(JX,JY-1)-2.*B*Q(JX,JY)+R1*Q(JX+1,JY+1)
      +R2*Q(JX-1,JY+1)+R3*Q(JX-1,JY-1)+R4*Q(JX+1,JY-1)+R5*(Q(JX,JY+2)-Q(JX,JY
20  2-Q(JX,JY-2))+A*B*(Q(JX-1,JY)-Q(JX+1,JY))
      DHX=DHX/DX
      DHY=DHY/DY
      RETURN
      END

      SUBROUTINE DH(A,B,DHX,DHY)
      COMMON DX,DY,Q(50,50),JX,JY
      S=(Q(JX+1,JY+1)-Q(JX-1,JY+1)-Q(JX+1,JY-1)+Q(JX-1,JY-1))/4.
      R=2.*Q(JX,JY)
5     DHX=G(JX+1,JY)*(A+0.500000)+Q(JX-1,JY)*(A-0.500000)-R*A+B*S
      DHX=DHX/DX
      DHY=G(JX,JY+1)*(B+0.500000)+Q(JX,JY-1)*(B-0.500000)-R*B+A*S
      DHY=DHY/DY
      RETURN
10     END

      SUBROUTINE DG(A,B,DHX,DHY)
      COMMON DX,DY,Q(50,50),JX,JY
      DHX=G(JX+1,JY)-Q(JX,JY)
      DHA=G(JX+1,JY+1)-Q(JX,JY+1)
5     DHX=((1.-B)*DHX+B*DHA)/DX
      DHY=G(JX,JY+1)-Q(JX,JY)
      DHA=G(JX+1,JY+1)-Q(JX+1,JY)
      DHY=((1.-A)*DHY+A*DHA)/DY
10     RETURN
      END

```