Environment Canada | Environnement Canada

PROGRAM ESOPH—Extended SOPH, Simulation of Time-Variant Piezometric Surface in a Confined, Leaky Aquifer Subjected to Pumping

A. Vandenberg

INLAND WATERS DIRECTORATE,
WATER RESOURCES BRANCH,
OTTAWA, CANADA, 1976.

(Résumé en français)

CONTRACT #02KXKL327-3-8061
THORN PRESS LIMITED

Abstract

Modifications to the mathematical development, to the computer program, and to the input requirements are described which extend program SOPH to program ESOPH for the simulation of leaky aquifers and for the use of variable dimensions of the elementary rectangle of the finite-difference grid.

This publication is not complete in itself; it should be read in conjunction with Vanden Berg (1974a) on "A Digital Simulation of Horizontal Salt-Water Encroachment Induced by Fresh-Water Pumping" and Vanden Berg (1974b) on "Program SOPH - Simulation of Time-Variant Piezometric Surface in a Confined Aquifer Subjected to Pumping".

Résumé

La présente publication décrit les modifications apportées à l'élaboration mathématique, au programme informatique ainsi qu'aux exigences relatives aux données à traiter qui transforment le programme SOPH en programme ESOPH en augmentant ses possibilités de manière à lui permettre de simuler des aquifères à nappe semi-perméable et d'utiliser les dimensions variables du rectangle élémentaire de la grille des différences finies.

La publication n'est pas exhaustive; le lecteur est prié de se reporter aux publications suivantes: *A Digital Simulation of Horizontal Salt-Water Encroachment Induced by Fresh-Water Pumping* de Vanden Berg (1974a) et *Program SOPH - Simulation of Time-Variant Piezometric Surface in a Confined Aquifer Subjected to Pumping* du même auteur (1974b).

PROGRAM ESOPH—Extended SOPH, Simulation of Time-Variant Piezometric Surface in a Confined, Leaky Aquifer Subjected to Pumping

A. Vandenberg

INTRODUCTION

Program ESOPH extends the capabilities of program SOPH (Vanden Berg, 1974b) to include the simulation of leaky aquifers and to give the option of using variable elementary grid dimensions. The latter feature allows a better description of the aquifer and better definition of the piezometric head in the neighbourhood of critical points. It also makes it possible to move the model boundaries, which are often not wanted, to great distances, where they do not influence the drawdown in the area of interest - as in the case of the sample run, where an aquifer of infinite areal extent is to be simulated.

Both new features have been tested extensively against analytical solutions and give accurate results; a sample run of the program shows the results of one of these tests.

This write-up describes the modifications that were made in the mathematical development (Vanden Berg, 1974a) and in the programming and the input requirements (Vanden Berg, 1974b). The complete new program listing is appended.

The new program, which requires considerably more central memory storage and computer time, has been given a new name ESOPH, in preference to treating it as a new version of program SOPH. In many cases program SOPH will be adequate in view of the limited amount of data available for the aquifer to be simulated; program SOPH is therefore maintained as described by Vanden Berg (1974a, 1974b) as a separate program.

MODIFICATIONS TO THE MATHEMATICAL DEVELOPMENT

In this section frequent reference will be made to the earlier publication by Vanden Berg (1974a), specifically to the numbered equations of that publication. Thus, unless stated otherwise, equation numbers in brackets will refer to equations in Vanden Berg (1974a); corresponding equations developed for ESOPH will have the same number followed by an a.

The symbols used in this publication, where they duplicate those appearing in Vanden Berg (1974a), have - with one exception - the same significance and are defined in the list of symbols given in that publication. A few symbols (H_o, K', b') are new and one symbol (L) duplicates one used in the earlier publication, although it has a different meaning. These last four symbols are defined at the point at which they are introduced; the dual use of L causes no practical difficulty and its usage in this publication conforms to conventional hydrogeological usage.

(a) Variable Elementary Dimensions

In order to include variable elementary dimensions Δx and Δy in the finite-difference formulation of (3) the terms $\Delta_x^2(T,H)$ and $\Delta_y^2(T,H)$ in (5) must be rewritten (Fig. 1):

$$(T\ \partial H/\partial x)_B \approx (\tfrac{1}{2})(T_{i,j} + T_{i+1,j})(H_{i+1,j} - H_{i,j})/\Delta x_i$$

$$(T\ \partial H/\partial x)_A \approx (\tfrac{1}{2})(T_{i,j} + T_{i-1,j})(H_{i,j} - H_{i-1,j})/\Delta x_{i-1}$$

where the subscripts A and B indicate that the values are approximated at the points A and B, respectively, of Figure 1. Thus,

$$\Delta_x^2(T,H) = \partial/\partial x(T\ \partial H/\partial x) \approx [(T\ \partial H/\partial x)_B - (T\ \partial H/\partial x)_A]/[(\tfrac{1}{2})(\Delta x_{i-1} + \Delta x_i)]$$

$$\approx (T_{i,j} + T_{i+1,j})(H_{i+1,j} - H_{i,j})/[\Delta x_i(\Delta x_{i-1} + \Delta x_i)]$$

$$-(T_{i,j} + T_{i-1,j})(H_{i,j} - H_{i-1,j})/[\Delta x_{i-1}(\Delta x_{i-1} + \Delta x_i)]$$

Similarly,

$$\Delta_y^2(T,H) \approx (T_{i,j} + T_{i,j+1})(H_{i,j+1} - H_{i,j})/[\Delta y_j(\Delta y_{j-1} + \Delta y_j)]$$

$$-(T_{i,j} + T_{i,j-1})(H_{i,j} - H_{i,j-1})/[\Delta y_{j-1}(\Delta y_{j-1} + \Delta y_j)]$$

Thus, the coefficients $A_{i,j}$, $B_{i,j}$, $C_{i,j}$ and $D_{i,j}$ in (11) and (12) become

$$A_{i,j} = (T_{i,j} + T_{i+1,j})/[\Delta x_i(\Delta x_{i-1} + \Delta x_i)]$$

$$B_{i,j} = (T_{i,j} + T_{i-1,j})/[\Delta x_{i-1}(\Delta x_{i-1} + \Delta x_i)]$$

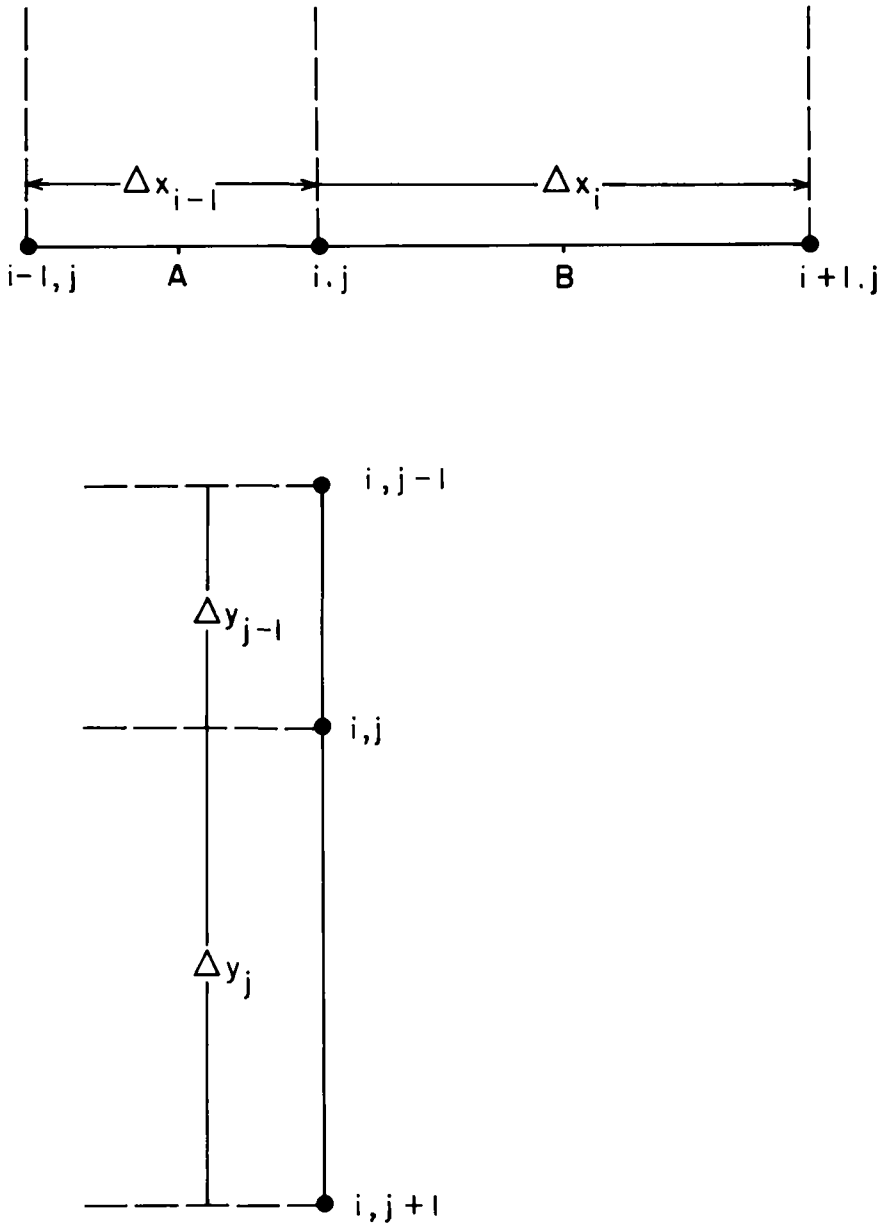

Figure 1. Numbering of the internodal distances.

$$C_{i,j} = (T_{i,j} + T_{i,j+1})/[\Delta y_j(\Delta y_{j-1} + \Delta y_j)]$$

$$D_{i,j} = (T_{i,j} + T_{i,j-1})/[\Delta y_{j-1}(\Delta y_{j-1} + \Delta y_j)]$$

(b) <u>The Leaky Aquifer</u>

To adapt the model to the leaky aquifer the leakance term must be introduced in (3), which then becomes (compare, for example, with Hantush and Jacob, 1955)

$$\partial/\partial x(T\ \partial H/\partial x) + \partial/\partial y(T\ \partial H/\partial y) = S(\partial H/\partial t) + f_{i,j} + (H - H_o)L \qquad (3a)$$

where

H_o = the head at the top of the overlying confining bed, which is constant in time but may be spatially variable,

L = K'/b', the "specific leakage" or "leakance" of the confining bed,

K' = the vertical permeability of the confining bed,

b' = the thickness of the confining bed.

Over the time interval Δt the piezometric head H in the aquifer is best represented by the average of H_n and H_{n+1} - the heads

at the beginning and end of the time interval. The finite difference expression for $(H - H_o)L$ then becomes

$$[(H_{n+1} + H_n)/2 - H_o]L$$

which, added to the right-hand side of (5), gives

$$\Delta_x^2(T,H) + \Delta_y^2(T,H) = S_{i,j}(H_{n+1} - H_n)/\Delta t + f_{i,j} - LH_o + (L/2)(H_n + H_{n+1})$$

$$= H_{n+1}(S_{i,j}/\Delta t + L/2) - H_n(S_{i,j}/\Delta t - L/2) + f_{i,j} - LH_o$$

---------- (5a)

If H_o and L are variable in the xy-plane, their values in (5a) will depend on the location of the node (i,j).

Continuation with the development of the analogous equations for (6), (7), (8), (9), (10), (11) and (12), yields:

$$\Delta_x^2(T,H)_n/2 + \Delta_x^2(T,H^*)_{n+1}/2 + \Delta_y^2(T,H)_n$$
$$= H^*_{n+1}(S_{i,j}/\Delta t + L/2) - H_n(S_{i,j}/\Delta t - L/2) + f_{i,j} - LH_o \quad . \tag{6a}$$

$$\Delta_x^2(T,H)_n/2 + \Delta_x^2(T,H^*)_{n+1}/2 + \Delta_y^2(T,H)_n/2 + \Delta_y^2(T,H)_{n+1}/2$$
$$=H_{n+1}(S_{i,j}/\Delta t + L/2) - H_n(S_{i,j}/\Delta t - L/2) + f_{i,j} - LH_o \quad . \tag{7a}$$

$$-\Delta_y^2(T,H)_n/2 + \Delta_y^2(T,H)_{n+1}/2 = (H_{n+1} - H^*_{n+1})(S_{i,j}/\Delta t + L/2) \ . \tag{8a}$$

$$\Delta_x^2(T,H^*)_{n+1} - (2S_{i,j}/\Delta t + L)H^*_{n+1} = -\Delta_x^2(T,H)_n - 2\Delta_y^2(T,H)_n$$
$$-(2S_{i,j}/\Delta t - L)H_n + 2f_{i,j} - 2LH_o \ . \tag{9a}$$

$$\Delta_y^2(T,H)_{n+1} - (2S_{i,j}/\Delta t + L)H_{n+1} = \Delta_y^2(T,H)_n - (2S_{i,j}/\Delta t + L)H^*_{n+1} . \tag{10a}$$

$$A_{i,j}\ H^*_{i+1,j,n+1} - (A_{i,j} + B_{i,j} + 2S_{i,j}/\Delta t + L)H^*_{i,j,n+1} + B_{i,j}\ H^*_{i-1,j,n+1}$$
$$= -A_{i,j}\ H_{i+1,j,n} - B_{i,j}\ H_{i-1,j,n} - 2C_{i,j}\ H_{i,j+1,n} - 2D_{i,j}\ H_{i,j-1,n}$$

$$+ (A_{i,j} + B_{i,j} + 2C_{i,j} + 2D_{i,j} - 2S_{i,j}/\Delta t + L)H_{i,j,n}$$

$$+ 2f_{i,j} - 2LH_o \ . \qquad (11a)$$

$$C_{i,j}\, H_{i,j+1,n+1} - (C_{i,j} + D_{i,j} + 2S_{i,j}/\Delta t + L)H_{i,j,n+1} + D_{i,j}\, H_{i,j-1,n+1}$$

$$= C_{i,j}\, H_{i,j+1,n} + D_{i,j}\, H_{i,j-1,n} - (C_{i,j} + D_{i,j})H_{i,j,n}$$

$$- (2S_{i,j}/\Delta t + L)H^{*}_{i,j,n+1} \qquad (12a)$$

MODIFICATIONS TO THE COMPUTER PROGRAM

Several modifications have to be made in order to accommodate the use of variable dimensions for the rectangular grid elements. Storage space for the two vectors holding the values of Δx_i, i = 1 to 50 and Δy_j, j = 1 to 50 has to be provided; the values of the vector elements have to be read in; and the factors $A_{i,j}$, $B_{i,j}$, $C_{i,j}$ and $D_{i,j}$ (C1, C3, D1 and D3 in the program) have to be recalculated. Minor modifications are required in the calculation of the drawdown at the well node and in the calculation of the source/sink term f. Furthermore, the use of variable dimensions is provided as a yes or no option, the choice being governed by the value given to the new logical variable VARXY. Statements in the program pertaining to this option are easily recognized since they all begin with: IF(VARXY)..... or: IF(.NOT.VARXY).....

The modifications to include leaky-aquifer simulation consist of providing storage matrices for the leakance and for the head H_o, placing these values in the input deck and calculating the coefficients of $H^*_{i,j,n+1}$, $H_{i,j,n}$ and $H_{i,j,n+1}$ in (11a) and (12a).

Under normal operation of the program, the head H_o at the top of the confining bed will be the same as the initial head H in the aquifer, and therefore does not need to be read in separately. However, a slight problem presents itself if the natural, nonpumping piezometric surface in the aquifer is not known. In that case the program can be used in a first stage to determine the steady-state, natural piezometric surface in the aquifer compatible with the given boundary conditions and aquifer properties, and only in a second stage is pumping introduced, with the initial piezometric head in the aquifer equal to the final head calculated in the first stage. In the second stage H_o must be reset to the original head matrix at the beginning of the first stage, or else piezometric-head (H) adjustments will take place which are the result of the natural piezometric surface tending toward a new equilibrium rather than the result of pumping only. In the program the variable IHO is used to specify one of three alternatives:

1) $H_o = 0$,
2) H_o = will be read in from cards or,
3) H_o = the initial value of H.

It must be realized, however, that the value of H_o will strongly influence the equilibrium configuration of H sought in the first

stage of the simulation; even to the extent that for an aquifer which is totally enclosed by impermeable boundaries, H at equilibrium will be equal to H_0 at all nodes. Thus, in this case the specification of H_0 completely replaces the first stage.

INPUT

(a) Necessary Input Data

Only those additional input data are listed which are not already described for program SOPH.

VARXY - Logical variable: if True, variable dimensions are used, the vectors containing these dimensions are expected in the data stream, and the values of DX and DY will be ignored; if False, no variable dimensions are expected and the values of DX and DY are used.

IHO - integer variable. If

IHO = 1, H_0 is assumed 0 at all nodes;

IHO = 2, values for the H_0 matrix are expected in the input-data stream; and

IHO $\neq$ 1 or 2, H_0 is set equal to the initial value of H.

DDXX(50) - Floating point variable vector; DDXX(I) is the distance between the nodes (I,J) and the nodes (I+1,J), I = 1,2,.. ,MX (feet, if the English system of units is used - see Section D.3, option (i)(a) (Vanden Berg, 1974b)). If VARXY is True, MX values must be given.

DDYY(50) - Floating point variable vector; DDYY(J) is the distance between the nodes (I,J) and the nodes (I,J+1), J = 1,2,.. ,MY (feet, if option (i)(a) is used). If VARXY is True, MY values must be given.

B(50,50) - Leakance (minutes^{-1}, if option (i)(a) is used). These values are read in following the values of transmissivity and storage, either in rectangular blocks if IMODE is False, or as single values if IMODE is True.

BE - The value of the leakance for a rectangular subarea (minutes^{-1}, if option (i)(a) is used); only used if IMODE is False, i.e. the input mode by rectangular blocks is selected.

HO(50,50) - Head at the top of the overlying leaky bed (feet, if option (i)(a) is used). These values are read in following values of H, only if IHO = 2.

(b) Data Deck Instructions

The following table specifies the setup of data cards for use in the program. In the table the group names (single capitals) used in program SOPH are retained for those data which are identical in both programs; group names for data not occurring in program SOPH consist of two capitals. Card numbers in each group are the same as used for program SOPH if the data on the card are identical to the data on the corresponding card in program SOPH; modified cards and new cards are given a number and lower-case alphabetic character.

Group	Card	FORMAT	Columns	Data and Type
A	1	(8A10)	1-30	FM1, alphameric.
			31-60	FM2, alphameric.
			61-80	FM3, alphameric.
B	1	(7E10.0)	1-70	DX,DY,DT,TINC,TMAX, VAR,FAC: floating point, 10 columns per variable.
	2a	(13I5,6L1,I1)	1-65	NF,MX,MY,I7,I8,I9 J7, J8,J9,M7,M8,K7,K8, integer, 5 columns per variable.
			66-71	HRDE,PCH,SYSTEM,IMODE, FRONT,VARXY: logical, 1 column per variable.
			72	IHO, integer.
BA, only if VARXY = T(rue)				
	1a	(13E6.0)	1-78	DDXX(1),DDXX(2)...DDXX(MX);
	2a	'	'	floating point, 6 columns
	'	'	'	per value, MX values

Group	Card	FORMAT	Columns	Data and Type
BB, only if VARXY = T(rue)				
	1a	(13E6.0)	1-78	DDYY(1),DDYY(2)...DDYY(MY);
	2a	'	'	floating point, 6 columns
	'	'	'	per value, MY values.
C	1	6(I1,6I2)	1	IX(1,1); integer.
			2-13	IX(1,2), IX(1,3)...IX(1,7); integer, 2 columns per variable.
			14	IX(2,1); integer.
			15-26	IX(2,2),IX(2,3)...IX(2,7); integer, 2 columns per variable.
			27-39	IX(3,1)...IX(3,7)
			etc.	
			66-78	IX(6,1)...IX(6,7)
	2	6(I1,6I2)	1-78	IX(7,1)...IX(12,7)
	'			'
	'			'
	'			' ...IX(MY,7)
D	1	6(I1,6I2)	as for group C	IY(1,1)...IY(MX,7); integer.
	2	6(I1,6I2)		
	'	'		
	'	'		
	'	'		
	'	'		

Group	Card	FORMAT	Columns	Data and Type
E, only if IMODE = F(alse)				
	1	(F10.0,4I2)	1-10	TE; floating point.
			11-18	LX,LXA,LY,LYA; integer,
	'	'		2 columns per variable.
	2	(F10.0,4I2)	1-18	as for card 1.
	'	'		
	last card	-	-	blank card.
F, only if IMODE = F(alse)				
	1	(F10.0,4I2)	1-10	SE; floating point.
			11-18	LX,LXA,LY,LYA: integer,
				2 columns per variable.
	2	(F10.0,4I2)	1-18	as for card 1.
	'			
	'			
	last card	-	-	blank card.
FA, only if IMODE = F(alse)				
	1a	(F10.0,4I2)	1-10	BE; floating point.
			11-18	LX,LXA,LY,LYA: integer,
				2 columns per variable.
	2a	(F10.0,4I2)	1-18	as for card 1a.
	'			
	'			
	last card	-	-	blank card.

Group	Card	FORMAT	Columns	Data and Type
G, only if IMODE = T(rue)				
	1	(13E6.0)	1-78	T(1,1),T(2,1)...T(MX,1),
	2	(13E6.0)		T(1,2),T(2,2)...T(MX-1,MY),
	'	'		T(MX,MY); floating point,
	'	'		MX x MY variables, 13 per
	'	'		card, 6 columns per variable.
H, only if IMODE = T(rue)				
	1	(13E6.0)	1-78	S(1,1),S(2,1)...S(MX,1),
	2	(13E6.0)		S(1,2),S(2,2)...S(MX-1,MY),
	'	'		S(MX,MY): floating point,
	'	'		MX x MY variables, 13 per
				card, 6 columns per variable.
HA, only if IMODE = T(rue)				
	1a	(13E6.0)	1-78	B(1,1),B(2,1)...B(MX,1),
	2a	(13E6.0)		B(1,2),B(2,2)...B(MX-1,MY),
	'	'		B(MX,MY): floating point,
	'	'		MX x MY variables, 13 per
	'	'		card, 6 columns per variable.
I, only if HRDE = T(rue)				
	1	FM1, user specified		H(1,1),H(2,1)...H(MX,1),
	'			H(1,2),H(2,2)... H(MX-1,MY),
	'			H(MX,MY); floating point,
	'			MX x MY variables.

Group	Card	FORMAT	Columns	Data and Type
IA, only if IHO = 2				
	1	FM1, user specified		HO(1,1),HO(2,1)...HO(MX,1),
	'			HO(1,2),HO(2,2)...HO(MX-1,MY),
	'			HO(MX,MY); floating point,
	'			MX x MY variables.
J, only if NF > 0				
	1	(2F2.0,4F10.0)	1-4	F(1,1),F(1,2); floating point,
				2 columns per variable.
			5-44	F(1,3), F(1,4), F(1,5),
				F(1,6); floating point,
				10 columns per variable.
	2	(2F2.0,4F10.0)	1-44	F(2,1)...F(2,6).
	'	'		
	'	'		
	'	'		
	NF			F(NF,1)...F(NF,6).

(c) Sample Data Deck

As an example the necessary input data are given for the simulation of a pump test of one week duration. The aquifer is homogeneous, leaky and quasi infinite; that is, aquifer constants and model dimensions are chosen such that for observation wells within a practical distance (approximately 500 m) from the pumping well the influence of the boundaries on the drawdown is negligible.

Table 1 shows the data deck for a model in which the full 50 x 50-node grid is used to represent a square aquifer with sides of 7290 m. A well is pumping at a rate of 1 m^3/min at node (25,25) at the center of the aquifer. The nodes immediately surrounding the well are at a distance of 10 m, and the distances between nodes in both the x- and y-directions gradually increase outward (Fig. 2). Since the well is slightly off centre on the grid, the distances between the boundary and the first inside nodes may be be either 450 or 500 m. The aquifer coefficients are constants and each needs to be input only once for the whole grid area:

Transmissivity: 0.05 m^2/min

Storativity : 0.0001

Leakance : 5×10^{-8} min^{-1}.

The initial time step is 0.01 min, incremented at each step by a factor of 1.2. The nonpumping head in the aquifer and the head at the top of the leaky bed are both assumed to be zero (HRDE is false and IHO = 0).

Table 1. Input deck for the sample run.

ESOPH, Sample Input Deck

Name A. Vandenberg

Date July 15, 1975

Page 1 of 1

```
A 1     (5E15.7)                      (11E10.3)                     (50I2/)
B 1            100       100       .01      1.20  1000000.     .0001       .50
  2         1   50   50   15   25    1   25   31    3    1   50    1   50FFFFFT0
BA 1a      450   400   350   300   250   220   200   180   160   140   120   100    90
   2a       80    70    60    50    40    35    30    25    20    15    10    10    15
   3a       20    25    30    35    40    50    60    70    80    90   100   120   140
   4a      160   180   200   220   250   300   350   400   450   500
BB        Four cards, exactly as for group BA
C 1     10150        10150        10150        10150        10150        10150
  2-8     Seven cards, exactly as card C1
  9     10150        10150
D         nine cards, exactly as for group C
E 1           0.0501500150
  2
F 1          .000101500150
  2
FA 1a        5.E-801500150
   2a
J 1     2525       -1.             2000000
```

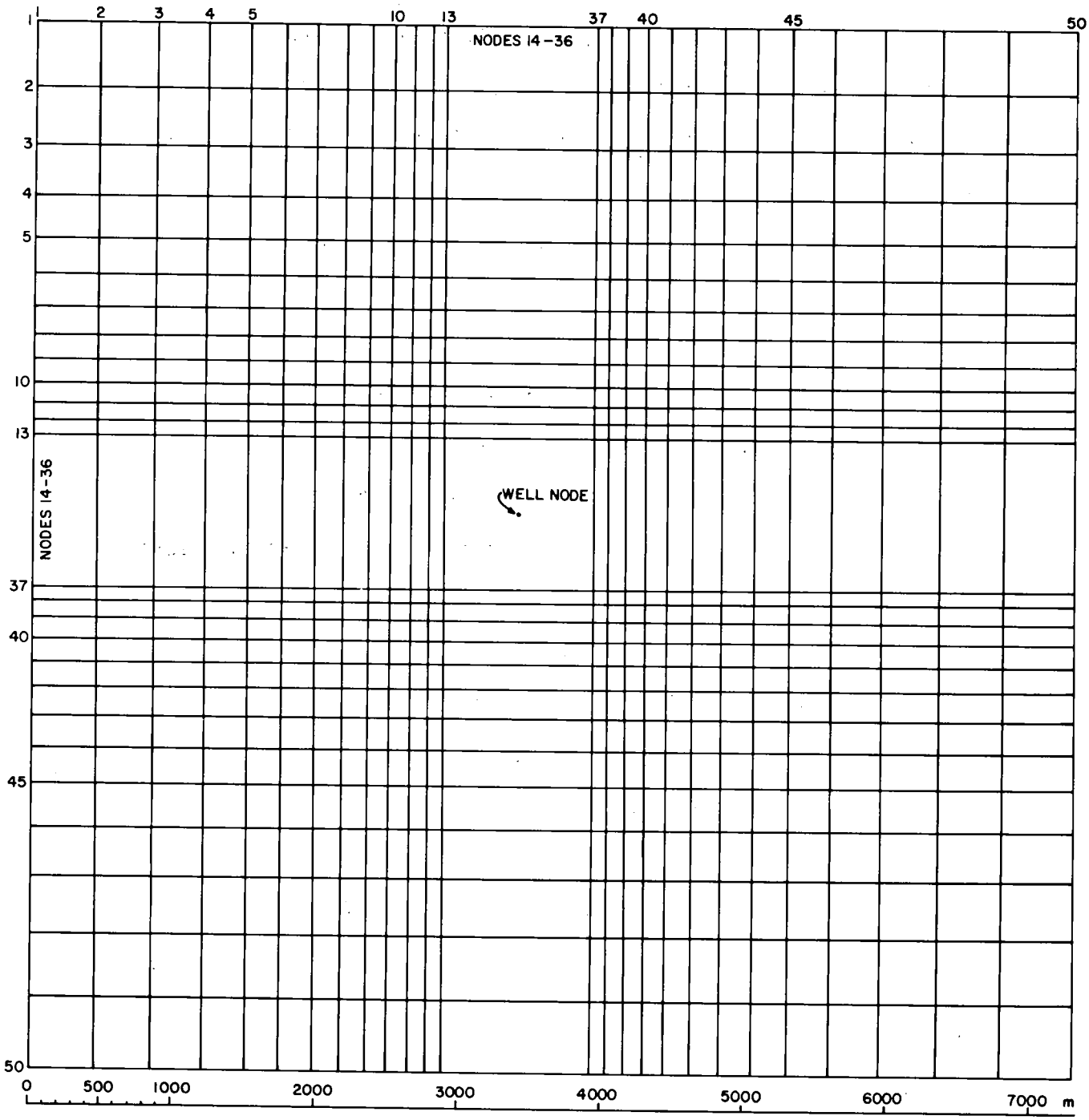

Figure 2. The finite-difference grid.

The format specification FM1 is ignored since no values of H or HO are read in; format specification FM2 provides for a selected 33 values of piezometric head to be printed at each time step to occupy 3 lines of 11 values each; format specification FM3 provides for 50 values of the last piezometric head matrix to be printed on one line, while the slash (/) indicates that one blank line is inserted between each row.

OUTPUT

No major changes have been made in the output format. If VARXY is True, the DDXX and DDYY vectors are printed out and if IHO = 2 and a set of values of HO is read in, the HO-matrix will be printed immediately following the H-matrix. The H-matrix itself will only be printed if not zero.

COMPARISON OF RESULTS WITH THE ANALYTICAL SOLUTION

The analytical solution for the drawdown s at a distance r from a well pumping at a constant rate Q from an infinite leaky aquifer is given by Hantush and Jacob (1955):

$$s = (Q/4\pi T) \int_u^\infty (1/x) \exp(-x - r^2/4B^2x)\, dx$$

where T = the transmissivity,

$u = r^2S/4Tt$,

S = the storativity,

t = the time since pumping started,

B = the leakage factor = $\sqrt{T/L}$,

L = the leakance.

In Figure 3 drawdowns calculated by the program from the sample data for the nodes (22,25), (20,25) and (15,25), at distances of 45 m, 100 m and 355 m, respectively, from the well node are plotted. The continuous curves in Figure 3 represent drawdown against time as calculated from the analytical expression. The discrepancy between the model and analytical solutions at early times is characteristic of the finite-difference method and can only be reduced by specifying a smaller initial time step. The fit at later times is excellent and demonstrates that the infinite leaky aquifer has been satisfactorily simulated.

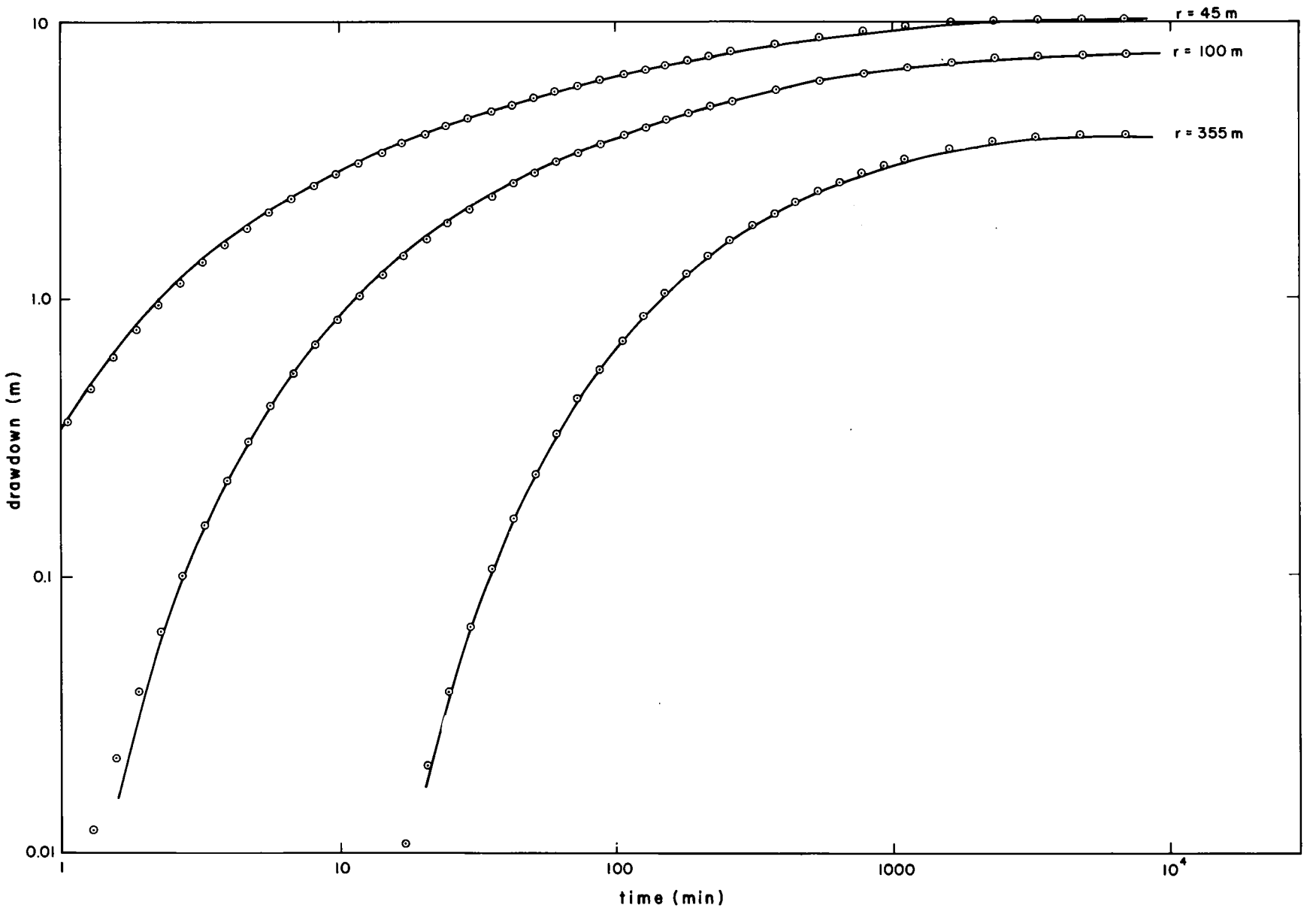

Figure 3. Time-drawdown curves from the sample model: analytical and finite-difference solutions.

REFERENCES

Hantush, M.S. and C.E. Jacob, 1955. Non-steady radial flow in an infinite leaky aquifer. Trans. Am. Geophys. Union, Vol. 36, No. 1, pp. 95-112.

Vanden Berg, A., 1974a. A digital simulation of horizontal salt-water encroachment induced by fresh-water pumping. Scientific Series No. 41, Inland Waters Directorate, Water Resources Branch, Ottawa, Canada (27 pages).

Vanden Berg, A., 1974b. Program SOPH - Simulation of time-variant piezometric surface in a confined aquifer subjected to pumping. Inland Waters Directorate, Water Resources Branch, Ottawa, Canada (56 pages).

Appendix

Program Listing

```
*          PROGRAM ESOPH (EXTENDED SIMULATION OF PIEZOMETRIC HEAD)      *
*                          BY A.VANDENBERG,                             *
*              ENVIRONMENT CANADA,INLAND WATERS DIRECTORATE.            *
*                                                                       *
*************************************************************************
*
***********              BRIEF PROGRAM DESCRIPTION             ************
*                                                                       *
*    THE TIME-VARIANT PIEZOMETRIC HEAD IN A HORIZONTAL AQUIFER  IS      *
*    CALCULATED AT THE NODES OF A RECTANGULAR GRID. THE METHOD          *
*    USED IS THE ALTERNATING DIRECTION IMPLICIT PROCEDURE (ADIP).       *
*    THE AQUIFER IS ISOTROPIC,BUT MAY BE INHOMOGENEOUS AND OF           *
*    IRREGULAR OUTLINE.UP TO 10 SINKS/SOURCES MAY BE  INCLUDED.         *
*                                                                       *
*    ESOPH IS AN EXTENSION OF PROGRAM SOPH,PERMITTING THE MODELLING     *
*    OF LEAKY AQUIFERS AND THE USE OF VARIABLE GRID SPACING.THE MATRIX  *
*    OF VALUES OF THE CONSTANT HEAD AT THE TOP OF THE OVERLYING LEAKY   *
*    BED CAN EITHER BE SET EQUAL TO THE ORIGINAL HEAD IN THE AQUIFER,   *
*    BE SET TO ZERO EVERYWHERE,OR READ IN SEPARATELY.                   *
*                                                                       *
*************************************************************************
*                                                                       *
***********            DEFINITION AND DATA STATEMENTS          ************
*                                                                       *
      PROGRAM ESOPH(INPUT,OUTPUT,PUNCH,TAPE1,TAPE5=INPUT,
     1TAPE6=OUTPUT,TAPE7=PUNCH)
      DIMENSION T(50,50),S(50,50),HH(50,50),F(10,6),FM1(3),FM2(3),A(150)
     1,D(50),FM3(2),TRES(10),H(50,50),IX(50,7),IY(50,7)
      DIMENSION DDXX(50),DDYY(50),B(50,50),HO(50,50)
      LOGICAL HRDE,PCH,SW2,SYSTEM,IMODE,FRONT,VARXY
      DATA SW2/.FALSE./
*************************************************************************
*                                                                       *
***********          READIN OF FORMATS AND CONTROL DATA          ************
*                                                                       *
      READ(5,102)FM1,FM2,FM3
  102 FORMAT(8A10)
      READ(5,100)DX,DY,DT,TINC,TMAX,VAR,FAC,NF,MX,MY,I7,I8,I9,J7,
     1J8,J9,M7,M8,K7,K8,HRDE,PCH,SYSTEM,IMODE,FRONT,VARXY,IHO
  100 FORMAT(7E10.6,/,13I5,6L1,I1)
      PRINT 87
   87 FORMAT(1H1,*DX,DY,DT,TINC,TMAX,VAR,      FAC,NF,MX,MY,I7,I8,I9,J7,
     1J8,J9,M7,M8,K7,K8,HRDE,PCH,SYSTEM,IMODE,FRONT,VARXY,IHO*)
      PRINT  89, DX,DY,DT,TINC,TMAX,VAR,       FAC,NF,MX,MY,I7,I8,I9,J7,
     1J8,J9,M7,M8,K7,K8,HRDE,PCH,SYSTEM,IMODE,FRONT,VARXY,IHO
   89 FORMAT(1X,7E15.4/13I6,6L6,I6)
*************************************************************************
*                                                                       *
****  READ VARIABLE NODAL DISTANCES ONLY IF VARXY = TRUE    *************
*                                                                       *
      IF(.NOT.VARXY)GOTO 712
      READ 201,(DDXX(I),I=1,MX)
      READ 201,(DDYY(I),I=1,MY)
      PRINT 3013
 3013 FORMAT(*0VARIABLE DX VECTOR*)
      PRINT 203,(DDXX(I),I=1,MX)
      PRINT 3014
 3014 FORMAT(*0VARIABLE DY VECTOR*)
```

```
      PRINT 203,(DDYY(I),I=1,MY)
  712 CONTINUE
*                                                                       *
************************************************************************
***********  READ INTEGERS FOR LOCATING RECHARGE BOUNDARIES  ***********
*
      PRINT 96
   96 FORMAT("0IX-MATRIX")
      READ(5,66)((IX(I,J),J=1,7),I=1,MY)
      PRINT 67, ((IX(I,J),J=1,7),I=1,MY)
      PRINT 97
   97 FORMAT("0IY-MATRIX")
      READ(5,66)((IY(I,J),J=1,7),I=1,MX)
      PRINT 67, ((IY(I,J),J=1,7),I=1,MX)
   67 FORMAT(1X,42I3)
   66 FORMAT(6(I1,6I2))
************************************************************************
      DTBEG=DT
      MY1=MY-1
      MX1=MX-1
*
***********                INITIALIZE   S,T,AND H              ***********
*
      DO 55 I=1,MX
      DO 55 J=1,MY
      S(I,J)=1.
      T(I,J)=0.
   55 H(I,J)=0.
************************************************************************
      PRINT 90
   90 FORMAT("0TRANSMISSIVITY-MATRIX")
*
***********    READ TRANSMISSIVITY AND STORAGE IN BLOCKS     ***********
*
      IF(IMODE)GOTO 200
    3 READ  101,TE,LX,LXA,LY,LYA
  101 FORMAT(F10.0,4I2)
      IF(LX.EQ.0)GOTO 3004
      PRINT 108,TE,LX,LXA,LY,LYA
  108 FORMAT(E20.5,4I5)
      IF(SYSTEM) TE=TE*1.1143E-4
      DO 2 I=LX,LXA
      DO 2 J=LY,LYA
      T(I,J)=TE
    2 CONTINUE
      GOTO 3
  200 READ 201,((T(I,J),I=1,MX),J=1,MY)
      PRINT 90
      PRINT203,((T(I,J),I=1,MX),J=1,MY)
      READ 201,((S(I,J),I=1,MX),J=1,MY)
      PRINT 91
      PRINT203,((S(I,J),I=1,MX),J=1,MY)
      READ 201,((B(I,J),I=1,MX),J=1,MY)
      PRINT 98
   98 FORMAT(*0LEAKANCE*)
      PRINT203,((B(I,J),I=1,MX),J=1,MY)
  201 FORMAT(13F6.0)
  203 FORMAT(1X,10F13.5)
```

```
      IF(.NOT.SYSTEM)GOTO  3000
      DO 202 I=1,MX
      DO 202 J=1,MY
  202 T(I,J)=T(I,J)*1.1143E-4
      GOTO 3000
 3004 PRINT 91
   91 FORMAT("0STORAGE COEFFICIENTS")
 3005 READ  101,SE,LX,LXA,LY,LYA
      IF(LX.EQ.0)GOTO 3002
      PRINT 108,SE,LX,LXA,LY,LYA
       DO 3001 I=LX,LXA
      DO 3001 J=LY,LYA
 3001 S(I,J)=SE
      GOTO 3005
 3002 PRINT 98
 3003 READ  101,SE,LX,LXA,LY,LYA
      IF(LX.EQ.0)GOTO 3000
      PRINT 108,SE,LX,LXA,LY,LYA
      DO 3006 I=LX,LXA
      DO 3006 J=LY,LYA
 3006 B(I,J)=SE
      GOTO 3003
 3008 DO 3011 I=1,MX
      DO 3011 J=1,MY
 3011 HO(I,J)=0.
      GOTO 3010
 3009  READ(5,FM1)((HO(I,K),I=1,MX),K=1,MY)
      WRITE(6,FM2)((HO(I,K),I=1,MX),K=1,MY)
      DO 3012 I=1,MX
      DO 3012 J=1,MY
 3012 HO(I,J)=-HO(I,J)*B(I,J)*2.
      GOTO 3010
*************************************************************************
 3000 WRITE(6,93)
   93 FORMAT("0ORIGINAL PRESSURE MATRIX")
*
***********  IF NOT 0 EVERYWHERE,READ ORIGINAL HEAD MATRIX   ***********
      IF(HROE)READ(5,FM1)((H(I,K),I=1,MX),K=1,MY)
      IF(HROE)WRITE(6,FM2)((H(I,J),I=1,MX),J=1,MY)
      IF(IHO.EQ.1)GOTO 3008
      IF(IHO.EQ.2)GOTO 3009
      DO 3007 I=1,MX
      DO 3007 J=1,MY
 3007 HO(I,J)=H(I,J)*B(I,J)*(-2.)
*************************************************************************
 3010 WRITE(6,92)
      IF(NF.EQ.0)GOTO 204
   92 FORMAT("0WELL ARRAY"/"     X       Y      RATE  FROM     TO")
*
***********    READ SOURCES AND SINKS.F(I,1)=X-COORDINATE    ***********
***********                           F(I,2)=Y-COORDINATE    ***********
***********                           F(I,3)=STRENGTH        ***********
***********                           F(I,4)=START TIME      ***********
***********                           F(I,5)=STOP  TIME      ***********
***********                           F(I,6)=WELL RADIUS     ***********
*
      DO 4 N=1,NF
      READ(5,104)(F(N,I),I=1,6)
```

```
      WRITE(6,105)(F(N,I),I=1,6)
  104 FORMAT(2F2.0,4F10.0)
  105 FORMAT("0",6E12.4)
*****************************************************************
      J=F(N,1)
      K=F(N,2)
*
*********** CALCULATE FINITE WELL EXCESS DRAWDOWN,STORING    ***********
*********** RESULT IN F(N,6).RECALCULATE STRENGTH IN F(N,3)  ***********
*
      IF(VARXY)DX=DDXX(J)
      IF(VARXY)DY=DDYY(K)
      IF(F(N,6).EQ.0.)GOTO 44
      F(N,6)=F(N,3)*5.881E-2/T(J,K)*ALOG10(DX/(4.81*F(N,6)))
   44 F(N,3)=2.*F(N,3)/(DX*DY)
      IF(SYSTEM)F(N,3)=F(N,3)*.16048
    4 CONTINUE
*****************************************************************
      PRINT 5702,(F(N,6),N=1,NF)
 5702 FORMAT(*0FINITE WELL DRAWDOWNS*/1X,10F12.3//)
204   DO 6 I=1,MX
      DO 6 J=1,MY
    6 HH(I,J)=H(I,J)
      DXS=DX*DX
      DYS=DY*DY
      TIME=0.
*
***********          WRITE PRELIMINARY DATA ON FILE 1          ***********
*
      IF(FRONT)
     1WRITE(1)DX,DY,MX,MY,NF,((T(I,J),S(I,J),H(I,J),I=1,MX),J=1,MY),
     2((F(N,I),I=1,5),N=1,NF),VARXY
      IF(VARXY)WRITE(1)(DDXX(I),I=1,MX),(DDYY(J),J=1,MY)
*****************************************************************
*
***********           ADIP,SWEEPING IN X-DIRECTION             ***********
*
   16 DO 9 J=2,MY1
      IF(IX(J,1).EQ.0)GOTO 9
      IBNDR=IX(J,1)
      DO 5010 L=1,IBNDR
      IBE=IX(J,L*2)+1
      IEN=IX(J,L*2+1)-1
      IK=2
      DO 10 I=IBE,IEN
      C1=(T(I-1,J)+T(I,J))/(DXS*2.)
      IF(VARXY)C1=C1*2.*DXS/(DDXX(I-1)*(DDXX(I-1)+DDXX(I)))
      IF(T(I-1,J).EQ.0.)C1=0.
      C3=(T(I+1,J)+T(I,J))/(DXS*2.)
      IF(VARXY)C3=C3*2.*DXS/(DDXX(I)*(DDXX(I-1)+DDXX(I)))
      IF(T(I+1,J).EQ.0.)C3=0.
      C2=C1+C3
      C4=S(I,J)/DT*2.
      C4=C4+B(I,J)
      C5=C4-2.*B(I,J)
      I6=(IK-2)*3
      A(I6+1)=-C4-C2
      A(I6+2)=C3
```

```
      IF(IK.GT.2)A(I6)=C1
      D1=(T(I,J-1)+T(I,J))/DYS
      IF(VARXY)D1=D1*2.*DYS/(DDYY(J-1)*(DDYY(J-1)+DDYY(J)))
      IF(T(I,J-1).EQ.0.)D1=0.
      D3=(T(I,J+1)+T(I,J))/DYS
      IF(VARXY)D3=D3*2.*DYS/(DDYY(J  )*(DDYY(J-1)+DDYY(J)))
      IF(T(I,J+1).EQ.0.)D3=0.
      D2=D1+D3
*
***********       TEST FOR SOURCE/SINK TERM AT NODE (I,J)      ***********
*
      IF(NF.EQ.0)GOTO 205
      DO 77 K=1,NF
      IF(F(K,1).EQ.FLOAT(I).AND.F(K,2).EQ.FLOAT(J))GO TO 88
      GOTO 77
   88 IF(TIME.GE.F(K,4).AND.TIME.LE.F(K,5))GO TO 99
   77 CONTINUE
  205 FT=HO(I,J)
      GOTO 1100
   99 FT=F(K,3)+HO(I,J)
 1100 IK=IK+1
   10 D(IK-2)=-C1*H(I-1,J)-C3*H(I+1,J)-D1*H(I,J-1)-D3*H(I,J+1)-FT+
     1H(I,J)*(C2+D2-C5)
      IF(VARXY)DXS=DDXX(IBE-1)*(DDXX(IBE-1)+DDXX(IBE))/2.
      D(1)=D(1)-H(IBE-1,J)*(T(IBE-1,J)+T(IBE,J))/(2.*DXS)
      D(IK-2 )=D(IK-2 )-C3*H(IEN+1,J)
      CALL MASOC8 (D,A,IK-2,1,1,1,5.E-7,IER)
      IO=1
      DO 11 I=IBE,IEN
      HH(I,J)=D(IO)
       IO=IO+1
   11 CONTINUE
 5010 CONTINUE
    9 CONTINUE
*****************************************************************
*
***********          ADIP,SWEEPING IN Y-DIRECTION          ***********
*
      DO 12  I=2,MX1
      IF(IY(I,1).EQ.0)GOTO 12
      IBNOR=IY(I,1)
      DO 5020 L=1,IBNOR
      IBE=IY(I,L*2)+1
      IEN=IY(I,L*2+1)-1
      IK=2
      DO 13 J=IBE,IEN
      C1=(T(I,J)+T(I,J-1))/(2.*DYS)
      IF(VARXY)C1=C1*2.*DYS/(DDYY(J-1)*(DDYY(J-1)+DDYY(J)))
      IF(T(I,J-1).EQ.0.)C1=0.
      C3=(T(I,J)+T(I,J+1))/(2.*DYS)
      IF(VARXY)C3=C3*2.*DYS/(DDYY(J  )*(DDYY(J-1)+DDYY(J)))
      IF(T(I,J+1).EQ.0.)C3=0.
      C2=C1+C3
      C4=S(I,J)*2./DT
      C4=C4+B(I,J)
      I6=(IK-2)*3
      A(I6+1)=-C4-C2
      A(I6+2)=C3
```

```
      IF(IK.GT.2)A(I6)=C1
      IK=IK+1
   13 D(IK-2)= C1*H(I,J-1)+H(I,J+1)*C3-H(I,J)*C2-C4*HH(I,J)
      IF(VARXY)DYS=DDYY(IBE-1)*(DDYY(IBE-1)+DDYY(IBE))/2.
      D(1)=D(1)-H(I,IBE-1)*(T(I,IBE-1)+T(I,IBE))/(2.*DYS)
      D(IK-2 )=D(IK-2 )-H(I,IEN+1)*C3
      CALL MAS008 (D,A,IK-2,1,1,1,5.E-7,IER)
      IO=1
      DO 14 J=IBE,IEN
      HH(I,J)=D(IC)
       IO=IO+1
   14 CONTINUE
 5020 CONTINUE
   12 CONTINUE
*****************************************************************
*
***********                TEST FOR STEADY STATE            ***********
*
      IF(VAR.LE.0.) GO TO 1008
      DO 1007 I=2,MX1
      DO 1007 J=2,MY1
      IF(ABS(HH(I,J)-H(I,J)).GT.ABS(VAR))GOTO 1008
 1007 CONTINUE
      FMIN=TMAX+1.E5
      IF(NF.EQ.0)GOTO 206
      DO 1009 I=1,NF
      IF(TIME.LT.F(I,4))FMIN=AMIN1(FMIN,F(I,4))
      IF(TIME.GT.F(I,4).AND.TIME.LT.F(I,5))FMIN=AMIN1(FMIN,F(I,5))
 1009 CONTINUE
206   WRITE(6,1011)
 1011 FORMAT(1H0,"STEADY STATE REACHED"/)
      IF(FMIN.GE.TMAX)GOTO 15
      DT=FMIN-TIME+1.E-5*DTBEG
      SW2=.TRUE.
*****************************************************************
 1008 DO 1013 I=1,MX
      DO 1013 J=1,MY
 1013 H(I,J)=HH(I,J)
1012  TIME=TIME+DT
      IF(DT.LT.1.E-15)GOTO 15
      WRITE(6,111)TIME,DT
      IF(NF.EQ.0)GOTO 207
      DO 5700 I=1,NF
      J=F(I,1)
      K=F(I,2)
      TRES(I)=H(J,K)
      IF(TIME.GE.F(I,4).AND.TIME.LT.F(I,5))H(J,K)=H(J,K)+F(I,6)
 5700 CONTINUE
*
*********** WRITE TIME,TIME-INTERVAL,AND HEAD MATRIX ON UNIT 1 *********
*********** WRITE SAME TO PRINTER-OUTPUT FILE                ***********
*                                                                      *
  111 FORMAT("0TOTAL TIME=",F15.5,"  INCREMENT=",F15.5)
207   WRITE(6,FM2)((H(I,J),I=I7,I8,I9),J=J7,J8,J9)
      IF(FRONT)
     1WRITE(1)TIME,DT,((H(I,J),I=1,MX),J=1,MY)
*****************************************************************
      IF(NF.EQ.0)GOTO 706
```

```
      DO 5701 I=1,NF
      J=F(I,1)
      K=F(I,2)
 5701 H(J,K)=TRES(I)
  706 IF(TIME.GT.TMAX)GO TO 15
*
*********** INCREMENT DT AND TEST TO SEE IF ANY SOURCE/SINK  ***********
*********** IS STARTING OR STOPPING DURING THAT TIME INTERVAL **********
*
      DT=DT*TINC
      IF(SW2)DT=DTBEG
      SW2=.FALSE.
      IBEG=1
      IF(NF.EQ.0)GOTO 16
  711 DO 709 I=IBEG,NF
      ITEST=-1
      IF(TIME.GE.F(I,4))ITEST=1
      IF(TIME.GT.F(I,5))ITEST=2
      TM2=TIME+DT
      JTEST=-1
      IF(TM2.GE.F(I,4))JTEST=1
      IF(TM2.GT.F(I,5))JTEST=2
      IF(JTEST.NE.ITEST)GO TO 710
  709 CONTINUE
      GO TO 16
  710 SW2=.TRUE.
      IF(ITEST.EQ.-1)GOTO 750
      DT=F(I,5)-TIME+DTBEG*1.E-5
      IBEG=I+1
      IF(IBEG.GT.NF)GOTO 16
      GO TO 711
  750 DT=F(I,4)-TIME+DTBEG*1.E-5
      IBEG=I+1
      IF(IBEG.GT.NF)GOTO 16
      GO TO 711
   15 IF(PCH)WRITE(7,110)((H(I,J),I=  M7,M8),J=K7,K8)
  110 FORMAT(5E15.7)
      DO 94 J=K7,K8
      DO 95 I=M7,M8
   95 IX(I,1)=FAC*H(I,J)
   94 WRITE(6,FM3)(IX(I,1),I=M7,M8)
      STOP
      END
```

```
***********          K3405DV  //// END OF LIST ////
***********          K3405DV  //// END OF LIST ////
```

Date Due
