DIGITAL MODEL FOR SIMULATING STEADY-STATE GROUND-WATER AND HEAT FLOW

By J. E. Reed

U. S. GEOLOGICAL SURVEY

Water-Resources Investigations Report 85-4248

Denver, Colorado

1985

CONTENTS

ILLUSTRATIONS

CONVERSION FACTORS

Multiply unit	By	To obtain metric unit
inch (in.)	25.4	millimeter (mm)
foot (ft)	0.3048	meter (m)
inch per year (in./yr)	8.049×10^{-7}	millimeter per second (mm/s)
cubic foot per day (ft^3/d)	3.277×10^{-7}	cubic meter per second (m^3/s)
degree Fahrenheit (^{o}F) - 32	0.5555	degree Celsius (^{o}C)
British thermal unit per day (BTU/d)	12.20	milliwatts (mW)
British thermal unit per cubic foot per degree Fahrenheit ($BTU/ft^3-^{o}F$)	67.02	joule per liter per degree Celsius ($J/L-^{o}C$)
British thermal unit per day per square foot per degree Fahrenheit ($BTU/d-ft^2-^{o}F$)	236.4	milliwatts per square meter per degree Celsius ($mW/m^2-^{o}C$)
heat flow unit (HFU); 10^{-6} calories per square centimeter per second	41.84	milliwatt per square meter (mW/m^2)
conductance unit (CU); 10^{-3} calories per centimeter per second per degree Celsius	418.4	milliwatt per meter per degree Cesius ($mW/m-^{o}C$)

DIGITAL MODEL FOR SIMULATING STEADY-STATE GROUND-WATER AND HEAT FLOW

By J. E. Reed

ABSTRACT

The computer program models steady-state water and heat flow in an isotropic, heterogeneous, three-dimensional aquifer system with uniform thermal properties and no change of state. Driving forces on the system are external hydrologic conditions of recharge from precipitation and fixed hydraulic-head boundaries. Heat flux includes geothermal heat-flow, conduction to the land surface, advection (heat conveyed by water) from recharge, and advection to or from fixed-head boundaries.

The program uses an iterative procedure that alternately solves the ground-water-flow and heat-flow equations, updating advective flux after solution of the ground-water-flow equation, and updating hydraulic conductivity after solution of the heat-flow equation. Direct solution is used for each equation.

Time of travel is determined by particle tracking through the modeled space. Velocities within blocks are linear interpolations of velocities at block faces.

Models of cross-sections display additional information along selected flow paths and for lithology within the section.

INTRODUCTION

This report documents the techniques used in the program HOTWTR to model the coupled, three-dimensional, steady-state flow of water and heat through isotropic heterogeneous porous media. This program was developed as an aid in screening large ground-water-flow systems as prospects for underground waste storage. The structure and content of the program reflect that task and the program is not intended to be a general purpose approach to a considerable range of ground-water problems.

Most of the techniques are variations of commonly accepted modeling practices. Sources for program concepts include Trescott, Pinder, and Larson (1976), Konikow and Bredehoeft (1978),and Faust and Mercer (1977). An earlier version of this model is discussed in Bedinger and others (1979). Only an outline of the mathematical development is included in this report. Discussion of alternative methods is included in places to justify the procedure selected. Readers unfamiliar with the concepts of finite-difference methods will find a comprehensive discussion of these methods in Bennett (1976, p. 119).

The program is based on separate finite-difference approximations of the ground-water-flow equation and of the convective-dispersive heat-flow equation. The ground-water-flow and heat-flow models are interdependent because hydraulic conductivity in the ground-water-flow model is a function of temperature and convective flow in the heat-flow model is a function of ground-water flow. The program iterates on the coupling between the two equations. The water- and heat-flow equations are solved alternately, with water flux through the block faces updated after solution of the ground-water-flow equation, and hydraulic conductivity updated after solution of the heat-flow equation. The variation of hydraulic conductivity with temperature is principally due to the change in viscosity, and, therefore, density can be assumed to be sensibly constant. The computer program also may be used as an isothermal model in which only the ground-water flow equation solved.

The program allows the hydraulic conductivity and porosity of the rock to be discretized in space, but the thermal conductivity of the fluid-saturated rock and the specific heat capacity of the fluid need to be assigned constant values. Also, for the program, the fluid is assumed to be of constant density. Hence, the program cannot be used to model free convection, which is one of two types of convection in porous media identified by Sorey (1978, p.D8). Forced convection results from stresses on the ground-water system. Free convection results from buoyancy effects caused by fluid-density variations due to temperature differences. Geothermal systems contain both types of convection (Sorey, p.D9). This model only considers forced convection, and is not applicable to systems where free convection is significant as indicated by geysers, steam, large temperature gradients or substantial heat flow.

Necessary data for this program consist of the geometric distribution of lithologies within the modeled space, the hydraulic conductivity and porosity for each lithologic type, the water level at recharge or discharge sites such as lakes or streams, recharge from infiltration of precipitation, temperature of recharged water, land-surface temperature, geothermal heat-flow, average thermal conductivity of the water-saturated rock, and volumetric specific heat of the fluid.

The ground-water-flow and heat-flow models are block-centered, finite-difference approximations with interblock hydraulic transmissivity calculated as the harmonic mean of the two adjacent blocks.

The finite-difference method considers the space of interest to be represented by a finite number of blocks. Variable properties of the flow system, such as hydraulic conductivity, are replaced in the model by their average value within each block. The average value for the variable to be calculated, hydraulic head or temperature, is assigned to a position called the node. Each node is enclosed by a subdomain of the flow system.

In a block-centered grid, the node is always centered between adjacent block faces and each flow subdomain is identical to a block. For point-distributed grids, each subdomain face is centered between adjacent nodes. Point-distributed grids with each subdomain identical to a block also are called face-centered grids.

For grids having uniform spacing in all directions, block-centered and face-centered grids are equivalent. If block size (therefore node spacing) is variable, the nodes in a face-centered grid will not be in the center of all blocks. In general, a face-centered grid with a minimum number of nodes cannot be designed because the block faces will not always coincide with natural boundaries in the modeled space. However, a block-centered grid can always be designed to match natural boundaries.

Point-distributed grids can be designed that match natural boundaries (R. L. Cooley, U.S. Geological Survey, oral commun., 1985). Such grids have nodes at block corners and flow subdomains include parts of four adjacent blocks. Equations developed for a point-distributed grid are a better approximation than are equations developed for a block-centered grid. However, velocity distribution in the flow subdomain could be more complex if the subdomain includes blocks of differing hydraulic conductivity or porosity. Examples of these grids are illustrated in Figure 1.

Subscript notation is used here to reference a quantity's position in the finite-difference grid. Integer subscripts refer to blocks or to nodes within blocks, whereas fractional-half subscripts refer to block faces. For example, in reference to node i, j, and k, where i, j, and k are integer numbers, the head (h) at node i, j, k is denoted by $h_{i,j,k}$, the head at the adjacent node in the i increasing direction is denoted by $h_{i+1,j,k}$, the specific discharge, q, at the block face in the i increasing direction is denoted by $q_{i+1/2,j,k}$, and so forth. The coordinate axes of a finite-difference block are illustrated in Figure 2.

Dimensions for coefficients in equations are given in general units, not in specific units. These abbreviations (and their units) are L (length), T (time), E (energy), and Deg. (temperature). Energy is used here as a general unit although energy has dimensions of M*L**2/T**2.

```
+-----+-----------+--------------------+
|     |===========|                    |
|  *  |=====*=====|         *          |
|     |===========|                    |
+-----+-----------+--------------------+
|     |           |                    |
|     |           |                    |
|  *  |     *     |         *          |
|     |           |                    |
|     |           |                    |
+-----+-----------+--------------------+
```

Block-centered grid

```
+-----+-----------+--------------------+
|     |===========|                    |
|  *  |==*========|        *           |
|     |===========|                    |
+-----+-----------+--------------------+
|     |           |                    |
|  *  |  *        |        *           |
|     |           |                    |
|     |           |                    |
|     |           |                    |
+-----+-----------+--------------------+
```

Face-centered grid

```
*-----*-----------*--------------------*
|     |           |                    |
|     |     +---------------+          |
|     |     |===============|          |
*-----*-----|=====*=========|----------*
|     |     |===============|          |
|     |     |===============|          |
|     |     +---------------+          |
|     |           |                    |
|     |           |                    |
*-----*-----------*--------------------*
```

Cooley's point-distributed grid

Explanation

```
*       Nodes

|
| , --- Block boundaries
|

+--+
|==|    One selected flow subdomain
+--+
```

Figure 1.-Three types of finite-difference grids.

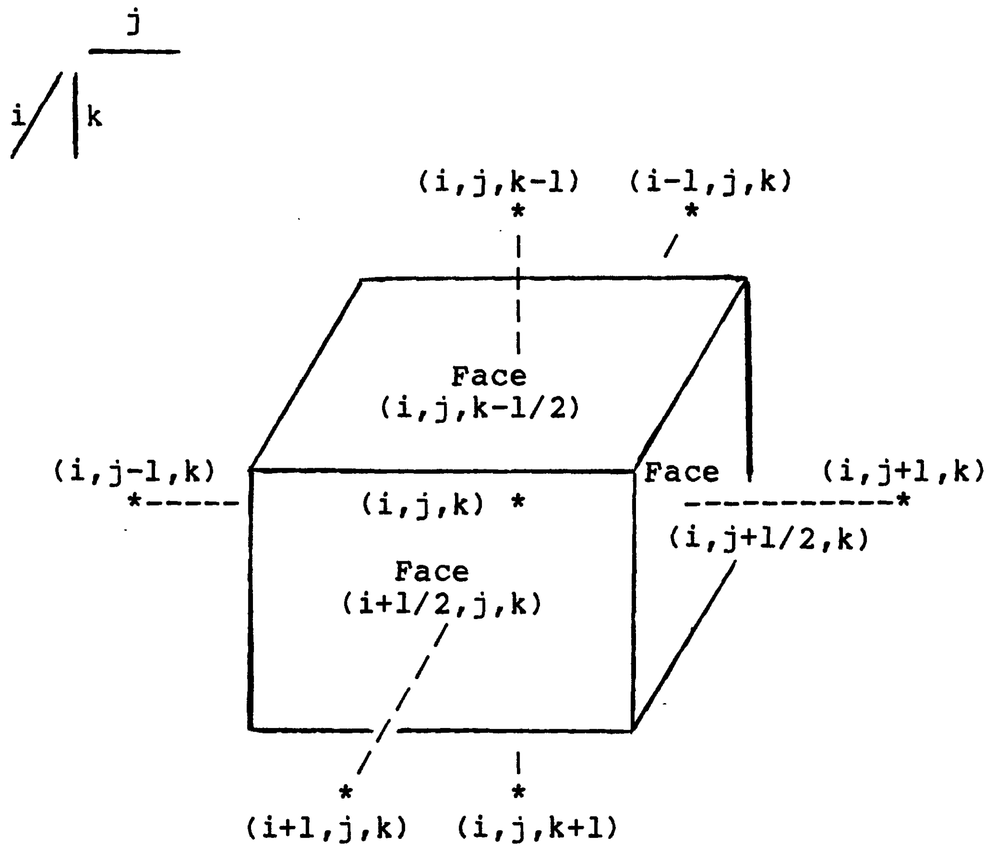


Figure 2.--Finite-difference block for three-dimensional ground-water flow. Location indices for the three orthogonal corrdinates are i,j.k. Nodes are indicated by an asterisk.

GROUND-WATER-FLOW MODEL

The equation of steady-state water flow in an isotropic aquifer for three dimensions can be written as

$$\frac{\partial\left(K\frac{\partial h}{\partial y}\right)}{\partial y}+\frac{\partial\left(K\frac{\partial h}{\partial x}\right)}{\partial x}+\frac{\partial\left(K\frac{\partial h}{\partial z}\right)}{\partial z}+Ws=0 \qquad (1)$$

where

y, x, z are the three orthogonal coordinate directions (L),

K is the hydraulic conductivity (L/T),

h is the hydraulic head or water-level elevation (L), and

Ws is the recharge per unit depth (1/T).

A finite-difference approximation to equation 1 can be written as

$$\frac{\dfrac{K_{i+1/2,j,k}(h_{i+1,j,k} - h_{i,j,k})}{(Dy_{i+1} + Dy_{i})/2} - \dfrac{K_{i-1/2,j,k}(h_{i,j,k} - h_{i-1,j,k})}{(Dy_{i} + Dy_{i-1})/2}}{Dy_{i}}$$

$$+ \frac{\dfrac{K_{i,j+1/2,k}(h_{i,j+1,k} - h_{i,j,k})}{(Dx_{j+1} + Dx_{j})/2} - \dfrac{K_{i,j-1/2,k}(h_{i,j,k} - h_{i,j-1,k})}{(Dx_{j} + Dx_{j-1})/2}}{Dx_{j}}$$

$$+ \frac{\dfrac{K_{i,j,k+1/2}(h_{i,j,k+1} - h_{i,j,k})}{(Dz_{k+1} + Dz_{k})/2} - \dfrac{K_{i,j,k-1/2}(h_{i,j,k} - h_{i,j,k-1})}{(Dz_{k} + Dz_{k-1})/2}}{Dz_{k}}$$

$$+ Ws_{i,j,k} = 0 \qquad (2)$$

In equation 2, the subscripts represent position in the finite-difference grid,

i,j,k indicates a specific node or block,

i-1 and i+1 indicate the preceding and succeeding nodes or blocks in the y direction,

j-1,j+1, k-1, and k+1 are likewise for the x and z directions respectively,

i-1/2 and i+1/2 indicate preceding and succeeding block faces in the y direction,

j-1/2, j+1/2, k-1/2, and k+1/2 indicate likewise for the x and z directions, respectively,

Dy, Dx, and Dz are block lengths in the y,x, and z directions, respectively,

and other symbols are as defined previously. The above finite-difference approximation applied to a region of interest results in a system of n (number of blocks) linear equations for n unknown hydraulic heads.

The hydraulic conductivity at the block face is the thickness-weighted harmonic mean of the hydraulic conductivities for the adjacent blocks. For the y direction this is

$$K_{i-1/2,j,k} = \frac{K_{i-1,j,k} K_{i,j,k} (Dy_i + Dy_{i-1})}{K_{i-1,j,k} Dy_i + K_{i,j,k} Dy_{i-1}}$$

and

$$K_{i+1/2,j,k} = \frac{K_{i+1,j,k} K_{i,j,k} (Dy_i + Dy_{i+1})}{K_{i+1,j,k} Dy_i + K_{i,j,k} Dy_{i+1}}$$

Similar expressions may be developed for the hydraulic conductivity at block faces in the x and z directions.

The finite-difference equation for a block multiplied by the volume of the block ($Dy_i * Dx_j * Dz_k$) becomes a balance equation with each term representing a flow into or out of the block. The balance equation requires that total inflow to the block must equal total outflow. Equation 2 becomes

$$\left[\frac{K_{i+1/2,j,k}(h_{i+1,j,k} - h_{i,j,k})}{(Dy_{i+1} + Dy_i)/2} - \frac{K_{i-1/2,j,k}(h_{i,j,k} - h_{i-1,j,k})}{(Dy_i + Dy_{i-1})/2}\right] Dx_j Dz_k$$

$$+\left[\frac{K_{i,j+1/2,k}(h_{i,j+1,k} - h_{i,j,k})}{(Dx_{j+1} + Dx_j)/2} - \frac{K_{i,j-1/2,k}(h_{i,j,k} - h_{i,j-1,k})}{(Dx_j + Dx_{j-1})/2}\right] Dy_i Dz_k$$

$$+\left[\frac{K_{i,j,k+1/2}(h_{i,j,k+1} - h_{i,j,k})}{(Dz_{k+1} + Dz_k)/2} - \frac{K_{i,j,k-1/2}(h_{i,j,k} - h_{i,j,k-1})}{(Dz_k + Dz_{k-1})/2}\right] Dy_i Dx_j$$

$$+ Ws_{i,j,k} Dy_i Dx_j Dz_k = 0 \qquad (3)$$

where all terms are as previously defined.

Collecting terms in equation 3 gives the form

$$A_{i,j,k} h_{i-1,j,k} + B_{i,j,k} h_{i,j-1,k} + C_{i,j,k} h_{i,j,k-1} + D_{i,j,k} h_{i,j,k}$$

$$+ E_{i,j,k} h_{i,j,k+1} + F_{i,j,k} h_{i,j+1,k} + G_{i,j,k} h_{i+1,j,k}$$

$$+ Ws_{i,j,k} Dy_i Dx_j Dz_k = 0. \qquad (4)$$

The terms in equation 4 are:

$$A_{i,j,k} = \frac{K_{i-1/2,j,k}\, Dx_j\, Dz_k}{(Dy_i + Dy_{i-1})/2},$$

$$B_{i,j,k} = \frac{K_{i,j-1/2,k}\, Dy_i\, Dz_k}{(Dx_j + Dx_{j-1})/2},$$

$$C_{i,j,k} = \frac{K_{i,j,k-1/2}\, Dy_i\, Dx_j}{(Dz_k + Dz_{k-1})/2},$$

$$E_{i,j,k} = \frac{K_{i,j,k+1/2}\, Dy_i\, Dx_j}{(Dz_{k+1} + Dz_k)/2},$$

$$F_{i,j,k} = \frac{K_{i,j+1/2,k}\, Dy_i\, Dz_k}{(Dx_{j+1} + Dx_j)/2},$$

$$G_{i,j,k} = \frac{K_{i+1/2,j,k}\, Dx_j\, Dz_k}{(Dy_{i+1} + Dy_i)/2},$$

and

$$D_{i,j,k} = -(A_{i,j,k} + B_{i,j,k} + C_{i,j,k} + E_{i,j,k} + F_{i,j,k} + G_{i,j,k}).$$

Equation 4 can be written for each of the n active blocks in the model resulting in a system of n equations for n unknown hydraulic heads. This system, in matrix notation, is

$$[B]\{h\}=\{-(W+H)\}$$

where B is the matrix of coefficients of the hydraulic heads in the system of equations, h is the vector (n by 1 matrix) of unknown hydraulic heads, -(W+H) is a vector of known terms, formed by moving the recharge and constant hydraulic-head terms to the right side of the equations.

In residual form (Trescott and others, 1976, p.16, eq. 17d) the system becomes

$$[B]\{hi+Dh\}=\{-(W+H)\}$$

where h is separated into hi, the initial hydraulic head, and Dh, the computed change in hydraulic head. The residual form decreases rounding errors in the computation when change in hydraulic head is smaller in magnitude than hydraulic head. Transfering the known terms to the right side gives

$$[B]\{Dh\}=-([B]\{hi\}+\{W+H\})$$

or

$$[B]\{Dh\}=\{R\}$$

where R is a vector composed of all known terms.

HEAT-FLOW MODEL

The equation for steady-state, convective-diffusive heat flow in three dimensions for conditions of uniform thermal conductivity and specific heat may be written

$$Kt\left(\frac{\partial^2 t}{\partial y^2} + \frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial z^2}\right) + St\left[\frac{\partial (qt)}{\partial y} + \frac{\partial (qt)}{\partial x} + \frac{\partial (qt)}{\partial z}\right] + Ht = 0 \qquad (5)$$

where

Kt is the thermal conductivity of the rock saturated with water, [E/(L*T*Deg.)],

t is the temperature (degrees on some scale),

y, x, z are the three orthogonal coordinates (L),

St is the volumetric specific heat for the fluid [E/(L**3*Deg.)],

q is the specific discharge, flow per unit area, of ground water at a point (L/T),

and Ht is a heat source term which includes the geothermal heat flow per unit depth [E/(L**3*T)].

A finite-difference approximation of equation 5 can be written for a block numbered i,j,k as

$$Kt\left[\frac{\dfrac{t_{i+1,j,k}-t_{i,j,k}}{(Dy_{i+1}+Dy_{i})/2}-\dfrac{t_{i,j,k}-t_{i-1,j,k}}{(Dy_{i}+Dy_{i-1})/2}}{Dy_{i}}\right.$$

$$+\frac{\dfrac{t_{i,j+1,k}-t_{i,j,k}}{(Dx_{j+1}+Dx_{j})/2}-\dfrac{t_{i,j,k}-t_{i,j-1,k}}{(Dx_{j}+Dx_{j-1})/2}}{Dx_{j}}$$

$$\left.+\frac{\dfrac{t_{i,j,k+1}-t_{i,j,k}}{(Dz_{k+1}+Dz_{k})/2}-\dfrac{t_{i,j,k}-t_{i,j,k-1}}{(Dz_{k}+Dz_{k-1})/2}}{Dz_{k}}\right]$$

$$+\ St\left(\frac{q_{i+1/2,j,k}\,t_{i+1/2,j,k}-q_{i-1/2,j,k}\,t_{i-1/2,j,k}}{Dy_{i}}\right.$$

$$+\frac{q_{i,j+1/2,k}\,t_{i,j+1/2,k}-q_{i,j-1/2,k}\,t_{i,j-1/2,k}}{Dx_{j}}$$

$$\left.+\frac{q_{i,j,k+1/2}\,t_{i,j,k+1/2}-q_{i,j,k-1/2}\,t_{i,j,k-1/2}}{Dz_{k}}\right)+Ht_{i,j,k}=0. \quad (6)$$

In equation 6, the subscript notation is as discussed previously and t, Kt, St, q, and Ht are as defined above.

The specific discharge, q, is calculated for the y direction as

$$q_{i-1/2,j,k} = \frac{K_{i-1/2,j,k}(h_{i,j,k} - h_{i-1,j,k})}{(Dy_{i} + Dy_{i-1})/2}$$

and

$$q_{i+1/2,j,k} = \frac{K_{i+1/2,j,k}(h_{i,j,k} - h_{i+1,j,k})}{(Dy_{i} + Dy_{i+1})/2} .$$

Similar expressions may be developed for the specific discharge at block faces in the x and z directions.

The effective temperature for convective flow by upgradient weighting is

$$t_{i+1/2,j,k} = t_{i,j,k} \quad \text{if } q_{i+1/2,j,k} > 0$$

or

$$t_{i+1/2,j,k} = t_{i+1,j,k} \quad \text{if } q_{i+1/2,j,k} < 0$$

and

$$t_{i-1/2,j,k} = t_{i-1,j,k} \quad \text{if } q_{i-1/2,j,k} > 0$$

or

$$t_{i-1/2,j,k} = t_{i,j,k} \quad \text{if } q_{i-1/2,j,k} < 0$$

and similarly for the other two directions. Upgradient weighting for convective heat transport assigns face temperature to the "upgradient" node. For central weighting, face temperature is computed as an average for the two adjacent nodes. Central weighting is more accurate for small grid spacing than upgradient weighting, but produces oscillatory, unstable solutions when the grid spacing is large. Upgradient weighting provides a stable solution for large grid spacing.

Multiplying the finite-difference equation, equation 6, by the volume of the block results in a balance equation for heat flow which is:

$$Kt[(\frac{t_{i+1,j,k} - t_{i,j,k}}{(Dy_{i+1} + Dy_i)/2} - \frac{t_{i,j,k} - t_{i-1,j,k}}{(Dy_i + Dy_{i-1})/2})Dx_j\ Dz_k$$

$$+ (\frac{t_{i,j+1,k} - t_{i,j,k}}{(Dx_{j+1} + Dx_j)/2} - \frac{t_{i,j,k} - t_{i,j-1,k}}{(Dx_j + Dx_{j-1})/2})Dy_i\ Dz_k$$

$$+ (\frac{t_{i,j,k+1} - t_{i,j,k}}{(Dz_{k+1} + Dz_k)/2} - \frac{t_{i,j,k} - t_{i,j,k-1}}{(Dz_k + Dz_{k-1})/2})Dx_j\ Dy_i\ ]$$

$$+ St[(q_{i+1/2,j,k}\ t_{i+1/2,j,k} - q_{i-1/2,j,k}\ t_{i-1/2,j,k})Dx_j\ Dz_k$$

$$+ (q_{i,j+1/2,k}\ t_{i,j+1/2,k} - q_{i,j-1/2,k}\ t_{i,j-1/2,k})Dy_i\ Dz_k$$

$$+ (q_{i,j,k+1/2}\ t_{i,j,k+1/2} - q_{i,j,k-1/2}\ t_{i,j,k-1/2})Dy_i\ Dx_j\ ]$$

$$+ Ht_{i,j,k}\ Dy_i\ Dx_j\ Dz_k = 0. \qquad (7)$$

Each term in the preceding balance equation is a heat flow (dimensions E/T) into or out of the block, with total inflow equal to total outflow.

Collecting terms in equation 7 gives the form

$$A_{i,j,k}\, t_{i-1,j,k} + B_{i,j,k}\, t_{i,j-1,k} + C_{i,j,k}\, t_{i,j,k-1} + D_{i,j,k}\, t_{i,j,k}$$

$$+ E_{i,j,k}\, t_{i,j,k+1} + F_{i,j,k}\, t_{i,j+1,k} + G_{i,j,k}\, t_{i+1,j,k}$$

$$+ Ht_{i,j,k}\, Dy_i\, Dx_j\, Dz_k = 0 \tag{8}$$

with

$$A_{i,j,k} = \frac{Kt\, Dx_j\, Dz_k}{(Dy_i + Dy_{i-1})/2} - St\, q_{i-1/2,j,k}\, Dx_j\, Dz_k$$

if $q_{i-1/2,j,k} > 0$

or

$$A_{i,j,k} = \frac{Kt\, Dx_j\, Dz_k}{(Dy_i + Dy_{i-1})/2}$$

if $q_{i-1/2,j,k} < 0$

and

$$B_{i,j,k} = \frac{Kt\, Dy_i\, Dz_k}{(Dx_j + Dx_{j-1})/2} - St\, q_{i,j-1/2,k}\, Dy_i\, Dz_k$$

if $q_{i,j-1/2,k} > 0$

or

$$B_{i,j,k} = \frac{Kt\, Dy_i\, Dz_k}{(Dx_j + Dx_{j-1})/2}$$

if $q_{i,j-1/2,k} < 0$.

Also,

$$C_{i,j,k} = \frac{Kt\ Dy_i\ Dx_j}{(Dz_k + Dz_{k-1})/2} - St\ q_{i,j,k-1/2}\ Dy_i\ Dx_j$$

if $q_{i,j,k-1/2} > 0$

or

$$C_{i,j,k} = \frac{Kt\ Dy_i\ Dx_j}{(Dz_k + Dz_{k-1})/2}$$

if $q_{i,j,k-1/2} < 0$

and

$$E_{i,j,k} = \frac{Kt\ Dy_i\ Dx_j}{(Dz_k + Dz_{k+1})/2}$$

if $q_{i,j,k+1/2} > 0$

or

$$E_{i,j,k} = \frac{Kt\ Dy_i\ Dx_j}{(Dz_k + Dz_{k+1})/2} - St\ q_{i,j,k+1/2}\ Dy_i\ Dx_j$$

if $q_{i,j,k+1/2} < 0$.

Also,

$$F_{i,j,k} = \frac{Kt\ Dy_i\ Dz_k}{(Dx_j + Dx_{j+1})/2}$$

if $q_{i,j+1/2,k} > 0$

or

$$F_{i,j,k} = \frac{Kt\ Dy_i\ Dz_k}{(Dx_j + Dx_{j+1})/2} - St\ q_{i,j+1/2,k}\ Dy_i\ Dz_k$$

if $q_{i,j+1/2,k} < 0$

and

$$G_{i,j,k} = \frac{Kt\ Dx_j\ Dz_k}{(Dy_i + Dy_{i+1})/2}$$

if $q_{i+1/2,j,k} > 0$

or

$$G_{i,j,k} = \frac{Kt\ Dx_j\ Dz_k}{(Dy_i + Dy_{i+1})/2} - St\ q_{i+1/2,j,k}\ Dx_j\ Dz_k$$

if $q_{i+1/2,j,k} < 0$

and

$D_{i,j,k}$ =-(sum of the alternate terms for $A_{i,j,k}$, $B_{i,j,k}$, $C_{i,j,k}$, $E_{i,j,k}$, $F_{i,j,k}$, and $G_{i,j,k}$ with siqn of q terms reversed if present).

Equation 8 can be written for each of the n active blocks in the model, resulting in a system of n equations for n unknown temperatures.

This system, in matrix notation, is

$$[B]\{t\}=\{-(H'+T')\}$$

where B is the matrix of coefficients of the temperatures in the system of equations, t is the vector of unknown temperatures, -(H'+T') is a vector of known terms, formed by moving the heat flow, H', and constant temperature, T', terms to the right side of the equations. This system in residual form is

$$[B]\{Dt\}=-([B]\{ti\}+\{H'+T'\})$$

or

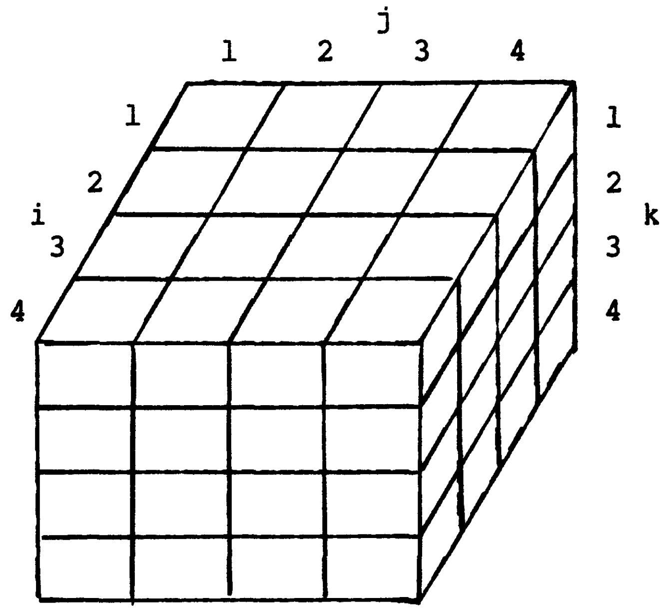
$$[B]\{Dt\}=\{R\}$$

where Dt is the change in temperature, ti is the initial temperature, and R is the vector of known terms.

SOLUTION PROCEDURE

The computer program numbers the balance equations alternately, proceeding in the order of the smallest dimensions of the model grid. An example of the equation numbering is shown in Figure 3. This produces an equation coefficent matrix with the form shown in Figure 4. This matrix is similar to that produced by the "D4" ordering of Larson (1978). However, the ordering, when applied to two dimensions, is less efficient than the "D4" ordering.

Coefficient matrices for both water and heat balances have the form shown in figure 4. Of course, the coefficients have different values. In particular, the matrix for water balance is symmetric, whereas the matrix for heat balance is not. The same procedure is used to solve both.



k=1

i \ j	1	2	3	4
1	1	35	5	39
2	41	11	45	15
3	17	51	21	55
4	57	27	61	31

k=2

i \ j	1	2	3	4
1	33	3	37	7
2	9	43	13	47
3	49	19	53	23
4	25	59	29	63

k=3

i \ j	1	2	3	4
1	2	36	6	40
2	42	12	46	16
3	18	52	22	56
4	58	28	62	32

k=4

i \ j	1	2	3	4
1	34	4	38	8
2	10	44	14	48
3	50	20	54	24
4	26	60	30	64

Figure 3.--Equation numbering for a 4 by 4 by by 4 grid

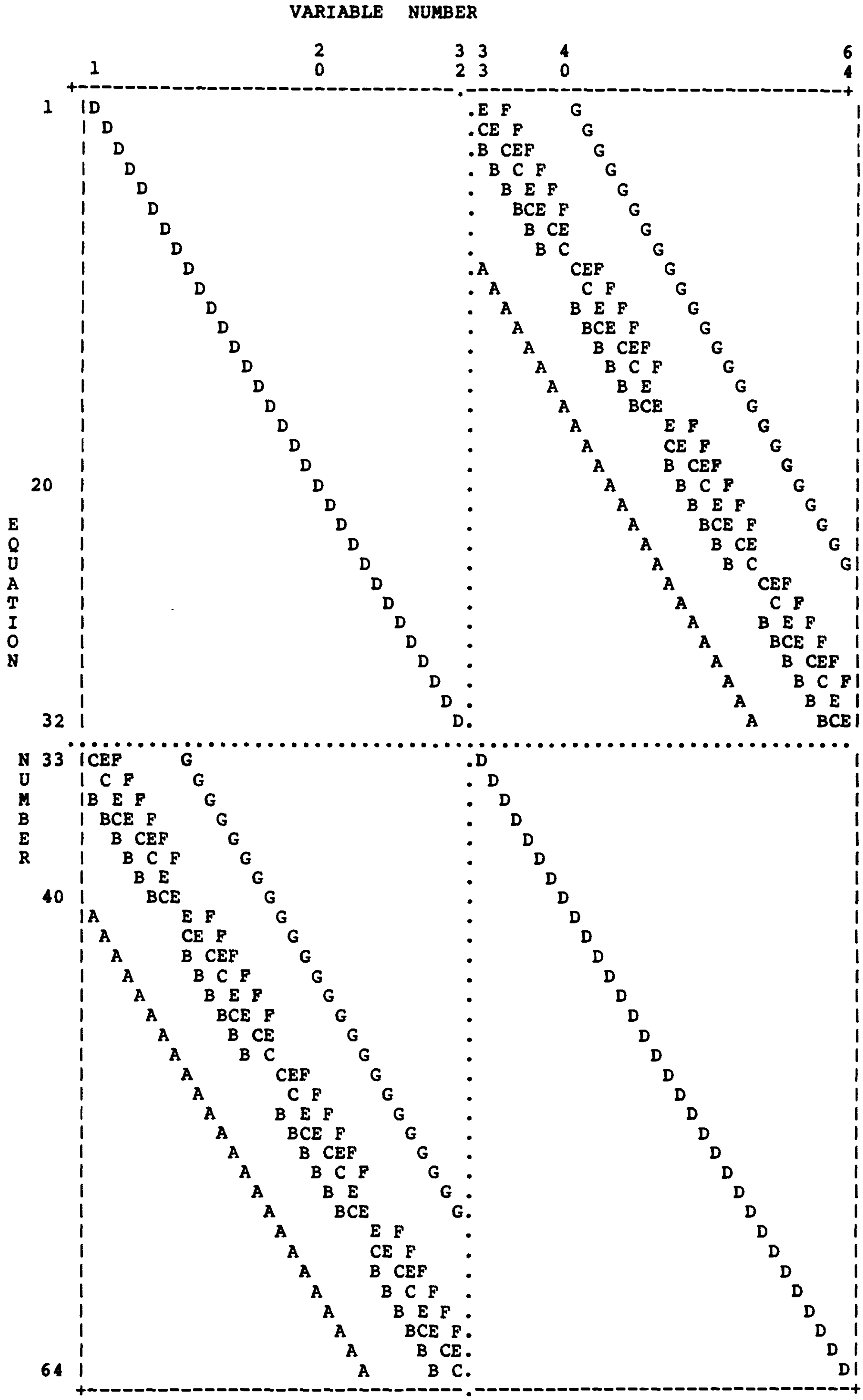


Figure 4.--Structure of coefficient matrix for a 4 by 4 by 4 grid. Symbols A,B,C,D,E,F, and G refer to coefficients in equations 4 and 8. Blanks indicate zero coefficients. Dotted lines indicate the partitioning of the matrix.

The coefficient matrix is partitioned into upper and lower, and further into left and right submatrices (fig. 4)

$$[B] = \frac{[DU] \mid [AU]}{[AL] \mid [DL]}$$

The two submatrices, DU and DL, are both diagonal matrices (terms off the main diagonal are zero). The unknown vector, U, and known vector, R, are divided by the upper-lower partitioning into two vectors, called UU and UL,and RU and RL, respectively. Eliminating terms in AL by adding multiples of DU gives

$$[B']\{U\}=\{R'\}$$

where R' consists of RU and RL' and

$$[B'] = \frac{[DU] \mid [AU]}{[0] \mid [A]}$$

and 0 is a submatrix consisting entirely of zeros. The matrix, A, is formed by the addition of multiples of AU to DL; likewise, the vector RL' is formed by the addition of the same multiples of RU to RL. This elimination is similar to that of Larson (1978, p. 3 and fig. 2).

The lower partition of the equation now is

$$[A]\{UL\}=\{RL'\}$$

which may be solved for UL by the Gauss algorithm (Kreyszig, 1972, p. 235). Elimination of terms below the main diagonal in A results in

$$[A']\{UL\}=\{RL''\}$$

where A' is an upper triangular matrix and RL'' is a new known-vector produced by the elimination. UL is then computed by back substitution.

The remaining equations may be expressed as

$$[DU]\{UU\}=\{RU\}-[AU]*UL$$

and, because DU is a diagonal matrix, each equation has only one unknown and UU may be calculated directly.

TRAVEL TIME

Travel time is computed in the code by moving a particle of water through the model. The velocity at a block face is

$$v_{i-1/2,j,k} = \frac{q_{i-1/2,j,k}}{p_{i,j,k}}$$

and

$$v_{i+1/2,j,k} = \frac{q_{i+1/2,j,k}}{p_{i,j,k}}$$

and similarly for the other two directions, where the subscripts indicate location as defined previously,

v is the velocity (L/T),

q is the specific discharge (L/T), and

p is the porosity (dimensionless).

The velocity in a given direction within a block is estimated as a linear interpolation between the velocities at the two opposing faces. For example,

$$vy = ay+by*y$$

where vy is the velocity in the y direction, and ay and by are constants defining the linear function. Using a local coordinate system for each block with y =0 at face i-1/2,j,k and y = Dy_i at face i+1/2,j,k the two constants are

$$ay = v_{i-1/2,j,k}$$

and

$$by = \frac{v_{i+1/2,j,k} - v_{i-1/2,j,k}}{Dy_i}$$

Velocities are computed in a similar manner for the other two directions.

The time, $Dty_{i,j,k}$, that it takes to travel from a point within a block, yo, to a point on a block face, yf, along the component of movement in the y direction is given by

$$Dty_{i,j,k} = \int_{yo}^{vf} \frac{1}{ay + by*y}\, dy$$

where yo is the lower limit of integration, yf is the upper limit, 1/(ay+by*y) is the integrand, dy is the differential of y, and ay+by*y is the linear velocity function. Performing the integration

$$Dty_{i,j,k} = \frac{1}{by} \ln\frac{ay + by*yf}{ay + by*yo} \qquad (9)$$

where ln is the natural (base e) logarithm. Similar development results in similar expressions for the x and z directions.

The exit point (xf,yf, zf) from the block is unknown. Substituting the maximum possible values for xf,yf, and xf into equation 9 results, in general, in three different travel times. The flow path intersects the face in the direction of the coordinate associated with the minimum time. The other two coordinates of the exit point may be determined by substituting the minimum time into equation 9.

The travel time across a block is the minimum of the travel times obtained by applying eq. 9 in the three coordinate directions. The total travel time along a flow path through the model is the sum of the travel times through the blocks traversed by the flow path.

COMPUTER PROGRAM

The program, written in Fortran 77, consists of the main routine, HOTWTR, and 27 subroutines. The code is listed in attachment 1. Definition of selected program variables is in attachment 2. Description of the input data is given in attachment 3. An outline of the output data is given in attachment 4. An example simulation showing input data and selected parts of the PRINT output data is in attachment 5. A generalized flow chart for the program is in attachment 6. All attachments are at the back of the report. A tabulation of the subroutines follows:

Name	Called from	Description
BACK	HOTWTR	Back substitutes to determine upper part of unknowns, adds change to old value, and determines largest change.
BALM	HOTWTR	Computes heat and water balance.
COMPQ	HOTWTR	Computes interblock ground-water flow.
DATAIN	HOTWTR	Reads input data.
DREAD	DATAIN	Reads double-precision arrays.
HEATEQ	HOTWTR	Computes coefficient matrix and known vector for the heat equation.
MOVE	HOTWTR	Determines time of travel and point of discharge for each active node.
MSUB	MOVE and PATH	Tracks water particle through the model.
MSUB1	MSUB	Calculates position, velocity, and travel time to block face in direction of movement.

Name	Called from	Description
MSUB2	MSUB	Determines new coordinates.
NUMBER	DATAIN	Numbers equations and sets pointers from model grid to equation.
PA	HOTWTR	Adjusts hydraulic conductivity for temperature change.
PARTN	HOTWTR	Sets pointers from equation to model grid.
PATH	HOTWTR	Traces flow paths from top of model to discharge point.
PLINE1	PRINT	Constructs one line of flow-path cross-section.
PLINE2	PRINT	Constructs one line of head or travel-time cross-section.
PLINE4	PRINT	Constructs one line of lithology cross-section.
PRINT	HOTWTR	Constructs cross-sections of flow paths, head, travel time, and lithology.
PSUB	MSUB	Saves flow-path coordinates.
READ	DATAIN	Reads single-precision arrays.
SCALEH	PRINT	Scales head values.
SCALET	PRINT	Scales time of travel.
SOLVE	HOTWTR	Reduces the A matrix to A' and back substitutes to determine the lower part of the unknowns.
SORT	HOTWTR	Sorts flow-path coordinates.
VELO	HOTWTR	Computes velocities.
WATEQ	HOTWTR	Computes the coefficient matrix and the known vector for the ground-water equations.
ZER	MSUB2 and PATH	Locates limiting points near points of zero velocity.

Size of the A array (253000) permits a 60*60 two-dimensional or a 12*12*12 three-dimensional model. Storage requirements for the A array can be estimated as (mn*2+1)*(mnl/2) where mn is the product of the two smallest of M, N, and L, and mnl is the product of all three.

REFERENCES

Bedinger, M.S., Pearson, F.J., Jr., Reed, J.E., Sniegocki, R.T., and Stone, C.G., 1979, The waters of Hot Springs National Park, Arkansas--Their nature and origin: U.S. Geological Survey Professional Paper 1044-C, 33 p.

Bennett, G.D., 1976, Introduction to ground-water hydraulics: U.S. Geological Survey Techniques of Water Resources Investigations, Book 3, Chapter B2, 172 p.

Faust, C. W., and Mercer, J. W., 1977, Finite difference model of two-dimensional single-, and two-phase heat transport in a porous medium--Version I: U.S. Geological Survey Open-File Report 77-234, 84 p.

Konikow, L.F., and Bredehoeft, J.D., 1978, Computer model of two-dimensional solute transport and dispersion in ground water: U.S. Geological Survey Techniques of Water Resources Investigations, Book 7, Chapter C2, 90 p.

Kreyszig, Erwin, 1972, Advanced engineering mathematics: New York, John Wiley and Sons, 888 p.

Larson, S.P., 1978, Direct solution algorithm for the two dimensional ground-water flow model: U.S. Geological Survey Open-File Report 79-202, 22 p.

Lohman, S.W., and others, 1972, Definitions of selected ground-water terms-revisions and conceptual refinements: U.S. Geological Survey Water-Supply Paper 1988, 21 p.

Sorey, M.L., 1978, Numerical modeling of liquid geothermal systems: U.S. Geological Survey Professional Paper 1044-D, 25 p.

Trescott, P.D., Pinder, G.F., and Larson, S.P., 1976, Finite-difference model for aquifer simulation in two dimensions, with results of numerical experiments: U.S. Geological Survey Techniques of Water Resources Investigations, Book 7, Chapter C1, 116 p.

Attachment 1. Program listing

Attachment 1.

```
      PROGRAM HOTWTR
      LOGICAL LFIN,LSTOP,LHCPT,LVPT
      CHARACTER G8
      CHARACTER*12 FILE,TEMP
      CHARACTER*18 FILE1,FILE2,FILE3,FILE4,FILE5,FILE6
      REAL KT
      REAL*8 A
      CHARACTER*8 XMESUR,ZMESUR
      COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
     c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
     c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
      COMMON//A(253000)
      COMMON/G1/G1(32768)
      COMMON/G2/G2(32768)
      COMMON/G3/G3(32768)
      COMMON/G4/G4(32768)
      COMMON/G5/G5(32768)
      COMMON/G6/G6(32768)
      COMMON/G7/G7(32768)
      COMMON/G8/G8(131072)
      DATA LREC,LSTOP/4,.FALSE./
      LFIN=.FALSE.
    1 WRITE(1,7)
      READ(1,8)FILE
    2 IBP=INDEX(FILE,' ')
      IF(IBP.EQ.0)THEN
           IBP=13
      ELSE IF(IBP.EQ.1)THEN
           IF(FILE.EQ.'            ')THEN
                WRITE(1,9)
                GOTO1
           ENDIF
           TEMP=FILE
           FILE=TEMP(2:12)
           GOTO2
      ENDIF
      FILE1=FILE(1:IBP-1)//'.INPUT'
      FILE2=FILE(1:IBP-1)//'.BAL'
      FILE3=FILE(1:IBP-1)//'.VELO'
      FILE4=FILE(1:IBP-1)//'.MOVE'
      FILE5=FILE(1:IBP-1)//'.PATH'
      FILE6=FILE(1:IBP-1)//'.PRINT'
      OPEN(5,FILE=FILE)
      READ(5,10) M,N,L,MAX,IREF,LHCPT,LVPT
      MN=M*N
      LMN=L*MN
C     SET POINTERS FOR G1 ARRAY
      IHD=1
      ITMP=IHD+LMN*2
      IDX=ITMP+LMN*2
      IDY=IDX+N
      IDZ=IDY+M
```

```
      IDISTX=IDZ+L
      IDISTZ=IDISTX+N
      IEND1=IDISTZ+L-1
C     SET POINTERS FOR G2 ARRAY
      IIH=1
      IIT=IIH+LMN
      INP=IIT+LMN
      IIR=INP+LMN
C     SET POINTERS FOR G4 ARRAY
      IHC=1
      IP=IHC+LMN
      IPOR=IP+LMN
      IRCH=IPOR+LMN
      IRJ=IRCH+MN
      IDP=IRJ+N*3
      IHF=IDP+MN
      IEND4=IHF+MN-1
C     SET POINTERS FOR G7 ARRAY
      IVX1=1
      IVX2=IVX1+LMN
      IVY1=IVX2+LMN
      IVY2=IVY1+LMN
      IVZ1=IVY2+LMN
      IVZ2=IVZ1+LMN
      IEND7=IVZ2+LMN-1
C     SET POINTERS FOR G8 ARRAY
      ISEC=1
      IF(M.GT.1)THEN
           ITIM=1
      ELSE
           ITIM=ISEC+LMN+MOD(LMN,4)
      ENDIF
      IIDS=ITIM+4*LMN
      IPATH=IIDS+4*LMN
      IF(IEND1.GT.32768)THEN
           WRITE(1,11)IEND1
           LSTOP=.TRUE.
      END IF
      IF(IIR.GT.32768)THEN
           WRITE(1,12)IIR
           LSTOP=.TRUE.
      END IF
      IF(IEND4.GT.32768)THEN
           WRITE(1,13)IEND4
           LSTOP=.TRUE.
      END IF
      IF(IEND7.GT.32768)THEN
           WRITE(1,14)IEND7
           LSTOP=.TRUE.
      END IF
      IF(IPATH.GT.131072)THEN
           WRITE(1,15)IPATH
```

```
          LSTOP=.TRUE.
      END IF
      IF(LSTOP)GOTO6
      WRITE(1,16)
      OPEN(6,FILE=FILE1)
      CALL DATAIN(G4(IRCH),G1(ITMP),G4(IHF),G2(IIT),G4(IP),
     c G4(IPOR),G4(IHC),G4(IDP),G2(IIH),G1(IHD),G1(IDX),G1(IDY),G1(IDZ),
     c G2(INP),G8(ISEC))
C     MORE POINTERS FOR G2 ARRAY
      IJR=IIR+NEQ
      IKR=IJR+NEQ
      IEND2=IKR+NEQ-1
C     SET POINTERS FOR G3 ARRAY
      IQ=1
      IEND3=NEQ*6
C     SET POINTERS FOR G5 ARRAY
      IAU=1
      IEND5=NU*12
C     SET POINTERS FOR G6 ARRAY
      IDU=1
      IRU=IDU+NU*2
      IRL=IRU+NU*2
      IEND6=IRL+NL*2-1
      IENDA=NL*IBW
      IF(IEND2.GT.32768)THEN
          WRITE(1,12)IEND2
          LSTOP=.TRUE.
      END IF
      IF(IEND3.GT.32768)THEN
          WRITE(1,17)IEND3
          LSTOP=.TRUE.
      END IF
      IF(IEND5.GT.32768)THEN
          WRITE(1,18)IEND5
          LSTOP=.TRUE.
      END IF
      IF(IEND6.GT.32768)THEN
          WRITE(1,19)IEND6
          LSTOP=.TRUE.
      END IF
      IF(IENDA.GT.253000)THEN
          WRITE(1,20)IENDA
          LSTOP=.TRUE.
      END IF
      IF(LSTOP)GOTO6
      CALL PARTN(G2(INP),G2(IIR),G2(IJR),G2(IKR),G3(IQ))
      CLOSE(6)
      OPEN(6,FILE=FILE2)
    3 WRITE(1,21)
      CALL PA(G1(ITMP),G2(IIT),G2(IIH),G2(INP),G4(IP),G4(IHC),G3(IQ))
      IF(LFIN)GOTO4
      WRITE(1,22)
```

Attachment 1, continued

```
      CALL WATEQ(G6(IRL),G6(IRU),G6(IDU),G5(IAU),                       HOTW157
     c G2(IIR),G2(IJR),G2(IKR),G2(INP),G4(IHC),G1(IDX),G1(IDY),G1(IDZ), HOTW158
     c G2(IIH),G1(IHD),G4(IRCH))                                        HOTW159
      WRITE(1,23)                                                       HOTW160
      IGATE = IGATE + 1                                                 HOTW161
      WRITE(1,24) IGATE                                                 HOTW162
      CALL SOLVE(G6(IRL))                                               HOTW163
      CALL BACK(G2(IIR),G2(IJR),G2(IKR),G2(INP),G1(IHD),G6(IRL),        HOTW164
     c G6(IRU),G5(IAU),G6(IDU),DELV)                                    HOTW165
      DELH=DELV                                                         HOTW166
      WRITE(1,25)DELH                                                   HOTW167
      WRITE(6,26)IGATE,DELH                                             HOTW168
      IF(ABS(DELT).LT.TCONV.AND.ABS(DELH).LT.HCONV)LFIN=.TRUE.          HOTW169
      IF(IGATE.GE.MAX) LFIN=.TRUE.                                      HOTW170
    4 WRITE(1,27)                                                       HOTW171
      CALL COMPQ(G2(IIR),G2(IJR),G2(IKR),G2(IIH),G1(IHD),               HOTW172
     c G4(IHC),G1(IDX),G1(IDY),G1(IDZ),G3(IQ))                          HOTW173
      IF(LFIN)GOTO5                                                     HOTW174
      WRITE(1,28)                                                       HOTW175
      CALL HEATEQ(G6(IRL),G6(IRU),G6(IDU),G5(IAU),G2(IIR),G2(IJR),      HOTW176
     c G2(IKR),G2(INP), G1(ITMP),G2(IIT),G1(IDX),G1(IDY),G1(IDZ),       HOTW177
     c G4(IRCH),G4(IDP),G4(IHF),G3(IQ))                                 HOTW178
      WRITE(1,23)                                                       HOTW179
      IGATE = IGATE + 1                                                 HOTW180
      WRITE(1,24) IGATE                                                 HOTW181
      CALL SOLVE(G6(IRL))                                               HOTW182
      CALL BACK(G2(IIR),G2(IJR),G2(IKR),G2(INP),G1(ITMP),G6(IRL),       HOTW183
     c G6(IRU),G5(IAU),G6(IDU),DELV)                                    HOTW184
      DELT=DELV                                                         HOTW185
      WRITE(1,29)DELT                                                   HOTW186
      WRITE(6,30)IGATE,DELT                                             HOTW187
      IF(ABS(DELT).LT.TCONV.AND.ABS(DELH).LT.HCONV)LFIN=.TRUE.          HOTW188
      IF(IGATE.GE.MAX) LFIN=.TRUE.                                      HOTW189
      IF(LFIN)GOTO5                                                     HOTW190
      GOTO3                                                             HOTW191
    5 WRITE(1,31)                                                       HOTW192
      CALL BAL(G1(ITMP),G4(IHC),G1(IDX),G1(IDY),G1(IDZ),G2(IIT),G4(IHF),HOTW193
     c G4(IDP),G4(IRCH),G2(IIH),G1(IHD))                                HOTW194
      WRITE(1,32)                                                       HOTW195
      WRITE(6,32)                                                       HOTW196
      CLOSE(6)                                                          HOTW197
      IF(LVPT)OPEN(6,FILE=FILE3)                                        HOTW198
      CALL VELO(G2(IIH),G1(IHD),G4(IHC),G1(IDX),G1(IDY),                HOTW199
     c G1(IDZ),G7(IVX1),G7(IVX2),G7(IVY1),G7(IVY2),                     HOTW200
     c G7(IVZ1),G7(IVZ2),G4(IPOR),G4(IRCH))                             HOTW201
      IF(LVPT)CLOSE(6)                                                  HOTW202
      OPEN(6,FILE=FILE4)                                                HOTW203
      WRITE(1,33)                                                       HOTW204
      CALL MOVE(G2(IIH),G7(IVX1),G7(IVX2),G7(IVY1),                     HOTW205
     c G7(IVY2),G7(IVZ1),G7(IVZ2),G1(IDX),G1(IDY),G1(IDZ),              HOTW206
     c G8(ITIM),G8(IIDS))                                               HOTW207
      CLOSE(6)                                                          HOTW208
```

Attachment 1, continued

```
      IF(M.GT.1)GOTO6                                                   HOTW209
      NR=(131073-IPATH)/3                                               HOTW210
      NREC=NR/4                                                         HOTW211
      WRITE(1,34)                                                       HOTW212
      OPEN(6,FILE=FILE5)                                                HOTW213
      CALL PATH(G4(IRCH),G2(IIH),G7(IVX1),G7(IVX2),                     HOTW214
     c G7(IVZ1),G7(IVZ2),G1(IDX),G1(IDY),G1(IDZ),                       HOTW215
     c G1(IDISTX),G1(IDISTZ),G4(IRJ),NREC,G8(IPATH),                    HOTW216
     c G3(IQ),G2(INP))                                                  HOTW217
      CLOSE(6)                                                          HOTW218
      WRITE(1,35)                                                       HOTW219
      NR=NREC*4                                                         HOTW220
C     MORE POINTERS FOR G8 ARRAY                                        HOTW221
      IIPOS=IPATH+NR                                                    HOTW222
      ISRT=IIPOS+NR                                                     HOTW223
      IEND8=ISRT+NR-1                                                   HOTW224
      IF(IEND8.GT.131072)THEN                                           HOTW225
           WRITE(1,15)IEND8                                             HOTW226
           LSTOP=.TRUE.                                                 HOTW227
      END IF                                                            HOTW228
      IF(LSTOP)GOTO6                                                    HOTW229
      CALL SORT(G8(IIPOS),G8(IPATH),G8(ISRT),NREC,LREC)                 HOTW230
      WRITE(1,36)                                                       HOTW231
      OPEN(6,FILE=FILE6)                                                HOTW232
      CALL PRINT(G8(ITIM),G8(ISRT),G1(IHD),G2(IIH),G1(IDX),             HOTW233
     c G1(IDZ),G8(ISEC),NR)                                             HOTW234
      CLOSE(6)                                                          HOTW235
    6 STOP                                                              HOTW236
    7 FORMAT(' FILENAME=')                                              HOTW237
    8 FORMAT(A12)                                                       HOTW238
    9 FORMAT(' FILE NAME IS BLANK')                                     HOTW239
   10 FORMAT(5I3,2L1)                                                   HOTW240
   11 FORMAT(' SIZE OF G1 ARRAY = ',I6,' , DIMENSIONED AS 32768')       HOTW241
   12 FORMAT(' SIZE OF G2 ARRAY = ',I6,' , DIMENSIONED AS 32768')       HOTW242
   13 FORMAT(' SIZE OF G4 ARRAY = ',I6,' , DIMENSIONED AS 32768')       HOTW243
   14 FORMAT(' SIZE OF G7 ARRAY = ',I6,' , DIMENSIONED AS 32768')       HOTW244
   15 FORMAT(' SIZE OF G8 ARRAY = ',I6,' , DIMENSIONED AS 131072')      HOTW245
   16 FORMAT(' DATAIN')                                                 HOTW246
   17 FORMAT(' SIZE OF G3 ARRAY = ',I6,' , DIMENSIONED AS 32768')       HOTW247
   18 FORMAT(' SIZE OF G5 ARRAY = ',I6,' , DIMENSIONED AS 32768')       HOTW248
   19 FORMAT(' SIZE OF G6 ARRAY = ',I6,' , DIMENSIONED AS 32768')       HOTW249
   20 FORMAT(' SIZE OF A ARRAY = ',I6,' , DIMENSIONED AS 253000')       HOTW250
   21 FORMAT(' PA')                                                     HOTW251
   22 FORMAT(' WATEQ')                                                  HOTW252
   23 FORMAT(' SOLVE')                                                  HOTW253
   24 FORMAT(/1X,'ITERATION NUMBER ',I3/1X,20(1H-))                     HOTW254
   25 FORMAT(/1X,'MAX. CHANGE IN HEAD = ',1PE12.4)                      HOTW255
   26 FORMAT(/1X,'ITERATION NUMBER ',I3,                                HOTW256
     c ', MAX. CHANGE IN HEAD = ',1PE12.4)                              HOTW257
   27 FORMAT(' COMPQ')                                                  HOTW258
   28 FORMAT(' HEATEQ')                                                 HOTW259
   29 FORMAT(/1X,'MAX. CHANGE IN TEMP. = ',1PE12.4)                     HOTW260
```

Attachment 1, continued

```
   30 FORMAT(/1X,'ITERATION NUMBER ',I3,
     c ', MAX. CHANGE IN TEMP. = ',1PE12.4)
   31 FORMAT(' BAL')
   32 FORMAT(' VELO')
   33 FORMAT(' MOVE')
   34 FORMAT(' PATH')
   35 FORMAT(' SORT')
   36 FORMAT(' PRINT')
      END
```

Attachment 1, continued

```
C       DATAIN - Reads input data.
        SUBROUTINE DATAIN(RCH,TMP,HF,IT,P,POR,HC,DP,IH,HD,DX,DY,DZ,
      c NP,SEC)
        LOGICAL LFIN,LHCPT,LVPT
        REAL KT
        REAL*8 HD,TMP
        CHARACTER*8 XMESUR,ZMESUR
        DIMENSION RCH(M,N),TMP(M,N,L),HF(M,N),
      cIT(M,N,L),P(M,N,L),POR(M,N,L),
      cHC(M,N,L),DP(M,N),
      cIH(M,N,L),HD(M,N,L),NP(M,N,L)
        DIMENSION DX(N),DY(M),DZ(L)
        CHARACTER SEC(N,L)
        COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
      c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
      c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
        WRITE(6,8)M,N,L,MAX,IREF,LVPT,LHCPT
        READ(5,9) KT,SP,TTOP,TEMPR,TCONV,HCONV
        WRITE(6,10) KT,SP,TTOP,TEMPR,TCONV,HCONV
C       CONVERT FROM CONDUCTANCE UNITS TO BTU/(DAY*FT**2*(DEG.F/FT))
        KT=KT*5.8058
        READ(5,9)DX
        WRITE(6,11)
        WRITE(6,12)DX
        READ(5,9)DY
        WRITE(6,13)
        WRITE(6,12)DY
        READ(5,9)DZ
        WRITE(6,14)
        WRITE(6,12)DZ
        DO 1 K=1,L
        DO 1 I=1,M
      1 READ(5,15)(IT(I,J,K),J=1,N)
        WRITE(6,16)
        DO 2 K=1,L
        WRITE(6,17)K
        DO 2 I=1,M
      2 WRITE(6,18)(IT(I,J,K),J=1,N)
        DO 3 K=1,L
        DO 3 I=1,M
      3 READ(5,15)(IH(I,J,K),J=1,N)
        WRITE(6,19)
        DO 4 K=1,L
        WRITE(6,17)K
        DO 4 I=1,M
      4 WRITE(6,18)(IH(I,J,K),J=1,N)
        WRITE(6,20)
        CALL READ(DP,M,N,1)
        WRITE(6,21)
        CALL READ(RCH,M,N,1)
        WRITE(6,22)
        CALL READ(HF,M,N,1)
```

Attachment 1, continued

```
      WRITE(6,23)
      CALL DREAD(HD,M,N,L)
      WRITE(6,24)
      CALL DREAD(TMP,M,N,L)
      WRITE(6,25)
      CALL READ(P,M,N,L)
      WRITE(6,26)
      CALL READ(POR,M,N,L)
      IF(M.EQ.1)THEN
          WRITE(6,27)
          DO 5 K=1,L
          READ(5,28)(SEC(J,K),J=1,N)
          WRITE(6,29)(SEC(J,K),J=1,N)
    5     CONTINUE
          READ(5,30)XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
          WRITE(6,31)XSCALE,DINCHX,XMESUR,ZSCALE,DINCHZ,ZMESUR
      ENDIF
      CLOSE(5)
      IGATE = 0
      DO 6 J=1,N
      DXJ=DX(J)
      DO 6 I=1,M
      XMUL1=DXJ*DY(I)
      XMUL2=XMUL1/(12.*365.25)
      XMUL3=XMUL1*.31853
C     CONVERT FROM HFU TO BTU/DAY
      HF(I,J)=HF(I,J)*XMUL3
C     CONVERT FROM IN/YR TO FT**3/DAY
      RCH(I,J)=RCH(I,J)*XMUL2
C     CONVERT FROM FEET TO (BTU/DAY)/DEG.F
      IF(DP(I,J).GT.0.)DP(I,J)=KT*XMUL1/DP(I,J)
      DO 6 K=1,L
      HC(I,J,K) = 0.
    6 CONTINUE
      DELH=HCONV
      DELT=TCONV
      DO 7 K=1,L
      DO 7 J=1,N
      DO 7 I=1,M
      NP(I,J,K)=0
    7 CONTINUE
      CALL NUMBER(IT,NP)
      RETURN
    8 FORMAT(' NUMBER OF ROWS = ',I3/' NUMBER OF COLS = ',I3
     c /' NUMBER OF LAYERS = ',I3/' MAXIMUM ITERATIONS = ',I3
     c /' REFERENCE TEMP. FOR HYD. COND. = ',I3,' DEG. C'
     c /' VELOCITY PRINT OPTION = ',L1
     c /' HYD. COND. PRINT OPTION = ',L1)
    9 FORMAT(16F5.0)
   10 FORMAT(/' THERMAL CONDUCTIVITY=',1PE12.4,
     c ' C.U. - 10**-3 CAL./(CM.*S.',
     c '*DEG. C)'/' SPECIFIC HEAT=',E12.4,' BTU/CUBIC FT/DEG. F'/
```

Attachment 1, continued

```
     c ' SURFACE TEMP.=',E12.4,' DEG. F'/                                DATA104
     c ' RECHARGE TEMP. = ',E12.4,' DEG. F'/                             DATA105
     c ' TEMP. CONV. CRITERION = ',E12.4,' DEG. F'/                      DATA106
     c ' HEAD CONV. CRITERION = ',E12.4,' FEET')                         DATA107
  11 FORMAT(/' HORIZONTAL (X) NODE SPACING, FEET'/1H ,33(1H-))           DATA108
  12 FORMAT(1X,1P10E12.4)                                                DATA109
  13 FORMAT(/' HORIZONTAL (Y) NODE SPACING, FEET'/1H ,33(1H-))           DATA110
  14 FORMAT(/' VERTICAL NODE SPACING, FEET'/1H ,27(1H-))                 DATA111
  15 FORMAT(80I1)                                                        DATA112
  16 FORMAT(/' NODE LEVEL FOR TEMP.'/1H ,20(1H-))                        DATA113
  17 FORMAT(/' LAYER ',I2/1H ,8(1H-))                                    DATA114
  18 FORMAT(1H ,80I1)                                                    DATA115
  19 FORMAT(/' NODE LEVEL FOR HEAD'/1H ,19(1H-))                         DATA116
  20 FORMAT(/' DEPTH BELOW LAND SURFACE, FEET'/1H ,30(1H-))              DATA117
  21 FORMAT(/' RECHARGE RATE, IN/YR'/1H ,20(1H-))                        DATA118
  22 FORMAT(/' HEAT FLOW, H.F.U. - 10**-6 CAL./(SQ. CM.*S.)'/1H ,        DATA119
     c 44(1H-))                                                          DATA120
  23 FORMAT(/' INITIAL HEAD, FEET'/1H ,18(1H-))                          DATA121
  24 FORMAT(/' INITIAL TEMPERATURE, DEG. F'/1H ,27(1H-))                 DATA122
  25 FORMAT(/' HYDRAULIC CONDUCTIVITY (FT./DAY) AT REF. TEMP.'/          DATA123
     c 1H ,46(1H-))                                                      DATA124
  26 FORMAT(/' POROSITY'/1H ,8(1H-))                                     DATA125
  27 FORMAT(/' CROSS SECTION'/1H ,13(1H-))                               DATA126
  28 FORMAT(80A1)                                                        DATA127
  29 FORMAT(1X,80A1)                                                     DATA128
  30 FORMAT(4F10.0,A8,2X,A8)                                             DATA129
  31 FORMAT(/' HORIZONTAL SCALE UNIT = ',G12.5,' FEET'                   DATA130
     c /' NUMBER PER INCH= ',G12.5/' NAME = ',A8                         DATA131
     c /' VERTICAL SCALE UNIT = ',G12.5,' FEET'                          DATA132
     c /' NUMBER PER INCH= ',G12.5/' NAME = ',A8)                        DATA133
      END                                                                DATA134
```

Attachment 1, continued

```
C     READ - Reads REAL*4 arrays.
      SUBROUTINE READ(ARRAY,M,N,KL)
      DIMENSION ARRAY(1),TEMP(100)
      MN=M*N
      READ(5,4)CONS,IVAR
      IF(IVAR.EQ.0)THEN
           NARY=MN*KL
           DO 1 IARY=1,NARY
    1      ARRAY(IARY)=CONS
           WRITE(6,5)CONS
      ELSE
           KK=-M-MN
           DO 3 K=1,KL
           KK=KK+MN
           II=KK
           IF(KL.GT.1)WRITE(6,6)K
           DO 3 I=1,M
           II=II+1
           IJK=II
           READ(5,7)(TEMP(J),J=1,N)
           DO 2 J=1,N
           IJK=IJK+M
C     IJK=I+(J-1)*M+(K-1)*M*N
           ARRAY(IJK)=TEMP(J)*CONS
    2      TEMP(J)=ARRAY(IJK)
           WRITE(6,8)(TEMP(J),J=1,N)
    3      CONTINUE
      ENDIF
      RETURN
    4 FORMAT(F10.0,I5)
    5 FORMAT(' UNIFORMLY =',1PE12.3)
    6 FORMAT(/' LAYER ',I2/1H ,8(1H-))
    7 FORMAT(20F4.0)
    8 FORMAT(1H ,1P10E12.3)
      END
```

Attachment 1, continued

```
C      DREAD - Reads REAL*8 arrays.
       SUBROUTINE DREAD(ARRAY,M,N,KL)
       REAL*8 ARRAY(1),TEMP(100)
       MN=M*N
       READ(5,4)CONS,IVAR
       IF(IVAR.EQ.0)THEN
            NARY=MN*KL
            DO 1 IARY=1,NARY
    1       ARRAY(IARY)=CONS
            WRITE(6,5)CONS
       ELSE
            KK=-M-MN
            DO 3 K=1,KL
            KK=KK+MN
            II=KK
            IF(KL.GT.1)WRITE(6,6)K
            DO 3 I=1,M
            II=II+1
            IJK=II
            READ(5,7)(TEMP(J),J=1,N)
            DO 2 J=1,N
            IJK=IJK+M
C      IJK=I+(J-1)*M+(K-1)*M*N
            ARRAY(IJK)=TEMP(J)*CONS
    2       TEMP(J)=ARRAY(IJK)
            WRITE(6,8)(TEMP(J),J=1,N)
    3       CONTINUE
       ENDIF
       RETURN
    4  FORMAT(F10.0,I5)
    5  FORMAT(' UNIFORMLY =',1PE12.3)
    6  FORMAT(/' LAYER ',I2/1H ,8(1H-))
    7  FORMAT(20F4.0)
    8  FORMAT(1H ,1P10E12.3)
       END
```

Attachment 1, continued

```
C     NUMBER - Numbers the equations, sets pointers from model grid
C        to equations.
      SUBROUTINE NUMBER(IT,NP)
      LOGICAL LFIN,LHCPT,LVPT
      REAL KT
      CHARACTER*8 XMESUR,ZMESUR
      DIMENSION IT(1),NP(1)
      COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
     c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
     c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
      I1L=MAX0(M,N,L)
      I3L=MIN0(M,N,L)
      IF(M.EQ.I1L)THEN
           I1I=1
           IF(L.EQ.I3L)THEN
                I2L=N
                I2I=M
                I3I=MN
           ELSE
                I2L=L
                I2I=MN
                I3I=M
           ENDIF
      ELSE IF(N.EQ.I1L)THEN
           I1I=M
           IF(L.EQ.I3L)THEN
                I2L=M
                I2I=1
                I3I=MN
           ELSE
                I2L=L
                I2I=MN
                I3I=1
           ENDIF
      ELSE
           I1I=MN
           IF(N.EQ.I3L)THEN
                I2L=M
                I2I=1
                I3I=M
           ELSE
                I2L=N
                I2I=M
                I3I=1
           ENDIF
      ENDIF
      IBW=2*I2L*I3L+1
      MDM=IBW/2
      IEQ=0
      DO 2 NC=1,2
      IF(NC.EQ.1)THEN
           I3C=2
```

Attachment 1, continued

```
            I3INC=2*I3I
      ELSE
            I3S=1
            I3C=1
            I3INC=I3I
      ENDIF
      II1=-M-M*N
      DO 1 I1=1,I1L
      II1=II1+I1I
      II2=II1
      DO 1 I2=1,I2L
      II2=II2+I2I
      IF(NC.EQ.1)THEN
            IF(MOD(I1+I2,2).EQ.0)THEN
                  I3S=1
                  II3=II2-I3I
            ELSE
                  I3S=2
                  II3=II2
            ENDIF
      ELSE
            II3=II2
      ENDIF
      DO 1 I3=I3S,I3L,I3C
      II3=II3+I3INC
C     II3=I1*I1I+I2*I2I+I3*I3I-M-M*N
      IF(IT(II3).NE.1)GOTO1
      IF(NP(II3).NE.0)GOTO1
      IEQ=IEQ+1
      NP(II3)=IEQ
    1 CONTINUE
      IF(NC.EQ.1)THEN
            NU=IEQ
      ELSE
            NEQ=IEQ
            NL=NEQ-NU
      ENDIF
    2 CONTINUE
      RETURN
      END
```

Attachment 1, continued

```
C     PARTN - Sets pointers from equation to model grid, zeros Q array.
      SUBROUTINE PARTN(NP,IR,JR,KR,Q)
      LOGICAL LFIN,LHCPT,LVPT
      CHARACTER*8 XMESUR,ZMESUR
      DIMENSION NP(M,N,L),IR(NEQ),JR(NEQ),KR(NEQ),Q(NEQ,6)
      COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
     c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
     c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
      DO 1 K=1,L
      DO 1 J=1,N
      DO 1 I=1,M
      IEQ=NP(I,J,K)
      IF(IEQ.EQ.0)GOTO1
      IR(IEQ)=I
      JR(IEQ)=J
      KR(IEQ)=K
    1 CONTINUE
      WRITE(6,3)
      WRITE(6,4)(IEQ,IR(IEQ),JR(IEQ),IEQR(IEQ),IEQ=1,NEQ)
      DO 2 JJ=1,6
      DO 2 IEQ=1,NEQ
    2 Q(IEQ,JJ)=0.
      RETURN
    3 FORMAT(/' BLOCK NUMBER AND ROW, COL., AND LAYER NUMBER'/
     c 1H ,44(1H-))
    4 FORMAT(1H ,9(I4,'=',I2,',',I2,',',I2,1H ))
      END
```

Attachment 1, continued

```
C       PA - Adjusts hydraulic conductivity for temperature change.
        SUBROUTINE PA(TMP,IT,IH,NP,P,HC,Q)
        LOGICAL LFIN,LHCPT,LVPT
        REAL KT
        REAL*8 TMP
        CHARACTER*8 XMESUR,ZMESUR
        DIMENSION TMP(M,N,L),IT(M,N,L),IH(M,N,L),NP(M,N,L),
       c P(M,N,L),HC(M,N,L)
        DIMENSION IC(6),JC(6),KC(6),IV(6),IVT(6)
        DIMENSION Q(NEQ,6),XKV(100)
        COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
       c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
       c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
        DATA XKV/1.7322,1.6741,1.6194,1.5676,1.5189,1.4727,1.4289,
       c 1.3874,1.3479,1.3101,1.2740,1.2396,1.2069,1.1757,1.1457,1.1168,
       c 1.0889,1.0618,1.0357,1.0105,.9863,.9629,.9403,.9186,.8976,
       c .8774,.8581,.8395,.8214,.8039,.7871,.7668,.7551,.7399,.7251,
       c .7109,.6971,.6839,.6711,.6587,.6468,.6352,.6240,.6132,.6029,
       c .5929,.5832,.5739,.5647,.5558,.5473,.5389,.5307,.5225,.5146,
       c .5069,.4994,.4921,.4849,.4779,.4711,.4644,.4579,.4516,.4455,
       c .4394,.4334,.4276,.4219,.4164,.4110,.4057,.4005,.3954,.3904,
       c .3855,.3808,.3762,.3718,.3674,.3631,.3589,.3547,.3504,.3466,
       c .3427,.3389,.3352,.3316,.3280,.3245,.3211,.3178,.3145,.3113,
       c .3082,.3051,.3020,.2991,.2962/
        DATA IC/+1,-1,0,0,0,0/,JC/0,0,+1,-1,0,0/
        DATA KC/0,0,0,0,+1,-1/,IVT/6*1/
        IVT(1)=M
        IVT(3)=N
        IVT(5)=L
        DO 2 K=1,L
        DO 2 J=1,N
        DO 2 I=1,M
        IF(IH(I,J,K).NE.3)THEN
             TC=(TMP(I,J,K)-32.)/1.8
             IF((IH(I,J,K).EQ.2).AND.(IT(I,J,K).EQ.3))THEN
                  TSUM=0.
                  QSUM=0.
                  IV(1)=I
                  IV(2)=I
                  IV(3)=J
                  IV(4)=J
                  IV(5)=K
                  IV(6)=K
                  DO 1 II=1,6
                  IF(IV(II).NE.IVT(II))THEN
                       IIC=I+IC(II)
                       JJC=J+JC(II)
                       KKC=K+KC(II)
                       NPIJK=NP(IIC,JJC,KKC)
                       IF(NPIJK.NE.0)THEN
                            QQ=Q(NPIJK,II)
                            IF(QQ.LT.0.)THEN
```

Attachment 1, continued

```
                                    TSUM=TSUM-QQ*TMP(IIC,JJC,KKC)
                                    QSUM=QSUM-QQ
                               ENDIF
                          ENDIF
                     ENDIF
    1                CONTINUE
                     IF(QSUM.GT.0.)THEN
                          TAVE=TSUM/QSUM
                          TC=(TAVE-32.)/1.8
                     ENDIF
                ENDIF
                IF(TC.GT.100.)THEN
                     TC=100.
                ELSE IF(TC.LT.1.)THEN
                     TC=1.
                ENDIF
                IXKV=TC
                IXKV1=IXKV+1
                IF(IXKV1.GT.100)IXKV1=100
                XKVTC=XKV(IXKV)+(TC-IXKV)*(XKV(IXKV1)-XKV(IXKV))
                HC(I,J,K)=XKV(IREF)/XKVTC*P(I,J,K)
           ENDIF
    2 CONTINUE
      RETURN
      END
```

Attachment 1, continued

```
C     WATEQ - Computes coefficient matrix and known vector for
C         ground-water-flow equations.
      SUBROUTINE WATEQ(RL,RU,DU,AU,IR,JR,KR,NP,HC,DX,DY,DZ,
     c IH,HD,RCH)
      LOGICAL LFIN,LHCPT,LVPT
      REAL KT
      REAL*8 HD
      REAL*8 A,RL,RU,DU,AU,AL
      REAL*8 REQ,DCOEF,ACOEF,GCOEF,BCOEF,FCOEF,CCOEF,ECOEF
      CHARACTER*8 XMESUR,ZMESUR
      DIMENSION RL(NL),RU(NU),DU(NU),AU(6,NU),AL(6),
     c IR(NEQ),JR(NEQ),KR(NEQ),NP(M,N,L),HC(M,N,L),DX(N),DY(M),DZ(L),
     c IH(M,N,L),HD(M,N,L),RCH(M,N)
      DIMENSION IC(6),JC(6),KC(6)
      COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
     c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
     c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
      COMMON//A(253000)
      DATA IC/-1,+1,0,0,0,0/,JC/0,0,-1,+1,0,0/,KC/0,0,0,0,-1,+1/
      DO 1 IEQ=1,NU
      DU(IEQ)=0.
      RU(IEQ)=0.
      DO 1 JJ=1,6
    1 AU(JJ,IEQ)=0.
      DO 2 JJ=1,6
    2 AL(JJ)=0.
      DO 3 IL=1,NL
    3 RL(IL)=0.
      IL=0
      IA=-IBW
      MD=MDM+1
      DO 7 IEQ=1,NEQ
      IF(IEQ.GT.NU)IL=IL+1
      I=IR(IEQ)
      J=JR(IEQ)
      K=KR(IEQ)
      HCIJK=HC(I,J,K)
      DYI=DY(I)
      DXJ=DX(J)
      DZK=DZ(K)
      AREAY=2.*DXJ*DZK
      AREAX=2.*DYI*DZK
      AREAZ=2.*DXJ*DYI
      DCOEF=0.
      REQ=0.
      IF(I.NE.1)THEN
           IHIM1=IH(I-1,J,K)
           IF(IHIM1.NE.3)THEN
                ACOEF=HC(I-1,J,K)*HCIJK/(HC(I-1,J,K)*DYI+HCIJK*DY(I-1))
     c          *AREAY
                DCOEF=DCOEF-ACOEF
                IF(IHIM1.NE.2)THEN
```

```
                    IF(IEQ.GT.NU)THEN
                         AL(1)=ACOEF
                    ELSE
                         AU(1,IEQ)=ACOEF
                    ENDIF
               ENDIF
               REQ=REQ-ACOEF*HD(I-1,J,K)
          ENDIF
     ENDIF
     IF(J.NE.1)THEN
          IHJM1=IH(I,J-1,K)
          IF(IHJM1.NE.3)THEN
               BCOEF=HC(I,J-1,K)*HCIJK/(HC(I,J-1,K)*DXJ+HCIJK*DX(J-1))
    c          *AREAX
               DCOEF=DCOEF-BCOEF
               IF(IHJM1.NE.2)THEN
                    IF(IEQ.GT.NU)THEN
                         AL(3)=BCOEF
                    ELSE
                         AU(3,IEQ)=BCOEF
                    ENDIF
               ENDIF
               REQ=REQ-BCOEF*HD(I,J-1,K)
          ENDIF
     ENDIF
     IF(J.NE.N)THEN
          IHJP1=IH(I,J+1,K)
          IF(IHJP1.NE.3)THEN
               FCOEF=HC(I,J+1,K)*HCIJK/(HC(I,J+1,K)*DXJ+HCIJK*DX(J+1))
    c          *AREAX
               DCOEF=DCOEF-FCOEF
               IF(IHJP1.NE.2)THEN
                    IF(IEQ.GT.NU)THEN
                         AL(4)=FCOEF
                    ELSE
                         AU(4,IEQ)=FCOEF
                    ENDIF
               ENDIF
               REQ=REQ-FCOEF*HD(I,J+1,K)
          ENDIF
     ENDIF
     IF(I.NE.M)THEN
          IHIP1=IH(I+1,J,K)
          IF(IHIP1.NE.3)THEN
               GCOEF=HC(I+1,J,K)*HCIJK/(HC(I+1,J,K)*DYI+HCIJK*DY(I+1))
    c          *AREAY
               DCOEF=DCOEF-GCOEF
               IF(IHIP1.NE.2)THEN
                    IF(IEQ.GT.NU)THEN
                         AL(2)=GCOEF
                    ELSE
                         AU(2,IEQ)=GCOEF
```

Attachment 1, continued

```
                    ENDIF                                                    WATE103
                 ENDIF                                                       WATE104
                 REQ=REQ-GCOEF*HD(I+1,J,K)                                   WATE105
            ENDIF                                                            WATE106
       ENDIF                                                                 WATE107
       IF(K.NE.1)THEN                                                        WATE108
            IHKM1=IH(I,J,K-1)                                                WATE109
            IF(IHKM1.NE.3)THEN                                               WATE110
                 CCOEF=HC(I,J,K-1)*HCIJK/(HC(I,J,K-1)*DZK+HCIJK*DZ(K-1))     WATE111
     c           *AREAZ                                                      WATE112
                 DCOEF=DCOEF-CCOEF                                           WATE113
                 IF(IHKM1.NE.2)THEN                                          WATE114
                      IF(IEQ.GT.NU)THEN                                      WATE115
                           AL(5)=CCOEF                                       WATE116
                      ELSE                                                   WATE117
                           AU(5,IEQ)=CCOEF                                   WATE118
                      ENDIF                                                  WATE119
                 ELSE                                                        WATE120
                      REQ=REQ-RCH(I,J)                                       WATE121
                 ENDIF                                                       WATE122
                 REQ=REQ-CCOEF*HD(I,J,K-1)                                   WATE123
            ELSE                                                             WATE124
                 REQ=REQ-RCH(I,J)                                            WATE125
            ENDIF                                                            WATE126
       ELSE                                                                  WATE127
            REQ=REQ-RCH(I,J)                                                 WATE128
       ENDIF                                                                 WATE129
       IF(K.NE.L)THEN                                                        WATE130
            IHKP1=IH(I,J,K+1)                                                WATE131
            IF(IHKP1.NE.3)THEN                                               WATE132
                 ECOEF=HC(I,J,K+1)*HCIJK/(HC(I,J,K+1)*DZK+HCIJK*DZ(K+1))     WATE133
     c           *AREAZ                                                      WATE134
                 DCOEF=DCOEF-ECOEF                                           WATE135
                 IF(IHKP1.NE.2)THEN                                          WATE136
                      IF(IEQ.GT.NU)THEN                                      WATE137
                           AL(6)=ECOEF                                       WATE138
                      ELSE                                                   WATE139
                           AU(6,IEQ)=ECOEF                                   WATE140
                      ENDIF                                                  WATE141
                 ENDIF                                                       WATE142
                 REQ=REQ-ECOEF*HD(I,J,K+1)                                   WATE143
            ENDIF                                                            WATE144
       ENDIF                                                                 WATE145
       IF(IEQ.LE.NU)THEN                                                     WATE146
            DU(IEQ)=DCOEF                                                    WATE147
            RU(IEQ)=REQ-DCOEF*HD(I,J,K)                                      WATE148
       ELSE                                                                  WATE149
            RL(IL)=REQ-DCOEF*HD(I,J,K)                                       WATE150
            IA=IA+IBW                                                        WATE151
            DO 4 ISUB=IA+1,IA+IBW                                            WATE152
   4        A(ISUB)=0.                                                       WATE153
            IMD=IA+MD                                                        WATE154
```

Attachment 1, continued

```
      A(IMD)=DCOEF                                              WATE155
      DO 6 II=1,6                                               WATE156
      IIC=I+IC(II)                                              WATE157
      IF(IIC.LE.0.OR.IIC.GT.M)GOTO6                             WATE158
      JJC=J+JC(II)                                              WATE159
      IF(JJC.LE.0.OR.JJC.GT.N)GOTO6                             WATE160
      KKC=K+KC(II)                                              WATE161
      IF(KKC.LE.0.OR.KKC.GT.L)GOTO6                             WATE162
      NPQ=NP(IIC,JJC,KKC)                                       WATE163
      IF(NPQ.EQ.0)GOTO6                                         WATE164
      DO 5 JJ=1,6                                               WATE165
      ICC=IIC+IC(JJ)                                            WATE166
      IF(ICC.LE.0.OR.ICC.GT.M)GOTO5                             WATE167
      JCC=JJC+JC(JJ)                                            WATE168
      IF(JCC.LE.0.OR.JCC.GT.N)GOTO5                             WATE169
      KCC=KKC+KC(JJ)                                            WATE170
      IF(KCC.LE.0.OR.KCC.GT.L)GOTO5                             WATE171
      NPP=NP(ICC,JCC,KCC)                                       WATE172
      IF(NPP.EQ.0)GOTO5                                         WATE173
      NN=IA+MD-IEQ+NPP                                          WATE174
      A(NN)=A(NN)-AU(JJ,NPQ)*AL(II)/DU(NPQ)                     WATE175
5     CONTINUE                                                  WATE176
      RL(IL)=RL(IL)-RU(NPQ)*AL(II)/DU(NPQ)                      WATE177
6     CONTINUE                                                  WATE178
    ENDIF                                                       WATE179
7   CONTINUE                                                    WATE180
    RETURN                                                      WATE181
    END                                                         WATE182
```

Attachment 1, continued

```
C       SOLVE - Reduces coefficient matrix to upper-diagonal form
C          and back substitutes to compute values for the unknowns.
        SUBROUTINE SOLVE(RL)
C       ASYMMETRIC BAND MATRIX EQUATION SOLVER
C       ORIGINALLY PROGRAMED BY JAMES O. DUGUID
        LOGICAL LFIN,LHCPT,LVPT
        REAL KT
        REAL*8 A,RL,PIVOT,C,SUM
        CHARACTER*8 XMESUR,ZMESUR
        DIMENSION  RL(NL)
        COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
       c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
       c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
        COMMON//A(253000)
        NRS=NL-1
        MD=MDM+1
C       TRIANGULARIZE MATRIX A USING DOOLITTLE METHOD
        IPIV=MD-IBW
        DO 4 IL=1,NRS
        IPIV=IPIV+IBW
        PIVOT=A(IPIV)
        IELIM=IPIV
        DO 2 IMOD=IL+1,IL+MDM
        IF(IMOD.GT.NL)GOTO3
        IELIM=IELIM+IBW-1
        C=-A(IELIM)/PIVOT
C       MODIFY KNOWN VECTOR R
        RL(IMOD)=RL(IMOD)+C*RL(IL)
        DO 1 JJ=1,MDM
      1 A(IELIM+JJ)=A(IELIM+JJ)+C*A(IPIV+JJ)
      2 CONTINUE
      3 CONTINUE
      4 CONTINUE
C       BACK SUBSTITUTE TO SOLVE
        ISOLV=NL*IBW+MD
        DO 6 IL=NL,1,-1
        SUM=0.
        ISOLV=ISOLV-IBW
        DO 5 I1=1,MDM
        I2=IL+I1
        IF(I2.GT.NL)GOTO6
      5 SUM=SUM+RL(I2)*A(ISOLV+I1)
      6 RL(IL)=(RL(IL)-SUM)/A(ISOLV)
        RETURN
        END
```

Attachment 1, continued

```
C      BACK - Back substitutes to determine the remaining unknowns,
C         adds change to old value, and determines largest change.
       SUBROUTINE BACK(IR,JR,KR,NP,V,RL,RU,AU,DU,DELV)
       LOGICAL LFIN,LHCPT,LVPT
       REAL*8 V,RL,RU,AU,DU
       REAL KT
       CHARACTER*8 XMESUR,ZMESUR
       DIMENSION IR(NEQ),JR(NEQ),KR(NEQ),NP(M,N,L),V(M,N,L),
      c RL(NL),RU(NU),AU(6,NU),DU(NU)
       DIMENSION IC(6),JC(6),KC(6)
       COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
      c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
      c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
       DATA IC/-1,+1,0,0,0,0/,JC/0,0,-1,+1,0,0/,KC/0,0,0,0,-1,+1/
       DELV=0.
       DO 1 IL=1,NL
       IEQ=IL+NU
       I=IR(IEQ)
       J=JR(IEQ)
       K=KR(IEQ)
       TEST=DABS(RL(IL))
       IF(TEST.GT.ABS(DELV))DELV=RL(IL)
     1 V(I,J,K)=V(I,J,K)+RL(IL)
       DO 3 IEQ=1,NU
       I=IR(IEQ)
       J=JR(IEQ)
       K=KR(IEQ)
       DO 2 II=1,6
       IIC=I+IC(II)
       IF(IIC.GE.1.AND.IIC.LE.M)THEN
            JJC=J+JC(II)
            IF(JJC.GE.1.AND.JJC.LE.N)THEN
                 KKC=K+KC(II)
                 IF(KKC.GE.1.AND.KKC.LE.L)THEN
                      NPIJK=NP(IIC,JJC,KKC)
                      IF(NPIJK.NE.0)THEN
                           IL=NPIJK-NU
                           RU(IEQ)=RU(IEQ)-AU(II,IEQ)*RL(IL)
                      ENDIF
                 ENDIF
            ENDIF
       ENDIF
     2 CONTINUE
       RU(IEQ)=RU(IEQ)/DU(IEQ)
       TEST=DABS(RU(IEQ))
       IF(TEST.GT.ABS(DELV))DELV=RU(IEQ)
     3 V(I,J,K)=V(I,J,K)+RU(IEQ)
       RETURN
       END
```

Attachment 1, continued

```
C      COMPQ - Calculates the interblock ground-water flow.
       SUBROUTINE COMPQ(IR,JR,KR,IH,HD,HC,DX,DY,DZ,Q)
       LOGICAL LFIN,LHCPT,LVPT
       REAL KT
       REAL*8 HD,HIJ
       CHARACTER*8 XMESUR,ZMESUR
       DIMENSION IR(NEQ),JR(NEQ),KR(NEQ),IH(M,N,L),
     c HD(M,N,L),HC(M,N,L),DX(N),DY(M),DZ(L)
       DIMENSION Q(NEQ,6)
       COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
     c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
     c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
       DO 1 IEQ=1,NEQ
       I=IR(IEQ)
       J=JR(IEQ)
       K=KR(IEQ)
       IF(IH(I,J,K).EQ.1)THEN
            HIJ=HD(I,J,K)
            HCIJK=HC(I,J,K)
            DZK=DZ(K)
            DXJ=DX(J)
            DYI=DY(I)
            AREAX=2.*DYI*DZK
            AREAY=2.*DXJ*DZK
            AREAZ=2.*DXJ*DYI
            IF((I.EQ.1).OR.(IH(I-1,J,K).EQ.3))THEN
                 Q(IEQ,1)=0.
            ELSE
                 Q(IEQ,1)=(HD(I-1,J,K)-HIJ)*HC(I-1,J,K)*HCIJK/
     c           (HC(I-1,J,K)*DYI+HCIJK*DY(I-1))*AREAY
            ENDIF
            IF((I.EQ.M).OR.(IH(I+1,J,K).EQ.3))THEN
                 Q(IEQ,2)=0.
            ELSE
                 Q(IEQ,2)=(HD(I+1,J,K)-HIJ)*HC(I+1,J,K)*HCIJK/
     c           (HC(I+1,J,K)*DYI+HCIJK*DY(I+1))*AREAY
            ENDIF
            IF((J.EQ.1).OR.(IH(I,J-1,K).EQ.3))THEN
                 Q(IEQ,3)=0.
            ELSE
                 Q(IEQ,3)=(HD(I,J-1,K)-HIJ)*HC(I,J-1,K)*HCIJK/
     c           (HC(I,J-1,K)*DXJ+HCIJK*DX(J-1))*AREAX
            ENDIF
            IF((J.EQ.N).OR.(IH(I,J+1,K).EQ.3))THEN
                 Q(IEQ,4)=0.
            ELSE
                 Q(IEQ,4)=(HD(I,J+1,K)-HIJ)*HC(I,J+1,K)*HCIJK/
     c           (HC(I,J+1,K)*DXJ+HCIJK*DX(J+1))*AREAX
            ENDIF
            IF((K.EQ.1).OR.(IH(I,J,K-1).EQ.3))THEN
                 Q(IEQ,5)=0.
            ELSE
```

Attachment 1, continued

```
             Q(IEQ,5)=(HD(I,J,K-1)-HIJ)*HC(I,J,K-1)*HCIJK/
     c        (HC(I,J,K-1)*DZK+HCIJK*DZ(K-1))*AREAZ
          ENDIF
          IF((K.EQ.L).OR.(IH(I,J,K+1).EQ.3))THEN
             Q(IEQ,6)=0.
          ELSE
             Q(IEQ,6)=(HD(I,J,K+1)-HIJ)*HC(I,J,K+1)*HCIJK/
     c        (HC(I,J,K+1)*DZK+HCIJK*DZ(K+1))*AREAZ
          ENDIF
       ENDIF
     1 CONTINUE
       RETURN
       END
```

Attachment 1, continued

```
C     HEATEQ - Computes coefficient matrix and known vector for
C         heat-flow equations.
      SUBROUTINE HEATEQ(RL,RU,DU,AU,IR,JR,KR,NP,
     c TMP,IT,DX,DY,DZ,RCH,DP,HF,Q)
      REAL*8 A,RL,RU,DU,AU,AL,TMP
      REAL*8 REQ,DCOEF,ACOEF,GCOEF,BCOEF,FCOEF,CCOEF,ECOEF,DPART
      REAL KT,KXY2,KXZ2,KYZ2
      LOGICAL LFIN,LHCPT,LVPT
      CHARACTER*8 XMESUR,ZMESUR
      COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
     c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
     c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
      COMMON//A(253000)
      DIMENSION RL(NL),RU(NU),DU(NU),AU(6,NU),AL(6),
     c IR(NEQ),JR(NEQ),KR(NEQ),NP(M,N,L),TMP(M,N,L),IT(M,N,L),
     c DX(N),DY(M),DZ(L),RCH(M,N),DP(M,N),HF(M,N)
      DIMENSION IC(6),JC(6),KC(6)
      DIMENSION Q(NEQ,6)
      DATA IC/-1,+1,0,0,0,0/,JC/0,0,-1,+1,0,0/,KC/0,0,0,0,-1,+1/
      DO 1 IEQ=1,NU
      DU(IEQ)=0.
      RU(IEQ)=0.
      DO 1 JJ=1,6
    1 AU(JJ,IEQ)=0.
      DO 2 JJ=1,6
    2 AL(JJ)=0.
      DO 3 IL=1,NL
    3 RL(IL)=0.
      IL=0
      IA=-IBW
      MD=MDM+1
      DO 7 IEQ=1,NEQ
      IF(IEQ.GT.NU)IL=IL+1
      I=IR(IEQ)
      J=JR(IEQ)
      K=KR(IEQ)
      DYI=DY(I)
      DXJ=DX(J)
      DZK=DZ(K)
      KXZ2=KT*DXJ*DZK*2.
      KYZ2=KT*DYI*DZK*2.
      KXY2=KT*DXJ*DYI*2.
      DCOEF=0.
      REQ=0.
      ACOEF=0.
      DPART=0.
      IF(I.GT.1)THEN
           ITIM1=IT(I-1,J,K)
           IF(ITIM1.NE.3)THEN
                ACOEF=KXZ2/(DY(I-1)+DYI)
                DPART=-ACOEF
           ENDIF
```

Attachment 1, continued

```
            Q1=Q(IEQ,1)
            IF(Q1.GT.0.)THEN
                  ACOEF=ACOEF+SP*Q1
            ELSE IF(Q1.LT.0.)THEN
                  DPART=DPART+SP*Q1
            ENDIF
            DCOEF=DCOEF+DPART
            IF(ITIM1.EQ.1)THEN
                  IF(IEQ.GT.NU)THEN
                        AL(1)=ACOEF
                  ELSE
                        AU(1,IEQ)=ACOEF
                  ENDIF
             ENDIF
            REQ=REQ-ACOEF*TMP(I-1,J,K)
      ENDIF
      BCOEF=0.
      DPART=0.
      IF(J.NE.1)THEN
            ITJM1=IT(I,J-1,K)
            IF(ITJM1.NE.3)THEN
                  BCOEF=KYZ2/(DX(J-1)+DXJ)
                  DPART=-BCOEF
            ENDIF
            Q3=Q(IEQ,3)
            IF(Q3.GT.0.)THEN
                  BCOEF=BCOEF+SP*Q3
            ELSE IF(Q3.LT.0.)THEN
                  DPART=DPART+SP*Q3
            ENDIF
            DCOEF=DCOEF+DPART
            IF(ITJM1.EQ.1)THEN
                  IF(IEQ.GT.NU)THEN
                        AL(3)=BCOEF
                  ELSE
                        AU(3,IEQ)=BCOEF
                  ENDIF
             ENDIF
            REQ=REQ-BCOEF*TMP(I,J-1,K)
       ENDIF
      FCOEF=0.
      DPART=0.
      IF(J.LT.N)THEN
            ITJP1=IT(I,J+1,K)
            IF(ITJP1.NE.3)THEN
                  FCOEF=KYZ2/(DXJ+DX(J+1))
                  DPART=-FCOEF
            ENDIF
            Q4=Q(IEQ,4)
            IF(Q4.GT.0.)THEN
                  FCOEF=FCOEF+SP*Q4
            ELSE IF(Q4.LT.0.)THEN
```

Attachment 1, continued

```
               DPART=DPART+SP*Q4
          ENDIF
          DCOEF=DCOEF+DPART
          IF(ITJP1.EQ.1)THEN
               IF(IEQ.GT.NU)THEN
                    AL(4)=FCOEF
               ELSE
                    AU(4,IEQ)=FCOEF
               ENDIF
           ENDIF
          REQ=REQ-FCOEF*TMP(I,J+1,K)
     ENDIF
     GCOEF=0.
     DPART=0.
     IF(I.LT.M)THEN
          ITIP1=IT(I+1,J,K)
          IF(ITIP1.NE.3)THEN
               GCOEF=KXZ2/(DYI+DY(I+1))
               DPART=-GCOEF
          ENDIF
          Q2=Q(IEQ,2)
          IF(Q2.GT.0.)THEN
               GCOEF=GCOEF+SP*Q2
          ELSE IF(Q2.LT.0.)THEN
               DPART=DPART+SP*Q2
          ENDIF
          DCOEF=DCOEF+DPART
          IF(ITIP1.EQ.1)THEN
               IF(IEQ.GT.NU)THEN
                    AL(2)=GCOEF
               ELSE
                    AU(2,IEQ)=GCOEF
               ENDIF
           ENDIF
          REQ=REQ-GCOEF*TMP(I+1,J,K)
      ENDIF
     IF(K.EQ.1)THEN
          DCOEF=DCOEF-DP(I,J)
          REQ=REQ-DP(I,J)*TTOP-SP*TEMPR*RCH(I,J)
     ELSE
          CCOEF=0.
          DPART=0.
          ITKM1=IT(I,J,K-1)
          IF(ITKM1.NE.1)THEN
               DCOEF=DCOEF-DP(I,J)
               REQ=REQ-DP(I,J)*TTOP-SP*TEMPR*RCH(I,J)
          ENDIF
          IF(ITKM1.NE.3)THEN
               CCOEF=KXY2/(DZK+DZ(K-1))
               DPART=-CCOEF
          ENDIF
          Q5=Q(IEQ,5)
```

Attachment 1, continued

```
          IF(Q5.GT.0.)THEN
               CCOEF=CCOEF+SP*Q5
          ELSE IF(Q5.LT.0.)THEN
               DPART=DPART+SP*Q5
          ENDIF
          DCOEF=DCOEF+DPART
          IF(IEQ.LE.NU)THEN
               AU(5,IEQ)=CCOEF
          ELSE
               AL(5)=CCOEF
          ENDIF
          REQ=REQ-CCOEF*TMP(I,J,K-1)
     ENDIF
     IF(K.EQ.L)THEN
          REQ=REQ-HF(I,J)
     ELSE
          ITKP1=IT(I,J,K+1)
          ECOEF=0.
          DPART=0.
          IF(ITKP1.NE.3)THEN
               ECOEF=KXY2/(DZK+DZ(K+1))
               DPART=-ECOEF
          ENDIF
          Q6=Q(IEQ,6)
          IF(Q6.GT.0.)THEN
               ECOEF=ECOEF+SP*Q6
          ELSE IF(Q6.LT.0.)THEN
               DPART=DPART+SP*Q6
          ENDIF
               DCOEF=DCOEF+DPART
          IF(ITKP1.EQ.1)THEN
               IF(IEQ.GT.NU)THEN
                    AL(6)=ECOEF
               ELSE
                    AU(6,IEQ)=ECOEF
               ENDIF
          ENDIF
          REQ=REQ-ECOEF*TMP(I,J,K+1)
     ENDIF
     IF(IEQ.LE.NU)THEN
          RU(IEQ)=REQ-DCOEF*TMP(I,J,K)
          DU(IEQ)=DCOEF
     ELSE
          RL(IL)=REQ-DCOEF*TMP(I,J,K)
          IA=IA+IBW
          DO 4 ISUB=IA+1,IA+IBW
 4        A(ISUB)=0.
          A(IA+MD)=DCOEF
          DO 6 II=1,6
          IXII=I+IC(II)
          IF((IXII.LE.0).OR.(IXII.GT.M))GOTO6
          JXII=J+JC(II)
```

Attachment 1, continued

```
           IF((JXII.LE.0).OR.(JXII.GT.N))GOTO6                          HEAT207
           KXII=K+KC(II)                                                HEAT208
           IF((KXII.LE.0).OR.(KXII.GT.L))GOTO6                          HEAT209
           NPQ=NP(IXII,JXII,KXII)                                       HEAT210
           IF(NPQ.EQ.0)GOTO6                                            HEAT211
           DO 5 JJ=1,6                                                  HEAT212
           IXJJ=I+IC(JJ)+IC(II)                                         HEAT213
           IF((IXJJ.LE.0).OR.(IXJJ.GT.M))GOTO5                          HEAT214
           JXJJ=J+JC(JJ)+JC(II)                                         HEAT215
           IF((JXJJ.LE.0).OR.(JXJJ.GT.N))GOTO5                          HEAT216
           KXJJ=K+KC(JJ)+KC(II)                                         HEAT217
           IF((KXJJ.LE.0).OR.(KXJJ.GT.L))GOTO5                          HEAT218
           NPP=NP(IXJJ,JXJJ,KXJJ)                                       HEAT219
           IF(NPP.EQ.0)GOTO5                                            HEAT220
           NN=IA+MD-IEQ+NPP                                             HEAT221
           A(NN)=A(NN)-AU(JJ,NPQ)*AL(II)/DU(NPQ)                        HEAT222
 5         CONTINUE                                                     HEAT223
           RL(IL)=RL(IL)-RU(NPQ)*AL(II)/DU(NPQ)                         HEAT224
 6         CONTINUE                                                     HEAT225
      ENDIF                                                             HEAT226
 7    CONTINUE                                                          HEAT227
      RETURN                                                            HEAT228
      END                                                               HEAT229
```

```
C      BAL - Computes water and heat balances.
       SUBROUTINE BAL(TMP,HC,DX,DY,DZ,IT,HF,DP,RCH,IH,HD)
       LOGICAL LFIN,LHCPT,LVPT
       REAL KT,KXY,KXZ,KYZ
       REAL*8 HD,TMP
       CHARACTER*8 XMESUR,ZMESUR
       DIMENSION TMP(M,N,L),HC(M,N,L),DX(N),DY(M),DZ(L),IT(M,N,L),
     c HF(M,N),DP(M,N),RCH(M,N),IH(M,N,L),HD(M,N,L)
       COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
     c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
     c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
       CTO=.0
       CTI=0.
       CFO=0.
       CFI=0.
       CVO=0.
       CVI=0.
       CHO=0.
       CHI=0.
       CSO=0.
       CSI=0.
       RECH=0.
       DO 1 K=1,L
       DO 1 J=1,N
       DO 1 I=1,M
       IF(IT(I,J,K).EQ.1)THEN
            IF(I.NE.1)THEN
                 IF(IT(I-1,J,K).EQ.2)THEN
                      CT=(TMP(I,J,K)-TMP(I-1,J,K))*KT*DX(J)*DZ(K)*2.
     c                /(DY(I)+DY(I-1))
                      IF(CT.LT.0.)THEN
                           CTI=CTI-CT
                      ELSE IF(CT.GT.0.)THEN
                           CTO=CTO+CT
                      ENDIF
                 ENDIF
                 IF(IH(I-1,J,K).EQ.2)THEN
                      CH=(HD(I,J,K)-HD(I-1,J,K))*HC(I-1,J,K)*HC(I,J,K)
     c                *2.*DX(J)*DZ(K)
     c                /(HC(I-1,J,K)*DY(I)+HC(I,J,K)*DY(I-1))
                      IF(CH.LT.0.)THEN
                           CHI=CHI-CH
                           CVI=CVI-CH*SP*TMP(I-1,J,K)
                      ELSE IF(CH.GT.0.)THEN
                           CHO=CHO+CH
                           CVO=CVO+CH*SP*TMP(I,J,K)
                      ENDIF
                 ENDIF
            ENDIF
            IF(I.NE.M)THEN
                 IF(IT(I+1,J,K).EQ.2)THEN
                      CT=(TMP(I,J,K)-TMP(I+1,J,K))*KT*DX(J)*DZ(K)*2.
```

Attachment 1, continued

```
     c               /(DY(I)+DY(I+1))
                     IF(CT.LT.0.)THEN
                          CTI=CTI-CT
                     ELSE IF(CT.GT.0.)THEN
                          CTO=CTO+CT
                     ENDIF
                ENDIF
                IF(IH(I+1,J,K).EQ.2)THEN
                     CH=(HD(I,J,K)-HD(I+1,J,K))*HC(I+1,J,K)*HC(I,J,K)
     c               *2.*DX(J)*DZ(K)
     c               /(HC(I+1,J,K)*DY(I)+HC(I,J,K)*DY(I+1))
                     IF(CH.LT.0.)THEN
                          CHI=CHI-CH
                          CVI=CVI-CH*SP*TMP(I+1,J,K)
                     ELSE IF(CH.GT.0.)THEN
                          CHO=CHO+CH
                          CVO=CVO+CH*SP*TMP(I,J,K)
                     ENDIF
                ENDIF
           ENDIF
           IF(J.NE.1)THEN
                IF(IT(I,J-1,K).EQ.2)THEN
                     CT=(TMP(I,J,K)-TMP(I,J-1,K))*KT*DY(I)*DZ(K)*2.
     c               /(DX(J)+DX(J-1))
                     IF(CT.LT.0.)THEN
                          CTI=CTI-CT
                     ELSE IF(CT.GT.0.)THEN
                          CTO=CTO+CT
                     ENDIF
                ENDIF
                IF(IH(I,J-1,K).EQ.2)THEN
                     CH=(HD(I,J,K)-HD(I,J-1,K))*HC(I,J-1,K)*HC(I,J,K)
     c               *2.*DY(I)*DZ(K)
     c               /(HC(I,J-1,K)*DX(J)+HC(I,J,K)*DX(J-1))
                     IF(CH.LT.0.)THEN
                          CHI=CHI-CH
                          CVI=CVI-CH*SP*TMP(I,J-1,K)
                     ELSE IF(CH.GT.0.)THEN
                          CHO=CHO+CH
                          CVO=CVO+CH*SP*TMP(I,J,K)
                     ENDIF
                ENDIF
           ENDIF
           IF(J.NE.N)THEN
                IF(IT(I,J+1,K).EQ.2)THEN
                     CT=(TMP(I,J,K)-TMP(I,J+1,K))*KT*DY(I)*DZ(K)*2.
     c               /(DX(J)+DX(J+1))
                     IF(CT.LT.0.)THEN
                          CTI=CTI-CT
                     ELSE IF(CT.GT.0.)THEN
                          CTO=CTO+CT
                     ENDIF
```

```
      ENDIF                                                             BAL104
      IF(IH(I,J+1,K).EQ.2)THEN                                          BAL105
```

Attachment 1, continued

```
                 CH=(HD(I,J,K)-HD(I,J+1,K))*HC(I,J+1,K)*HC(I,J,K)
c                *2.*DY(I)*DZ(K)
c                /(HC(I,J+1,K)*DX(J)+HC(I,J,K)*DX(J+1))
                 IF(CH.LT.0.)THEN
                      CHI=CHI-CH
                      CVI=CVI-CH*SP*TMP(I,J+1,K)
                 ELSE IF(CH.GT.0.)THEN
                      CHO=CHO+CH
                      CVO=CVO+CH*SP*TMP(I,J,K)
                 ENDIF
            ENDIF
       ENDIF
       IF(K.EQ.1.OR.IT(I,J,K-1).NE.1)THEN
            CS=(TTOP-TMP(I,J,K))*DP(I,J)
            IF(CS.LT.0.)THEN
                 CSO=CSO-CS
            ELSE IF(CS.GT.0.)THEN
                 CSI=CSI+CS
            ENDIF
            RECH=RECH+RCH(I,J)
            CVI=CVI+SP*TEMPR*RCH(I,J)
       ENDIF
       IF(K.NE.1)THEN
            IF(IT(I,J,K-1).EQ.2)THEN
                 CT=(TMP(I,J,K)-TMP(I,J,K-1))*KT*DX(J)*DY(I)*2.
c                /(DZ(K)+DZ(K-1))
                 IF(CT.LT.0.)THEN
                      CTI=CTI-CT
                 ELSE IF(CT.GT.0.)THEN
                      CTO=CTO+CT
                 ENDIF
            ENDIF
            IF(IH(I,J,K-1).EQ.2)THEN
                 CH=(HD(I,J,K)-HD(I,J,K-1))*HC(I,J,K-1)*HC(I,J,K)
c                *2.*DX(J)*DY(I)
c                /(HC(I,J,K-1)*DZ(K)+HC(I,J,K)*DZ(K-1))
                 IF(CH.LT.0.)THEN
                      CHI=CHI-CH
                      CVI=CVI-CH*SP*TMP(I,J,K-1)
                 ELSE IF(CH.GT.0.)THEN
                      CHO=CHO+CH
                      CVO=CVO+CH*SP*TMP(I,J,K)
                 ENDIF
            ENDIF
       ENDIF
       IF(K.EQ.L)THEN
            IF(HF(I,J).LT.0.)THEN
                 CFO=CFO-HF(I,J)
            ELSE IF(HF(I,J).GT.0.)THEN
                 CFI=CFI+HF(I,J)
            ENDIF
       ELSE
```

Attachment 1, continued

```
            IF(IT(I,J,K+1).EQ.2)THEN
                 CT=(TMP(I,J,K)-TMP(I,J,K+1))*KT*DX(J)*DY(I)*2.
     c           /(DZ(K)+DZ(K+1))
                 IF(CT.LT.0.)THEN
                      CTI=CTI-CT
                 ELSE IF(CT.GT.0.)THEN
                      CTO=CTO+CT
                 ENDIF
            ENDIF
            IF(IH(I,J,K+1).EQ.2)THEN
                 CH=(HD(I,J,K)-HD(I,J,K+1))*HC(I,J,K+1)*HC(I,J,K)
     c           *2.*DX(J)*DY(I)
     c           /(HC(I,J,K+1)*DZ(K)+HC(I,J,K)*DZ(K+1))
                 IF(CH.LT.0.)THEN
                      CHI=CHI-CH
                      CVI=CVI-CH*SP*TMP(I,J,K+1)
                 ELSE IF(CH.GT.0.)THEN
                      CHO=CHO+CH
                      CVO=CVO+CH*SP*TMP(I,J,K)
                 ENDIF
            ENDIF
       ENDIF
      ENDIF
    1 CONTINUE
      HBAL=0.
      DENOM=CVO+CVI+CSO+CSI+CFO+CFI+CTO+CTI
      HNUM=CVI+CSI+CFI+CTI-(CVO+CSO+CFO+CTO)
      IF(DENOM.NE.0.)HBAL=100.*HNUM/DENOM
      WBAL=0.
      DENOM=CHO+CHI+RECH
      WNUM=CHI+RECH-CHO
      IF(DENOM.NE.0.)WBAL=100.*WNUM/DENOM
      WRITE(6,11)CVI,CVO,CFI,CFO,CSI,CSO,CTI,CTO,HNUM,HBAL,
     c RECH,CHI,CHO,WNUM,WBAL
      WRITE(6,7)
      DO 2 K=1,L
      WRITE(6,6)K
      DO 2 I=1,M
    2 WRITE(6,9) (TMP(I,J,K),J=1,N)
      WRITE(6,8)
      DO 3 K=1,L
      WRITE(6,6)K
      DO 3 I=1,M
    3 WRITE(6,9)(HD(I,J,K),J=1,N)
      IF(.NOT.LHCPT)GOTO5
      WRITE(6,10)
      DO 4 K=1,L
      WRITE(6,6)K
      DO 4 I=1,M
    4 WRITE(6,9)(HC(I,J,K),J=1,N)
    5 CONTINUE
      RETURN
```

Attachement 1, continued

```
    6 FORMAT(/1X,'LAYER ',I2/1X,8(1H-))                                      BAL210
    7 FORMAT(/1X,'TEMPERATURE, DEG. F'/1X,19(1H-))                           BAL211
    8 FORMAT(/1X,'HEAD, FEET'/1X,10(1H-))                                    BAL212
    9 FORMAT(1X,1P10E12.4)                                                   BAL213
   10 FORMAT(/1X,'HYD. COND. (FT./DAY) AT COMP. TEMP.'/1X,35(1H-))           BAL214
   11 FORMAT(/13X,'STRESS COMPONENTS FOR HEAT, UNITS = BTU/DAY'/             BAL215
     c 13X,43(1H-)/                                                          BAL216
     c 33X,'IN',11X,'OUT'/19X,'ADVECTION = ',1PE12.4,1X,E12.4/               BAL217
     c 8X,'HEAT FLOW FROM BELOW = ',E12.4,1X,E12.4/                          BAL218
     c 8X,'HEAT FLOW TO SURFACE = ',E12.4,1X,E12.4/                          BAL219
     c 3X,'CONSTANT TEMP. BOUNDARIES = ',E12.4,1X,E12.4/                     BAL220
     c 22X,'IN-OUT = ',E12.4/14X,'HEAT IMBALANCE = ',0PF7.2,' PERCENT'/      BAL221
     c /12X,'STRESS COMPONENTS FOR WATER, UNITS = CUBIC FT./DAY'/            BAL222
     c 12X,50(1H-)/                                                          BAL223
     c 33X,'IN',11X,'OUT'/                                                   BAL224
     c 20X,'RECHARGE = ',1PE12.4/                                            BAL225
     c 4X,'CONSTANT HEAD BOUNDARIES = ',E12.4,1X,E12.4/                      BAL226
     c 22X,'IN-OUT = ',E12.4/13X,'WATER IMBALANCE = ',0PF7.2,' PERCENT')     BAL227
      END                                                                    BAL228
```

Attachment 1, continued

```
C      VELO - Determines velocities.
       SUBROUTINE VELO(IH,HD,HC,DX,DY,DZ,
     c VX1,VX2,VY1,VY2,VZ1,VZ2,POR,RCH)
       LOGICAL LFIN,LHCPT,LVPT
       REAL KT
       REAL*8 HD,HIJK
       CHARACTER*8 XMESUR,ZMESUR
       DIMENSION IH(M,N,L),HD(M,N,L),HC(M,N,L),
     c DX(N),DY(M),DZ(L),VX1(M,N,L),VX2(M,N,L),VY1(M,N,L),
     c VY2(M,N,L),VZ1(M,N,L),VZ2(M,N,L),POR(M,N,L),RCH(M,N)
       COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
     c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
     c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
       DO 1 K=1,L
       DO 1 J=1,N
       DO 1 I=1,M
       IHIJK=IH(I,J,K)
       IF(IHIJK.EQ.1)THEN
            HIJK=HD(I,J,K)
            HCIJK=HC(I,J,K)
            PORIJK=POR(I,J,K)
            IF((J.EQ.1).OR.(IH(I,J-1,K).EQ.3))THEN
                 VX1(I,J,K)=0.
            ELSE
                 VX1(I,J,K)=2.*HCIJK*HC(I,J-1,K)/(HCIJK*DX(J-1)
     c           +HC(I,J-1,K)*DX(J))*
     c           (HD(I,J-1,K)-HIJK)/PORIJK
            ENDIF
            IF((J.EQ.N).OR.(IH(I,J+1,K).EQ.3))THEN
                 VX2(I,J,K)=0.
            ELSE
                 VX2(I,J,K)=2.*HCIJK*HC(I,J+1,K)/(HCIJK*DX(J+1)
     c           +HC(I,J+1,K)*DX(J))*
     c           (HIJK-HD(I,J+1,K))/PORIJK
            ENDIF
            IF((I.EQ.1).OR.(IH(I-1,J,K).EQ.3))THEN
                 VY1(I,J,K)=0.
            ELSE
                 VY1(I,J,K)=2.*HCIJK*HC(I-1,J,K)/(HCIJK*DY(I-1)
     c           +HC(I-1,J,K)*DY(I))*
     c           (HD(I-1,J,K)-HIJK)/PORIJK
            ENDIF
            IF((I.EQ.M).OR.(IH(I+1,J,K).EQ.3))THEN
                 VY2(I,J,K)=0.
            ELSE
                 VY2(I,J,K)=2.*HCIJK*HC(I+1,J,K)/(HCIJK*DY(I+1)
     c           +HC(I+1,J,K)*DY(I))*
     c           (HIJK-HD(I+1,J,K))/PORIJK
            ENDIF
            IF((K.EQ.1).OR.(IH(I,J,K-1).EQ.3))THEN
                 VZ1(I,J,K)=RCH(I,J)/(DX(J)*DY(I)*PORIJK)
            ELSE
```

Attachment 1, continued

```
               VZ1(I,J,K)=2.*HCIJK*HC(I,J,K-1)/(HCIJK*DZ(K-1)
     c         +HC(I,J,K-1)*DZ(K))*
     c         (HD(I,J,K-1)-HIJK)/PORIJK
            ENDIF
            IF((K.EQ.L).OR.(IH(I,J,K+1).EQ.3))THEN
               VZ2(I,J,K)=0.
            ELSE
               VZ2(I,J,K)=2.*HCIJK*HC(I,J,K+1)/(HCIJK*DZ(K+1)
     c         +HC(I,J,K+1)*DZ(K))*
     c         (HIJK-HD(I,J,K+1))/PORIJK
            ENDIF
        ELSE
            VX1(I,J,K)=0.
            VX2(I,J,K)=0.
            VY1(I,J,K)=0.
            VY2(I,J,K)=0.
            VZ1(I,J,K)=0.
            VZ2(I,J,K)=0.
        ENDIF
    1   CONTINUE
        IF(LVPT)THEN
            WRITE(6,5)
            DO 2 K=1,L
            WRITE(6,6)K
            DO 2 I=1,M
    2       WRITE(6,7)(VX1(I,J,K),VX2(I,J,K),J=1,N)
            WRITE(6,8)
            DO 3 K=1,L
            WRITE(6,6)K
            DO 3 I=1,M
    3       WRITE(6,7)(VY1(I,J,K),VY2(I,J,K),J=1,N)
            WRITE(6,9)
            DO 4 K=1,L
            WRITE(6,6)K
            DO 4 I=1,M
    4       WRITE(6,7)(VZ1(I,J,K),VZ2(I,J,K),J=1,N)
        ENDIF
        RETURN
    5   FORMAT (1H1,21HX VELOCITIES, FT./DAY)
    6   FORMAT(1X,'LAYER ',I2/1X,8(1H-))
    7   FORMAT (1H ,8G12.3)
    8   FORMAT (1H1,21HY VELOCITIES, FT./DAY)
    9   FORMAT (1H1,21HZ VELOCITIES, FT./DAY)
        END
```

Attachment 1, continued

```
C     MOVE - Determines location of discharge and travel time to
C        discharge point for each active node.
      SUBROUTINE MOVE(IH,VX1,VX2,VY1,VY2,VZ1,VZ2,DX,DY,DZ,TIM,IDS)
      LOGICAL LFIN,LHCPT,LVPT,LPATH
      REAL KT
      CHARACTER*8 XMESUR,ZMESUR
      DIMENSION IH(M,N,L),VX1(M,N,L),VX2(M,N,L),VY1(M,N,L),
     c VY2(M,N,L),VZ1(M,N,L),VZ2(M,N,L),
     c DX(N),DY(M),DZ(L),TIM(M,N,L),IDS(M,N,L)
      COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
     c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
     c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
      DATA LPATH/.FALSE./
      DO 1 K=1,L
      DO 1 J=1,N
      DO 1 I=1,M
      IDSYXZ=0
      TISUM=0.
      IF(IH(I,J,K).EQ.1)THEN
           INY=I
           INX=J
           INZ=K
           X0=DX(J)/2.
           Y0=DY(I)/2.
           Z0=DZ(K)/2.
           CALL MSUB(X0,Y0,Z0,INY,INX,INZ,IDSYXZ,VX1,VX2,
     c      VY1,VY2,VZ1,VZ2,DX,DY,DZ,IH,TISUM,LPATH,
     c      DISTX,DISTZ,NREC,NRLIM,PATH,IPA,KSYM,XSF,YSF)
      ENDIF
      TISUM=TISUM/365.25
      TIM(I,J,K)=TISUM
      IDS(I,J,K)=IDSYXZ
    1 CONTINUE
      J1=1
      J2=12
    2 IF(J2.GT.N)J2=N
      WRITE(6,8)
      IF(M.EQ.1)WRITE(6,9)(J,J=J1,J2)
      DO 3 K=1,L
      WRITE(6,10)K
      IF(M.GT.1)WRITE(6,9)(J,J=J1,J2)
      DO 3 I=1,M
    3 WRITE(6,11)I,(TIM(I,J,K),J=J1,J2)
      J1=J1+12
      IF(J1.GT.N)GOTO4
      J2=J2+12
      GOTO2
    4 J1=1
      J2=12
    5 IF(J2.GT.N)J2=N
      WRITE(6,12)
      IF(M.EQ.1)WRITE(6,9)(J,J=J1,J2)
```

Attachment 1, continued

```
      DO  6 K=1,L
      WRITE(6,10)K
      IF(M.GT.1)WRITE(6,9)(J,J=J1,J2)
      DO  6 I=1,M
    6 WRITE(6,13)I,(IDS(I,J,K),J=J1,J2)
      J1=J1+12
      IF(J1.GT.N)GOTO7
      J2=J2+12
      GOTO5
    7 CONTINUE
      RETURN
    8 FORMAT(1H1,'TIME OF TRAVEL TO DISCHARGE POINT, YEARS')
    9 FORMAT(6X,12I10)
   10 FORMAT(1X,'LAYER ',I2/1X,8(1H-))
   11 FORMAT(1H ,I3,2X,1P12E10.2)
   12 FORMAT(1H1,'DISCHARGE POINT:Y,X,Z')
   13 FORMAT(1H ,I3,2X,12I10)
      END
```

Attachment 1, continued

```
C       MSUB - Moves a water particle from initial point to discharge.
        SUBROUTINE MSUB(X0,Y0,Z0,I,J,K,IDSYXZ,VX1,VX2,
      c VY1,VY2,VZ1,VZ2,DX,DY,DZ,IH,TISUM,LPATH,
      c DISTX,DISTZ,NREC,NRLIM,PATH,IPATH,KSYM,XSF,ZSF)
        LOGICAL LFIN,LHCPT,LVPT,LPATH
        REAL KT
        CHARACTER*8 XMESUR,ZMESUR
        DIMENSION IH(M,N,L),VX1(M,N,L),VX2(M,N,L),VY1(M,N,L),
      c VY2(M,N,L),VZ1(M,N,L),VZ2(M,N,L),
      c DX(N),DY(M),DZ(L),DISTX(N),DISTZ(L)
        COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
      c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
      c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
        IF(LPATH)THEN
             IF(KSYM.GE.50)THEN
                  KSYM=1
             ELSE
                  KSYM=KSYM+1
             ENDIF
        ENDIF
        X=X0
        Y=Y0
        Z=Z0
        IORG=I
        JORG=J
        KORG=K
        LIMIT=30*(N+M+L)
        NMOVE=0
        TISUM=0.
        TX=1.E+38
        TY=1.E+38
        TZ=1.E+38
        IF(LPATH)THEN
             WRITE(6,2)J,K,X0,Z0,TISUM
             CALL PSUB(DISTX,DX,XSF,DISTZ,ZSF,DZ,
      c       IPATH,PATH,NREC,NRLIM,KSYM,X0,Z0,J,K)
        ENDIF
      1 DXJ=DX(J)
        DYI=DY(I)
        DZK=DZ(K)
        IF(N.GT.1)THEN
             VX01=VX1(I,J,K)
             VX02=VX2(I,J,K)
             CALL    MSUB1(X0,DXJ,VX0,VX01,VX02,BX,TX,J,INX)
        ENDIF
        IF(M.GT.1)THEN
             VY01=VY1(I,J,K)
             VY02=VY2(I,J,K)
             CALL    MSUB1(Y0,DYI,VY0,VY01,VY02,BY,TY,I,INY)
        ENDIF
        IF(L.GT.1)THEN
             VZ01=VZ1(I,J,K)
```

Attachment 1, continued

```
          VZ02=VZ2(I,J,K)
          CALL   MSUB1(Z0,DZK,VZ0,VZ01,VZ02,BZ,TZ,K,INZ)
      ENDIF
      TIME=AMIN1(TX,TY,TZ)
      IF(TIME.EQ.1.E+38)THEN
          WRITE(1,3)I,J,K,IORG,JORG,KORG
          WRITE(6,3)I,J,K,IORG,JORG,KORG
          RETURN
      ENDIF
      IF(TIME.LT.TX)INX=J
      IF(TIME.LT.TY)INY=I
      IF(TIME.LT.TZ)INZ=K
      TISUM=TISUM+TIME
      IF(N.GT.1)CALL  MSUB2(X,X0,DXJ,VX0,VX01,VX02,BX,TIME)
      IF(M.GT.1)CALL  MSUB2(Y,Y0,DYI,VY0,VY01,VY02,BY,TIME)
      IF(L.GT.1)CALL  MSUB2(Z,Z0,DZK,VZ0,VZ01,VZ02,BZ,TIME)
      X0=X
      Y0=Y
      Z0=Z
      IF((INY.LT.1).OR.(INY.GT.M).OR.(INX.LT.1).OR.(INX.GT.N).OR.
     c (INZ.LT.1).OR.(INZ.GT.L).OR.(IH(INY,INX,INZ).NE.1))THEN
          IDSYXZ=100*(100*INY+INX)+INZ
          IF(LPATH)THEN
              WRITE(6,2)J,K,X,Z,TISUM
              CALL PSUB(DISTX,DX,XSF,DISTZ,ZSF,DZ,
     c        IPATH,PATH,NREC,NRLIM,KSYM,X,Z,J,K)
              TISUM=TISUM/365.25
              WRITE(6,4)J,K,INX,INZ,TISUM
          ENDIF
          RETURN
      ELSE
          NMOVE=NMOVE+1
          IF(NMOVE.GT.LIMIT)THEN
              WRITE(1,5)I,J,K,IORG,JORG,KORG
              WRITE(6,5)I,J,K,IORG,JORG,KORG
              RETURN
          ELSE
              IF(INX.GT.J)THEN
                  X0=0.
              ELSE IF(INX.LT.J)THEN
                  X0=DX(INX)
              ENDIF
              IF(INY.GT.I)THEN
                  Y0=0.
              ELSE IF(INY.LT.I)THEN
                  Y0=DY(INY)
              ENDIF
              IF(INZ.GT.K)THEN
                  Z0=0.
              ELSE IF(INZ.LT.K)THEN
                  Z0=DZ(INZ)
              ENDIF
```

Attachment 1, continued

```
            J=INX
            I=INY
            K=INZ
            IF(LPATH)THEN
                 WRITE(6,2)J,K,X0,Z0,TISUM
                 CALL PSUB(DISTX,DX,XSF,DISTZ,ZSF,DZ,
c                 IPATH,PATH,NREC,NRLIM,KSYM,X0,Z0,J,K)
            ENDIF
            GOTO1
       ENDIF
  ENDIF
2 FORMAT(1X,2I3,2F7.0,1P4E10.3)
3 FORMAT(1H ,'STAGNATION POINT AT',3I5,' ORIGIN AT',3I5)
4 FORMAT(1X,4I3,1PE10.3)
5 FORMAT(1H ,'MOVE LIMIT REACHED AT ',3I5,' ORIGIN AT ',3I5)
  END
```

Attachment 1, continued

```
C     MSUB1 - Determines position, velocity, and travel time to
C        next block face.
      SUBROUTINE MSUB1(W0,DW,V0,V1,V2,B,TIME,IW,INW)
      DV=V2-V1
      B=DV/DW
      IF((W0/DW).LT..5)THEN
           V0=V1+B*W0
      ELSE
           V0=V2-B*(DW-W0)
      ENDIF
      IF(V0.LT.0.)THEN
           IF(V1.LT.0.)THEN
                INW=IW-1
                VRL=ALOG(V1/V0)
                IF(ABS(VRL).LT.1.5E-3)GOTO1
                TIME=VRL/B
                IF(TIME.LE.0.)GOTO1
           ELSE
                INW=IW
                TIME=1.E+38
           ENDIF
      ELSE IF((V0.EQ.0.).OR.(V2.LE.0.))THEN
           INW=IW
           TIME=1.E+38
      ELSE
           INW=IW+1
           VRL=ALOG(V2/V0)
           IF(ABS(VRL).LT.1.5E-3)GOTO1
           TIME=VRL/B
           IF(TIME.LE.0.)GOTO1
      ENDIF
      RETURN
    1 IF(V0.GT.0.)THEN
           INW=IW+1
           TIME=(DW-W0)/V0
      ELSE
           INW=IW-1
           TIME=-W0/V0
      ENDIF
      RETURN
      END
```

Attachment 1, continued

```
C     MSUB2 - Determines new position after an increment of time.
      SUBROUTINE MSUB2(W,W0,DW,V0,V1,V2,B,TIME)
      IF(V0.EQ.0.)THEN
           W=W0
      ELSE
           BT=B*TIME
           IF(ABS(BT).LT.1.5E-3)THEN
                WT=W0+V0*TIME
           ELSE
                WT=(V0*EXP(BT)-V1)/B
           ENDIF
           IF(WT.GT.DW)THEN
                WT=DW
           ELSE IF(WT.LT.0.)THEN
                WT=0.
           ENDIF
           IF(V1.EQ.0.)THEN
                ZERO=0.
           ELSE IF(V2.EQ.0.)THEN
                ZERO=DW
           ELSE
                IF(B.EQ.0.)THEN
                     W=WT
                     RETURN
                ELSE
                     ZERO=-V1/B
                ENDIF
           ENDIF
           IF(V0.LT.0.)THEN
                IF((ZERO.GE.W0).OR.(ZERO.LT.0.))THEN
                     W=AMAX1(WT,0.)
                ELSE
                     CALL ZER(WTT,ZERO,+1)
                     W=AMAX1(WT,WTT)
                ENDIF
           ELSE IF((ZERO.LE.W0).OR.(ZERO.GT.DW))THEN
                W=AMIN1(WT,DW)
           ELSE
                CALL ZER(WTT,ZERO,-1)
                W=AMIN1(WT,WTT)
           ENDIF
      ENDIF
      RETURN
      END
```

Attachment 1, continued

```
                IF(JM.GT.4194304)THEN
                     IF(JE.LT.255)THEN
                          JM=4194303
                          JE=JE+1
                     ELSE
                          JM=0
                     ENDIF
                ENDIF
           ENDIF
      ENDIF
      JM=OR(JM,JS)
      DO 2 ISUB=1,3
    2 CR(ISUB)=CM(ISUB+1)
      CR(4)=CE(4)
      WNEAR=R
      RETURN
      END
```

Attachment 1, continued

```
C     ZER - Determines a position near, but upstream from, a point of
C        zero velocity.
      SUBROUTINE ZER(WNEAR,WZERO,IDIR)
      CHARACTER CR(4),CM(4),CE(4),CS(4)
      EQUIVALENCE (R,CR(1)),(JM,CM(1)),(JE,CE(1)),(JS,CS(1))
      JM=0
      JE=0
      JS=0
      R=WZERO
      CE(4)=CR(4)
      CS(2)=CR(1)
      JS=AND(JS,8388608)
      IF(R.EQ.0.)THEN
           IF(IDIR.GT.0)THEN
                JM=4194304
           ELSE
                JM=4194303
                JS=8388608
           ENDIF
      ELSE
           DO 1 ISUB=1,3
    1      CM(ISUB+1)=CR(ISUB)
           JM=AND(JM,8388607)
           JM=JM+IDIR
           IF(JS.EQ.0)THEN
                IF(JM.GT.8388607)THEN
                     IF(JE.LT.255)THEN
                          JM=4194304
                          JE=JE+1
                     ELSE
                          JM=8388607
                     ENDIF
                ENDIF
                IF(JM.LT.4194304)THEN
                     IF(JE.GT.0)THEN
                          JM=8388607
                          JE=JE-1
                     ELSE
                          JM=0
                     ENDIF
                ENDIF
           ELSE
                JM=AND(JM,16777215)
                IF(JM.EQ.4194304)THEN
                     IF(JE.GT.0)THEN
                          JM=0
                          JE=JE-1
                     ELSE
                          JM=0
                          JS=0
                     ENDIF
                ENDIF
```

```
C     PATH - Traces selected flow lines from recharge area to
C        discharge area.
      SUBROUTINE PATH(RCH,IH,VX1,VX2,VZ1,VZ2,
     c DX,DY,DZ,DISTX,DISTZ,RJ,NREC,PATH,Q,NP)
      LOGICAL LFIN,LHCPT,LVPT,LLEFT,LVSW,LPATH
      REAL KT
      CHARACTER PATH(1)
      CHARACTER*8 XMESUR,ZMESUR
      DIMENSION RCH(N),IH(N,L),VX1(N,L),VX2(N,L),VZ1(N,L),VZ2(N,L),
     c DX(N),DY(M),DZ(L),DISTX(N),DISTZ(L)
      DIMENSION RJ(N,3),PER(7),Q(NEQ,6),NP(N,L)
      COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
     c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
     c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
      DATA LPATH/.TRUE./
      DATA PER/.5,.1,.01,.001,.0001,.00001,.000001/
      DATA NPER/7/
      Y0=DY(1)/2.
      IY=1
      NRLIM=NREC
      IPATH=0
      NREC=0
      LLEFT=.FALSE.
      LVSW=.FALSE.
      LFIN=.FALSE.
      RL=0.
      XSF=DINCHX*XSCALE
      ZSF=DINCHZ*ZSCALE
      RJS=0.
      DO 2 J=1,N
      DO 1 K=1,L
      IF(IH(J,K).EQ.1)THEN
           IEQ=NP(J,K)
           IF((J.GT.1).AND.(IH(J-1,K).EQ.2).AND.
     c     (Q(IEQ,3).GT.0.))RJS=RJS+Q(IEQ,3)
           RJ(J,1)=RJS
           IF((K.GT.1).AND.(IH(J,K-1).EQ.2).AND.
     c     (Q(IEQ,5).GT.0.))THEN
                RJS=RJS+Q(IEQ,5)
           ELSE
                RJS=RJS+RCH(J)
           ENDIF
           RJ(J,2)=RJS
           IF((J.LT.N).AND.(IH(J+1,K).EQ.2).AND.
     c     (Q(IEQ,4).GT.0.))RJS=RJS+Q(IEQ,4)
           RJ(J,3)=RJS
           GOTO2
      ENDIF
    1 CONTINUE
    2 CONTINUE
      WRITE(6,16)(J,(RJ(J,JJ),JJ=1,3),J=1,N)
      DISTX(1)=0.
```

Attachment 1, continued

```
      DO 3 J=2,N
    3 DISTX(J)=DISTX(J-1)+DX(J-1)
      DISTZ(L)=DZ(L)
      DO 4 K=L-1,1,-1
    4 DISTZ(K)=DISTZ(K+1)+DZ(K)
      DO 5 J=1,N
      DO 5 K=1,L
      IF(IH(J,K).EQ.1)GOTO6
    5 CONTINUE
    6 IF((VZ1(J,K).LT.0.).OR.(VX1(J,K).LT.0.))LLEFT=.TRUE.
      J=1
      K=0
    7 IF(LFIN)RETURN
      IF(J.EQ.1)THEN
           K=K+1
           IF(K.GT.L)THEN
                K=L
                J=J+1
                TEST=VX1(J,K)*VX2(J,K)
           ELSE
                TEST=VZ1(J,K)*VZ2(J,K)
           ENDIF
      ELSE IF(K.EQ.L)THEN
           J=J+1
            IF(J.GT.N)THEN
                J=N
                K=K-1
                TEST=VZ1(J,K)*VZ2(J,K)
           ELSE
                TEST=VX1(J,K)*VX2(J,K)
           ENDIF
      ELSE IF(J.EQ.N)THEN
           K=K-1
           IF((K.EQ.0).OR.(IH(J,K).NE.1))THEN
                RR=RJ(N,3)
                LFIN=.TRUE.
                GOTO10
           ELSE
                TEST=VZ1(J,K)*VZ2(J,K)
           ENDIF
      ENDIF
      IF(TEST.GE.0.)GOTO7
      JBEG=J
      KBEG=K
      IF((J.GT.1).AND.(J.LT.N))THEN
           X0=-VX1(J,K)/(VX2(J,K)-VX1(J,K))*DX(J)
           CALL ZER(Z0,DZ(L),-1)
      ELSE
           Z0=-VZ1(J,K)/(VZ2(J,K)-VZ1(J,K))*DZ(K)
           IF(J.EQ.1)THEN
                CALL ZER(X0,0.,1)
           ELSE IF(J.EQ.N)THEN
```

```
                CALL ZER(X0,DX(N),-1)
             ENDIF
        ENDIF
        IX=J
        IZ=K
        WRITE(6,17)J,K,LLEFT
        IF(LLEFT)THEN
             LVSW=.TRUE.
             DO 8 J=1,N
             DO 8 K=1,L
             VX1(J,K)=-VX1(J,K)
             VZ1(J,K)=-VZ1(J,K)
             VX2(J,K)=-VX2(J,K)
             VZ2(J,K)=-VZ2(J,K)
 8           CONTINUE
        ENDIF
        KSYM=49
        CALL MSUB(X0,Y0,Z0,IY,IX,IZ,IDSYXZ,VX1,VX2,
       cVY1,VY2,VZ1,VZ2,DX,DY,DZ,IH,TISUM,LPATH,
       cDISTX,DISTZ,NREC,NRLIM,PATH,IPATH,KSYM,XSF,ZSF)
        IF(LVSW)THEN
             LVSW=.FALSE.
             DO 9 J=1,N
             DO 9 K=1,L
             VX1(J,K)=-VX1(J,K)
             VZ1(J,K)=-VZ1(J,K)
             VX2(J,K)=-VX2(J,K)
             VZ2(J,K)=-VZ2(J,K)
 9           CONTINUE
        ENDIF
        J=IX
        K=IZ
        ZTEST=Z0/DZ(K)
        XTEST=X0/DX(J)
        IF(XTEST.LT..5)THEN
            IF(XTEST.LT.ZTEST)THEN
                 RR=RJ(J-1,3)+ZTEST*
       c         (RJ(J,1)-RJ(J-1,3))
                 WRITE(6,18)J,K,X0,Z0,RJ(J-1,3),RJ(J,1)
            ELSE
                 RR=RJ(J,1)+XTEST*
       c         (RJ(J,2)-RJ(J,1))
                 WRITE(6,18)J,K,X0,Z0,RJ(J,1),RJ(J,2)
            ENDIF
        ELSE
            IF((1.-XTEST).LT.ZTEST)THEN
                 RR=RJ(J,2)+ZTEST*
       c         (RJ(J,3)-RJ(J,2))
                 WRITE(6,18)J,K,X0,Z0,RJ(J,2),RJ(J,3)
            ELSE
                 RR=RJ(J,1)+XTEST*
       c         (RJ(J,2)-RJ(J,1))
```

Attachment 1, continued

```
                WRITE(6,18)J,K,X0,Z0,RJ(J,1),RJ(J,2)
          ENDIF
      ENDIF
   10 CONTINUE
      IF(LLEFT)THEN
           R1=RR
           R2=RL
      ELSE
           R1=RL
           R2=RR
      ENDIF
      WRITE(6,19)R1,R2
      KSYM=0
      DO 15 IPER=1,NPER
      RECH=R1+(R2-R1)*PER(IPER)
      WRITE(6,20)RECH
      DO 11 J=1,N
      DO 11 JK=1,3
      IF(RJ(J,JK).GT.RECH)GOTO12
   11 CONTINUE
   12 DO 13 K=1,L
      IF(IH(J,K).EQ.1)GOTO14
   13 CONTINUE
   14 CONTINUE
      IF((J.EQ.1).AND.(JK.EQ.1))THEN
           X0=DX(1)*RECH/RJ(1,1)
           Z0=0.
      ELSE
           IF(JK.EQ.1)THEN
                Z0=DZ(K)*(RECH-RJ(J-1,3))/(RJ(J,1)-RJ(J-1,3))
                X0=0.
           ELSE IF(JK.EQ.2)THEN
                X0=DX(J)*(RECH-RJ(J,1))/(RJ(J,2)-RJ(J,1))
                Z0=0.
           ELSE
                Z0=DZ(K)*(RECH-RJ(J,2))/(RJ(J,3)-RJ(J,2))
                X0=DX(J)
           ENDIF
      ENDIF
      IF(X0.LT.0.)X0=0.
      IF(Z0.LT.0.)Z0=0.
      IF(X0.GT.DX(J))X0=DX(J)
      IF(Z0.GT.DZ(K))Z0=DZ(K)
      IX=J
      IZ=K
      CALL MSUB(X0,Y0,Z0,IY,IX,IZ,IDSYXZ,VX1,VX2,
     cVY1,VY2,VZ1,VZ2,DX,DY,DZ,IH,TISUM,LPATH,
     cDISTX,DISTZ,NREC,NRLIM,PATH,IPATH,KSYM,XSF,ZSF)
   15 CONTINUE
      RL=RR
      LLEFT=.NOT.LLEFT
      J=JBEG
```

Attachment 1, continued

```
      K=KBEG
      GOTO7
   16 FORMAT(1X,I2,1P3E11.3)
   17 FORMAT(1X,2I3,L2)
   18 FORMAT(1X,2I3,2F7.0,2F6.1)
   19 FORMAT(' R1=',1PE10.3,',R2=',E10.3)
   20 FORMAT(' RECH=',1PE10.3)
      END
```

Attachment 1, continued

```
C       PSUB - Saves coordinates along selected flow-paths.
        SUBROUTINE PSUB(DISTX,DX,XSF,DISTZ,ZSF,DZ,
      c IPATH,PATH,NREC,NRLIM,KSYM,X0,Z0,J,K)
        DIMENSION DISTX(N),DX(N),DISTZ(L),DZ(L)
        CHARACTER PATH(1),IDXC(4),IDZC(4),SYM(50)
        LOGICAL LFIN,LHCPT,LVPT
        REAL KT
        CHARACTER*8 XMESUR,ZMESUR
        COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
      c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
      c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
        EQUIVALENCE (IDX,IDXC(1)),(IDZ,IDZC(1))
        DATA SYM/'A','B','C','D','E','F','G','H','I','J','K','L','M','N',
      c 'O','P','Q','R','S','T','U','V','W','X','Y','Z','a','b','c','d',
      c 'e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t',
      c 'u','v','w','x'/
        IF(NREC.EQ.NRLIM)THEN
             WRITE(1,1)
             WRITE(6,1)
             STOP
        ENDIF
        IDX=6.*(DISTX(J)+X0)/XSF+1.5
        IDZ=10.*(DISTZ(K)-Z0)/ZSF+3.5
        IPATH=IPATH+1
        PATH(IPATH)=IDXC(3)
        IPATH=IPATH+1
        PATH(IPATH)=IDXC(4)
        IPATH=IPATH+1
        PATH(IPATH)=IDZC(4)
        IPATH=IPATH+1
        PATH(IPATH)=SYM(KSYM)
        NREC=NREC+1
        RETURN
      1 FORMAT(' G8 ARRAY FILLED TO LIMIT OF 131072')
        END
```

Attachment 1, continued

```
C       SORT - Sorts flow-path coordinates.
        SUBROUTINE SORT(IPOS,PATH,SRT,NREC,LREC)
        CHARACTER PATH(1),A1,A2,SRT(1)
        DIMENSION IPOS(1)
        JC=0
        DO 1 IC=1,NREC
        JC=JC+LREC
        IPOS(IC)=JC
      1 CONTINUE
        IBEG=1
        IEND=NREC
      2 CONTINUE
        ISW=0
        ISW2=0
        ISTART=IBEG+1
        ILAST=IEND
        IF(ILAST.LT.ISTART)GOTO7
        IP1=IPOS(IBEG)
        J1=IP1-LREC
        DO 4 IC=ISTART,ILAST
        IP2=IPOS(IC)
        JJ1=J1
        J2=IP2-LREC
        JJ2=J2
        DO 3 JC=1,LREC
        JJ1=JJ1+1
        JJ2=JJ2+1
        A1=PATH(JJ1)
        A2=PATH(JJ2)
        IF(A1.GT.A2)THEN
             ISW=IC
             ISW2=1
             IPOS(IC-1)=IP2
             GOTO4
        ELSE IF(A1.LT.A2)THEN
             J1=J2
             IF(ISW2.NE.0)THEN
                  IPOS(IC-1)=IP1
                  ISW2=0
             ENDIF
             IP1=IP2
             GOTO4
        ENDIF
      3 CONTINUE
      4 CONTINUE
        IEND=IEND-1
        IF(ISW.EQ.0)GOTO7
        IF(ISW.GE.ILAST)THEN
             IPOS(ILAST)=IP1
        ENDIF
        ISW=0
        ISW2=0
```

Attachment 1, continued

```
      ISTART=IBEG
      ILAST=IEND-1
      IF(ILAST.LT.ISTART)GOTO7
      IP1=IPOS(IEND)
      J1=IP1-LREC
      DO 6 IB=ISTART,ILAST
      IC=ILAST-IB+ISTART
      IP2=IPOS(IC)
      JJ1=J1
      J2=IP2-LREC
      JJ2=J2
      DO 5 JC=1,LREC
      JJ1=JJ1+1
      JJ2=JJ2+1
      A1=PATH(JJ1)
      A2=PATH(JJ2)
      IF(A1.LT.A2)THEN
           ISW=IC
           ISW2=1
           IPOS(IC+1)=IP2
           GOTO6
      ELSE IF(A1.GT.A2)THEN
           J1=J2
           IF(ISW2.NE.0)THEN
                IPOS(IC+1)=IP1
                ISW2=0
           ENDIF
           IP1=IP2
           GOTO6
      ENDIF
    5 CONTINUE
    6 CONTINUE
      IBEG=IBEG+1
      IF(ISW.NE.0)THEN
           IF(ISW.GE.ISTART)THEN
                IPOS(ISTART)=IP1
           ENDIF
           GOTO2
      ENDIF
    7 CONTINUE
      ISRT=0
      DO 9 IC=1,NREC
      KC=IPOS(IC)-LREC+1
      KPLM1=K+LREC-1
      DO 8 JC=KC,KPLM1
      ISRT=ISRT+1
      SRT(ISRT)=PATH(JC)
    8 CONTINUE
    9 CONTINUE
      RETURN
      END
```

Attachment 1, continued

```
C     PRINT - Constructs cross-sections showing selected flow-paths,
C        head, travel time, and lithology.
      SUBROUTINE PRINT(TIM,SRT,HD,IH,DX,DZ,SEC,NR)
      LOGICAL LFIN,LHCPT,LVPT
      REAL KT
      REAL*8 HD(N,L),XN1
      CHARACTER*8 XLABEL,ZLABEL*35,XMESUR,ZMESUR
      CHARACTER*20 TITLE(4)
      CHARACTER VF4*24,VF3*24,VF1*21,VF5*22,VF2*21
      CHARACTER SRT(1),PRNT(123),SEC(N,L)
      DIMENSION TIM(N,L),IH(N,L),DX(N),DZ(L)
      DIMENSION NA(4),XLABEL(3),ZN(13),XN(100),INDX(3)
      COMMON/VAR/NU,NL,IBW,NEQ,IGATE,MDM,MAX,L,M,N,LFIN,MN,
     c LHCPT,LVPT,IREF,SP,KT,TTOP,TEMPR,
     c DELH,DELT,TCONV,HCONV,XSCALE,ZSCALE,DINCHX,DINCHZ,XMESUR,ZMESUR
      DATA PRNT/123*' '/
      DATA XLABEL/' X DIS- ','TANCE IN',' MILES '/
      DATA ZLABEL/'DISTANCE IN Z DIRECTION IN MILES   '/
      DATA VF1/'(''(1H ,'',I3,''X,A20)'')'/
      DATA VF2/'(''(1H ,'',I3,''X,A35)'')'/
      DATA VF3/'(''(1H ,2X,A8,'',I3,''A1)'')'/
      DATA VF4/'(''(1H ,F10.2,'',I3,''A1)'')'/
      DATA VF5/'(''(1H ,10X,'',I3,''A1)'')'/
      DATA TITLE/' PLOT OF FLOWLINES  ','    PLOT OF HEAD    ',
     c 'PLOT OF TRAVEL TIME ',' MODELED LITHOLOGY  '/
      DATA N1,N2,N3,NA(4),XN1/6,10,133,1000,.833333333D-1/
      DATA ISRT/0/
    1 XSF=DINCHX*XSCALE
      ZSF=DINCHZ*ZSCALE
      RXSF=XSF/N1
      RZSF=ZSF/N2
      XDIM=0.
      DO 2 J=1,N
    2 XDIM=XDIM+DX(J)
      ZDIM=0.
      DO 3 K=1,L
    3 ZDIM=ZDIM+DZ(K)
      NZD=ZDIM/ZSF
      IF(NZD*ZSF.LT.ZDIM)NZD=NZD+1
      IF(NZD.GT.12)THEN
           NZD=12
           ZSF=ZDIM/NZD
           DINCHZ=ZSF/ZSCALE
           RZSF=ZSF/N2
           WRITE(6,15)DINCHZ
           WRITE(1,15)DINCHZ
      ENDIF
    4 NXD=XDIM/XSF+.5
      IF(NXD*XSF.LT.XDIM)NXD=NXD+1
      NLIN=NXD*N1+1
      NEND=N1*XDIM/XSF+.5+1
      N5=NXD+1
```

```
      N6=NZD+1
      NPRNT=N2*ZDIM/ZSF+3.5
      NA(1)=NLIN/2-1
      NA(2)=NLIN/2
      NA(3)=NLIN/2+3
      NB=NZD*5
      NC=NB-7
      IF(NC.LT.10)NC=10
      WRITE(VF1,VF1)NB
      WRITE(VF2,VF2)NC
      WRITE(VF3,VF3)NPRNT
      WRITE(VF4,VF4)NPRNT
      WRITE(VF5,VF5)NPRNT
      XLABEL(3)=XMESUR
      ZLABEL(28:35)=ZMESUR
      DO 5 I2=1,N6
    5 ZN(I2)=ZSF*(I2-1)/ZSCALE
      DO 6 I2=1,N5
    6 XN(I2)=XSF*(I2-1)/XSCALE
      DO 14 NG=1,4
      IF(NG.EQ.1)THEN
           NEXT=1
      ELSE IF(NG.EQ.2)THEN
           CALL SCALEH(HD,N,L,FACT)
           DO 7 J=1,N
           WRITE(6,17)(HD(J,K),K=L,1,-1)
    7      CONTINUE
           D1=DX(1)/2.
           NEXT=D1/RXSF+1.5
      ELSE IF(NG.EQ.3)THEN
           DO 8 J=1,N
           WRITE(6,18)(TIM(J,K),K=L,1,-1)
    8      CONTINUE
           CALL SCALET(TIM,N,L)
           D1=DX(1)/2.
           NEXT=D1/RXSF+1.5
      ELSE
           D2=DX(1)
           NEXT=1
      ENDIF
      J=1
      LL=1
      WRITE(6,VF1)TITLE(NG)
      WRITE(6,VF2)ZLABEL
      WRITE(6,19)(ZN(I2),I2=1,N6)
      DO 13 ILIN=1,NLIN
      DO 9 IPRNT=1,NPRNT
    9 PRNT(IPRNT)=' '
      IF(ILIN.GT.NEND)GOTO12
      IF((ILIN.EQ.1).OR.(ILIN.EQ.NEND))THEN
           DO 10 IPRNT=3,NPRNT
   10      PRNT(IPRNT)='-'
```

```
          DO 11 IPRNT=3,NPRNT,N2
   11     PRNT(IPRNT)='+'
          PRNT(NPRNT)='+'
      ELSE
          IF(MOD(ILIN,N1).EQ.1)THEN
               PRNT(3)='+'
               PRNT(NPRNT)='+'
          ELSE
               PRNT(3)='|'
               PRNT(NPRNT)='|'
          ENDIF
      ENDIF
      IF(ILIN.NE.NEXT)GOTO12
      IF(NG.EQ.1)THEN
          CALL PLINE1(ILIN,NEXT,ISRT,NR,PRNT,SRT,NPRNT)
      ELSE IF(NG.EQ.2)THEN
          CALL PLINE2(ILIN,NEXT,HD,TIM,IH,PRNT,DX,DZ,D1,RXSF,RZSF,
     c     NG,N,L,J)
      ELSE IF(NG.EQ.3)THEN
          CALL PLINE2(ILIN,NEXT,HD,TIM,IH,PRNT,DX,DZ,D1,RXSF,RZSF,
     c     NG,N,L,J)
      ELSE
          CALL PLINE4(ILIN,NEXT,SEC,IH,PRNT,DX,DZ,D2,RXSF,RZSF,N,L,J,
     c    NPRNT)
      ENDIF
   12 CONTINUE
      IF(ILIN-NA(LL).EQ.0)THEN
          WRITE(6,VF3)XLABEL(LL),(PRNT(IPRNT),IPRNT=1,NPRNT)
          LL=LL+1
      ELSE
          IF(MOD(ILIN,N1).EQ.1)THEN
               INDEX=1+(ILIN-1)/6
               WRITE(6,VF4)XN(INDEX),(PRNT(IPRNT),IPRNT=1,NPRNT)
          ELSE
               WRITE(6,VF5)(PRNT(IPRNT),IPRNT=1,NPRNT)
          ENDIF
      ENDIF
   13 CONTINUE
      IF(NG.EQ.2)WRITE(6,20)FACT
      IF(NG.EQ.3)WRITE(6,21)
   14 CONTINUE
      RETURN
   15 FORMAT(' ',25X,10('*'),' TO FIT MAP WITHIN 12 INCHES, DINCHZ SHOUL
     cD BE GREATER THAN',G15.7,1X,10('*'))
   17 FORMAT(1X,20F6.1)
   18 FORMAT(1X,1P7E10.3)
   19 FORMAT(1H ,5X,13F10.2)
   20 FORMAT(' MULTIPLICATION FACTOR =',1PE8.1)
   21 FORMAT(' EXPONENT, 5 INDICATES 10.**5 <= VALUE < 10.**6')
      END
```

Attachment 1, continued

```
C     PLINE1 - Constructs one line of flow-path cross-section.
      SUBROUTINE PLINE1(ILIN,NEXT,ISRT,NR,PRNT,SRT,NPRNT)
      CHARACTER SRT(1),PRNT(1),BFX(4),BFZ(4)
      EQUIVALENCE (IBX,BFX(1)),(IBZ,BFZ(1))
      DATA IBX,IBZ/0,0/
    1 IF(ISRT.GE.NR)GOTO3
      ISRT=ISRT+1
      BFX(3)=SRT(ISRT)
      ISRT=ISRT+1
      BFX(4)=SRT(ISRT)
      ISRT=ISRT+1
      BFZ(4)=SRT(ISRT)
      ISRT=ISRT+1
      IF(IBX.LT.ILIN)GOTO1
      IF(IBX.GT.ILIN)GOTO2
      IF((IBZ.LT.3).OR.(IBZ.GT.NPRNT))GOTO1
      PRNT(IBZ)=SRT(ISRT)
      GOTO1
    2 ISRT=ISRT-4
      NEXT=IBX
      GOTO4
    3 NEXT=NPRNT+1
    4 RETURN
      END
```

Attachment 1, continued

```
C      SCALET - Scales travel time to three digits or less.
       SUBROUTINE SCALET(TIM,N,L)
       DIMENSION TIM(N,L)
       DO 2 K=1,L
       DO 2 J=1,N
       TIJK=TIM(J,K)
       IF(TIJK.LT.0.)GOTO3
       IF(TIJK.GE.10.)GOTO1
       ILOG=0
       GOTO2
     1 ILOG=ALOG10(TIJK)
     2 TIM(J,K)=ILOG
     3 RETURN
       END
```

```
C     PLINE2 - Constructs one line of head or travel-time cross-section.
      SUBROUTINE PLINE2(ILIN,NEXT,HD,TIM,IH,PRNT,DX,DZ,D1,RXSF,RZSF,
     c NG,N,L,J)
      CHARACTER PRNT(1),SYM(0:11),PRCH
      REAL*8 HD(N,L)
      DIMENSION TIM(N,L),IH(N,L),DX(1),DZ(1),INDX(3)
      DATA SYM/'0','1','2','3','4','5','6','7','8','9','-',' '/
      DISTZ=0.
      DO 4 K=L,1,-1
      IF(K.EQ.L)THEN
           DISTZ=DISTZ+DZ(L)/2.
      ELSE
           DISTZ=DISTZ+(DZ(K+1)+DZ(K))/2.
      ENDIF
      IZ=DISTZ/RZSF+3.5
      IF(IH(J,K).NE.1)GOTO4
      IF(NG.EQ.2)THEN
           VAR=HD(J,K)
      ELSE
           VAR=TIM(J,K)
      ENDIF
      IF(VAR.LT.0)THEN
           IN=-VAR+.5
           IF(IN.GE.10)THEN
                INDX(3)=10
                INDX(2)=IN/10
                INDX(1)=MOD(IN,10)
           ELSE
                INDX(3)=11
                INDX(2)=10
                INDX(1)=IN
           ENDIF
      ELSE IF(VAR.EQ.0)THEN
           INDX(1)=0
           INDX(2)=11
           INDX(3)=11
      ELSE
           IN=VAR+.5
           INDX(1)=MOD(IN,10)
           IF(IN.GE.100)THEN
                INDX(3)=IN/100
                INDX(2)=MOD(IN,100)/10
           ELSE IF(IN.GE.10)THEN
                INDX(3)=11
                INDX(2)=IN/10
           ELSE
                INDX(3)=11
                INDX(2)=11
           ENDIF
      ENDIF
      IF(INDX(3).NE.11)THEN
           I2=3
```

Attachment 1, continued

```
      ELSE IF(INDX(2).NE.11)THEN
           I2=2
      ELSE
           I2=1
      ENDIF
      IF((IZ-I2).LT.0)GOTO4
      DO 2 I1=1,I2
      PRCH=PRNT(IZ-I2+I1)
      IF(PRCH.GE.'0'.AND.PRCH.LE.'9')GOTO4
    2 CONTINUE
      DO 3 I1=1,I2
    3 PRNT(IZ-I1+1)=SYM(INDX(I1))
    4 CONTINUE
   45 J=J+1
      IF(J.LE.N)THEN
           D1=D1+(DX(J-1)+DX(J))/2.
           NEXT=D1/RXSF+1.5
      ELSE
           NEXT=NZ+1
           GOTO5
      ENDIF
      IF(NEXT.EQ.ILIN)GOTO45
    5 RETURN
      END
```

Attachment 1, continued

```
C     SCALEH - Scales head to three digits or less.
      SUBROUTINE SCALEH(HD,N,L,FACT)
      REAL*8 HD(N,L)
      HMAX=-1.E+35
      HMIN=1.E+35
      DO 1 K=1,L
      DO 1 J=1,N
      IF(HD(J,K).GT.HMAX)THEN
           HMAX=HD(J,K)
      ELSE IF(HD(J,K).LT.HMIN)THEN
           HMIN=HD(J,K)
      ENDIF
    1 CONTINUE
      IF(HMAX.LE.-HMIN)HMAX=-HMIN
      IF(HMAX.EQ.0.)THEN
           FACT=1.
      ELSE
           HLOG=ALOG10(HMAX)-2.
           ILOG=HLOG
           IF(HLOG.LT.0.)ILOG=ILOG-1
           IF(HMAX.EQ.-HMIN)ILOG=ILOG+1
           FACT=10.**ILOG
           DO 2 K=1,L
           DO 2 J=1,N
           HD(J,K)=HD(J,K)/FACT
    2      CONTINUE
      ENDIF
      RETURN
      END
```

Attachment 1, continued

```
C     PLINE4 - Constructs one line of lithologic cross-section.
      SUBROUTINE PLINE4(ILIN,NEXT,SEC,IH,PRNT,DX,DZ,D2,RXSF,RZSF,N,L,J,
     c NPRNT)
      CHARACTER SEC(N,L),PRNT(1)
      DIMENSION IH(N,L),DX(1),DZ(1)
      DISTX=(ILIN-1)*RXSF
      IF(DISTX.GT.D2)THEN
           J=J+1
           IF(J.LE.N)THEN
                D2=D2+DX(J)
           ELSE
                J=N
           ENDIF
      ENDIF
      K=L
      D3=0.
      D4=DZ(L)
      DO 2 IPRNT=3,NPRNT
      DISTZ=(IPRNT-3)*RZSF
    1 IF(DISTZ.GT.D4)THEN
           K=K-1
           IF(K.GT.0)THEN
                D3=D4
                D4=D4+DZ(K)
                GOTO1
           ELSE
                K=1
           ENDIF
      ENDIF
      IF(IH(J,K).NE.3)PRNT(IPRNT)=SEC(J,K)
    2 CONTINUE
      NEXT=ILIN+1
      RETURN
      END
```

Attachment 2. Definition of selected program variables

Attachment 2.

A - Array (NL*IBW) containing the reduced lower-right partition of the coefficient matrix.
AL - Array (6) containing the coefficients, for one block, in the lower-left partition of the matrix.
ARRAY - Input array to be read.
AU - Array (6*NU) containing the upper-right partition of the coefficient matrix.
B - The second parameter in the linear velocity function, v=a+b*w.
DELH - Largest change in head at last solution.
DELT - Largest change in temperature at last solution.
DELV - Largest change (either head or temperature) at last solution.
DINCHX - Number of horizontal scale units per inch, for output.
DINCHZ - Number of vertical scale units per inch, for output.
DISTX - Array (N) containing horizontal distance (ft) of the decreasing face of each block from the left edge of the section.
DISTZ - Array (N) containing vertical distance (ft) of the decreasing face of each block from the bottom of the section.
DP - Array (M*N) containing depths below surface for the top nodes of the model.
DU - Array (NU) containing the upper-left partition of the coefficient matrix.
DW - Distance across a model block, in the direction of one of the coordinate axes.
DX - Array (N) containing the dimensions of the blocks in the horizontal-x direction.
DY - Array (M) containing the dimensions of the blocks in the horizontal-y direction.
DZ - Array (L) containing the dimensions of the blocks in the vertical direction.
D1 - The distance, from the left side of the section, to the next line to be plotted.
D2 - The distance, from the left side of the section, to the next block face.
FACT - Scaling factor for the head values displayed on a cross-section.
G1 - Array containing HD (head), TMP (temperature), DX (horizontal-x dimensions), DY (horizontal-y dimensions), DZ (vertical dimensions), DISTX (horizontal distances), and DISTZ (vertical distances.
G2 - Array containing IH (block types for water equation), IT (block types for heat equation), NP (block number), IR (row number corresponding to the block number), JR (column number), and KR (layer number).
G3 - Array containing Q (interblock ground- water flow).
G4 - Array containing HC (hydraulic conductivity at computed temperature), P (hydraulic conductivity at reference temperature), POR (porosity), RCH (recharge from rainfall), RJ (recharge from all sources at the top of the model), DP (depth below surface for the top model layer), and HF (geothermal heat flow at the bottom of the model).
G5 - Array containing AU (upper-right partition of the coefficient matrix).

G6 - Array containing DU (upper-left partition of the coefficient matrix), RU (upper part of the known vector), and RL (lower part of known vector).
G7 - Array containing VX1 and VX2 (x velocities), VY1 and VY2 (y velocities), and VZ1 and VZ2 (z velocities).
G8 - Array containing SEC (block lithology), TIM (travel time), IDS (disharge points), PATH (flow path coordinates), IPOS (data for sorting routine), and SRT (sorted coordinates).
HD - Array (M*N*L) containing elevation of the water level for a node.
HC - Array (M*N*L) containing hydraulic conductivity adjusted to the temperature at each block.
HCONV - Convergence test for head.
solution if changes in computed head are smaller.
HF - Array (M*N) containing heat-flow rates at a node in the bottom layer of the model.
I - Location index in the y direction.
IAU - Pointer to the AU array.
IBW - Bandwidth of reduced lower-right part of coefficient matrix.
IDIR - The upstream direction near a stagnation point, +1 indicates increasing direction, -1 indicates decreasing direction.
IDISTX - Pointer to the DISTX array.
IDISTZ - Pointer to the DISTZ array.
IDP - Pointer to the DP array.
IDS - Array (M*N*L) containing discharge points for each node.
IDSYXZ - The discharge point for one beginning point.
IDU - Pointer to the DU array.
IDX - Pointer to the DX array.
IDY - Pointer to the DY array.
IDZ - Pointer to the DZ array.
IEQ - Index indicating equation number.
IGATE - Number of times equations have been solved.
IH - Array (M*N*L) containing integer codes indicating the type of model block for the ground-water flow equation. Codes are: 1, active block, 2, constant-head, and 3, inactive block.
IHC - Pointer to the HC array.
IHD - Pointer to the HD array.
IHF - Pointer to the HF array.
IIDS - Pointer to the IDS array.
IIH - Pointer to the IH array.
IIR - Pointer to the IR array.
IIT - Pointer to the IT array.
IJR - Pointer to the JR array.
IKR - Pointer to the KR array.
IL - Index indicating equation number relative to lower partition.
ILIN - Number (horizontal direction) of a line on a cross-section.
INP - Pointer to the NP array.
INW - New block number (in one direction) after a move, either one greater or less than previous number.
IQ - Pointer to the Q array.
IP - Pointer to the P array.
IPATH - Pointer to location within PATH array.

IPOR - Pointer to the POR array.
IPOS - Array (NREC) containing pointers to flow-path data.
IR - Array (NEQ) containing pointers from equation number to row.
IRCH - Pointer to the RCH array.
IREF - Reference temperature (Celsius) for input values of hydraulic conductivity.
IRJ - Pointer to the RJ array.
IRL - Pointer to the RL array.
IRU - Pointer to the RU array.
ISEC - Pointer to the SEC array.
ISRT - Pointer to location within SRT array.
IT - Array (M*N*L) containing integer codes indicating the type of model block for the heat flow equation. Codes are: 1, active block, 2, constant temperature, and 3, inactive block.
ITIM - Pointer to the TIM array.
ITMP - Pointer to the TMP array.
IVX1 - Pointer to the VX1 array.
IVX2 - Pointer to the VX2 array.
IVY1 - Pointer to the VY1 array.
IVY2 - Pointer to the VY2 array.
IVZ1 - Pointer to the VZ1 array.
IVZ2 - Pointer to the VZ2 array.
IW - Location index (in one direction) before a move.
J - Location index in the x direction.
JR - Array (NEQ) containing pointers from equation number to column.
K - Location index in the z direction.
KL - Number of layers for an array, KL=1 for DP, RCH, and HF, and KL=L for other arrays.
KR - Array (NEQ) containing pointers from equation number to layer.
KSYM - Number for flow-path symbol.
KT - Thermal conductivity, input in conductance units (10**-3 cal./(cm.*s*deg. C).
L - Number of blocks in the vertical direction.
LFIN - Logical, T indicates final solution.
LHCPT - Logical, LHCPT=T displays values of adjusted hydraulic conductivity in "input-file-name".BAL.
LMN - L*M*N, number of blocks in the model.
LPATH - Logical, LPATH=T indicates that flow paths are to be traced.
LREC - Number of bytes in one flow-path coordinate.
LSTOP - Logical, T causes program stop due to insufficient storage allocation.
LVPT - Logical, LVPT=T displays velocities in "input-file-name".VELO.
M - Number of blocks in the horizontal-y direction.
MAX - Maximum repetitions of solution procedure.
MD - Row position of the main diagonal term in the A matrix.
MDM - IBW/2 (half bandwidth).
MN - M*N (number of rows * number of columns).
N - Number of blocks in the horizontal-x direction.
NEQ - Number of active blocks (equations).
NEXT - Next line of the output section that will display data.
NG - Section counter in PRINT routine, 1 - flow paths, 2 - head, 3 - travel time, and 4 - lithology.

Attachment 2, continued

NL - Number of blocks (equations) in the lower partition of the matrix.
NP - Array (M*N*L) containing pointers from row, column, and layer to equation number.
NR - Number of bytes in PATH or SRT arrays, equal to 4*NREC.
NREC - Number of flow-path coordinates.
NRLIM - Number of bytes available in G8 array.
NU - Number of blocks (equations) in the upper partition of the matrix.
NPRNT - Number of characters in one vertical line of the output sections.
P - Array (M*N*L) containing hydraulic conductivity for each block at the reference temperature.
PATH - Array (NR bytes) containing flow-path data.
POR - Array (M*N*L) containing porosity for each block.
PRNT - Characters in one vertical line of a cross-section.
Q - Array (NEQ*6) containing discharge of water for each active block through each of the block faces.
RCH - Array (M*N) containing recharge rate for the top nodes.
RJ - Array (N*3) containing inflow to the top of a section from recharge and constant-head nodes.
RL - Array (NL) containing the lower part of the known vector.
RU - Array (NU) containing the upper part of the known vector.
RXSF - Horizontal distance represented by one character on a cross-section.
RZSF - Vertical distance represented by one character on a cross-section.
SEC - Array (N*L bytes) containing one-character codes representing the lithologic type of each block.
SP - Volumetric specific heat for fluid, input in btu/(ft**3*deg.f)
SRT - Array (NR bytes) containing sorted flow-path data.
TMP - Array (M*N*L) containing temperatures for a node.
TCONV - Convergence test for temperature. solution if changes in computed temperature are smaller.
TEMPR - Temperature of recharge from rainfall, deg. F.
TIM -Array (M*N*L) containing time of travel (years) from a node to a discharge point.
TIME - The time of travel (days) across a block.
TISUM - The time of travel (days) along a flow path, equal to the sum of TIME along the path.
TTOP - Surface temperature, deg. f.
V - Either head or temperature, depending on which equation is solved.
VAR - Common block containing several variables needed in subroutines.
VX1 - Array (M*N*L) containing x velocity component at the face between i,j-1,k and i,j,k. Positive if movement is from i,j-1,k to i,j,k.
VX2 - Array (M*N*L) containing x velocity component at the face between i,j,k and i,j+1,k. Positive if movement is from i,j,k to i,j+1,k.
VY1 - Array (M*N*L) containing y velocity component at the face between i-1,j,k and i,j,k. Positive if movement is from i-1,j,k to i,j,k.
VY2 - Array (M*N*L) containing y velocity component at the face between i,j,k and i+1,j,k. Positive if movement is from i,j,k to i+1,j,k.
VZ1 - Array (M*N*L) containing z velocity component at the face between i,j,k-1 and i,j,k. Positive if movement is from i,j,k-1 to i,j,k.

VZ2 - Array (M*N*L) containing z velocity component at the face between i,j,k and i,j,k+1. Positive if movement is from i,j,k to i,j,k+1.
V0 - Velocity component, at a location (W0) within a block.
V1 - Velocity component, at a decreasing block face.
V2 - Velocity component, at an increasing block face.
W - Distance from decreasing block-face after a move step. Value is from zero up to the dimension of the block.
WNEAR - A location near a stagnation point but upstream from it.
WZERO - A stagnation point (point of zero velocity).
W0 - Distance from decreasing block-face before a move step. Represents X0, Y0, or Z0. Value is from zero to the dimension of the block.
XMESUR - Name of the horizontal scale unit, for output cross sections.
XSCALE - Number of feet in the horizontal scale unit, for output.
XSF - Horizontal scale (ft/in.) for the cross-sections.
XO - Distance (ft) from the decreasing block-face in the x direction.
YO - Distance (ft) from the decreasing block-face in the y direction.
ZMESUR - Name of the vertical scale unit, for output cross sections.
ZSCALE - Number of feet in the vertical scale unit, for output.
ZSF - Vertical scale (ft/in.) for the cross-sections.
ZO - distance (ft) from the decreasing block-face in the z direction.

Attachment 3. Input data

Attachment 3.

The input data was designed for card format (80 character lines) although the input data is read from disk files. For number of lines, the ratio A/B, if not an integer, is rounded up to the next larger integer. Explanation of the formats are in FORTRAN language manuals. HOTWTR prompts for the input file name and reads the data from it. The input file name must not be more than 12 characters.

Number of lines	Columns	Variable	Format	Definition
1	1-3	M	I3	Number of blocks in the y direction
	4-6	N	I3	Number of blocks in the x direction
	7-9	L	I3	Number of blocks in the z direction
	10-12	MAX	I3	Number of iterations
	13-15	IREF	I3	Base temperature, deg. C, for hydraulic conductivity
	16	LHCPT	L1	Display option for hydraulic conductivity, T=yes, F=no
	17	LVPT	L1	Display option for velocities, T=yes, F=no

Note.-None of M, N, or L should be greater than 99. M = 1 produces special output for cross-sections. MAX = 1 indicates a solution for the ground-water equation only. For a solution to both equations, MAX = 5 results in adequate solutions for many cross-section models. IREF indicates the temperature for the input hydraulic conductivity. Changes in temperature from IREF modify the hydraulic conductivity by changes in the kinematic viscosity of water. Viscosity data used in the program is from Lohman and others (1972, table 3).

Attachment 3, continued

Number of lines	Columns	Variable	Format	Definition
1	1-5	KT	F5.0	Thermal conductivity, in conductance units (0.001 calories/(cm.*sec.)
	6-10	SP	F5.0	Volumetric specific heat, BTU/cubic ft.
	11-15	TTOP	F5.0	Temperature, deg. F, at the point of conductive heat discharge out of the top of the model.
	16-20	TEMPR	F5.0	Temperature, deg. F, of recharge from precipitation.
	21-25	TCONV	F5.0	Temperature-change criterion, deg. F
	26-30	HCONV	F5.0	Head-change criterion, ft.
N/16	1-80	DX(J)	16F5.0	Block lengths, ft, x direction.
M/16	1-80	DY(I)	16F5.0	Block lengths, ft, y direction.
L/16	1-80	DZ(K)	16F5.0	Block lengths, ft, z direction.

Number of lines	Columns	Variable	Format	Definition
L*M*(N/80)	1-80	IT(I,J,K)	80I1	Block type for the heat-flow model.

Note.-The types are: 1, active node (unknown temperature to be solved for), 2, constant temperature node, and 3, no heat conduction across the block face but there may be convection across the block face. The data are grouped by layer (z index or K), by row (y index or I) within each layer, and by column (x index or J) within each row. Each row begins on a new line.

Number of lines	Columns	Variable	Format	Definition
L*M*(N/80)	1-80	IH(I,J,K)	80I1	Block type for the ground-water-flow model.

Note.-The types are: 1, active node, 2, constant head, and 3, no flow across block face. Active nodes should be the same for heat and water ($IT_{i,j,k}$=1 requires that $IH_{i,j,k}$=1). However, inactive nodes do not have to correspond ($IH_{i,j,k}$=2 but $IT_{i,j,k}$=3, for example). At least one node in the ground-water model must be constant head because head cannot be determined by flux alone. No constant-temperature nodes are necessary. The sequence is the same as for IT.

Attachment 3, continued

Each of the following data sets consists of a parameter card and, if the data set contains variable data, additional data cards. Each parameter card is coded as:

Number of lines	Columns	Variable	Format	Definition
1	1-10	CONS	F10.0	If IVAR=0 then all array values are equal to CONS. Otherwise, CONS is a multiplication factor for the data values that follow.
	11-15	IVAR	I5	If IVAR=0 then no data cards follow. Otherwise, data cards for this array follow.

Attachment 3, continued

When data cards are included, start each row on a new card. Multi-layered data sets are coded by layers. The first three of the following data sets are always one layer.

Number of lines	Columns	Variable	Format	Definition
M*(N/20)	1-80	DP(I,J)	20F4.0	Distance, ft, along the path of conductive heat-discharge out the top of the model.

Note.-If TTOP is the average air temperature then DP is the distance that the top active node is below land surface.

Number of lines	Columns	Variable	Format	Definition
M*(N/20)	1-80	RCH(I,J)	20F4.0	Recharge, in./yr, from precipitation.

Note.-Recharge should be coded as zero if the top active node is overlain by a constant-head node.

Number of lines	Columns	Variable	Format	Definition
M*(N/20)	1-80	HF(I,J)	20F4.0	Geothermal heat-flow, heat flow units (0.000001 calories /(square cm*sec)).
L*M*(N/20)	1-80	HD(I,J,K)	20F4.0	Head, ft.

Note.-Only head at constant-head nodes need be coded. Other heads can be blank. If there is only one constant-head node, it's head can be coded on the parameter card and the data cards can be omitted.

Number of lines	Columns	Variable	Format	Definition
L*M*(N/20)	1-80	TMP(I,J,K)	20F4.0	Temperature, deg. F.

Note.-Values should be specified for constant-temperature nodes (IT=2) or for constant-head nodes (IH=2) where flow is into the model.

Attachment 4. Description of output data files.

Attachment 4.

OUTPUT FILE NAME[1]	DESCRIPTION OF FILE
"FILE".INPUT	Contains programs interpretation of all input data; equation, row, column, and layer numbers.
"FILE".BAL	Heat and water balance, calculated temperature and head, and hydraulic conductivity (if LHCPT=T in input data).
"FILE".VELO	Velocities (if LVPT=T in input data) for x, y, and z directions.
"FILE".MOVE	Time of travel to discharge node, and location of discharge node.
"FILE".PATH	Inflow at top of model section, flow divides (if any), and points on selected flow paths.
"FILE".PRINT	Line-printer drawn sections showing selected flow paths, head, time of travel, and lithology.

[1] "FILE" will be replaced by the name of the input data.

Attachment 5. Example simulation

Attachment 5.

This simulation is for the cross-section shown on the last page of this attachment. The section is 200,000 ft long and 7,500 ft in vertical extent. Three rock types are shown on the section: basalt (B), crystalline rocks (G), and alluvial fill (A). There are two discharge areas for ground water, one at each end of the section. The basalt has a hydraulic conductivity of .5 ft/d and a porosity of .05. The crystalline rocks have a hydraulic conductivity of .0001 ft/d and porosity of .0001. The alluvial fill has a hydraulic conductivity of 10 ft/d and porosity of .3. The water-level elevation in both of the discharge areas is 1000 ft. The section is divided into 20 model blocks in the horizontal direction and 10 blocks in the vertical. The input data for this simulation is as follows.

Attachment 5, continued

```
    ---------------------------------------------------------------------------------
    |  1 20 10  5 20FF
    |   6. 62.5  68.  68.  .01  .01
    ---------------------------------------------------------------------------------
DX  |  1+4  1+4  1+4  1+4  1+4  1+4  1+4  1+4  1+4  1+4  1+4  1+4  1+4  1+4  1+4  1+4
    |  1+4  1+4  1+4  1+4
    ---------------------------------------------------------------------------------
DY  |1.
    ---------------------------------------------------------------------------------
DZ  |  500  500  500  500  500  500  500 1000 1000 2000
    ---------------------------------------------------------------------------------
    |33333333111133333333
    |33333331111113333333
    |33333311111111333333
    |11111111111111111111
IT  |11111111111111111111
    |11111111111111111111
    |11111111111111111111
    |11111111111111111111
    |11111111111111111111
    |11111111111111111111
    ---------------------------------------------------------------------------------
    |33333333111133333333
    |33333331111113333333
    |23333311111111333332
    |11111111111111111111
IH  |11111111111111111111
    |11111111111111111111
    |11111111111111111111
    |11111111111111111111
    |11111111111111111111
    |11111111111111111111
    ---------------------------------------------------------------------------------
DP  |250.
    ---------------------------------------------------------------------------------
RCH |.05       1
    |   0   1   1   1   1   1 1-3 2-1 2-1 2-1 2-1 2-1 2-1 1-3   1   1   1   1   1   0
    ---------------------------------------------------------------------------------
HF  |1.5
    ---------------------------------------------------------------------------------
HD  |1000.
    ---------------------------------------------------------------------------------
TMP |68.
    ---------------------------------------------------------------------------------
```

Attachment 5, continued

```
    ------------------------------------------------------------------------------
    |1.        1
    |  0   0   0   0   0   0   0   0 5-1 5-1 5-1 5-1   0   0   0   0   0   0   0   0
    |  0   0   0   0   0   0   0 5-1 5-1 5-1 5-1 5-1 5-1   0   0   0   0   0   0   0
    |1+4   0   0   0   0   0 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4   0   0   0   0   0 1+4
    |1+1 1+1 1+1 1+1 1+1 1+1 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1+1 1+1 1+1 1+1 1+1 1+1
P   |1+1 1+1 1+1 1+1 1+1 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1+1 1+1 1+1 1+1 1+1
    |1+1 1+1 1+1 1+1 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1+1 1+1 1+1 1+1
    |1+1 1+1 1+1 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1+1 1+1 1+1
    |1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4
    |1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4
    |1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4
    ------------------------------------------------------------------------------
    |1.        1
    |  0   0   0   0   0   0   0   0 .05 .05 .05 .05   0   0   0   0   0   0   0   0
    |  0   0   0   0   0   0   0 .05 .05 .05 .05 .05 .05   0   0   0   0   0   0   0
    |  0   0   0   0   0   0 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4   0   0   0   0   0   0
    |.30 .30 .30 .30 .30 .30 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 .30 .30 .30 .30 .30 .30
POR |.30 .30 .30 .30 .30 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 .30 .30 .30 .30 .30
    |.30 .30 .30 .30 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 .30 .30 .30 .30
    |.30 .30 .30 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 .30 .30 .30
    |1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4
    |1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4
    |1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4 1-4
    ------------------------------------------------------------------------------
    |00000000BBBB00000000
    |0000000BBBBBB0000000
    |200000GGGGGGGG000002
    |AAAAAAGGGGGGGGAAAAAA
SEC |AAAAAGGGGGGGGGGAAAAA
    |AAAAGGGGGGGGGGGGAAAA
    |AAAGGGGGGGGGGGGGGAAA
    |GGGGGGGGGGGGGGGGGGGG
    |GGGGGGGGGGGGGGGGGGGG
    |GGGGGGGGGGGGGGGGGGGG
    ------------------------------------------------------------------------------
    |5280.     1000.     5.        2.        MILES     1000 FT
    ------------------------------------------------------------------------------
```

Attachment 5, continued

Output for Cross Sections

For cross-section models (M=1) additional output is written to a disk file called "input-file-name.PRINT". The additional output is four cross-sections showing selected flow-lines, head, travel time, and lithology.

The first of these cross-sections shows selected flow paths through the cross-section. The cross-section is examined by the program to determine if it contains more than one flow system. The program does this by examining the velocity along the sides and bottom of the cross-section. If a reversal in direction of movement occurs at the side or base of the cross-section, a boundary between flow systems exists there. A flow line is traced backward from this point to the top of the model and it's position is indicated on the cross-section by "x". This marks the system boundary. Within each distinct flow system, seven flow paths are traced. These paths, denoted by A, B, C, D, E, F, and G, represent .5, .1, .01, .001, .0001, .00001, and .000001, respectively, of the flow entering the top of the flow system. Points on the flow paths are shown where paths cross block faces. Each path begins at the top of the model and ends on the face of the discharge block. Points on the flow paths may overlap on the plot and only the highest point in the sequence A, B, C, D, E, F, G, x will be shown on the cross-section.

Attachment 5, continued

```
                   PLOT OF FLOWLINES
           DISTANCE IN Z DIRECTION IN 1000 FT
         0.00      2.00      4.00      6.00      8.00
    0.00  +---------G----G----G--G--G--G--G-------+
          |         F    F    F  B    A A       |
          |G    F             EB   A            |
          |                        A            |
          |                                     |
          |G   F              EB    A           |
    5.00  +                   E  B    A         +
          GF        E   EE       DB   A         |
          |                        B            |
          GE        D    DD   D  D CB   A       |
          |                           B         |
          G D            C    CC C C  B         |
   10.00  +         C                           +
          |                                     |
          G C                          B        |
          |                                     |
          G C                           B  B B  |
          |                                     |
   15.00  GD        C  C                        +
          |              C    C  C C  C C  C    |
          GE     D                          C   |
          |         D                           |
 X DIS-   |              D    D  D D  D D  D C  C
TANCE IN  x         x    x    x  x x  x x  x x xG
   20.00  +         D    D    D  D D  D D  D C  +
          GE     D                          C   |
MILES     |                      C C  C C  C    |
          GD        C  C C    C                 |
          |                                     |
          |                                B B  |
   25.00  G C                           B       +
          |                                     |
          G C                          B        |
          |                                     |
          G D       C    C    CC C C  B         |
          |                           B         |
   30.00  GE        D    DD        CB           +
          |                   D  D B    A       |
          |                                     |
          GF        E   EE    E  DB   A         |
          |                      B              |
          G    F              EB    A           |
   35.00  +                        A            +
          |G    F             EB   A            |
          |         F    F            A A       |
          +---------G----G----G--G--G--G--G-------+


   40.00
```

Attachment 5, continued

The second cross-section shows computed head. The head is scaled and no more than three digits are displayed. The multiplying factor is printed at the end of the cross-section. The low order digit marks the location of the value. A check is made by the program to avoid overlapping displayed values. Consequently, depending on scale and configuration of the model, not all heads may be displayed.

Attachment 5, continued

```
                      PLOT OF HEAD
            DISTANCE IN Z DIRECTION IN 1000 FT
         0.00       2.00       4.00       6.00       8.00
  0.00   +----------+----------+----------+---------+
         |  100     100  100100100  100              |
         |                                           |
         |  100     100  100100100  100              |
         |                                           |
         |                                           |
  5.00   +  100     100  100100100  100              +
         |                                           |
         |  100     100  100100100  100              |
         |                                           |
         |  101     101  101100100  100              |
         |                                           +
 10.00   +                                           |
         |  111     109  107105103  100              |
         |                                           |
         |  171     171  171171171  171              |
         |                                           |
         |  230     231  233234236  239  243         |
 15.00   +                                           +
         |  241     241  241242242  243  243         |
         |                                           |
         |                                           |
 X DIS-  |  243     243  243243243  243  243         |
TANCE IN |                                           +
 20.00   +  243     243  243243243  243  243         |
         |                                           |
 MILES   |  241     241  241242242  243  243         |
         |                                           |
         |  230     231  233234236  239  243         |
         |                                           +
 25.00   +                                           |
         |  171     171  171171171  171              |
         |                                           |
         |  111     109  107105103  100              |
         |                                           |
         |  101     101  101100100  100              +
 30.00   +                                           |
         |                                           |
         |  100     100  100100100  100              |
         |                                           |
         |  100     100  100100100  100              |
         |                                           +
 35.00   +  100     100  100100100  100              |
         |                                           |
         |  100     100  100100100  100              |
         +----------+----------+----------+---------+

     40.00
MULTIPLICATION FACTOR = 1.0E+01
```

Attachment 5, continued

The third cross-section shows travel time in years. The displayed value is the exponent of the time. A display of "5" indicates that the time is greater or equal to 100,000 years and less than 1,000,000 years. The time is from the point to the discharge area. Travel time is computed for each active node. However, depending on scale and block dimensions, every time may not be displayed.

Attachment 5, continued

```
                PLOT OF TRAVEL TIME
             DISTANCE IN Z DIRECTION IN 1000 FT
            0.00      2.00      4.00      6.00      8.00
   0.00   +---------+---------+---------+--------+
          I    6       5    5  4  4 4  3          I
          I                                       I
          I    6       5    5  4  4 4  4          I
          I                                       I
          I                                       I
   5.00   +    6       5    5  5  4 4  4          +
          I    5       5    5  5  5 5  4          I
          I                                       I
          I    5       5    5  5  5 5  5          I
          I                                       I
  10.00   +                                       +
          I    5       5    5  5  5 5  5          I
          I                                       I
          I    5       5    5  5  5 5  5 5        I
          I                                       I
          I    5       5    5  5  5 5  5 5  5     I
  15.00   +                                       +
          I    5       5    5  5  5 5  5 5  5 5   I
          I                                       I
          I                                       I
 X DIS-   I    5       5    5  5  5 5  5 5  5 5   I
TANCE IN  I                                       I
  20.00   +    5       5    5  5  5 5  5 5  5 5   +
          I                                       I
 MILES    I    5       5    5  5  5 5  5 5  5 5   I
          I                                       I
          I    5       5    5  5  5 5  5 5  5     I
          I                                       I
  25.00   +                                       +
          I    5       5    5  5  5 5  5 5        I
          I                                       I
          I    5       5    5  5  5 5  5          I
          I                                       I
          I    5       5    5  5  5 5  5          I
  30.00   +                                       +
          I                                       I
          I    5       5    5  5  5 5  4          I
          I                                       I
          I    6       5    5  5  4 4  4          I
          I                                       I
  35.00   +    6       5    5  4  4 4  4          +
          I                                       I
          I    6       5    5  4  4 4  3          I
          +---------+---------+---------+--------+


  40.00
EXPONENT, 5 INDICATES 10.**5 <= VALUE < 10.**6
```

Attachment 5, continued

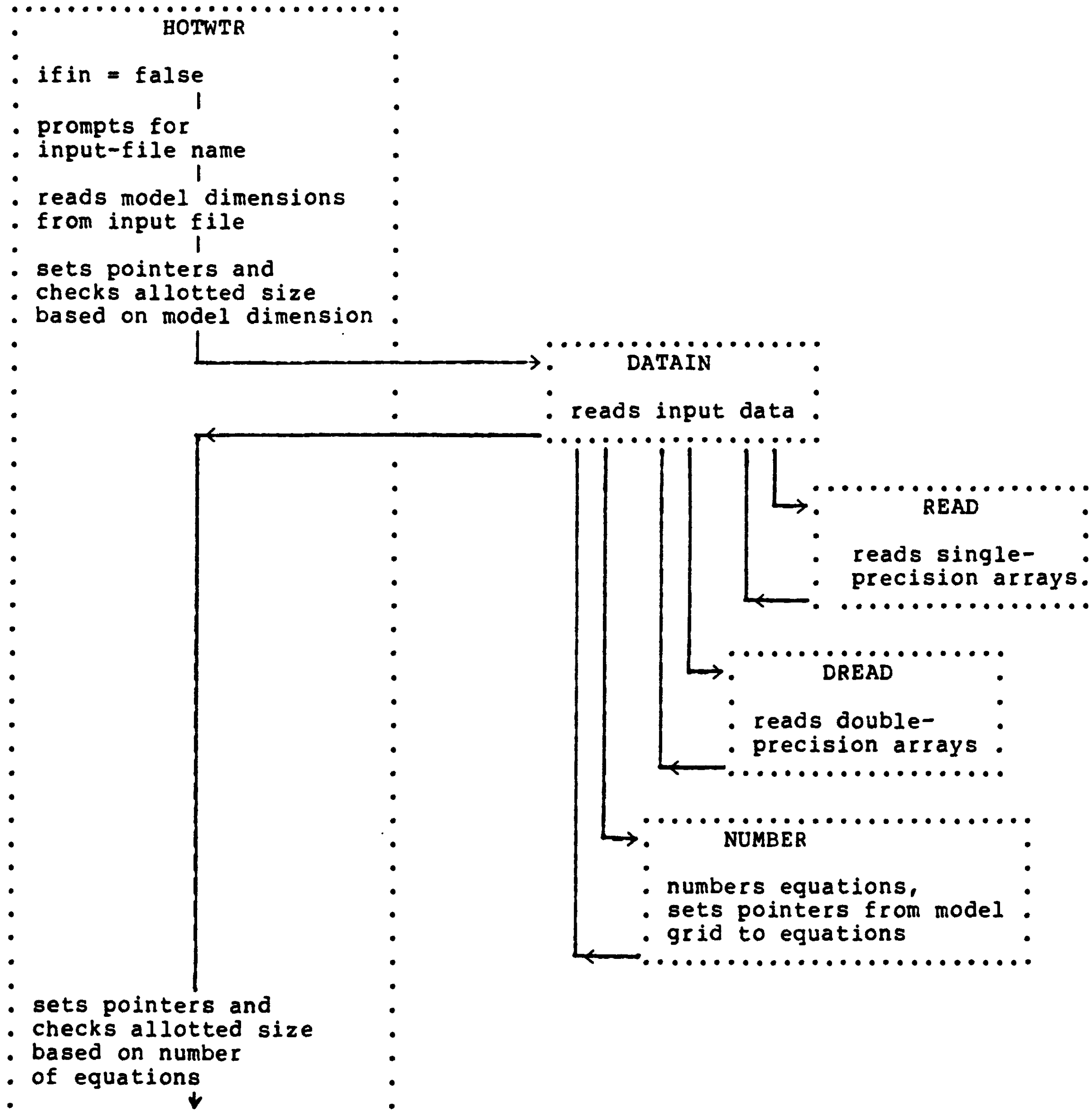
The last cross-section indicates the lithology of the cross-section. It shows the character data in the SEC array. It is also possible to indicate where the constant head boundaries are located. Depending on scale and block size, small blocks may not be displayed.

Attachment 5, continued

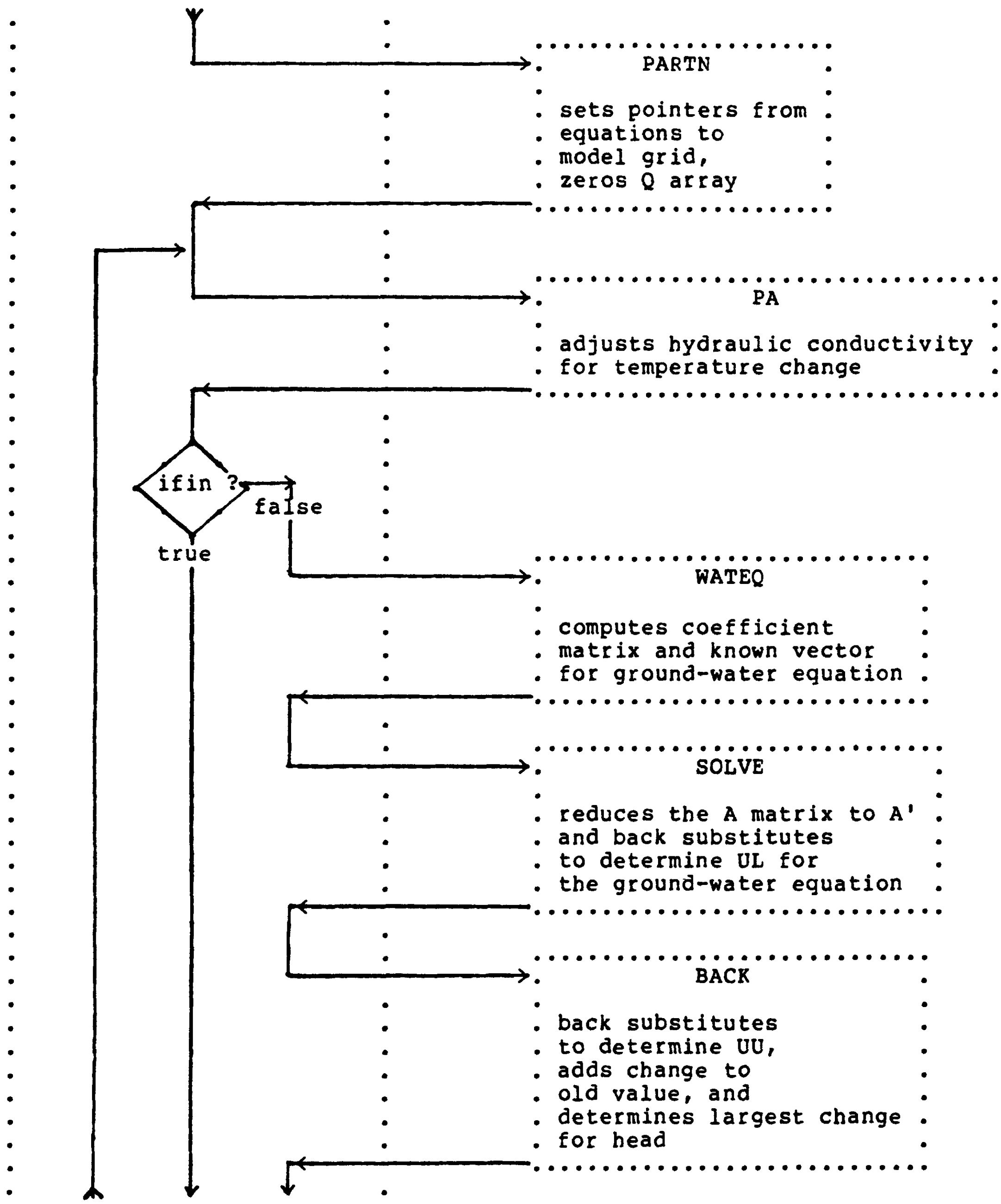
```
                       MODELED LITHOLOGY
                 DISTANCE IN Z DIRECTION IN 1000 FT
               0.00        2.00        4.00        6.00        8.00
     0.00  GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAAA22-----+
           GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAAA22     |
           GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAAA22     |
           GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAA        |
           GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAA        |
           GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAA        |
     5.00  GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAA        +
           GGGGGGGGGGGGGGGGGGGGGGAAAAAAAAA        |
           GGGGGGGGGGGGGGGGGGGGGGAAAAAAAAA        |
           GGGGGGGGGGGGGGGGGGGGGGAAAAAAAAA        |
           GGGGGGGGGGGGGGGGGGGGGGGGGGAAAAA        |
           GGGGGGGGGGGGGGGGGGGGGGGGGGAAAAA        |
    10.00  GGGGGGGGGGGGGGGGGGGGGGGGGGGGAAA        +
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGAAA        |
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG      |
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG      |
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBB   |
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBB   |
    15.00  GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBB   +
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBBBBB
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBBBBB
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBBBBB
   X DIS-  GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBBBBB
 TANCE IN  GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBBBBB
    20.00  GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBBBBB
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBBBBB
 MILES     GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBBBBB
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBBBBB
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBB   |
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGBBB   |
    25.00  GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG      +
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG      |
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGAAA        |
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGAAA        |
           GGGGGGGGGGGGGGGGGGGGGGGGGGGGAAA        |
           GGGGGGGGGGGGGGGGGGGGGGGGGGAAAAA        |
    30.00  GGGGGGGGGGGGGGGGGGGGGGGGGGAAAAA        +
           GGGGGGGGGGGGGGGGGGGGGGAAAAAAAAA        |
           GGGGGGGGGGGGGGGGGGGGGGAAAAAAAAA        |
           GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAA        |
           GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAA        |
           GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAA        |
    35.00  GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAA        +
           GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAA        |
           GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAAA22     |
           GGGGGGGGGGGGGGGGGGGGGAAAAAAAAAAA22-----+


    40.00
```

Attachment 6. Generalized flow chart for program.

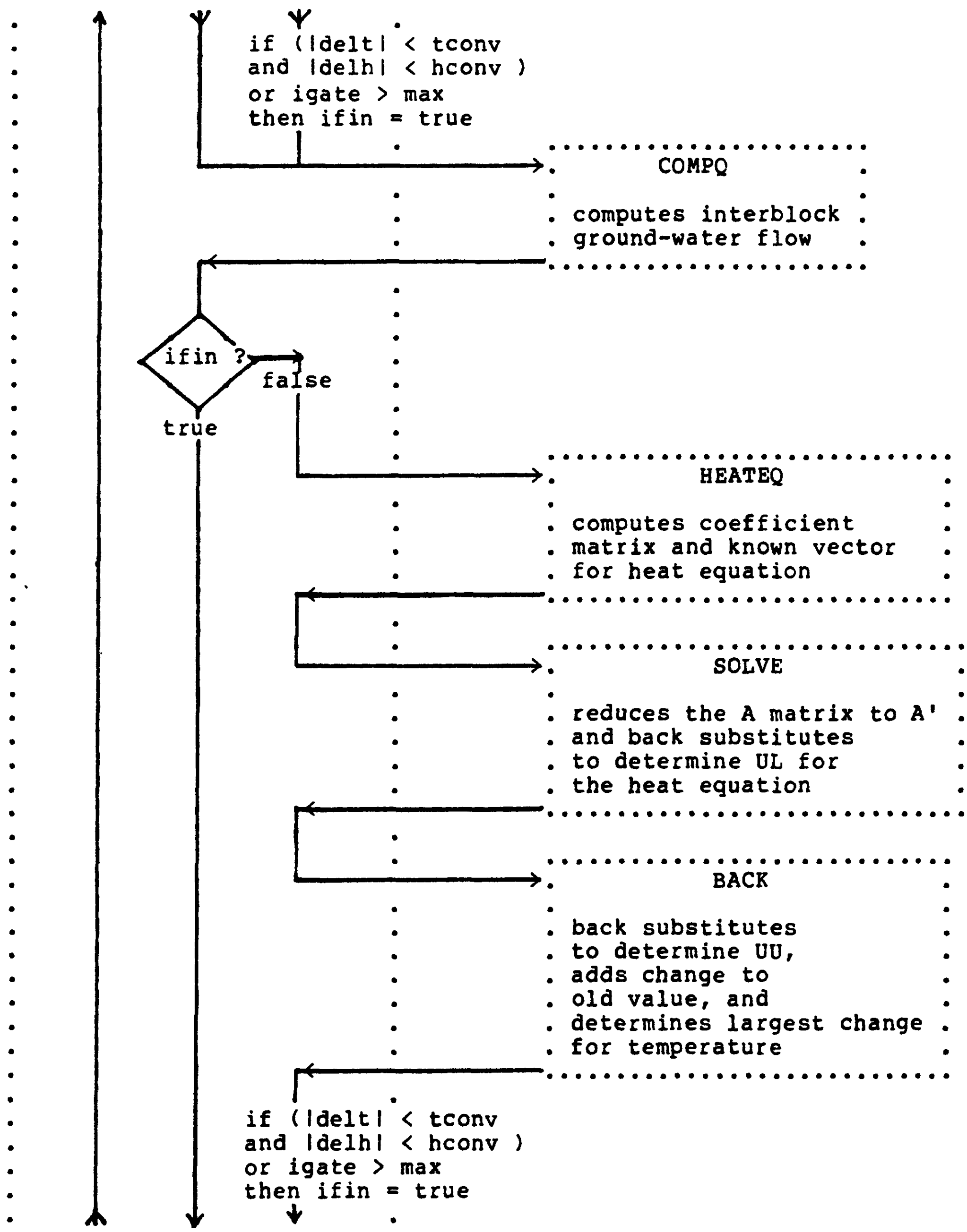
Attachment 6.



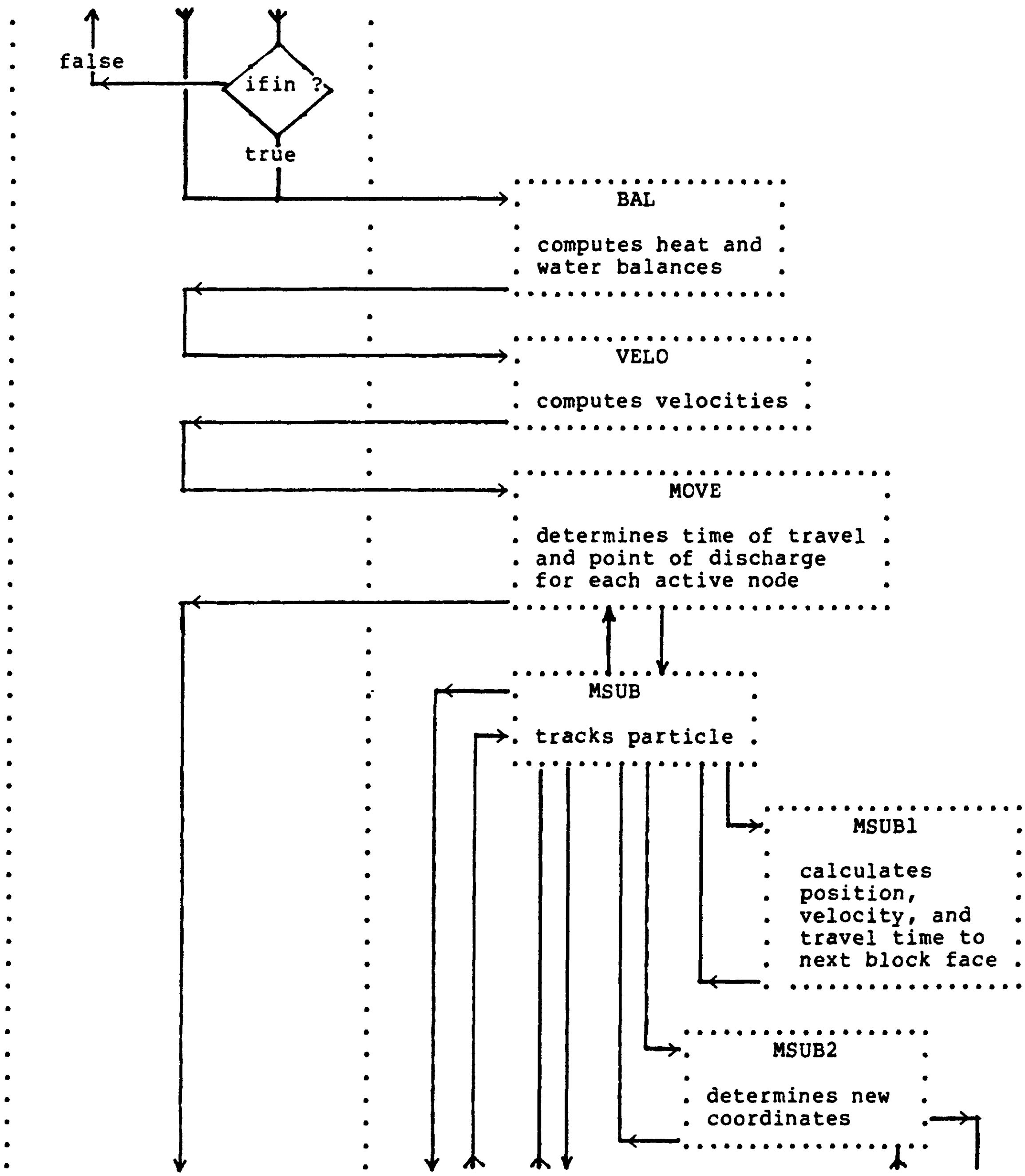
Attachment 6, continued



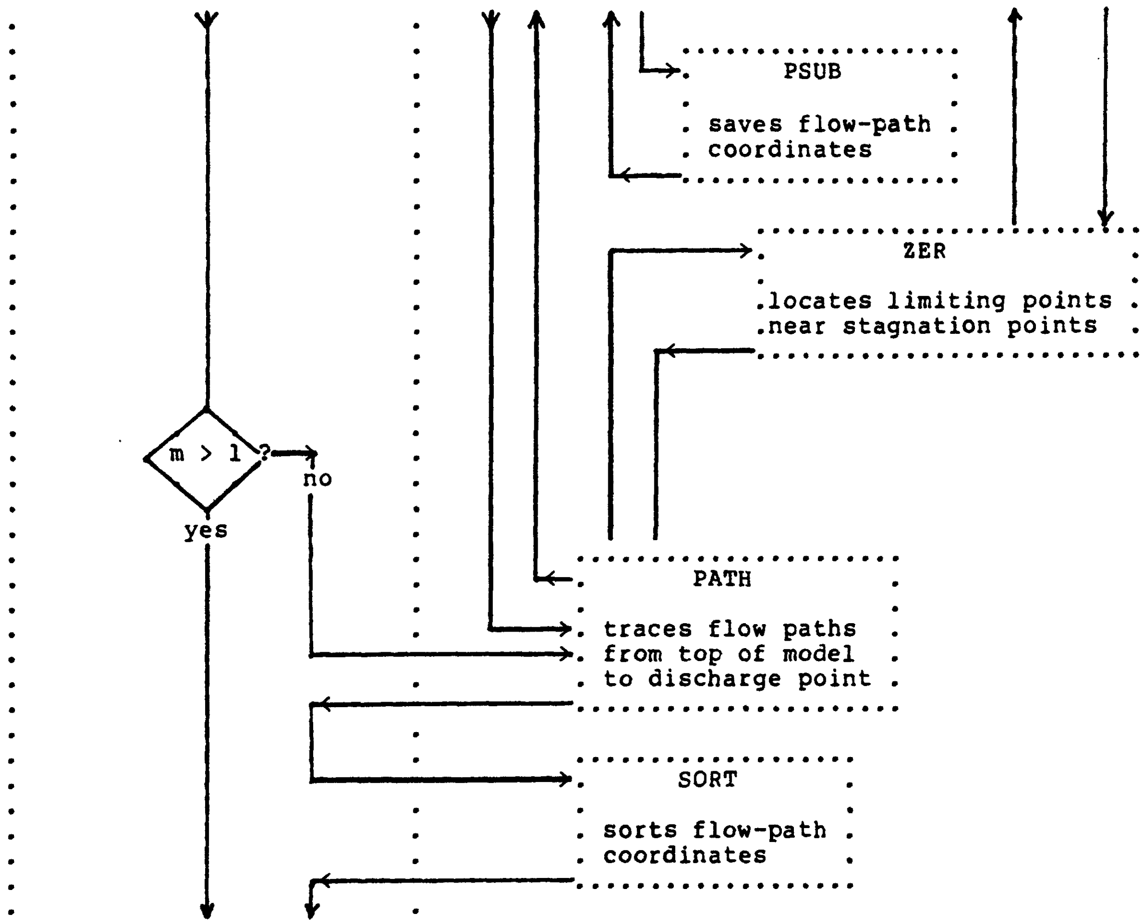
Attachment 6, continued



Attachment 6, continued



Attachment 6, continued



Attachment 6, continued