Ammerman

709

Waterflow in Soils: A Generalized Steady-State, Two-Dimensional Porous Media Flow Model

ARS-NC-30
April 1976

NORTH CENTRAL REGION

AGRICULTURAL RESEARCH SERVICE • U.S. DEPARTMENT OF AGRICULTURE

Acknowledgments

The writer is indebted to Dr. Donald Greenspan, University of Wisconsin, Madison, who provided much helpful advice regarding the use of finite difference methods. A large share of the cost of computing was borne by the Small Scale Waste Management Project, University of Wisconsin, Madison, directed by Dr. Johannes Bouma and funded by the State of Wisconsin and the Upper Great Lakes Regional Commission. Marion Mazzacco of the USDA-ARS Watershed Research unit, Columbia, Mo., has been most helpful in preparing the final version of the model.

In cooperation with

Wisconsin Agricultural Experiment Station

Missouri Agricultural Experiment Station

Contents

Waterflow in Soils: A Generalized Steady-State, Two-Dimensional Porous Media Flow Model

C. R. Amerman[1]

Introduction

Understanding water movement through soil and underlying unconsolidated material is basic to understanding important aspects of other such phenomena as the hydrologic cycle and the movement of waterborne substances through the landscape. Unfortunately, subsurface water movement under natural conditions cannot be directly observed. Methods of indirect observation are usually difficult, tedious, and expensive. Further, instrumentation for such methods may modify the system under observation. Although there is no real substitute for carefully made field observations to provide initial concepts and to check theoretical results, computational efforts can save large amounts of field labor and expense. Such methods also often provide clearer, broader concepts than would be available analyzing field data alone.

This report discusses a finite difference model of the hydraulic head distribution within two-dimensional regions of porous media subject to steady flow. Soil water content, water table position and shape, pathlines of flow, and flow velocities can be estimated from such a distribution.

The range of applicability and limitations of the model may be summarized as:

1. Flow system boundary geometries must be approximated with straight-line segments. Straight-line boundaries which do not parallel Cartesian coordinate axes can be modeled but require considerably more effort than those that do.
2. The Cartesian coordinate system may be rotated so that the major axis of the flow system is on a slope.
3. Boundary conditions must be in terms of pressure head or flux, or both; at least a portion of the boundary, however, must have a known pressure head.
4. A cross section may contain several soil units. The boundaries between units may be geometrically complex.
5. Soils within each unit of a modeled system are considered to be isotropic and homogeneous.
6. Hysteresis in the hydraulic conductivity pressure head relationship is ignored.
7. Spacing of nodes within the finite difference solution mesh may be irregular.

The usual assumptions regarding porous media flow apply to this model:

1. Inertial forces are not significant as compared with viscous forces.
2. Water is continuously connected throughout the system.
3. Flow is isothermal.
4. Air escapes freely from all parts of the flow system.

The flow equation modeled may be described as an elliptic partial differential equation with mixed boundary conditons. The finite difference method is used, and the resulting system of equations is solved by the successive overrelaxation (SOR) method. The model takes the form of a digital computer program written in USASI Fortran.

The following pages describe the model and its application in detail. A sample cross section is modeled; the sample input and results given may be used to check the operation of the model when implementing it for the first time. Appendixes document the program and describe two useful auxiliary programs.

Partial Differential Equation

A number of textbooks discuss the theory of soil water movement. Childs (*2*)[2] gives a detailed but quite readable mathematical description, while resumes are given by Hillel (*5*) and by Baver and associates (*1*). In general, porous media flow may be modeled

---

[1] Hydraulic Engineer, Watershed Research, Agricultural Research Service, U.S. Department of Agriculture, 207 Business Loop 70 East, Columbia, Mo. 65201

[2] Italic numbers in parentheses refer to Literature Cited, p. 27.

by a partial differential equation, called Richards' equation, and associated initial and boundary conditions. Solution of a steady-state version of Richards' equation in two dimensions requires only boundary conditions and is approximated by the model presented here.

Richards' equation is derived by combining equations of state and continuity with Darcy's law. For steady state, it may be written

$$\frac{\partial}{\partial x}\left(K\frac{\partial H}{\partial x}\right)+\frac{\partial}{\partial y}\left(K\frac{\partial H}{\partial y}\right)=0 \qquad [1]$$

in which

H = hydraulic head = $h + z$ for porous media flow (L)
h = soil water pressure head (L)
z = elevation above a datum (L)
$K = K(h)$ is hydraulic conductivity (LT^{-1})
x = distance parallel to the x-axis of the Cartesian coordinate system, positive to the right (L)
y = distance parallel to the y-axis, positive upward (L)

To accommodate a sloping soil, rotating the Cartesian coordinate axes through an angle, α, is convenient. The tangent of this angle should be equal to the slope of the prototype system. Because of the dependence of K upon h, the so-called h-based form of Richards' equation is used. Stated for general orientation, this form of Richards' equation is

$$\frac{\partial}{\partial x}\left(K\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(K\frac{\partial h}{\partial y}\right) + \sin\alpha\frac{\partial K}{\partial x}+\cos\alpha\frac{\partial K}{\partial y}=0 \qquad [2]$$

Equation [2] is a nonlinear, elliptic, partial differential equation.

Boundary conditions are essentially of two types: (1) pressure head (h) specified and (2) hydraulic gradient ($\partial h/\partial x$ or $\partial h/\partial y$) specified. Given a zero value, the latter represents impermeable boundaries or the coincidence of streamlines of flow with the boundaries. Given non-zero values, a hydraulic gradient boundary condition represents a flux boundary. The model to be described contains an algorithm that allows the flux boundary condition to be stated in terms of the flux itself.

There is no closed form solution for equation [2]. Analytical methods can be applied to the equation for certain special situations but, for the general case, it must be solved by such approximate methods as finite differences.

Finite Difference Model

Finite difference equation

Whereas equation [2] applies throughout a two-dimensional flow region, finite differencing provides another equation at each of a set of discrete points, superimposed on the cross section of that region. Associated boundary conditions are applied at discrete points along the boundaries. The set of discrete points, called nodes, is arranged in a grid, termed the solution mesh in this report, such as that in figure 1.

The sizes of the mesh increments, Δx and Δy, influence the precision with which the finite difference model represents the partial differential system. The smaller the mesh increment size, the greater the precision. These concepts are covered in many textbooks; for example, see Smith (*8*). In many flow systems, precision requirements are not uniform over the cross section and, for economic reasons, Δx and Δy are often varied. They may be varied independently.

The finite difference equation approximating equation [2] was developed using the central difference method. By this method, the pressure head at each node is a function of the heads at its four nearest neighboring nodes. For example, in figure 1

$$h_P = f(h_A, h_B, h_C, h_D)$$

To simplify and organize notation, I = 1, 2, 3, . . . gives column identification to all the nodes of each row, while J = 1, 2, 3, . . . gives row identification to all nodes in each column. Thus node P is node$_{7,5}$. If h at point P is identified as $h_{i,j}$, then h at point A is $h_{i-1,j}$ and h at point B is $h_{i,j-1}$. Note that the directions in which I and J increase are independent of those in which x and y increase. Because of Fortran limitations, I and J can take on only positive non-zero values. The x-axis most conveniently coincides with the soil surface and, in this notational scheme, J increases downward even

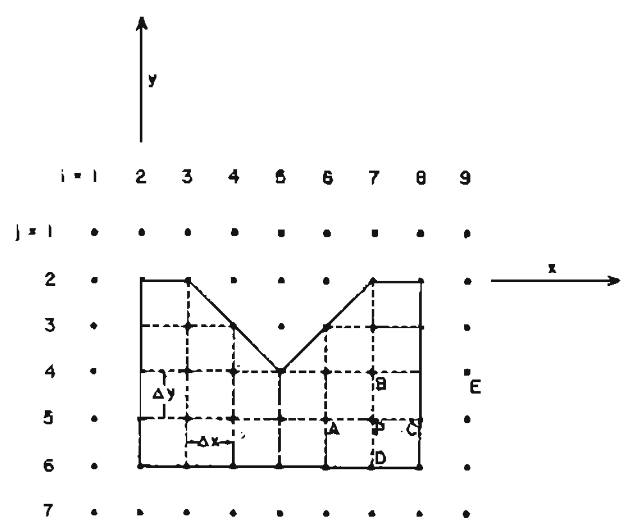


Figure 1.--Square mesh for finite differencing superimposed on a two-dimensional flow field.

though y is considered to be positive upward. Using this notation and conditions at the four neighboring nodes, a finite difference representation is expressed in equation [3].

$$\frac{2}{\Delta x_- + \Delta x_+}\left[\frac{K_{i-\frac{1}{2},j}h_{i-1,j}}{\Delta x_-} - \frac{(\Delta x_+ K_{i-\frac{1}{2},j} + \Delta x_- K_{i+\frac{1}{2},j})h_{i,j}}{\Delta x_- \cdot \Delta x_+} + \frac{K_{i+\frac{1}{2},j}h_{i+1,j}}{\Delta x_+}\right]$$
$$+ \frac{2}{\Delta y_- + \Delta y_+}\left[\frac{K_{i,j-\frac{1}{2}}h_{i,j-1}}{\Delta y_-} - \frac{(\Delta y_+ K_{i,j-\frac{1}{2}} + \Delta y_- K_{i,j+\frac{1}{2}})h_{i,j}}{\Delta y_- \cdot \Delta y_+} + \frac{K_{i,j+\frac{1}{2}}h_{i,j+1}}{\Delta y_+}\right]$$
$$+ \sin\alpha \frac{K_{i-1,j} - K_{i+1,j}}{\Delta x_- + \Delta x_+} + \cos\alpha \frac{K_{i,j-1} - K_{i,j+1}}{\Delta y_- + \Delta y_+} = 0 \qquad [3]$$

in which

$$K_{i-\frac{1}{2},j} = \frac{K_{i-1,j} + K_{i,j}}{2}, \qquad K_{i+\frac{1}{2},j} = \frac{K_{i,j} + K_{i+1,j}}{2}$$

$$K_{i,j-\frac{1}{2}} = \frac{K_{i,j-1} + K_{i,j}}{2}, \qquad K_{i,j+\frac{1}{2}} = \frac{K_{i,j} + K_{i,j+1}}{2}$$

and

Δx_- = mesh increment length to left of node i,j (L)

Δx_+ = mesh increment length to right of node i,j (L)

Δy_- = mesh increment length above node i,j (L)

Δy_+ = mesh increment length below node i,j (L)

Because of the appearance of $K = K(h)$ in equation [2], equation [3] was developed by inspection--simply transforming partial derivatives into ratios of differences. However, Forsythe and Wasow (3) on pages 187 and 188 give formal derivations of linear partial derivatives that support the validity of equation [3] for the nonlinear case. They also note that, if Δx, for example, varies over the cross section, one may expect an error of $O(\Delta x)$. For a square mesh in which $\Delta y = \Delta x$, the error is $O[(\Delta x)^2]$.

"Imaginary" rows and columns outside the cross section provide the fourth node for calculations involving a node on gradient-type boundaries. To illustrate, if the right side of the cross section in figure 1 is impermeable, then $\partial H/\partial x = 0$ and H at each node to the right of the boundary is set exactly equal to the H-value for the node horizontally to its left immediately inside the boundary, for example, $H_E = H_B$. Because $H = h + z$, to convert H to h at the imaginary node is a simple operation regardless of orientation of the system.

For a given geometry and set of boundary conditions, the flow regime may be defined, with some error or lack of precision as noted earlier, by the distribution of pressure head (h) that satisfies equation [3] at all nodes of the solution mesh. To find this distribution, a starting array of h-values, which may be completely arbitrary, is used. Equation [3] is then solved for each node of the solution mesh except those on h-specified boundaries and except for the imaginary nodes outside the gradient (flux) boundaries. If subsectioning, discussed later, is carried out properly, solution starts with the left-most node on the top row and proceeds to the right along that row. Lower rows are processed in succession, also from left to right. A complete cycle of solving equation [3] once for all nodes constitutes an iteration, the h-array at the end of an iteration being in some way a closer approximation to the solution array than that at the beginning. Many such iterations are usually necessary before convergence to the final distribution of h-values.

Overrelaxation

Experience has shown that an overrelaxation factor (ω) may speed convergence of a finite difference model of the type of equation [3]. If $h = f(g)$, then

$$\omega h + h = \omega f(g) + h$$

or

$$h = h(1-\omega) + \omega f(g)$$

where ω has a value between 1.0 and 2.0. When full convergence is reached, of course, h on the left side, h on the right side, and $f(g)$ are all equal. Before reaching convergence, h on the right side has the value calculated during the preceding iteration, whereas h on the left is the new estimate to be calculated during the current iteration.

Solving equation [3] for $h_{i,j}$ and introducing the overrelaxation factor give the equation that is solved for each node during operation of the model.

$$h_{i,j} = (1-\omega)\, h_{i,j}$$

$$+\ \omega\, [\frac{2}{\Delta x_- + \Delta x_+}\, (\frac{K_{i-\frac{1}{2},j}}{\Delta x_-}\, h_{i-1,j} + \frac{K_{i+\frac{1}{2},j}}{\Delta x_+}\, h_{i+1,j})$$

$$+\ \frac{2}{\Delta y_- + \Delta y_+}\, (\frac{K_{i,j-\frac{1}{2}}}{\Delta y_-}\, h_{i,j-1} + \frac{K_{i,j+\frac{1}{2}}}{\Delta y_+}\, h_{i,j+1})$$

$$+\ \sin\alpha\, \frac{K_{i-1,j} - K_{i+1,j}}{\Delta x_- + \Delta x_+} + \cos\alpha\, \frac{K_{i,j-1} - K_{i,j+1}}{\Delta y_- + \Delta y_+}]$$

$$\div\ [\frac{2}{\Delta x_- + \Delta x_+}\, (\frac{\Delta x_+ \cdot K_{i-\frac{1}{2},j} + \Delta x_- \cdot K_{i+\frac{1}{2},j}}{\Delta x_- \cdot \Delta x_+})$$

$$+\ \frac{2}{\Delta y_- + \Delta y_+}\, (\frac{\Delta y_+ \cdot K_{i,j-\frac{1}{2}} + \Delta y_- \cdot K_{i,j+\frac{1}{2}}}{\Delta y_- \cdot \Delta y_+})] \qquad [4]$$

Equation [4] is a successive overrelaxation (SOR) model of equation [2] and, with associated boundary conditions, approximates two-dimensional, steady-state, saturated, unsaturated, or partially saturated porous media flow using a finite difference mesh in which the mesh increment size may vary from one part of the flow region to another and in which the major axes of the mesh may be rotated to conform to the slope of the prototype. Greenspan (*4*) discusses SOR models as does Smith (*8*) and Forsythe and Wasow (*3*).

Interation scheme

The SOR model converges toward the actual h-distribution most rapidly if the new h-value at any node replaces the old value in the h-array as soon as it is calculated. Thus, when $h_{i,j}$ is being calculated, $h_{i-1,j}$ and $h_{i,j-1}$ are new values calculated during the current iteration, whereas $h_{i+1,j}$ and $h_{i,j+1}$ are old values from the preceding iteration. New K-values would seem appropriate for use with new h-values, but experience has shown that this practice is less efficient than the use of old K-values. The latter practice results in a larger maximum overrelaxation factor (ω_{max}), hence more rapid convergence, than is possible with new K-values. The concept of ω_{max} will be discussed in greater detail later.

Equation [4] may be modified to show the iteration scheme used. Also introduced at this point is simplified notation used in the computer program for the model.

$$h^m_{i,j} = (1-\omega)\, h^{m-1}_{i,j} + \omega\, [\frac{EX(AX^{m-1}h^m_{i-1,j} + CX^{m-1}h^{m-1}_{i+1,j}) + EY(AY^{m-1}h^m_{i,j-1} + CY^{m-1}h^{m-1}_{i,j+1}) + DELTA^{m-1}}{EX \cdot XB^{m-1} + EY \cdot YB^{m-1}}] \qquad [5]$$

where m refers to values obtained during the current iteration.

m-1 refers to values obtained in the preceding iteration.

$$AX = \frac{K^{m-1}_{i-\frac{1}{2},j}}{\Delta x_-}, \qquad CX = \frac{K^{m-1}_{i+\frac{1}{2},j}}{\Delta x_+}$$

$$AY = \frac{K^{m-1}_{i,j-\frac{1}{2}}}{\Delta y_-}, \qquad CY = \frac{K^{m-1}_{i,j+\frac{1}{2}}}{\Delta y_+}$$

$$XB = \frac{\Delta x_+ \cdot K^{m-1}_{i-\frac{1}{2},j} + \Delta x_- \cdot K^{m-1}_{i+\frac{1}{2},j}}{\Delta x_- \cdot \Delta x_+}$$

$$YB = \frac{\Delta y_+ \cdot K^{m-1}_{i,j-\frac{1}{2}} + \Delta y_- \cdot K^{m-1}_{i,j+\frac{1}{2}}}{\Delta y_- \cdot \Delta y_+}$$

$$DELTA = \sin\alpha\, \frac{K^{m-1}_{i-1,j} - K^{m-1}_{i+1,j}}{\Delta x_- + \Delta x_+} + \cos\alpha\, \frac{K^{m-1}_{i,j-1} - K^{m-1}_{i,j+1}}{\Delta y_- + \Delta y_+},$$

$$EX = \frac{2}{\Delta x_- + \Delta x_+}$$

$$EY = \frac{2}{\Delta y_- + \Delta y_+}$$

Nonlinearity and convergence

As noted earlier, the Richards' equation, hence its finite difference approximation, is nonlinear. Because finite difference theory has been developed almost exclusively in the linear context, there are no firm guidelines on the application or operation of nonlinear models.[3]

The lack of a body of theory covering nonlinear finite differencing is felt most keenly when considering questions of convergence and the rate of convergence. A model converges if it converts an initial guess regarding the distribution of the dependent variable to an approximation to the true distribution. Fortunately, experience indicates that many finite difference schemes developed for linear systems also converge for nonlinear systems even though there is no theoretical proof that they should. However, sometimes certain modifications are necessary.

The rate of convergence is a concept of some importance to the economical use of finite difference models. The overrelaxation factor (ω) was introduced to speed convergence. Forsythe and Wasow (*3*) show on page 257 that, for linear systems, as ω increases in value between 1.0 and 2.0, convergence rate increases until some maximum rate is reached. Further increases in ω result in decreasing convergence rate until at $\omega = 2.0$ there is essentially no improvement over $\omega = 1.0$. For linear systems, overestimating the optimum ω-value (ω_{opt}) is usually better than underestimating it.

Experience with nonlinear models of unsaturated porous media flow systems indicates that ω_{opt} cannot be estimated using the procedures that apply to a geometrically similar linear system. Further, Reisenauer and others (*6*) found that $\omega > 1.15$ led to instability of their model, that is, the solution did not converge for larger ω. The author's experience also indicates that, for nonlinear systems, the concept of ω_{opt} should be modified to one of ω_{max}, or the maximum ω-value with which convergence can be obtained. Apparently, increasing ω toward ω_{max} increases convergence rate. The value of ω_{max} differs between cases, that is, between different combinations of boundary geometry, boundary hydraulic conditions, $\Delta x, \Delta y$ magnitudes and h-K relationships, and may vary between the first and last iterations for a given case. For the several cases investigated thus far, its value has been less than 2.0. In certain cases of complex geometry, ω_{max} has had a value smaller than 1.0.

[3]Greenspan, D., 1973, personal communication.

The only method for approximating ω_{max} seems to be trial and error. This reduces the economic advantage of finding ω_{max}, so that exhaustive search for its value would probably be more expensive than simply running the model with some less exact value. Because different cases involve different convergence rates and different amounts of computational time per iteration, each user must develop from his own experience a feel for the amount of trial and error to be expended in approximating ω_{max}. He should keep in mind that ω simply influences convergence rate; it does not affect the accuracy of the approximation to the true h-distribution unless ω_{max} is exceeded.

For fully saturated flow in which K is independent of h, equations [1] and [2] become linear. For such systems, the model described here also becomes linear, and the methods for approximating ω_{opt}, given in the references previously cited, may contribute to considerable savings in the number of iterations necessary for convergence.

Digital Computer Model

Model philosophy

The only feasible way to apply equation [4] iteratively to a small mesh of few nodes is by digital computer. The objective of the effort reported here was to develop a computer program for the application of equation [4] using the iteration scheme portrayed in equation [5] to the solution of porous media flow problems under a variety of geometrical and hydraulic boundary conditions. Hopefully, users with little experience in computer programming and finite differencing can use the model. USASI Fortran was used to reduce problems when using the model on different computer facilities.

To model soil-water movement in all its complexity and to provide for all the possible contingencies encountered in hydrologic systems require a complex program difficult to understand, describe, or modify. Fortunately, considerable insight into porous media flow questions can often be gained without strict attention to emulating all details of the prototype flow system.

Some details cannot be measured with great enough precision nor at enough points in a given system to warrant trying to model them with great accuracy. For example, hysteresis effects in the hydraulic conductivity-pressure head relationships may exert less influence on the system than the errors inherent in establishing the relationships themselves, particularly if they are to apply to flow regions of large extent and exhibiting spatial variation.

Again, for larger systems, closely defining the exact positions and shapes of all boundaries in the prototype is not usually necessary or possible. In many cases, satisfactory results can be attained using only rough boundary approximations.

The computer program, called STDY2, is documented in appendix A. The following sections discuss concepts that are helpful or necessary to the use of the model.

Solution mesh

The solution mesh is represented in a digital computer by an array of storage locations identified with the variable PHED(I,J). The latter is the Fortran representation of the variable $h_{i,j}$ (pressure head at node I,J) in equation [4]. Each storage location corresponds to a node in the solution mesh. The effect of solving equation [4] for a given node for a given iteration is to replace the value of PHED(I,J) calculated during the previous iteration with a new, improved value.

Mesh increment size is not physically reflected in the PHED storage array but is controlled through the use of four Fortran variables representing Δx_-, Δx_+, Δy_-, and Δy_+.

Through use of the Fortran EQUIVALENCE statement, the HEAD(I,J)-array corresponding to hydraulic head ($H_{i,j}$) replaces PHED(I,J) at certain stages of program execution, thus avoiding the need for an additional storage array.

A second two-dimensional storage array, with a location for each node of the solution mesh, is occupied by HCON(I,J). The latter represents $K_{i,j}$, the hydraulic conductivity.

A fundamental concept necessary to understanding model control is that the model proceeds from an initially guessed array of PHED(I,J)-values by means of a series of iterations to a solution array of PHED(I,J). One may view the PHED-array at the end of any iteration as the initial guess for all the iterations to follow. Therefore, a computer run can be interrupted and restarted without loss of significant computer time if the PHED-array at the time of interruption can be returned as the initial guess when restarting.

Boundary geometry

Two characteristics define boundaries, their geometric shape and their hydraulic condition or status. To avoid excessive complexity and programming, the model was designed with the restriction that boundaries must cross rows and columns of the finite difference mesh at the nodes. Because the mesh is rectangular, boundary shapes must be composed of straight-line segments. Usually, these segments will coincide with portions of rows or columns, but placing them at an angle is possible by adjusting the relative size of horizontal and vertical mesh increments in the region crossed by this boundary. In a square mesh, for example, a boundary at a 45° angle will cross rows and columns only at their intersection nodes, as desired. Curved boundaries may be approximated in stairstep fashion.

Solution mesh and the cartesian coordinate system

Various data involving geometric information must be given as punchcard input for control of the model in a computer. The user will understand how to determine numerical values for these data if he thinks of the cross section of interest as being placed in the Cartesian coordinate system and the solution mesh superimposed thereon.

In a computer, control of the model is accomplished using the variables I and J, so the solution mesh must be placed on the model cross section in such a way that I and J may be calculated from x- and y-measurement This means that a column of nodes which is fixed in space and whose I-value is known, regardless of Δx, must be identified and related spatially to the y-axis of the Cartesian coordinate system. The same may be said for a row of nodes in the context of the x-axis. Rows and columns coincident with boundaries of the cross section are fixed spatially, and the top and left-hand boundaries, if straight lines, may be made to coincide with the x- and y-axes of the Cartesian coordinate system. Even if a boundary is complex, one or more of its straight-line segments may be made to coincid with an axis, as in figure 1.

Because I and J can take on only nonzero, positive values, and recalling that J is positive downward, a cross section must be contained entirely within quadrant IV of the Cartesian coordinate system. Thus, proper model control requires that the uppermost straight-line segment of the upper boundary of the cross section be made to coincide with the x-axis and the leftmost straight-line segment of the left boundary be made to coincide with the y-axis. The type of boundary condition to be applied is of no consequence in these considerations. The solution mesh will be adjusted by a computational algorithm in the model without loss of correspondence between x,y and I,J if the top row of nodes or the left-hand column of nodes, or both, must be imaginary.

Equation [1] was formulated for y positive upward, and this should be kept in mind for such purposes as assigning positive or negative sign to a surface boundary flux. But, to require that y-measurements for geometrical control be given a negative sign may lead to frequent errors of omission. Therefore, the model is programmed to accept positive y-measurements even though they are made downward from the x-axis.

Hydraulic boundary conditions

The model simulates hydraulic boundary conditions of the following types:

1. Hydraulic head on any boundary (may vary hydrostatically along vertical boundaries).
2. Steady flux across soil surface boundary only (infiltration or evapotranspiration rates).
3. Impermeable condition on any boundary (may also be the vertical streamline boundary between two halves of a symmetrical flow region).

Type 1 is the so-called Dirichlet boundary condition. Types 2 and 3 are each implemented in terms of the hydraulic gradient perpendicular to the boundary, called the Neumann boundary condition. Many porous media flow regions have boundaries that are combinations of the Dirichlet and Neumann types and thus belong to the general classification of mixed problems (in the context of elliptic partial differential equations).

A unique solution is assured for Dirichlet and mixed problems but not for the Neumann. Greenspan (4) and Remson and others (7) note that nonuniqueness of a Neumann problem is limited to an unknown additive constant. Thus, $h' = h + c$ would be calculated with c unknown so that the h-K relationship could not be used. Therefore, at least part of the boundary of any porous media flow model must have a known pressure head.

Neumann-type boundaries also have the disadvantage that their implementation in finite differences can only be done by approximation. The resultant errors add to the errors inherent in the finite differencing technique. In general, the greater the proportion of Neumann-type boundaries, the greater the model error. This can be partially overcome by using smaller mesh increments near such boundaries.[4]

Later discussion will be clearer if the user understands that the h-values at all nodes except those on imaginary rows and columns and nodes on h-specified boundaries are calculated by means of the same version of equation [4]. To implement Neumann-type boundary conditions, the proper h-values at imaginary nodes are calculated and assigned before solving for h at the boundary node.

The equation by which a flux boundary condition is applied at the soil surface is derived from Darcy's law as

$$h_{i,j-1} = h_{i,j+1} - \Delta z(1 + \frac{v}{K_a}) \qquad [6]$$

where v = flux (LT^{-1})

Δz = vertical separation between nodes i,j-1 and i,j+1 $= \cos\alpha\ (\Delta y_- + \Delta y_+)$

K_a = average hydraulic conductivity $= \dfrac{K_{i,j-1} + K_{i,j} + K_{i,j+1}}{3}$

When $v = 0$, equation [6] reduces to the equation for calculating h for an impervious surface.

Dirichlet-type boundary conditions are implemented by assigning h-values at appropriate nodes and making certain that equation [4] is not processed for those nodes. The procedures for this are outlined in the next section.

In cases where a water table (zero isobar) intersects such a pervious boundary exposed to the atmosphere as the bank of a stream or ditch, a surface of seepage develops. The boundary above the surface of seepage, being a boundary to an unsaturated zone, is usually considered impermeable for modeling purposes. Water leaving the flow system across the surface of seepage is assumed to run down that surface as a thin film. The latter is usually considered, then, to form a saturated boundary with a pressure head of 0 cm of water.

The position of a water table is usually not known before modeling and, thus, the limits of the surface of seepage are not known. Considerable checking and cross checking would be necessary to determine these limits by means of the model, and part of these checks would have to be made for all boundaries under all conditions. The model was not, therefore, designed to determine automatically the position of a surface of seepage and, thus, will not determine automatically the correct shape of a water table which intersects a pervious surface exposed to the atmosphere.

Where a surface of seepage is expected, the following procedure will approximate its

[4]See footnote 3, page 5.

correct limits and, thus, the correct shape of the water table:

1. Assume the position of the zero isobar intercept on the boundary and assign an impervious condition to the boundary nodes above and a zero pressure head condition to the boundary nodes below.

2. Run the model to obtain the solution for the given boundary conditions.

3. When the solution shows positive pressure heads at boundary nodes above the assumed intercept, move the intercept higher and run again with boundary conditions revised accordingly.

4. When the solution shows no positive pressure heads above the assumed intercept, it may have been placed too high. This possibility should be checked by lowering the intercept, revising boundary conditions, and running again.

5. The best location of the intercept is the lowest node for which the solution does not show positive pressure heads on the boundary above the intercept.

Subdivision of flow cross section

The key to making this model flexible regarding boundary geometry and boundary conditions is the concept of subdivision of the cross section. The parameters defining subsections are used to describe the geometry and boundary conditions of the cross section to be modeled. Specifically, they

1. Direct the flow of the program so that only appropriate nodes are processed by equation [4], that is, nodes outside the boundaries or on h-specified boundaries are not processed.

2. Control the calculation of h for nodes on imaginary columns and rows before applying equation [4] to the neighboring Neumann-type boundary nodes.

3. Cause the program to apply known or calculated pressure heads at nodes on h-specified boundaries.

There are two sets of subsections, one for rows and one for columns. The description of one suffices to describe the other. A given row subsection, for example, contains a group of rows which are identical from the program processing standpoint. That is, processing starts on the same column and ends on the same column. Beginning and ending boundary conditions are the same. Other such considerations as variable Δx and Δy and soil unit geometry do not affect the selection of subsections.

Consider, for example, figure 2 that portrays a half cross section of a typical septic tank disposal line. (A similar cross section will be modeled in the sample problem given later.)

Boundary $\overline{AB}$ represents the soil surface

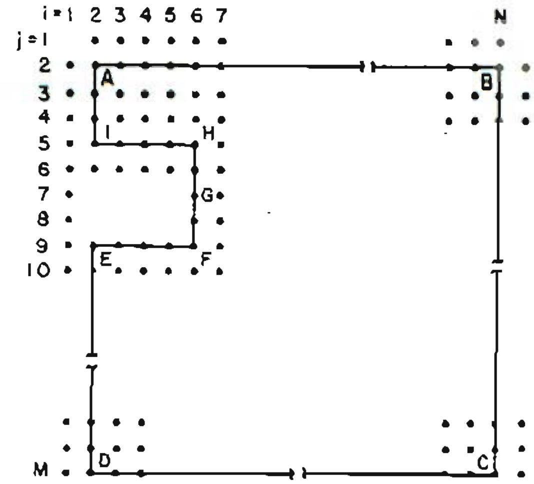


Figure 2.--Half section of septic tank disposal field indicating finite difference mesh overlay.

and may have applied to it any of the three types of boundary conditions mentioned.

Boundary $\overline{BC}$ may be a line of symmetry, hence a stream line, if the disposal field has several lines. When there is only one tile line, $\overline{BC}$ may be arbitrarily positioned or positioned by trial and error at such a distance that further outward movement affects the solution little in the region of the tile line--in effect an infinite boundary. In either case, the hydraulic gradient normal to the boundary would be given a value of zero (0).

Boundary $\overline{CD}$ may represent a water table by applying to it a zero pressure head. Boundaries $\overline{DE}$ and $\overline{IA}$ are boundaries of symmetry and, therefore, have zero normal hydraulic gradients. Boundary $\overline{EFG}$ represents a crusted infiltration zone, point G being at the approximate level of the fluid in the gravel-packed trench. The crust dissipates a large fraction of the head in the trench so that negative pressures are maintained on the soil side of the crust. In the absence of infiltration across the soil surface and the development of a saturated zone in proximity to it, boundary $\overline{GHI}$ acts essentially as an impermeable boundary, so a 0 gradient may be applied to it.

Pressure heads at nodes on h-specified boundaries $\overline{CD}$ and $\overline{EFG}$ are held constant; hence, these nodes must be eliminated from processing by equation [4]. Pressure heads at imaginary nodes are calculated by special equations, and those at nodes inside the notch (parts of rows 7 and 8) are not part of the solution mesh. All these nodes must also be eliminated from processing by equation [4]. They are eliminated by failing to include them in subsections.

The first subsection consists of rows 2, 3, 4, and 5. Processing of these rows starts on column 2 and ends on column N. For each row, the beginning boundary condition is $\partial h/\partial x = 0$ and the ending boundary condition is $\partial h/\partial x = 0$. Although row 2 and part of row 5 are themselves boundaries of the Neumann type, equation [4] is applied at each node on them just as it is at each node of rows 3 and 4 and the nonboundary part of row 5. This group of rows, then, forms a subsection for which the following parameters may be given as input to the model:

1. First row number
2. Last row number
3. Column on which row begins
4. Column on which row ends
5. Boundary condition at beginning of row
6. Boundary condition at end of row

Note that the first four items specify the first and last nodes in each mesh direction at which equation [4] is solved. Although row and column numbers have been mentioned for illustrative purposes, actual input data, as discussed in appendix A, are in terms of measured distances.

The beginning and ending boundary conditions on row 6 are the same as on the preceding rows, but processing begins on column 6 instead of column 2. So, row 6 must start a new subsection.

The beginning boundary condition on row 7 is a specified pressure head. This is different from the boundary condition on row 6 and also causes processing to begin on a different column. Either of these circumstances makes placing rows 6 and 7 in different subsections necessary. So, row 6 forms a subsection by itself. Row 8 has the same characteristics as row 7. The first few nodes of row 9 are part of a boundary, as in the case of row 5. But, in this case, the boundary portion is of the h-specified type and so equation [4] must not be applied to them. However, the portion of row 9 that is to be processed has the same characteristics as rows 7 and 8, so these three form the third subsection.

Rows 10 through M-1 have common characteristics and form a fourth subsection. Row M, being a pressure head-type boundary, must not be processed, so it is not part of any subsection.

Row subsectioning is illustrated in figure 3a. Column subsectioning proceeds using the same criteria as row subsectioning; an example case is illustrated in figure 3b.

As stated earlier, under the finite difference scheme used in this model, processing should be from left to right along rows and from the top to the bottom rows in succession. This is what happens, with no further user control, within a subsection. But the order in which sets of subsection parameters are given in the punchcard data deck specifies the order in which the several regions of the cross section are processed. Therefore, the user must be careful in arranging the order of these sets.

The significant concern here is that for any iteration no node should be processed before the node above it is processed. For example, if figure 2 is rotated 90° clockwise, so that the notch is vertically oriented, the column subsections of figure 3b will become row subsections. The long subsection IV would underlie subsections II and III so that both of these must be processed before subsection IV.

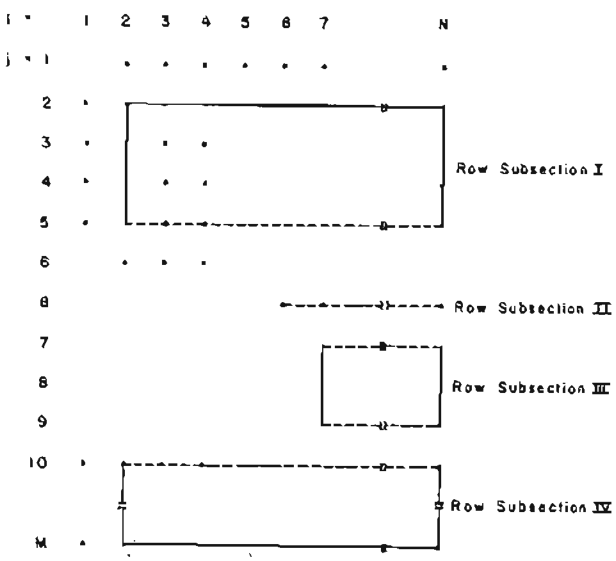


Figure 3a.--Row subsectioning.

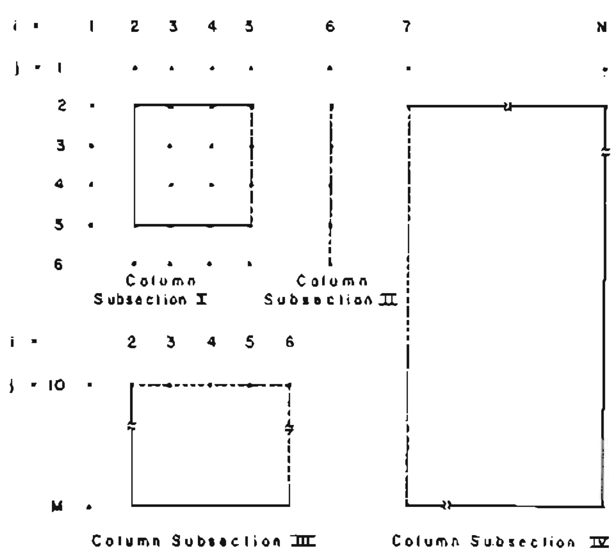


Figure 3b.--Column subsectioning.

Column subsection data are used only in setting boundary conditions, and the order of subsectioning is arbitrary.

The model is provided with the facility for setting hydrostatic boundary conditions along vertical boundaries. The boundary condition algorithms are flagged in the program listing in appendix A. If a user wishes to distribute pressure head in some other way, he may remove the hydrostatic algorithms and substitute others. As an alternative, he may define subsections whose boundaries coincide with changes in h, but this may become quite tedious if h varies continuously so that each row (or column) must form a separate subsection.

A boundary not parallel to one of the Cartesian coordinate axes involves a number of rows and columns of differing length. Each row and column then would form a separate subsection.

Soil units

A cross section may be composed of several different soil units. The boundaries of these units may have complex geometry and are given in punchcard input as straight-line segments. Unlike the cross section boundaries, soil unit boundaries need not intersect rows and columns at node points. However, the program will convert the straight-line input data so the unit boundaries are represented in stairstep fashion during processing. Each soil unit is considered homogeneous, and a single h-k table or equation must be included in the punchcard input for each.

Computer program

The program called STDY2 will be discussed in later sections as a source deck of punched cards. The program listing, a flow chart, and a glossary of variables are given in appendix A. Not shown are the job control cards which must precede and follow the program. These vary among computer facilities, and pertinent details may be obtained from consultants at the particular facility being used.

As noted earlier, USASI Fortran was used; but certain features of the program will require modification according to the computer facility being used. Again, facility consultants will be able to advise on the exact nature of the modifications needed. Program statements most likely to need modification are identified in the listing given in appendix A and discussed in a later section.

Model Control and Options

Case termination

A model run consists of execution of the program listed in appendix A together with input data and required job control cards. A given run may process, in sequence, a number of different cases or problems. A given case or problem is defined by a unique combination of geometry, boundary conditions and soil properties.

Telling in advance how much computer time will be required to reach convergence for any given case is not possible. Yet, on one of the job control cards, one must usually specify a period of time which, when elapsed will cause the run to be automatically terminated. Progress made on the case thus far would be lost if one underestimated the time required.

The input Fortran variable ESTIME is used to prevent loss of the PHED-array, provided the user wants it saved, when the elapsed computer time is close to exceeding the limit estimated for the case. KARPCH, discussed later, is used to effect saving the PHED-array in punched cards or on magnetic tape in the event that ESTIME is exceeded. These data may then be used to restart the case in another run. The time limit on the job card should exceed ESTIME by a small amount to allow for job compilation, for recording the PHED-array, and for the time interval between ESTIME checks.

The time interval between ESTIME checks depends upon the value of INTPRT, a variable equal to the number of iterations to be processed between each time check. Experience with the model at a particular computer facility will give a user a basis on which to estimate ESTIME and job card time.

In a multicase run, an ESTIME value must be given for each case, so the job card time must exceed the sum of the ESTIME values. If one case exceeds its ESTIME value, the run will continue with the next case after recording the PHED-array, if desired, of the case stopped.

The input variable ITMAX is the primary control variable for case termination. When starting a new case, it is given the value of the number of iterations to be processed for that case during the first run. When restarting a case, ITMAX should be equal to the number of iterations to be processed in the new run plus the iteration number corresponding to the restart PHED-array. One cannot predict in advance how many iterations will be needed for convergence, so ITMAX is a guess. The user may not want to set ITMAX to reach complete convergence, because he may wish to change the overrelaxation constant occasionally.

ITMAX is checked at the same frequency as ESTIME. When ITMAX is exceeded, processing of a case stops. The final PHED-array may be obtained in punched cards or on magnetic tape if desired. Again KARPCH effects this saving. ESTIME is a backup to ITMAX and only stops the case if the user has underestimated the amount of computer time necessary to process a number of iterations equal to ITMAX.

ITMAX also controls the segmentation of a case when a predetermined set of changes in the overrelaxation factor is desired. Its function in this regard will be discussed later.

Initial PHED-array

When considering a particular case for the first time, one usually has only a rough idea of how h is distributed over the cross section. Computational savings might be more than offset by the cost of keypunching an approximate initial PHED-array. The model has two alternative routines for initializing the PHED-array in the absence of read-in data. In one routine, PHED is given the same value at every node except those on h-specified boundaries. This value is given by the user as the input variable PHEDS. This routine is used for a case if the user gives the input variable INISIG the value 0. Any other value causes the alternative routine (below) to be used.

The other routine assigns PHED-values that are distributed smoothly in a direction parallel to the y-axis of the Cartesian coordinate system. It uses the input variable ELEV, which is defined in appendix A. This provides a starting estimate of the PHED-distribution that may have some advantages when the flow system is essentially one of drainage toward the water table.

The more closely the initial PHED-array approximates the converged (solution) array, the fewer the iterations needed for convergence. If, somehow, one has an initial PHED-array from which to start, it may be given as part of the input data via either punch cards or magnetic tape. This, in effect, occurs when restarting following a run that has not reached convergence.

When the PHED-array is to be initialized using data on cards or tape, the input variable KAREAD must be given the value 1. Any other value will cause one of the two other options discussed to be used.

When an input PHED-array is in punch cards, then the input variable IFILE must have the value 0. Data on a magnetic tape are arranged in "files," one PHED-array to a file. For each restarting case of a new run, IFILE must be assigned a number representing the position on the tape of the restart file to be read. Determination of this position is dependent upon whether preceding files on the same tape have been used by an earlier case of the same run. IFILE is defined in detail, and its use is illustrated in appendix A.

Pressure head—hydraulic conductivity relation

The program includes a table look-up routine as one means of determining hydraulic conductivity (K) as a function of pressure head (h). Linear interpolation is used between tabulated values. Alternatively, the user may insert into the program his own routine for solving an equation of the type

$$K = K(h)$$

The position for this insertion is noted in the program listing, appendix A. READ statements for parameter input may be inserted at the same place or among the other READ statements at the beginning of the program.

The program is set up for the insertion of only one equation. When a user wishes to use equations for several soil units, he must add the logic necessary to change equation parameters or the equation form from unit to unit.

Soil property data are given in a multiple card group of leader cards followed by one or more subgroups. Each subgroup contains the h-K data and coordinate data from which the geometry of the soil unit lower boundary may be specified. The first leader card specifies the number of soil units and contains a signal variable. The second leader card gives the number of h-K entries, NUMLIN(NS), in each table and the number of breakpoints in the lower boundary description, NUMBRK(NS), where NS = 1, 2, . . ., 5 is an index for identifying soil units. Appendix A gives a more detailed definition of these terms. When the h-K relation for a soil unit is given in equation form, then NUMLIN(NS) is given the value 999.

The signal variable KHPRNT on the first leader card is given a value other than 0 when the user wants to obtain a printout of the hydraulic conductivity assigned each node before setting boundary conditions at the beginning of a run. This feature may be used to check for correct positioning of soil units in the solution mesh. Note that, because the K-array is printed before boundary condition setting, K-values at imaginary nodes do not necessarily correspond to the h-values at those nodes.

Use of KHPRNT $\neq$ 0 during restart results in a meaningless K-array. It produces a useful array only when the processing of a case is being initialized or if MCHNGE ≠ 0.

If, during computation, h becomes smaller than the smallest h in the table, the case will be terminated after printing information helpful in locating the problem node. For some cases, however, the first few iterations produce overshoot with subsequent iterations converging smoothly toward a solution. Termination of such a case may be avoided by adding to the table an h-K pair for which h is smaller than the overshoot values. The value of NUMLIN(NS) must then be increased by 1. Of course, if the converged solution contains h-values outside the valid range of the table, it is considered a faulty solution.

Because h-K tables usually occupy a number of cards in the input and because more than one case may involve the same soil, the option of avoiding reading in a new table every time the processing of a different case begins is convenient. The input variable KTABLE, when given the value 1, causes a case to use the h-K table already in storage and used during the processing of the previous case. Any other value of KTABLE causes the case to read and store a new table. The first case of a run must, of course, have KTABLE = 0 or some value other than 1.

Overrelaxation factor

As mentioned earlier, the optimum (maximum) value of the overrelaxation factor (ω_{max}) can be determined for this nonlinear model only by trial and error. This could be accomplished by running a series of separate cases, each with a small number of iterations and each with a different value of the overrelaxation factor (ω).

Processing one case as a series of segments of a few iterations each (say 20 to 50) where each segment has a different ω-value is more economical than processing a series of individual runs. The PHED-array at the end of one segment serves as the initial PHED-array at the beginning of the next. Thus, when one has determined the value of ω_{max}, considerable convergence has been achieved.

Segmentation is accomplished by giving the input variable NOMEGA a value other than 0 and by adding segmentation cards to the input deck, as outlined in appendix A. NOMEGA must have the value 0 for a normal, unsegmented run.

The input variable ITMAX, the main function of which was discussed previously, is used to terminate the processing of each segment and to terminate the segmented case itself. Values for segments other than the first are given in the same series of segmentation cards as are subsequent OMEGA-values. ITMAX for each segment after the first must be equal to ITMAX for the preceding segment plus the number of iterations to be performed in the segment in question. When segments of 30 iterations each are processed, for example, then ITMAX = 30, 60, 90, . . . for segments 1, 2, 3, ...

To terminate a segmented case, an extra segmentation card must follow that for the last segment and must contain ITMAX = 0. The corresponding OMEGA may be blank or have any value. As in unsegmented cases, ESTIME is given only once and will terminate the segmented case (with the option of restart data in cards or tape) if its value is an underestimation of the time needed to process the total number of iterations wanted for the case.

As noted earlier, in this nonlinear model instability may develop when $\omega > \omega_{max}$. The h-arrays at the end of segments in which ω was too large will not be useful if the fluctuations covered too great a range. To preserve any progress made toward convergence, the model stores the PHED-array at the end of each segment on punched cards or on magnetic tape, provided KARPCH = 1, as discussed later. The most advanced PHED-array free of excessive fluctuation can then be used to restart the case in a later run.

Model output

Model output consists of printed material and data in punched cards or on magnetic tape. Examples of printed output will be given with the sample problem discussed later. The main objective of running STDY2 is to obtain distributions of pressure head (h) and hydraulic head (H). From these two distributions, one can deduce almost anything he needs regarding water content and hydraulic status of the modeled system.

The h- and H-arrays may be quite large, and the model is provided with options to control their printing. When the input variable IPSIG is given the value 1, the initial PHED-array will be printed. Any other value will suppress printing. When the input variable ILSIG is given the value 1, the final PHED-array will be printed. Any other value will suppress printing. When a case terminates because ESTIME is exceeded, the final PHED-array will not be printed. When ILSIG = 1, PHED-arrays will be printed after processing each segment of a segmented case.

Imaginary rows and columns are printed in the PHED- and HEAD-arrays. Nodes on imaginary columns do not reflect the hydraulic condition of the neighboring boundary columns. Nodes on imaginary rows have values that are dependent on Neumann-type boundary conditions along the neighboring boundary rows. For example, if a top boundary row is impermeable, HEAD-values at imaginary nodes above it will be equal to the HEAD-values at the nodes on the first row below the boundary. If a top or bottom boundary is partly Neumann and partly Dirichlet, then the imaginary nodes next to the Dirichlet boundaries will have meaningless values. One should keep in mind, then, that the real boundaries within a PHED- or HEAD-array may coincide with row 2 and column 2 and with the next to last row and the next to last column.

The final PHED-array at the end of a run may optionally be obtained in punchcard or magnetic tape form. For convenience, these data are called restart data in this report. These data may also be used as input for such other programs as convergence checking discussed later or machine plotting to produce isobars or to convert h to H for the purpose of plotting equipotential lines.

When the input variable KARPCH is given the value 1, restart data on punched cards or magnetic tape will be obtained. Any other value will suppress this form of output. When the input variable IFILE is given the value 0, the output will be in card form. Any other value will result in writing on magnetic tape provided the proper job control cards have been included so that tapes will be mounted. Computer facility personnel must be consulted for information on tape handling. The user will probably want magnetic tape for storage when the solution mesh is larger than 1,000 cards for the PHED-array.

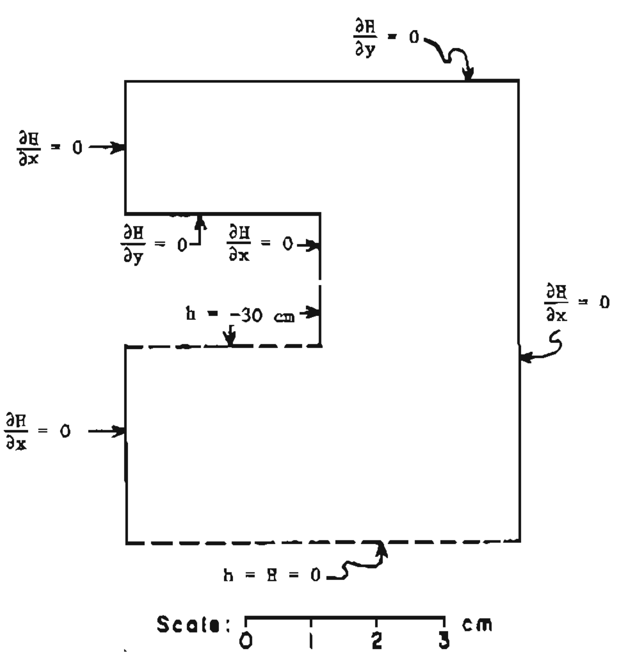


Figure 4.--Sample problem flow region with boundary conditions.

Sample problem

A small-scale, porous media cross section that has a geometry similar to that of figure 2 was modeled as an example. It, along with the boundary conditions, is shown in figure 4. The small scale was selected so that a user may, at small expense, verify the operation of the model on his computer.

Figure 5 shows the input data in a convenient assembly format. The input variables are defined and discussed in detail in appendix A. Before going further with the example, the reader should familiarize himself with that appendix.

The sample solution was accomplished in two runs. One run initialized the problem and was segmented to try various overrelaxation factors. The second run was an unsegmented restart of the first; its ω-value having been assigned on the basis of the first run's results.

The input data deck for the initialization run consisted of card groups 1-12 and card group 14. Card groups 1 and 3 each consisted of a single card punched with the data given on their value rows in figure 5.

For the restart run, card group 14 was replaced by card group 13. Card groups 1 and 3 were replaced by cards containing the data of the rows marked "Restart #1" in figure 5.

The printed portion of model output is illustrated in figures 6 and 7 which contain output for the sample problem. Printed output has three parts: (1) initialization data, (2) convergence monitoring data, and (3) pressure head and hydraulic head arrays. When the solution has converged acceptably, the latter arrays contain the data which portray the model's estimate of the prototype hydraulic regime.

Some data in the initialization part of the output are unmodified input data printed for the purpose of checking input and for recording a complete description of the conditions of the case. Other entries are derived from the input data. For example, card group 2 contains measured length and depth (in the sample, measurement was in centimeters because the units of K were cm/sec) of the cross section of figure 4. This group also contains variables which specify whether an imaginary row or column is needed at each extremity of the cross section. Card groups 9 and 10 contain specifications for Δx and Δy in various parts of the cross section. Using all these data, the model determines the total number of rows and columns needed in the solution mesh. These are printed and identified as MROW and MCOL, respectively.

Other noninput initialization data given in the output are of the same type as MROW and MCOL, that is, row and column equivalents

Figure 5.--Input data for STDY2 sample problem.

DATA SHEET STDY2 **Date** 3/74 **Page** 1 **of** 7

Case Small-Scale Septic Tank **Name**

Card Grp 1										
Variable	ESTIME	KARPAD	KARPCH	ITER	IFILR	IPSIG	ILSIG	KTABLE	MCHNGE	
Format	F5.0	I5	←					→	I5	
Value	15	0	1	0	0	1	0	0	0	
Restart #1	15	1	1	150	0	0	1	0	0	
Card Grp 2										
Variable	LGTH	DEPTH	SLOPE	IMGTOP	IMGBOT	IMGLSD	IMGRSD	INISIG	PHEDS	ELEV
Format	F10.2	F10.2	F5.2	I5	←		→	I5	F10.2	F10.2
Value	6.00	7.00	0.00	1	0	1	1	1	0.00	7.00
Card Grp 3										
Variable	ITMAX	INTPRT	OMEGA	NOMEGA	NNODES	IDBLE	NCARDY	NCARDX	JGEOM	IGEOM
Format	I5	I5	F10.2	I5	←				→	I5
Value	30	1	1.00	1	5	0	3	3	4	4
Restart #1	175	1	1.60	0	5	1	3	3	4	4
Card Grp 4										
Variable	COORDI	COORDJ	COORDI	COORDJ	COORDI	COORDJ	COORDI	COORDJ		
Format	F10.2	←					→	F10.2		
Value Cd 1	0.00	0.00	6.00	0.00	4.50	3.00	0.00	6.00		
Cd 2	6.00	6.00								

DATA SHEET STDY2 **Date** 3/74 **Page** 2 **of** 7

Case Small-Scale Septic Tank **Name**

Card Grp 5										
Variable	COMENT									
Format	(20A4)									
Value Cd 1	STDY2 2D SOR STEADY POROUS MEDIA FLOW MODEL									
Cd 2	SMALL-SCALE SEPTIC TANK DISPOSAL, 6 x 7 CM, FINER MESH NEAR NOTCH									
Cd 3	HOMOGENEOUS SOIL--IIC (TILL) HORIZON, SAYBROOK, S. L.									
Cd 4	SAMPLE CASE--INITIALIZATION RUN									
Cd 5	Blank									
Card Grp 6										
Variable	LUNITS	KHPRT								
Format	I5	I5								
Value	1	0								
Card Grp 7										
Variable	NUMLIN	NUMBRK	NUMLIN	NUMBRK	NUMLIN	NUMBRK	NUMLIN	NUMBRK	NUMLIN	NUMBRK
Format	I5	←							→	I5
Value	37	2								

DATA SHEET STDY2 Date 3/74 Page 3 of 7

Case Small-Scale Septic Tank Name

Card Grp 8a										
Variable	XBRK	YBRK	XBRK	YBRK	XBRK	YBRK	XBRK	YBRK		
Format	F10.2	←					→	F10.2		
Value	0.00	7.00	6.00	7.00						

Card Grp 8b										
Variable	PTAB	PTAB	PTAB	PTAB	PTAB	PTAB	PTAB	PTAB		
Format	E10.3	←					→	E10.3		
Value Cd 1	.000+00	-.300+01	-.700+01	-.800+01	-.900+01	-.100+02	-.120+02	-.130+02		
Cd 2	-.150+02	-.160+02	-.170+02	-.180+02	-.190+02	-.210+02	-.230+02	-.240+02		
Cd 3	-.250+02	-.260+02	-.280+02	-.290+02	-.300+02	-.310+02	-.340+02	-.380+02		
Cd 4	-.400+02	-.430+02	-.450+02	-.500+02	-.650+02	-.680+02	-.770+02	-.100+03		
Cd 5	-.300+03	-.500+03	-.790+03	-.100+04	-.100+11					

DATA SHEET STDY2 Date 3/74 Page 4 of 7

Case Small-Scale Septic Tank Name

Card Grp 8c										
Variable	KTAB	KTAB	KTAB	KTAB	KTAB	KTAB	KTAB	KTAB		
Format	E10.3	←					→	E10.3		
Value Cd 1	.926-03	.532-03	.231-03	.197-03	.185-03	.174-03	.162-03	.150-03		
Cd 2	.127-03	.118-03	.116-03	.113-03	.110-03	.104-03	.984-04	.926-04		
Cd 3	.903-04	.856-04	.833-04	.810-04	.787-04	.752-04	.671-04	.579-04		
Cd 4	.544-04	.498-04	.463-04	.394-04	.185-04	.162-04	.104-04	.116-05		
Cd 5	.463-07	.231-08	.926-09	.347-10	.347-10					

Figure 5.--Continued.

DATA SHEET STDY2 Date 3/74 Page 5 of 7

Case Small-Scale Septic Tank Name

Card Grp 9										
Variable	DYLGTH	DELY	DYLGTH	DELY	DYLGTH	DELY	DYLGTH	DELY		
Format	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3		
Value	0.00	1.000	1.00	0.500	5.00	1.000				

Card Grp 10										
Variable	DXLGTH	DELX	DXLGTH	DELX	DXLGTH	DELX	DXLGTH	DELX		
Format	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3		
Value	0.00	1.000	2.00	0.500	5.00	1.000				

Card Grp 11										
Variable	STARTY	STOPY	BEGX	ENDX	JBETA	JETA	BCLJ	BCRJ		
Format	F10.2	←	→	F10.2	I5	I5	F10.2	F10.2		
Value Cd 1	0.00	2.00	0.00	6.00	1	1	0.00	0.00		
Cd 2	2.10	2.90	3.00	6.00	1	1	0.00	0.00		
Cd 3	3.00	4.00	3.10	6.00	0	1	-30.00	0.00		
Cd 4	4.10	6.90	0.00	6.00	1	1	0.00	0.00		

DATA SHEET STDY2 Date 3/74 Page 6 of 7

Case Small-Scale Septic Tank Name

Card Grp 12										
Variable	STARTX	STOPX	BEGY	ENDY	IBETA	IETA	BCUI	BCBI	FLUX	
Format	F10.2	←	→	F10.2	I5	I5	F10.2	F10.2	E10.2	
Value Cd 1	0.00	2.90	0.00	2.00	1	1	0.00	0.00	0.00+00	
Cd 2	3.00	3.00	0.00	2.90	1	0	0.00	-30.00	0.00+00	
Cd 3	0.00	3.00	4.05	6.90	0	0	-30.00	0.00	0.00+00	
Cd 4	3.10	6.00	0.00	6.90	1	0	0.00	0.00	0.00+00	

Card Grp 13						
Variable	PHED	PHED	PHED	PHED	PHED	PHED
Format	D13.6	←			→	D13.6
Value	This is usually a restart deck already punched or written on magnetic tape.					
	on initialization run: none					
	on restart run: use restart deck produced by initialization run					

Figure 5.--Continued.

DATA SHEET STDY2	Date 3/74	Page 7 of 7
Case Small-Scale Septic Tank	Name	

Card Grp 14	-									
Variable	OMEGA	ITMAX								
Format	F5.2	I5								
Value Cd 1	1.20	60	on initialization run							
Cd 2	1.40	90								
Cd 3	1.60	120								
Cd 4	1.80	150								
Cd 5	0.00	0								
	on restart run: none									

Figure 5.--Continued.

of measured data. These row and column equivalents are necessary because the pressure and hydraulic head arrays contain the values of those variables in row and column order but do not reflect mesh spacing and thus do not reflect x,y coordinate position. The row and column equivalents, then, aid in fixing the latter position.

In general, each derived datum in the initialization part of the output corresponds directly to an input entry in figure 5. The single exception is in the output labeled VARIABLE MESH INCREMENT DATA FOLLOW IN TRIPLETS AS XYZ. Where three pairs of values were punched in each of the card groups 9 and 10, four triplets resulted for each. In each case, the final triplet specifies that the last Δy- or Δx-value continues all the way to the lower or right-hand boundary, that is, until J = 13 or I = 12 (the values of MROW and MCOL, respectively).

The flow region was subsectioned according to an earlier part of this publication, Subdivision of Flow Cross Section. For a discussion of the measurement of some of the quantities in card groups 11 and 12, see appendix A, card group 11.

The solution mesh with row and column subsections outlined is shown in figures 8 and 9.

Card groups 6-8 describe the soil units in the prototype. In this case, only one soil unit was used, so its bottom boundary coincided with the bottom boundary of the cross section. Nevertheless, the coordinates for two breakpoints—the two bottom corners of the cross section—were necessary.

In some previous runs, fluctuations in pressure head with the first few iterations had produced values smaller than -10^6 cm of water, so the h-K table was extended by the addition of a much lower pressure head (higher suction).

In figure 5, input card group 4 identifies five nodes for printing pressure head values in convergence checking, a procedure explained later. INTPRT in card group 3 specifies that the PHED-values at the corresponding nodes are printed every iteration.

NOMEGA = 1 on the value line in card group 3 specifies that the first run is segmented. ITMAX and OMEGA, on the same value line, show that the first segment is 30 iterations long and that the first ω-value is 1.00. Card group 14 specifies four more segments, each consisting of 30 iterations and each with a progressively higher ω-value.

The convergence-checking data for node (2,2) in figure 6 were plotted as the upper curve of figure 10. Tic marks above the

Figure 6.--STDY2 printout for first run of sample problem.

```
STDY2  2D SOR STEADY POROUS MEDIA FLOW MODEL
   SMALL SCALE SEPTIC TANK DISPOSAL, 6 X 7 CM, FINER MESH NEAR NOTCH
   HOMOGENEOUS SOIL --IIC (TILL) HORIZON, SAYBROOK S.L.
   SAMPLE CASE -- INITIALIZATION RUN

ESTIME    KAREAD    KARPCH    ITER    IFILE    IPSIG    ILSIG    KTABLE    MCHNGE
 15.        0         1         0       0        1        0        0          0

 LGTH     DEPTH     SLOPE    IMGTOP   IMGBOT   IMGLSD   IMGRSD   INISIG    PHEDS    ELEV
 6.00      7.00     0.0        1        0        1        1        1        0.0     7.00

ITMAX    INTPRT    OMEGA    NOMEGA   NNODES    ICBLE    NCARDY   NCARDX    JGEOM    IGEOM
 30        1       1.00       1        5         0        3        3         4        4

GEOMETRY AND BOUNDARY CONDITION DATA

MCOL     MROW
 12       13

FOR ROW SUBSECTIONS
NSUBY     STARTY     STOPY     BEGX     ENDX     JBETA     JETA     BCLJ      BCRJ
  1        0.0        2.00     0.0      6.00       1         1       0.0       0.0
  2        2.10       2.90     3.00     6.00       1         1       0.0       0.0
  3        3.00       4.00     3.10     6.00       0         1     -30.00      0.0
  4        4.10       6.90     0.0      6.00       1         1       0.0       0.0
NSUBY   JSTART   JSTOP   IBEG   IEND
  1        2       5       2     11
  2        6       6       6     11
  3        7       9       7     11
  4       10      12       2     11

FOR COLUMN SUBSECTIONS
NSUBX     STARTX     STOPX     BEGY     ENDY     IBETA     IETA     BCUI      BCBI    FLUX
  1        0.0        2.90     0.0      2.00       1         1       0.0       0.0    0.0
  2        3.00       3.00     0.0      2.90       1         0       0.0     -30.00   0.0
  3        0.0        3.00     4.05     6.90       0         0     -30.00      0.0    0.0
  4        3.10       6.00     0.0      6.90       1         0       0.0       0.0    0.0
NSUBX   ISTART   ISTOP   JBEG   JEND
  1        2       5       2      5
  2        6       6       2      6
  3        2       6      10     12
  4        7      11       2     12

VARIABLE MESH INCREMENT DATA FOLLOW IN TRIPLETS AS XYZ
     WHERE X = MEASURED DISTANCE FROM AXIS (DXLGTH OR DYLGTH)
           Y = ROW OR COLUMN NUMBER (JY OR IX)
           Z = INCREMENT LENGTH (DELY OR DELX)
VERTICAL
    0.0    2   1.000          1.00   3   0.500          5.00   11   1.000          7.00   13   1.000
HORIZONTAL
    0.0    2   1.000          2.00   4   0.500          5.00   10   1.000          6.00   12   1.000

                          HYDRAULIC CONDUCTIVITY TABLE FOR SOIL UNIT    1
BOTTOM OF UNIT  LIES ALONG STRAIGHT LINES CONNECTING THE FOLLOWING COORDINATES
AS (X,Y) MEASURED FROM AXES
    0.0       7.00       6.00      7.00
AS COLUMN AND ROW NUMBER (I,J)
   2   13          12   13

          P             K              P             K              P             K              P             K
      0.0          0.926E-03     -0.170E+02    0.116E-03     -0.300E+02    0.787E-04     -0.770E+02    0.104E-04
     -0.300E+01    0.532E-03     -0.180E+02    0.113E-03     -0.310E+02    0.752E-04     -0.100E+03    0.116E-05
     -0.700E+01    0.231E-03     -0.190E+02    0.110E-03     -0.340E+02    0.671E-04     -0.300E+03    0.463E-07
     -0.800E+01    0.197E-03     -0.210E+02    0.104E-03     -0.380E+02    0.579E-04     -0.500E+03    0.231E-08
     -0.900E+01    0.185E-03     -0.230E+02    0.984E-04     -0.400E+02    0.544E-04     -0.790E+03    0.926E-09
     -0.100E+02    0.174E-03     -0.240E+02    0.926E-04     -0.430E+02    0.498E-04     -0.100E+04    0.347E-10
     -0.120E+02    0.162E-03     -0.250E+02    0.903E-04     -0.450E+02    0.463E-04     -0.100E+11    0.347E-10
     -0.130E+02    0.150E-03     -0.260E+02    0.856E-04     -0.500E+02    0.394E-04
     -0.150E+02    0.127E-03     -0.280E+02    0.833E-04     -0.650E+02    0.185E-04
     -0.160E+02    0.118E-03     -0.290E+02    0.810E-04     -0.680E+02    0.162E-04

                                     STARTING DISTRIBUTION OF PRESSURE HEAD
  1  -0.800000D+01-0.800000D+01-0.800000D+01-0.800000D+01-0.800000D+01-0.800000D+01-0.800000D+01-0.800000D+01-0.800000D+01-0.800000D+0
     -0.800000D+01-0.800000D+01
  2  -0.700000D+01-0.700000D+01-0.700000D+01-0.700000D+01-0.700000D+01-0.700000D+01-0.700000D+01-0.700000D+01-0.700000D+01-0.700000D+0
     -0.700000D+01-0.700000D+01
  3  -0.600000D+01-0.600000D+01-0.600000D+01-0.600000D+01-0.600000D+01-0.600000D+01-0.600000D+01-0.600000D+01-0.600000D+01-0.600000D+0
     -0.600000D+01-0.600000D+01
  4  -0.550000D+01-0.550000D+01-0.550000D+01-0.550000D+01-0.650000D+01-0.550000D+01-0.550000D+01-0.550000D+01-0.550000D+01-0.550000D+0
     -0.550000D+01-0.550000D+01
  5  -0.500000D+01-0.500000D+01-0.500000D+01-0.500000D+01-0.500000D+01-0.500000D+01-0.500000D+01-0.500000D+01-0.500000D+01-0.500000D+0
     -0.500000D+01-0.500000D+01
  6  -0.450000D+01-0.450000D+01-0.450000D+01-0.450000D+01-0.450000D+01-0.450000D+01-0.450000D+01-0.450000D+01-0.450000D+01-0.450000D+0
     -0.450000D+01-0.450000D+01
  7  -0.400000D+01-0.400000D+01-0.400000D+01-0.400000D+01-0.400000D+01-0.300000D+02-0.400000D+01-0.400000D+01-0.400000D+01-0.400000D+0
     -0.400000D+01-0.400000D+01
  8  -0.350000D+01-0.350000D+01-0.350000D+01-0.350000D+01-0.350000D+01-0.300000D+02-0.350000D+01-0.350000D+01-0.350000D+01-0.350000D+0
     -0.350000D+01-0.350000D+01
  9  -0.300000D+01-0.300000D+02-0.300000D+02-0.300000D+02-0.300000D+02-0.300000D+02-0.300000D+01-0.300000D+01-0.300000D+01-0.300000D+0
     -0.300000D+01-0.300000D+01
 10  -0.250000D+01-0.250000D+01-0.250000D+01-0.250000D+01-0.250000D+01-0.250000D+01-0.250000D+01-0.250000D+01-0.250000D+01-0.250000D+0
     -0.250000D+01-0.250000D+01
 11  -0.200000D+01-0.200000D+01-0.200000D+01-0.200000D+01-0.200000D+01-0.200000D+01-0.200000D+01-0.200000D+01-0.200000D+01-0.200000D+0
     -0.200000D+01-0.200000D+01
 12  -0.100000D+01-0.100000D+01-0.100000D+01-0.100000D+01-0.100000D+01-0.100000D+01-0.100000D+01-0.100000D+01-0.100000D+01-0.100000D+0
     -0.100000D+01-0.100000D+01
 13   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
      0.0          0.0
```

```
ITERATION NO. AND PRESSURE HEAD AT SELECTED NODES AS IDENTIFIED BELOW
COORDINATES AS MEASURED FROM AXES (X,Y)
    0.0 ,   0.0         6.00,   0.0         4.50,   3.00        0.0 ,   6.00        6.00,   6.00
COORDINATES AS ROW AND COLUMN NUMBER (I,J)
ITER    2,   2      11,   2      9,   7      2,  12      11,  12

   0 -0.70000D+01 -0.70000D+01 -0.40000D+01 -0.10000D+01 -0.10000D+01
   1 -0.70000D+01 -0.70000D+01 -0.43164D+01 -0.16979D+01 -0.10486D+01
   2 -0.70000D+01 -0.70000D+01 -0.46873D+01 -0.25288D+01 -0.11164D+01
   3 -0.70000D+01 -0.70000D+01 -0.50636D+01 -0.29155D+01 -0.11924D+01
   4 -0.70000D+01 -0.70000D+01 -0.53889D+01 -0.31399D+01 -0.12698D+01
   5 -0.70000D+01 -0.70062D+01 -0.56814D+01 -0.33065D+01 -0.13445D+01
   6 -0.70000D+01 -0.70326D+01 -0.59587D+01 -0.34284D+01 -0.14144D+01
   7 -0.70000D+01 -0.70762D+01 -0.62256D+01 -0.35143D+01 -0.14788D+01
   8 -0.70000D+01 -0.71389D+01 -0.64843D+01 -0.35760D+01 -0.15380D+01
   9 -0.70052D+01 -0.72186D+01 -0.67349D+01 -0.36211D+01 -0.15921D+01
  10 -0.70165D+01 -0.73126D+01 -0.69802D+01 -0.36548D+01 -0.16419D+01
  11 -0.70342D+01 -0.74183D+01 -0.72213D+01 -0.36798D+01 -0.16876D+01
  12 -0.70583D+01 -0.75337D+01 -0.74593D+01 -0.36985D+01 -0.17293D+01
  13 -0.70887D+01 -0.76568D+01 -0.76931D+01 -0.37129D+01 -0.17676D+01
  14 -0.71248D+01 -0.77864D+01 -0.79245D+01 -0.37242D+01 -0.18028D+01
  15 -0.71663D+01 -0.79217D+01 -0.81538D+01 -0.37333D+01 -0.18353D+01
  16 -0.72126D+01 -0.80623D+01 -0.83818D+01 -0.37406D+01 -0.18653D+01
  17 -0.72634D+01 -0.82075D+01 -0.86031D+01 -0.37468D+01 -0.18932D+01
  18 -0.73181D+01 -0.83560D+01 -0.88113D+01 -0.37519D+01 -0.19191D+01
  19 -0.73767D+01 -0.85073D+01 -0.90133D+01 -0.37563D+01 -0.19432D+01
  20 -0.74389D+01 -0.86610D+01 -0.92086D+01 -0.37600D+01 -0.19656D+01
  21 -0.75045D+01 -0.88168D+01 -0.93975D+01 -0.37633D+01 -0.19864D+01
  22 -0.75733D+01 -0.89746D+01 -0.96794D+01 -0.37662D+01 -0.20059D+01
  23 -0.76453D+01 -0.91344D+01 -0.97494D+01 -0.37688D+01 -0.20241D+01
  24 -0.77206D+01 -0.92954D+01 -0.99117D+01 -0.37711D+01 -0.20411D+01
  25 -0.77991D+01 -0.94571D+01 -0.10068D+02 -0.37732D+01 -0.20570D+01
  26 -0.78808D+01 -0.96190D+01 -0.10218D+02 -0.37751D+01 -0.20720D+01
  27 -0.79656D+01 -0.97806D+01 -0.10362D+02 -0.37768D+01 -0.20861D+01
  28 -0.80537D+01 -0.99415D+01 -0.10500D+02 -0.37784D+01 -0.20993D+01
  29 -0.81447D+01 -0.10101D+02 -0.10632D+02 -0.37798D+01 -0.21117D+01
  30 -0.82387D+01 -0.10260D+02 -0.10760D+02 -0.37812D+01 -0.21234D+01

RESTART PUNCHED

TOTAL CASE TIME =  0.976000 SECONDS.

CONTINUE WITH OMEGA =  1.20, ITMAX =    60
  31 -0.83546D+01 -0.10485D+02 -0.10944D+02 -0.37828D+01 -0.21513D+01
  32 -0.84809D+01 -0.10711D+02 -0.11124D+02 -0.37843D+01 -0.21679D+01
  33 -0.86168D+01 -0.10936D+02 -0.11296D+02 -0.37857D+01 -0.21829D+01
  34 -0.87639D+01 -0.11159D+02 -0.11461D+02 -0.37869D+01 -0.21962D+01
  35 -0.89221D+01 -0.11377D+02 -0.11619D+02 -0.37881D+01 -0.22084D+01
  36 -0.90857D+01 -0.11591D+02 -0.11770D+02 -0.37891D+01 -0.22195D+01
  37 -0.92545D+01 -0.11801D+02 -0.11914D+02 -0.37902D+01 -0.22296D+01
  38 -0.94277D+01 -0.12006D+02 -0.12051D+02 -0.37911D+01 -0.22389D+01
  39 -0.96048D+01 -0.12206D+02 -0.12184D+02 -0.37921D+01 -0.22475D+01
  40 -0.97854D+01 -0.12401D+02 -0.12311D+02 -0.37929D+01 -0.22555D+01
  41 -0.99687D+01 -0.12592D+02 -0.12434D+02 -0.37938D+01 -0.22630D+01
  42 -0.10154D+02 -0.12778D+02 -0.12551D+02 -0.37945D+01 -0.22700D+01
  43 -0.10342D+02 -0.12960D+02 -0.12664D+02 -0.37952D+01 -0.22767D+01
  44 -0.10530D+02 -0.13137D+02 -0.12774D+02 -0.37958D+01 -0.22832D+01
  45 -0.10719D+02 -0.13309D+02 -0.12881D+02 -0.37964D+01 -0.22895D+01
  46 -0.10908D+02 -0.13477D+02 -0.12984D+02 -0.37969D+01 -0.22956D+01
  47 -0.11098D+02 -0.13642D+02 -0.13084D+02 -0.37974D+01 -0.23013D+01
  48 -0.11288D+02 -0.13803D+02 -0.13181D+02 -0.37979D+01 -0.23069D+01
  49 -0.11477D+02 -0.13960D+02 -0.13275D+02 -0.37983D+01 -0.23122D+01
  50 -0.11665D+02 -0.14114D+02 -0.13366D+02 -0.37988D+01 -0.23173D+01
  51 -0.11852D+02 -0.14265D+02 -0.13455D+02 -0.37992D+01 -0.23223D+01
  52 -0.12038D+02 -0.14413D+02 -0.13542D+02 -0.37995D+01 -0.23270D+01
  53 -0.12222D+02 -0.14558D+02 -0.13628D+02 -0.37999D+01 -0.23316D+01
  54 -0.12404D+02 -0.14700D+02 -0.13712D+02 -0.38002D+01 -0.23359D+01
  55 -0.12585D+02 -0.14840D+02 -0.13795D+02 -0.38005D+01 -0.23402D+01
  56 -0.12764D+02 -0.14976D+02 -0.13875D+02 -0.38008D+01 -0.23443D+01
  57 -0.12941D+02 -0.15110D+02 -0.13953D+02 -0.38011D+01 -0.23483D+01
  58 -0.13116D+02 -0.15242D+02 -0.14030D+02 -0.38014D+01 -0.23521D+01
  59 -0.13289D+02 -0.15371D+02 -0.14105D+02 -0.38017D+01 -0.23558D+01
  60 -0.13461D+02 -0.15498D+02 -0.14178D+02 -0.38020D+01 -0.23594D+01

RESTART PUNCHED

TOTAL CASE TIME =  1.801999 SECONDS.

CONTINUE WITH OMEGA =  1.40, ITMAX =    90
  61 -0.13659D+02 -0.15681D+02 -0.14294D+02 -0.38023D+01 -0.23729D+01
  62 -0.13868D+02 -0.15866D+02 -0.14409D+02 -0.38027D+01 -0.23781D+01
  63 -0.14085D+02 -0.16053D+02 -0.14521D+02 -0.38030D+01 -0.23839D+01
  64 -0.14316D+02 -0.16238D+02 -0.14628D+02 -0.38033D+01 -0.23891D+01
  65 -0.14559D+02 -0.16418D+02 -0.14730D+02 -0.38035D+01 -0.23940D+01
  66 -0.14795D+02 -0.16592D+02 -0.14828D+02 -0.38039D+01 -0.23986D+01
  67 -0.15030D+02 -0.16761D+02 -0.14922D+02 -0.38042D+01 -0.24029D+01
  68 -0.15263D+02 -0.16926D+02 -0.15012D+02 -0.38045D+01 -0.24069D+01
  69 -0.15493D+02 -0.17086D+02 -0.15097D+02 -0.38049D+01 -0.24106D+01
  70 -0.15720D+02 -0.17241D+02 -0.15178D+02 -0.38053D+01 -0.24143D+01
  71 -0.15946D+02 -0.17391D+02 -0.15256D+02 -0.38056D+01 -0.24177D+01
  72 -0.16169D+02 -0.17537D+02 -0.15330D+02 -0.38058D+01 -0.24210D+01
  73 -0.16390D+02 -0.17677D+02 -0.15401D+02 -0.38060D+01 -0.24241D+01
  74 -0.16607D+02 -0.17813D+02 -0.15469D+02 -0.38063D+01 -0.24271D+01
  75 -0.16822D+02 -0.17944D+02 -0.15534D+02 -0.38065D+01 -0.24299D+01
  76 -0.17033D+02 -0.18070D+02 -0.15597D+02 -0.38067D+01 -0.24326D+01
  77 -0.17241D+02 -0.18192D+02 -0.15658D+02 -0.38068D+01 -0.24353D+01
  78 -0.17444D+02 -0.18309D+02 -0.15716D+02 -0.38070D+01 -0.24379D+01
  79 -0.17644D+02 -0.18423D+02 -0.15772D+02 -0.38072D+01 -0.24403D+01
  80 -0.17838D+02 -0.18532D+02 -0.15827D+02 -0.38074D+01 -0.24427D+01
  81 -0.18027D+02 -0.18638D+02 -0.15879D+02 -0.38075D+01 -0.24450D+01
  82 -0.18210D+02 -0.18740D+02 -0.15929D+02 -0.38077D+01 -0.24472D+01
  83 -0.18388D+02 -0.18839D+02 -0.15977D+02 -0.38078D+01 -0.24492D+01
  84 -0.18560D+02 -0.18934D+02 -0.16024D+02 -0.38079D+01 -0.24513D+01
  85 -0.18727D+02 -0.19026D+02 -0.16068D+02 -0.38081D+01 -0.24532D+01
  86 -0.18889D+02 -0.19115D+02 -0.16111D+02 -0.38082D+01 -0.24551D+01
  87 -0.19046D+02 -0.19200D+02 -0.16153D+02 -0.38083D+01 -0.24568D+01
  88 -0.19197D+02 -0.19283D+02 -0.16193D+02 -0.38084D+01 -0.24586D+01
  89 -0.19344D+02 -0.19362D+02 -0.16232D+02 -0.38086D+01 -0.24602D+01
  90 -0.19466D+02 -0.19439D+02 -0.16269D+02 -0.38087D+01 -0.24618D+01
```

Figure 6.--Continued.

Figure 6.—Continued.

```
RESTART PUNCHED

TOTAL CASE TIME =  2.594000 SECONDS.

CONTINUE WITH OMEGA =  1.60, ITMAX =    120
  91 -0.15843D+02 -0.19553D+02 -0.16339D+02 -0.38088D+01 -0.24724D+01
  92 -0.19816D+02 -0.19670D+02 -0.16404D+02 -0.38089D+01 -0.24734D+01
  93 -0.19974D+02 -0.19789D+02 -0.16468D+02 -0.38090D+01 -0.24758D+01
  94 -0.20154D+02 -0.19909D+02 -0.16528D+02 -0.38092D+01 -0.24777D+01
  95 -0.20347D+02 -0.20016D+02 -0.16583D+02 -0.38093D+01 -0.24793D+01
  96 -0.20518D+02 -0.20117D+02 -0.16633D+02 -0.38095D+01 -0.24807D+01
  97 -0.20686D+02 -0.20210D+02 -0.16680D+02 -0.38097D+01 -0.24821D+01
  98 -0.20849D+02 -0.20298D+02 -0.16722D+02 -0.38098D+01 -0.24833D+01
  99 -0.21001D+02 -0.20379D+02 -0.16759D+02 -0.38102D+01 -0.24844D+01
 100 -0.21146D+02 -0.20455D+02 -0.16792D+02 -0.38104D+01 -0.24856D+01
 101 -0.21283D+02 -0.20525D+02 -0.16822D+02 -0.38104D+01 -0.24867D+01
 102 -0.21412D+02 -0.20591D+02 -0.16851D+02 -0.38103D+01 -0.24878D+01
 103 -0.21536D+02 -0.20652D+02 -0.16877D+02 -0.38102D+01 -0.24887D+01
 104 -0.21653D+02 -0.20708D+02 -0.16902D+02 -0.38105D+01 -0.24896D+01
 105 -0.21764D+02 -0.20759D+02 -0.16925D+02 -0.38106D+01 -0.24904D+01
 106 -0.21869D+02 -0.20807D+02 -0.16947D+02 -0.38106D+01 -0.24911D+01
 107 -0.21968D+02 -0.20852D+02 -0.16967D+02 -0.38106D+01 -0.24918D+01
 108 -0.22061D+02 -0.20894D+02 -0.16986D+02 -0.38107D+01 -0.24924D+01
 109 -0.22149D+02 -0.20933D+02 -0.17004D+02 -0.38107D+01 -0.24930D+01
 110 -0.22232D+02 -0.20970D+02 -0.17020D+02 -0.38108D+01 -0.24936D+01
 111 -0.22309D+02 -0.21004D+02 -0.17036D+02 -0.38108D+01 -0.24941D+01
 112 -0.22381D+02 -0.21036D+02 -0.17050D+02 -0.38109D+01 -0.24946D+01
 113 -0.22449D+02 -0.21066D+02 -0.17063D+02 -0.38109D+01 -0.24951D+01
 114 -0.22512D+02 -0.21094D+02 -0.17076D+02 -0.38109D+01 -0.24955D+01
 115 -0.22571D+02 -0.21120D+02 -0.17087D+02 -0.38110D+01 -0.24959D+01
 116 -0.22626D+02 -0.21144D+02 -0.17098D+02 -0.38110D+01 -0.24963D+01
 117 -0.22678D+02 -0.21167D+02 -0.17108D+02 -0.38110D+01 -0.24966D+01
 118 -0.22726D+02 -0.21188D+02 -0.17118D+02 -0.38110D+01 -0.24970D+01
 119 -0.22771D+02 -0.21208D+02 -0.17126D+02 -0.38111D+01 -0.24973D+01
 120 -0.22813D+02 -0.21226D+02 -0.17135D+02 -0.38111D+01 -0.24975D+01

RESTART PUNCHED

TOTAL CASE TIME =  3.389999 SECONDS.

CONTINUE WITH OMEGA =  1.60, ITMAX =    150
 121 -0.22857D+02 -0.21255D+02 -0.17154D+02 -0.38111D+01 -0.25025D+01
 122 -0.22902D+02 -0.21265D+02 -0.17171D+02 -0.38111D+01 -0.25014D+01
 123 -0.22947D+02 -0.21317D+02 -0.17188D+02 -0.38112D+01 -0.25019D+01
 124 -0.22996D+02 -0.21354D+02 -0.17204D+02 -0.38112D+01 -0.25018D+01
 125 -0.23052D+02 -0.21380D+02 -0.17218D+02 -0.38112D+01 -0.25015D+01
 126 -0.23096D+02 -0.21403D+02 -0.17229D+02 -0.38113D+01 -0.25011D+01
 127 -0.23141D+02 -0.21421D+02 -0.17237D+02 -0.38114D+01 -0.25008D+01
 128 -0.23182D+02 -0.21438D+02 -0.17243D+02 -0.38114D+01 -0.25006D+01
 129 -0.23216D+02 -0.21451D+02 -0.17245D+02 -0.38117D+01 -0.25007D+01
 130 -0.23247D+02 -0.21462D+02 -0.17245D+02 -0.38118D+01 -0.25013D+01
 131 -0.23275D+02 -0.21471D+02 -0.17244D+02 -0.38113D+01 -0.25017D+01
 132 -0.23298D+02 -0.21477D+02 -0.17245D+02 -0.38109D+01 -0.25019D+01
 133 -0.23320D+02 -0.21481D+02 -0.17246D+02 -0.38107D+01 -0.25017D+01
 134 -0.23340D+02 -0.21482D+02 -0.17247D+02 -0.38121D+01 -0.25017D+01
 135 -0.23357D+02 -0.21481D+02 -0.17249D+02 -0.38117D+01 -0.25016D+01
 136 -0.23372D+02 -0.21481D+02 -0.17251D+02 -0.38113D+01 -0.25014D+01
 137 -0.23383D+02 -0.21483D+02 -0.17252D+02 -0.38111D+01 -0.25013D+01
 138 -0.23393D+02 -0.21485D+02 -0.17253D+02 -0.38113D+01 -0.25015D+01
 139 -0.23400D+02 -0.21487D+02 -0.17252D+02 -0.38117D+01 -0.25015D+01
 140 -0.23405D+02 -0.21488D+02 -0.17252D+02 -0.38114D+01 -0.25015D+01
 141 -0.23408D+02 -0.21490D+02 -0.17251D+02 -0.38113D+01 -0.25016D+01
 142 -0.23410D+02 -0.21490D+02 -0.17251D+02 -0.38113D+01 -0.25016D+01
 143 -0.23410D+02 -0.21490D+02 -0.17251D+02 -0.38114D+01 -0.25016D+01
 144 -0.23409D+02 -0.21489D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
 145 -0.23408D+02 -0.21488D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
 146 -0.23408D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25014D+01
 147 -0.23408D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25014D+01
 148 -0.23408D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
 149 -0.23409D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
 150 -0.23409D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01

RESTART PUNCHED

TOTAL CASE TIME =  4.195599 SECONDS.

END OF JOB
```

curve show where changes in ω-value took place. A steepening of the curve below each tic showed that each higher ω-value resulted in faster convergence. The other curves in figure 10 show convergence for unsegmented runs using various ω-values. Inspecting the convergence-checking data of figure 6 shows that with $\omega = 1.80$, there was a slight tendency to fluctuate in the interval from iteration 141 to iteration 150. Further trials indicated that 1.65 was the approximate ω_{max} for this case. Disregarding the minor fluctuation at the end of figure 6, though, the PHED-values seemed to have reached a plateau, indicating that convergence was essentially complete.

The notation at the end of each segment of convergence-checking data in figure 6 indicated that restart data were punched on cards at the end of each segment. To facilitate separating punched PHED-arrays, the model punches a card with five asterisks and the words END OF FILE after each restart deck. This resulted from the card group 1 data, KARPCH = 1, and IFILE = 0. These data would have been written on magnetic tape, had IFILE $\neq$ 0, but an end-of-file mark would not appear on the tape until the run was terminated. See the discussion for IFILE, card group 1, glossary of input variables, appendix A.

Because PHED fluctuation near the 150th iteration was minor, the portion of the output deck corresponding to that iteration was used as card group 13 for the second or restart run.

Figure 7.--STDY2 printout for second run of sample problem.

STDY2 2D FOR STEADY POROUS MEDIA FLOW MODEL
SMALL SCALE SEPTIC TANK DISPOSAL, 6 X 7 CM, FINER MESH NEAR NOTCH
HOMOGENEOUS SOIL —IIC (TILL) HORIZON, SAYBROOK S.L.
SAMPLE CASE — RESTART FROM ITERATION 150

ESTIME	KAREAD	KARPCH	ITER	IFILE	IPSIG	ILSIG	KTABLE	MCHNGE
15.	1	1	150	0	0	1	0	0

LGTH	DEPTH	SLOPE	IMGTOP	IMGBOT	IMGLSD	IMGRSD	INISIG	PHEDS	ELEV
6.00	7.00	0.0	1	0	1	1	1	0.0	7.00

ITMAX	INTPRT	OMEGA	NOMEGA	NNODES	IDBLE	NCARDY	NCARDX	JGEOM	IGEOM
175	1	1.60	0	5	1	3	3	4	4

GEOMETRY AND BOUNDARY CONDITION DATA

MCOL	MROW
12	13

FOR ROW SUBSECTIONS

NSUBY	STARTY	STOPY	BEGX	ENDX	JBETA	JETA	BCLJ	BCRJ
1	0.0	2.00	0.0	6.00	1	1	0.0	0.0
2	2.10	2.90	3.00	6.00	1	1	0.0	0.0
3	3.00	4.00	3.10	6.00	0	1	-30.00	0.0
4	4.10	6.90	0.0	6.00	1	1	0.0	0.0

NSUBY	JSTART	JSTOP	IBEG	IEND
1	2	5	2	11
2	6	6	6	11
3	7	9	7	11
4	10	12	2	11

FOR COLUMN SUBSECTIONS

NSUBX	STARTX	STOPX	BEGY	ENDY	IBETA	IETA	BCUI	BCBI	FLUX
1	0.0	2.90	0.0	2.00	1	1	0.0	0.0	0.0
2	3.00	3.00	0.0	2.90	1	0	0.0	-30.00	0.0
3	0.0	3.00	4.05	6.90	0	0	-30.00	0.0	0.0
4	3.10	6.00	0.0	6.90	1	0	0.0	0.0	0.0

NSUBX	ISTART	ISTOP	JBEG	JEND
1	2	5	2	5
2	6	6	2	6
3	2	6	10	12
4	7	11	2	12

VARIABLE MESH INCREMENT DATA FOLLOW IN TRIPLETS AS XYZ
WHERE X = MEASURED DISTANCE FROM AXIS (DXLGTH OR DYLGTH)
Y = ROW OR COLUMN NUMBER (JY OR IX)
Z = INCREMENT LENGTH (DELY OR DELX)

VERTICAL

0.0	2	1.000	1.00	3	0.500	5.00	11	1.000	7.00	13	1.000

HORIZONTAL

0.0	2	1.000	2.00	4	0.500	5.00	10	1.000	6.00	12	1.000

HYDRAULIC CONDUCTIVITY TABLE FOR SOIL UNIT 1

BOTTOM OF UNIT LIES ALONG STRAIGHT LINES CONNECTING THE FOLLOWING COORDINATES
AS (X,Y) MEASURED FROM AXES
0.0 7.00 6.00 7.00
AS COLUMN AND ROW NUMBER (I,J)
2 13 12 13

P	K	P	K	P	K	P	K
0.0	0.926E-03	-0.170E+02	0.116E-03	-0.300E+02	0.787E-04	-0.770E+02	0.104E-04
-0.300E+01	0.532E-03	-0.180E+02	0.113E-03	-0.310E+02	0.752E-04	-0.100E+03	0.116E-05
-0.700E+01	0.231E-03	-0.190E+02	0.110E-03	-0.340E+02	0.671E-04	-0.300E+03	0.463E-07
-0.800E+01	0.197E-03	-0.210E+02	0.104E-03	-0.380E+02	0.579E-04	-0.500E+03	0.231E-08
-0.900E+01	0.185E-03	-0.230E+02	0.984E-04	-0.400E+02	0.544E-04	-0.790E+03	0.926E-09
-0.100E+02	0.174E-03	-0.240E+02	0.926E-04	-0.430E+02	0.498E-04	-0.100E+04	0.347E-10
-0.120E+02	0.162E-03	-0.250E+02	0.903E-04	-0.450E+02	0.463E-04	-0.100E+11	0.347E-10
-0.130E+02	0.150E-03	-0.260E+02	0.856E-04	-0.500E+02	0.394E-04		
-0.150E+02	0.127E-03	-0.280E+02	0.833E-04	-0.650E+02	0.185E-04		
-0.160E+02	0.118E-03	-0.290E+02	0.810E-04	-0.680E+02	0.162E-04		

ITERATION NO. AND PRESSURE HEAD AT SELECTED NODES AS IDENTIFIED BELOW
COORDINATES AS MEASURED FROM AXES (X,Y)
0.0 , 0.0 6.00, 0.0 4.60, 3.00 0.0 , 6.00 6.00, 6.00
COORDINATES AS ROW AND COLUMN NUMBER (I,J)

ITER	2, 2	11, 2	9, 7	2, 12	11, 12
150	-0.23409D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
151	-0.23409D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
152	-0.23409D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
153	-0.23409D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
154	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
155	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
156	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
157	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
158	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
159	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
160	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
161	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
162	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
163	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
164	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
165	-0.23408D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
166	-0.23407D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
167	-0.23407D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
168	-0.23407D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
169	-0.23407D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
170	-0.23407D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
171	-0.23407D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
172	-0.23407D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
173	-0.23407D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
174	-0.23407D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01
175	-0.23407D+02	-0.21487D+02	-0.17251D+02	-0.38114D+01	-0.25015D+01

Figure 7.—Continued.

```
                              PRESSURE HEAD DISTRIBUTION AFTER   175 ITERATIONS
  1  -0.80000D+01-0.24423D+02-0.24421D+02-0.24406D+02-0.24359D+02-0.24230D+02-0.23920D+02-0.23472D+02-0.22992D+02-0.22583D+02
     -0.22192D+02-0.80000D+01
  2  -0.70000D+01-0.23407D+02-0.23392D+02-0.23317D+02-0.23214D+02-0.23040D+02-0.22773D+02-0.22437D+02-0.22086D+02-0.21784D+02
     -0.21487D+02-0.70000D+01
  3  -0.60000D+01-0.22423D+02-0.22421D+02-0.22406D+02-0.22359D+02-0.22230D+02-0.21920D+02-0.21472D+02-0.20992D+02-0.20583D+02
     -0.20192D+02-0.60000D+01
  4  -0.55000D+01-0.21932D+02-0.21941D+02-0.21989D+02-0.22052D+02-0.22097D+02-0.21706D+02-0.21046D+02-0.20350D+02-0.19778D+02
     -0.19261D+02-0.55000D+01
  5  -0.50000D+01-0.21435D+02-0.21451D+02-0.21545D+02-0.21761D+02-0.22404D+02-0.21769D+02-0.20667D+02-0.19599D+02-0.18778D+02
     -0.18091D+02-0.50000D+01
  6  -0.45000D+01-0.20932D+02-0.20941D+02-0.20989D+02-0.21052D+02-0.24067D+02-0.22331D+02-0.20285D+02-0.18637D+02-0.17468D+02
     -0.16607D+02-0.45000D+01
  7  -0.40000D+01-0.40000D+01-0.40000D+01-0.40000D+01-0.40000D+01-0.30000D+02-0.23347D+02-0.19603D+02-0.17251D+02-0.15783D+02
     -0.14777D+02-0.40000D+01
  8  -0.35000D+01-0.35000D+01-0.35000D+01-0.35000D+01-0.35000D+01-0.30000D+02-0.22337D+02-0.17840D+02-0.15168D+02-0.13649D+02
     -0.12679D+02-0.35000D+01
  9  -0.30000D+01-0.30000D+02-0.30000D+02-0.30000D+02-0.30000D+02-0.30000D+02-0.19395D+02-0.14853D+02-0.12494D+02-0.11195D+02
     -0.10347D+02-0.30000D+01
 10  -0.25000D+01-0.18809D+02-0.18741D+02-0.18416D+02-0.17870D+02-0.16571D+02-0.13326D+02-0.10953D+02-0.94255D+01-0.84911D+01
     -0.78641D+01-0.25000D+01
 11  -0.20000D+01-0.11062D+02-0.10982D+02-0.10634D+02-0.10183D+02-0.94279D+01-0.83240D+01-0.73077D+01-0.65625D+01-0.60817D+01
     -0.57292D+01-0.20000D+01
 12  -0.10000D+01-0.38114D+01-0.37774D+01-0.36458D+01-0.35104D+01-0.33241D+01-0.31000D+01-0.28875D+01-0.27184D+01-0.25983D+01
     -0.25015D+01-0.10000D+01
 13   0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
      0.0         0.0

RESTART PUNCHED
  176 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  177 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  178 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  179 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  180 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  181 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  182 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  183 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  184 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  185 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  186 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  187 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  188 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  189 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01
  190 -0.23407D+02 -0.21487D+02 -0.17251D+02 -0.38114D+01 -0.25015D+01

                              PRESSURE HEAD DISTRIBUTION AFTER   190 ITERATIONS
  1  -0.80000D+01-0.24423D+02-0.24421D+02-0.24406D+02-0.24359D+02-0.24230D+02-0.23920D+02-0.23472D+02-0.22992D+02-0.22583D+02
     -0.22192D+02-0.80000D+01
  2  -0.70000D+01-0.23407D+02-0.23392D+02-0.23317D+02-0.23214D+02-0.23039D+02-0.22773D+02-0.22436D+02-0.22085D+02-0.21784D+02
     -0.21487D+02-0.70000D+01
  3  -0.60000D+01-0.22423D+02-0.22421D+02-0.22406D+02-0.22359D+02-0.22230D+02-0.21920D+02-0.21472D+02-0.20992D+02-0.20583D+02
     -0.20192D+02-0.60000D+01
  4  -0.55000D+01-0.21931D+02-0.21941D+02-0.21989D+02-0.22052D+02-0.22097D+02-0.21706D+02-0.21046D+02-0.20350D+02-0.19778D+02
     -0.19261D+02-0.55000D+01
  5  -0.50000D+01-0.21435D+02-0.21451D+02-0.21545D+02-0.21761D+02-0.22404D+02-0.21769D+02-0.20667D+02-0.19599D+02-0.18778D+02
     -0.18091D+02-0.50000D+01
  6  -0.45000D+01-0.20931D+02-0.20941D+02-0.20989D+02-0.21052D+02-0.24067D+02-0.22331D+02-0.20285D+02-0.18637D+02-0.17468D+02
     -0.16607D+02-0.45000D+01
  7  -0.40000D+01-0.40000D+01-0.40000D+01-0.40000D+01-0.40000D+01-0.30000D+02-0.23347D+02-0.19602D+02-0.17251D+02-0.15783D+02
     -0.14777D+02-0.40000D+01
  8  -0.35000D+01-0.35000D+01-0.35000D+01-0.35000D+01-0.35000D+01-0.30000D+02-0.22337D+02-0.17840D+02-0.15168D+02-0.13649D+02
     -0.12679D+02-0.35000D+01
  9  -0.30000D+01-0.30000D+02-0.30000D+02-0.30000D+02-0.30000D+02-0.30000D+02-0.19396D+02-0.14853D+02-0.12494D+02-0.11195D+02
     -0.10347D+02-0.30000D+01
 10  -0.25000D+01-0.18809D+02-0.18741D+02-0.18416D+02-0.17870D+02-0.16571D+02-0.13326D+02-0.10953D+02-0.94255D+01-0.84911D+01
     -0.78641D+01-0.25000D+01
 11  -0.20000D+01-0.11062D+02-0.10982D+02-0.10634D+02-0.10183D+02-0.94279D+01-0.83240D+01-0.73077D+01-0.65625D+01-0.60816D+01
     -0.57292D+01-0.20000D+01
 12  -0.10000D+01-0.38114D+01-0.37774D+01-0.36458D+01-0.35104D+01-0.33241D+01-0.31000D+01-0.28875D+01-0.27184D+01-0.25983D+01
     -0.25015D+01-0.10000D+01
 13   0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
      0.0         0.0

RESTART PUNCHED

TOTAL CASE TIME =  1.235999 SECONDS.

                                        HYDRAULIC HEAD DISTRIBUTION
  1   0.0        -0.16423D+02-0.16421D+02-0.16406D+02-0.16359D+02-0.16230D+02-0.15920D+02-0.15472D+02-0.14992D+02-0.14583D+02
     -0.14192D+02 0.0
  2   0.0        -0.16407D+02-0.16392D+02-0.16317D+02-0.16214D+02-0.16039D+02-0.15773D+02-0.15436D+02-0.15085D+02-0.14784D+02
     -0.14487D+02 0.0
  3   0.0        -0.16423D+02-0.16421D+02-0.16406D+02-0.16359D+02-0.16230D+02-0.15920D+02-0.15472D+02-0.14992D+02-0.14583D+02
     -0.14192D+02 0.0
  4   0.0        -0.16431D+02-0.16441D+02-0.16489D+02-0.16552D+02-0.16597D+02-0.16206D+02-0.15546D+02-0.14850D+02-0.14278D+02
     -0.13761D+02 0.0
  5   0.0        -0.16435D+02-0.16451D+02-0.16545D+02-0.16761D+02-0.17404D+02-0.16769D+02-0.15667D+02-0.14599D+02-0.13778D+02
     -0.13091D+02 0.0
  6   0.0        -0.16431D+02-0.16441D+02-0.16489D+02-0.16552D+02-0.19567D+02-0.17831D+02-0.15785D+02-0.14137D+02-0.12968D+02
     -0.12107D+02 0.0
  7   0.0         0.0         0.0         0.0         0.0        -0.26000D+02-0.19347D+02-0.15602D+02-0.13251D+02-0.11783D+02
     -0.10777D+02 0.0
  8   0.0         0.0         0.0         0.0         0.0        -0.26500D+02-0.18837D+02-0.14340D+02-0.11668D+02-0.10149D+02
     -0.91787D+01 0.0
  9   0.0        -0.27000D+02-0.27000D+02-0.27000D+02-0.27000D+02-0.27000D+02-0.16395D+02-0.11853D+02-0.94942D+01-0.81945D+01
     -0.73465D+01 0.0
 10   0.0        -0.16309D+02-0.16241D+02-0.15916D+02-0.15370D+02-0.14071D+02-0.10826D+02-0.84535D+01-0.69255D+01-0.59911D+01
     -0.53641D+01 0.0
 11   0.0        -0.90622D+01-0.89818D+01-0.86342D+01-0.81826D+01-0.74279D+01-0.63240D+01-0.53077D+01-0.45625D+01-0.40816D+01
     -0.37292D+01 0.0
 12   0.0        -0.26114D+01-0.27774D+01-0.26458D+01-0.25104D+01-0.23241D+01-0.21000D+01-0.18875D+01-0.17184D+01-0.15983D+01
     -0.15015D+01 0.0
 13   0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
      0.0         0.0

END OF JOB
```

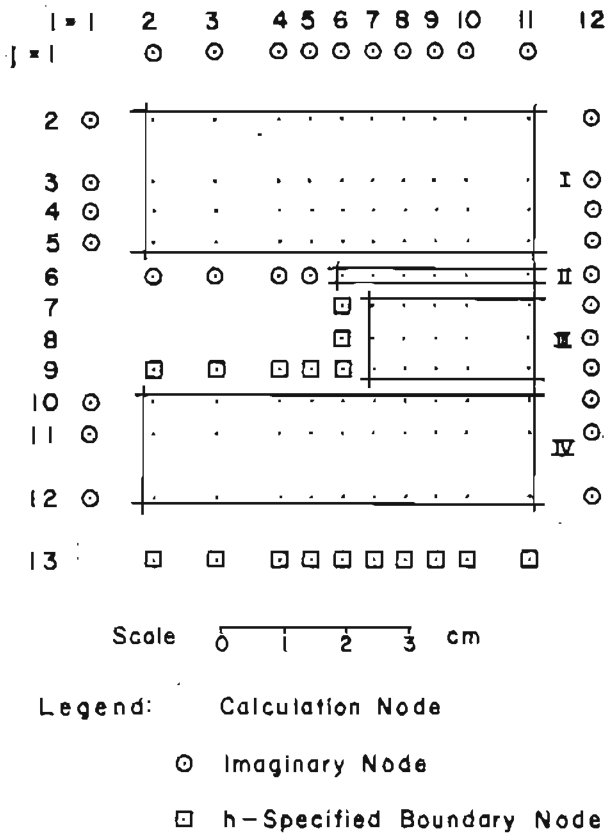


Figure 8.—Sample problem solution mesh and row subsections.

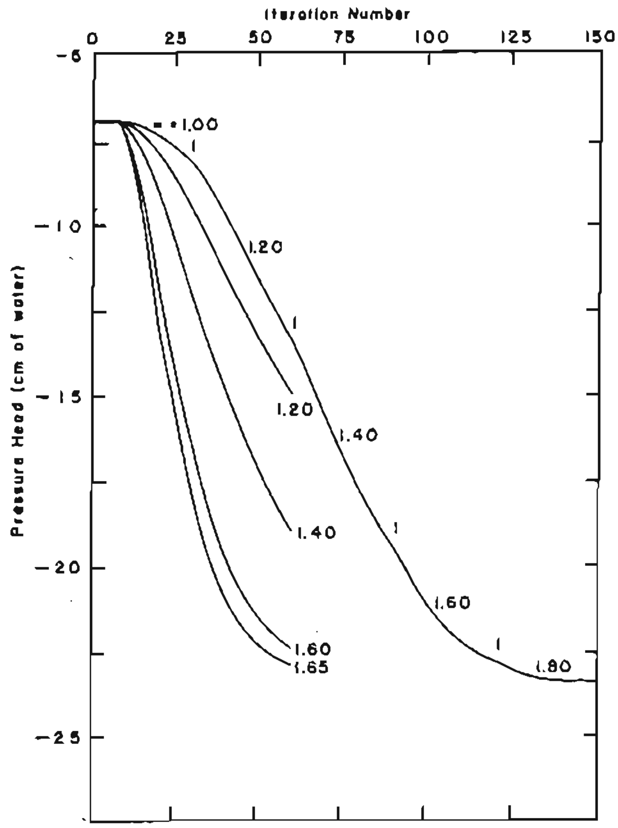


Figure 10.—Convergence as a function of overrelaxation factor.

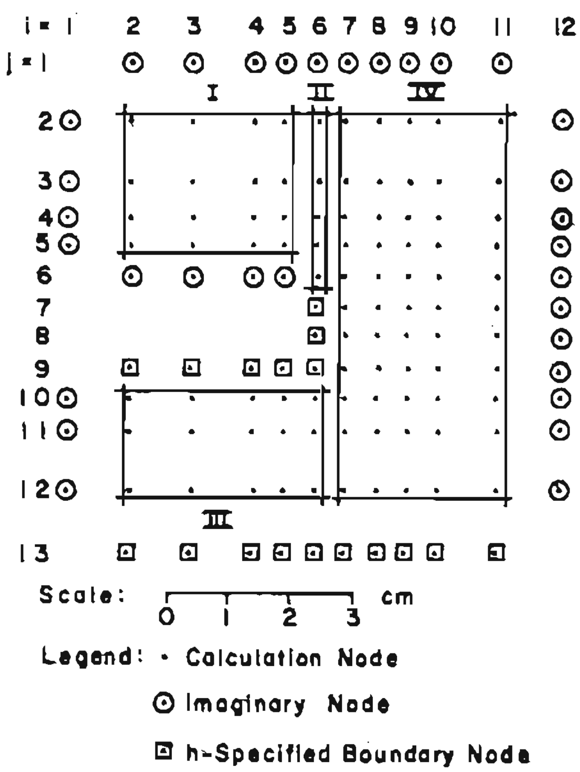


Figure 9.—Sample problem solution mesh and column subsections.

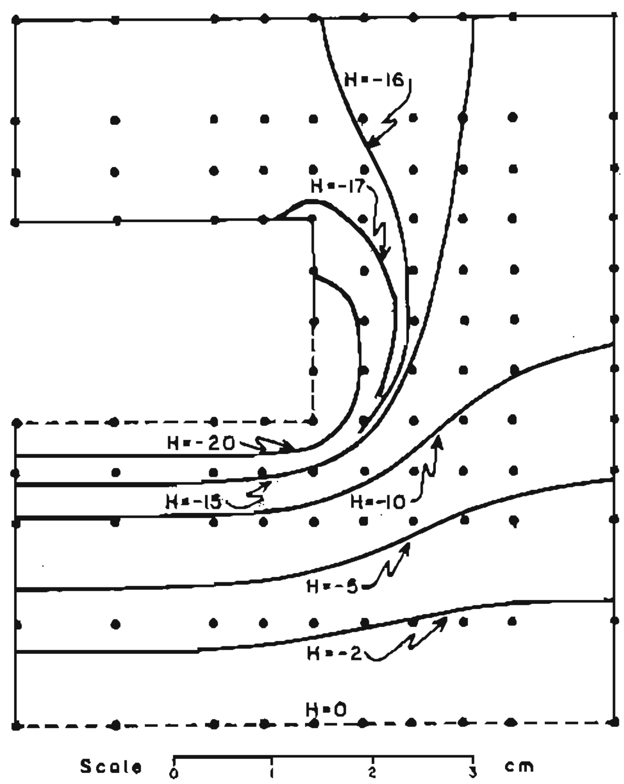


Figure 11.--Sample problem solution mesh showing equipotential distribution (imaginary nodes not shown).

Because $\omega = 1.80$ created some fluctuation and because $\omega = 1.60$ did not, the latter was used for the restart run. Complete convergence seemed nearly achieved, so the restart run was allowed only 25 iterations followed by an additional 15 for detailed convergence comparison. The need for 15 additional iterations and a second restart deck was signaled by IDBLE = 1 in card group 3.

Figure 7 contains initialization data, most of which are identical to those of figure 6. The exceptions reflect the changes made in card groups 1 and 3 before restarting. The convergence-checking data show that, on the basis of the five selected nodes, more than acceptable convergence was reached before the 175th iteration.

Node by node comparison of the two PHED-arrays (one at the 175th iteration, the other at the 190th) shows that the extra 15 iterations resulted in only six values being changed by 1 in the fifth significant digit.

Note that, because the left, top, and right boundaries are all impermeable, the left-hand column, the top row, and the right-hand column are all imaginary. So, in the PHED- and HEAD-arrays, the four outer corners of the cross section are at nodes (2,2), (11,2), (11,13), and (2,13) with the values -23.407, -21.487, 0.0, and 0.0, respectively, in the final PHED-distributions.

Data in the hydraulic head array of figure 7 were used to plot the lines of equal hydraulic head (equipotentials) in figure 11. Because of a boundary subject to -30 cm pressure head located only 3 cm higher than one subject to 0-cm pressure head, the hydraulic head gradient was directed from the lower to the higher boundary. Because the elevation datum was taken at the 0 pressure boundary, H on that boundary was 0 and therefore the H-values in the flow region were negative.

The sample cross section was also run with $\Delta x = \Delta y$ throughout the solution mesh for 0.5-cm and 1-cm mesh increment sizes. The equipotential lines of figure 11 almost exactly duplicated those produced by the 0.5-cm mesh increment case. For the 1-cm mesh increment, however, the equipotentials in the upper part of the flow region were irregular in shape and considerably displaced from their counterparts in figure 11. One may conclude, then, that the irregularity in the equipotential for which $H = -17$ would probably disappear if an even finer mesh was introduced near the notch.

Determination of acceptable convergence

Many hundreds of iterations may be necessary to reach convergence for large cross sections in which flow is partly unsaturated. To cause the program to keep track of the rate of convergence would consume a significant amount of computer time. Instead of incurring such costs, this model requires user interaction to determine when acceptable convergence has been reached. As one aid to this end, the program periodically prints the PHED-values for a user-selected set of nodes. The value of NNODES specifies the number of nodes selected. Through use of the input variable INTPRT, the user may select how often he wants these values printed.

For example, for INTPRT = 1, the selected set of PHED-values is printed every iteration. For INTPRT = 5, printing is obtained every fifth iteration. When the user wants to suppress such printing, he may give the input variable NNODES the value 0. But recall that INTPRT also controls the frequency of checking elapsed time against ESTIME and should be given a reasonable value even if NNODES = 0.

When selected nodes are printed, convergence rates may be examined by inspection or by plotting the manner in which PHED at a node varies with the number of iterations, as was done in the sample problem. If, after many iterations, convergence rate becomes slow, testing another set of overrelaxation factors may be worthwhile, because the optimum value found for the early iterations may not be optimum for later ones.

Selection of the printed nodes should be aimed at finding the node at which convergence is slowest. Experience with various cases will eventually guide the user in this respect, but one should probably start by considering a node from near each corner and at least one from near the center of the solution mesh. A maximum of eight nodes may be selected.

Scanning or plotting PHED as a function of iteration number for a few selected nodes is only an indication of how convergence is going. Some cases involving unsaturated flow have shown seemingly complete convergence in part of the flow region while in another part the PHED-values were still changing appreciably with each additional iteration. Hopefully, one or more nodes in the still-converging zone would have been included in the set selected for periodic printing. To be sure of this, after all the printed nodes have converged acceptably, one should compare two h-arrays that are separated by a few iterations.

Program COMPAR, documented in appendix C, was developed for the purpose of comparing PHED-arrays. To get two decks for comparison, a user may restart a STDY2 case for, say, 15 iterations and compare the PHED-distribution deck obtained with the one produced by the preceding run. Or, if he thinks he will be close to convergence at the end of a longer run, he may give the input variable IDBLE the value 1. This will cause the program to punch the PHED deck (or write it on tape) when ITMAX is reached, then run 15 more iterations and produce another PHED-array, both as written printout and in punchcard or magnetic tape form. When recording output on magnetic tape, one should use the IDBLE = 1 method. In this way, he obtains the two PHED-arrays within one logical tape file.

Experience has shown that a given ω-value may produce smooth convergence at most nodes, but certain nodes may begin to show instability as final convergence is approached. This will usually be detected when the result at the end of an odd-numbered iteration is compared with that of an even-numbered iteration. If fluctuation occurs and if the amplitude is too wide, then OMEGA should be reduced and further convergence obtained.

The subsectioning facility of the model may be used to save computer time if one portion of the solution mesh continues to change rapidly while the rest has apparently converged or is changing slowly. In this circumstance, rows and columns may be identified which are, in effect, boundaries between the converged and nonconverged parts of the solution mesh. These rows and columns may be considered as boundaries of known pressure head on the nonconverged part.

A restart run may then be set up in which:

1. Subsection parameters in input card groups 11 and 12, glossary of input variables, appendix A are given so that only nodes in the nonconverged part of the flow region are processed. The variables JBETA, JETA, IBETA, and IETA when on a boundary between converged and nonconverged regions should have the value 0 for known pressure head. When requirement 4 below is fulfilled, BCUI, BCBI, BCLJ, and BCRJ will not influence the solution.
2. Nodes selected under control of INTPRT should be specified inside the nonconverged part with one or two of them being located near the new boundaries.
3. All other such geometrical input data as overall length and depth, Δx and Δy, and so forth, remain unchanged from the run that produced the restart deck.
4. The input variable MCHNGE (card group 1) has the value 0.
5. The entire restart data deck, including the entire PHED-array whether on cards or on tape, is submitted with no changes other than those mentioned above.

Such a restart preserves the input PHED-values at all nodes on the boundary of and outside the nonconverged part of the mesh whereas further convergence is obtained in the zone of interest.

After reasonable convergence has been achieved in the truncated model, subsectioning data for the complete cross section should be put back into the input data deck and iterations continued until acceptable overall convergence is reached.

Definition of what constitutes an acceptable degree of convergence (the maximum acceptable difference between PHED-arrays) rests ultimately with the user. He should keep in mind that he is running a model. Regardless of how well successive PHED-distributions agree, his solution is only an approximation to the actual pressure distribution of the prototype. Besides model inaccuracies associated with non-zero Δx and Δy, the complexity of natural prototype systems and the difficulties of measuring their characteristics and properties are such that the modeler will be fortunate if he achieves better than 15- to 20-percent correspondence between model results and prototype truth. All he should be striving for, then, is a reasonable approximation.

Anomalies in the isobar or equipotential patterns will sometimes be observed. These are not necessarily because of incomplete convergence. For example, if the isobars or equipotentials plotted from final PHED- and HEAD-arrays are quite irregular and show abrupt changes in direction, without physical reason, the mesh increments may have been too coarse.

Changing mesh increment size

An auxiliary program, called CARRY and documented in appendix B, was developed to facilitate changing mesh increment size. It is useful when one already has a PHED-array in punched card form and wishes to refine the solution mesh either in total or in some localized area and then obtain further convergence without returning to a completely arbitrary starting distribution. CARRY produces an output PHED deck with the number of nodes needed for the refined mesh. PHED-values at extraneous nodes in the input PHED deck are eliminated from the output deck. PHED-values at new nodes inserted into the original mesh are interpolated from values at neighboring nodes in that original mesh. The output deck, then, portrays the same pressure head distribution as the input deck but in a differently arranged solution mesh.

CARRY concerns itself only with nodes inside and on the boundaries of the flow region. Its output deck does not contain the proper values for imaginary nodes. To restart STDY2 with a deck of CARRY output, one must give the STDY2 input variable MCHNGE some value other than 0. This assures that boundary conditions are properly set before further solution begins. For all normal restarts, MCHNGE should have the value 0.

Model dimensions

The DOUBLE PRECISION and the DIMENSION statements near the beginning of the program listing, appendix A, show the number of values which can be given each array variable used by the program. For example, PHED(60,70), HEAD(60,70), and HCON(60,70) indicate that a solution mesh can have 60 columns and 70 rows, including those containing imaginary nodes. NUMLIN(5) shows that a maximum of five soil layers may be modeled. PTAB(50,5) and KTAB(50,5) indicate that five *h-K* tables, each with a maximum of 50 lines can be read in.

The user is free, within the limits of computer storage available to him, to change these dimension values, thus changing the number of columns and rows, the number of subsections, and so on, that can be handled in the model. When making changes in dimension, one should be sure that dimensions in all associated variables are changed. For example, PHED and HEAD are equivalenced, so that their dimensions must be the same. Also, every node has associated with it a PHED-value and an HCON-value, so the dimensions of HCON should be the same as PHED and HEAD.

Computer facility adjustments

The program listed in appendix A was written in USASI Fortran, so should be compatible with most computer systems now in operation. However, each computer facility has certain unique characteristics that must be considered when implementing the model. Job control cards, not shown with the listing, will almost certainly vary from facility to facility. In addition, some program statements may also have to be modified to be compatible with a particular system. The statements likely to require modification are flagged in the listing by M1, M2, . . . The same flag is given for all statements of like category. They are discussed as follows:

M1.—Precision varies widely. The computer on which the listing was obtained and the sample problem run had single precision of four significant digits. If a computer with eight or more significant digits is used, the DOUBLE PRECISION statement could be removed to save on storage requirements. If this is done, the dimensioned variables PHED and HEAD should be added to the DIMENSION statement.

M2.—These statements pertain to time monitoring for comparison with ESTIME. Most computer facilities have a library routine which may be called to start the clock and another to return time accumulation to the program. These are often unique to the facility and probably will need changing when implementing this program for the first time at a given computer center. The routine in use with the computer producing the listing of appendix A and on which the sample problem was run returned time in milliseconds. The variable TTIME was introduced to convert to seconds for comparison with ESTIME. This would not be wanted at a facility where time is returned in seconds.

M3.—COMENT is a variable to which an A-format applies, that is, which can give alphanumeric information to a program. This information is read in as words of a certain number of characters each. On the computer used in connection with this report, a word contains only four characters (20 words per 80-column card). The statement under which COMENT is read is such that the total number of words read is specified in the DIMENSION statement, for example, 100. The FORMAT statement gives the number of words per card and the number of characters per word—20A4. Thus, with COMENT dimensioned with 100, the computer will read 100/20 = 5 cards. When running on a computer that has another word length, both the DIMENSION and the FORMAT statements must be changed to reflect the number of words and the number of characters per word in five 80-column cards. The user may want to change the number of COMENT cards. If his FORMAT statement is compatible with the computer he is using, then a change in the dimension of COMENT will change the number of cards required.

M4.—In READ(5, . . .), WRITE(6, . . .), and WRITE(7, . . .), the numerals refer to the read, write, and punch units, respectively. One or more of these numbers may differ from facility to facility. Because of the number of these statements, only one of each type was flagged. There are usually two ways to change the numeral assigning read, write, and punch units. One may replace the numerals in all read, write, and punch statements in the program so that they conform to the standard assignments at the facility being used. The other method involves the use of job control cards to reassign the read function of the computer being used to unit 5, the write function to unit 6, and the punch function to unit 7.

M5.—The two statements flagged by this symbol refer to reading and writing magnetic tapes. Some computer centers have special and unique routines for accomplishing these tasks.

The status of each of the above possible modifications, as well as questions regarding job control card requirements, should be discussed with consultants at the computer center. A user should also have a consultant check the punching in the source deck of Fortran statements. Such symbols as the equal sign or parentheses are represented by different punch combinations on different computers. A conversion routine is often available for converting the punching to a form compatible with the machine to be used.

Literature Cited

(1) Baver, L. D., Gardner, W. H., and Gardner, W. R. 1972. Soil Physics. Ed. 4. 498 pp., illus. John Wiley and Sons, New York, N.Y.

(2) Childs, E. C. 1969. An Introduction to the Physical Basis of Soil Water Phenomena. 493 pp., illus. John Wiley and Sons, New York, N.Y.

(3) Forsythe, G. E. and Wasow, W. R. 1960. Finite Difference Methods for Partial Differential Equations. 444 pp., illus. John Wiley and Sons, New York, N.Y.

(4) Greenspan, D. 1968. Lectures on the Numerical Solution of Linear, Singular, and Nonlinear Differential Equations. 185 pp., illus. Prentice Hall, Englewood Cliffs, N.J.

(5) Hillel, D. 1971. Soil and Water Physical Principles and Processes. 288 pp., illus. Academic Press, New York, N.Y.

(6) Reisenauer, A. E., Nelson, R. W., and Knudsen, C. N. 1963. Steady Darcian Transport of Fluids in Heterogeneous Partially Saturated Porous Media. AEC Res. and Develop. (Richland, Wash.) Rpt. HW-72335-PT2, 88 pp., illus.

(7) Remson, I., Hornberger, G. M., and Molz, F. J. 1971. Numerical Methods in Subsurface Hydrology with an Introduction to the Finite Element Method. 389 pp., illus. Wiley-Interscience, New York, N.Y.

(8) Smith, G. D. 1965. Numerical Solution of Partial Differential Equations. 179 pp., illus. Oxford University Press, New York, N.Y.

Appendix A:

STDY2 — The Generalized Steady-State, Two-Dimensional Porous Media Flow Model

The general philosophy and the key concepts of the model are presented in the preceding text. This appendix contains detailed documentation of the computer program that embodies the model. Included are (1) a program listing, (2) a flow chart for the program, (3) a glossary of input variables arranged in the order of their appearance in the input card deck, and (4) an alphabetical glossary of all other variables used by the program.

The following program listing contains a number of flags to alert the user to possible modifications that may be necessary before running the model on his computer. These modifications are discussed in the the text, Computer Facility Adjustments.

Program listing

```
      C  STDY2       - STEADY STATE GENERAL GEOMETRY, GENERAL BOUNDARY
      C    CONDITION MODEL.  VARIABLE DELTA X AND Y.  LAYERED SLOPING SOILS.
      C
      C  S.G.R. METHOD
      C
      C  3/18/74
      C
      C***** POSSIBLE MODIFICATION TYPE M1.
      C
0001        DOUBLE PRECISION PHED(60,70), HEAD(60,70), HEDA,HEDB,XA,XC,YA,YC
           1   ,DELTA,AY,CY,AX,CX,YB,XB,ELEV,A,B,SINAL,COSAL,XDIST
0002        REAL KTAB, KAVE, LGTH
      C
      C***** POSSIBLE MODIFICATION TYPE M2.
      C
0003        INTEGER CHKTM
      C
      C***** POSSIBLE MODIFICATION TYPE M3.
      C
0004        DIMENSION HCON(60,70),CCMENT(100),NUMLIN(5),PTAB(50,
           1   5), KTAB(50,5), JY(12),DELY(12),IX(12),DELX(12),JSTART(10),
           2   JSTOP(10),IBEG(10),IEND(10),JBETA(10),JETA(10),BCLJ(10),BCRJ(10
           3   ),ISTART(10),ISTOP(10),JBEG(10),JEND(10),IBETA(10),IETA(10),
           4   BCUT(10),BCBI(10),FLUX(10),COORDI(8),COORDJ(8),XBRK(8,5),
           5   YBRK(8,5),CYLGTH(12),DXLGTH(12),STARTY(10),STOPY(10),BEGX(10),
           6   ENDX(10),STARTX(10),STOPX(10),BEGY(10),ENDY(10),INODE(8),
           7   JNODE(8),XO(12),DELK(12),KEND(10),KSTART(10),JBRK(8,5),
           8   IBRK(8,5),NUMBRK(5),BCL(10),BCR(10)
```

Text continues on page 40.

```
      EQUIVALENCE (PHED(1,1),HEAD(1,1))
C
C***** POSSIBLE MODIFICATION TYPE M2.
C
      CALL TASKTM
C
C***** POSSIBLE MODIFICATION TYPE M4.
C
    5 READ (5,10,END=15) ESTIME,KAREAD,KARPCH,ITER,IFILE,
     1  IPSIG,ILSIG,KTABLE,MCHNGE
   10 FORMAT (F5.0,8I5)
      GO TO 25
C
C***** POSSIBLE MODIFICATION TYPE M4
C
   15 WRITE (6,20)
   20 FORMAT (1H0,10HEND OF JOB    )
      STOP
C
C***** POSSIBLE MODIFICATION TYPE M2 -- NEXT TWO STATEMENTS.
C
   25 CHKTM=0
      TTIME = 0.
      READ (5,30) LGTH,DEPTH,SLOPE,IMGTOP,IMGBOT,IMGLSD,IMGRSD,
     1  INISIG,PHEDS,ELEV
   30 FORMAT (2F10.2,F5.2,5I5,2F10.2)
      READ (5,35) ITMAX,INTPRT,OMEGA,NOMEGA,NNODES,IDBLE,NCARDY,
     1  NCARDX,JGEOM,IGEOM
   35 FORMAT (2I5,F10.2,7I5)
      NYCRD = NCARDY
      NXCRD = NCARDX
      ALPHA = ATAN(SLOPE)
      SINAL = SIN(ALPHA)
      COSAL = COS(ALPHA)
      ITMAXS = ITMAX
      IF (NNODES.NE.0) READ (5,40) (COORDI(K),COORDJ(K),K=1,NNODES)
   40 FORMAT (8F10.2)
      READ (5,45) COMENT
C
C***** POSSIBLE MODIFICATION TYPE M3.
C
   45 FORMAT (20A4)
      READ (5,50) LUNITS,KHPRNT
   50 FORMAT (2I5)
      READ (5,55)(NUMLIN(NS),NUMBRK(NS),NS=1,LUNITS)
   55 FORMAT (10I5)
      DO 60 NS = 1,LUNITS
           IF (NUMLIN(NS).EQ.999) GO TO 60
           IDUM = NUMLIN(NS)
           IDUMA = NUMBRK(NS)
           READ (5,65) (XBRK(K,NS),YBRK(K,NS),K=1,IDUMA)
           IF (KTABLE.EQ.1) GO TO 60
           READ (5,70) (PTAE(IT,NS),IT = 1,IDUM)
           READ (5,70) (KTAB (IT,NS),IT = 1,IDUM)
   60      CONTINUE
   65 FORMAT(8F10.2)
   70 FORMAT (8E10.3)
      READ (5,75) (DYLGTH(MY), DELY(MY),MY = 1,NCARDY)
      READ (5,75) (DXLGTH(MX),DELX(MX),MX = 1,NCARDX)
   75 FORMAT (4(F10.2,F10.3))
C
C*** CONVERT X,Y COORDINATES TO I,J COORDINATES **********
C
      NCARDY = NCARDY + 1
      DELY(NCARDY) = DELY(NCARDY-1)
      DYLGTH(NCARDY) = DEPTH
      NCARDX = NCARDX + 1
      DELX(NCARDX) = DELX(NCARDX - 1)
      DXLGTH(NCARDX) = LGTH
      ELEVS = ELEV
      IF(IMGTOP.EQ.1) ELEV = ELEV + DELY(1)
      IF (IMGTOP.EQ.1) GO TO 80
```

```
      JY(1) = 1
      GO TO 85
   80 JY(1) = 2
   85 DO 90 MY = 2,NCARDY
         IDUM = (DYLGTH(MY) - DYLGTH(MY-1)) / DELY(MY-1)
   90    JY(MY) = JY(MY-1) + IDUM
      IF (IMGBOT.EQ.1) JY(NCARDY) = JY(NCARCY) + 1
      MROW = JY(NCARDY)
      IF (IMGLSD.EQ.1) GO TO 95
      IX(1) = 1
      GO TO 100
   95 IX(1) = 2
  100 DO 105 MX = 2,NCARDX
         IDUM = (DXLGTH(MX) - DXLGTH(MX-1)) / DELX(MX-1)
  105    IX(MX) = IX(MX-1) + IDUM
      IF(IMGRSD.EQ.1) IX(NCARDX) = IX(NCARDX) + 1
      MCOL = IX(NCARDX)
      DO 130 K = 1,NNODES
         SUMX = 0.
         MX = 2
         IF (IMGLSD.EQ.1) GO TO 110
         I = 1
         GO TO 115
  110    I = 2
  115    IDUM = (COORDI(K) + .00005) * 1000.
         IF (IDUM.EQ.0) GO TO 125
  120    SUMX = SUMX + DELX(MX-1)
         IDUMA = (SUMX + .00005) * 1000.
         I = I + 1
         IF (I.GE.IX(MX)) MX = MX + 1
         IF (IDUMA.GE.IDUM) GO TO 125
         GO TO 120
  125      INODE(K) = I
  130    CONTINUE
      DO 155 K = 1,NNODES
         SUMY = 0.
         MY = 2
         IF (IMGTOP.EQ.1) GO TO 135
         J = 1
         GO TO 140
  135    J = 2
  140    IDUM = (COORDJ(K) + .00005) * 1000.
         IF (IDUM.EQ.0) GO TO 150
  145    SUMY = SUMY + DELY(MY-1)
         IDUMA = (SUMY + .00005) * 1000.
         J = J + 1
         IF (J.GE.JY(MY)) MY = MY + 1
         IF (IDUMA .GE.IDUM) GO TO 150
         GO TO 145
  150    JNODE(K) = J
  155    CONTINUE
      NS = 1
  160 IJK = NUMBRK(NS)
      DO 185 K = 1,IJK
         SUMX = 0.
         MX = 2
         IF (IMGLSD.EQ.1) GO TO 165
         I = 1
         GO TO 170
  165    I = 2
  170    IDUM = (XBRK(K,NS) + .00005) * 1000.
         IF (IDUM.EQ.0) GO TO 180
  175    SUMX = SUMX + DELX(MX-1)
         IDUNA = (SUMX + .00005) * 1000.
         I = I + 1
         IF (I.GE.IX(MX)) MX = MX + 1
         IF (IDUMA.GE.IDUM) GO TO 180
         GO TO 175
  180    IBRK(K,NS) = I
         IF (I.EQ.MCOL-1) IBRK(K,NS) = MCOL
  185    CONTINUE
      DO 210 K = 1,IJK
```

```
               SUMY = 0.
               MY = 2
               IF (IMGTOP.EQ.1) GO TO 190
               J = 1
               GO TO 195
      190      J = 2
      195      IDUM = (YBRK(K,NS) + .00005) * 1000.
               IF (IDUM.EQ.0) GO TO 205
      200      SUMY = SUMY + DELY(MY-1)
               IDUMA = (SUMY + .00005) * 1000.
               J = J + 1
               IF (J.GE.JY(MY)) MY = MY + 1
               IF (IDUMA.GE.IDUM) GO TO 205
               GO TO 200
      205      JBRK(K,NS) = J
               IF (J.EQ.MROW-1) JBRK(K,NS) = MROW
      210      CONTINUE
           NS = NS + 1
           IF (NS.LE.LUNITS) GO TO 160
           READ (5,215) (STARTY(NSUBY),STOPY(NSUBY),BEGX(NSUBY),ENDX(NSUBY),
          1   JBETA(NSUBY),JETA(NSUBY),BCLJ(NSUBY),BCRJ(NSUBY),NSUBY = 1,
          2   JGEOM)
      215  FORMAT(4F10.2,2I5,2F10.2)
           READ (5,220) (STARTX(NSUBX),STOPX(NSUBX),BEGY(NSUBX),ENDY(NSUBX),
          1   IBETA(NSUBX),IETA(NSUBX),BCUI(NSUBX),BCBI(NSUBX),FLUX(NSUBX),
          2   NSUBX = 1,IGEOM)
      220  FORMAT(4F10.2,2I5,2F10.2,E10.2)
           DO 222 NSUBY = 1,JGEOM
               BCL(NSUBY) = BCLJ(NSUBY)
      222      BCR(NSUBY) = BCRJ(NSUBY)
           NSIG = 1
      225  GO TO (230,240,245,255),NSIG
      230  ICHK = IMGTOP
           MCT = JGEOM
           DO 235 MY = 1,NCARDY
               KD(MY) = JY(MY)
      235      DELK(MY) = DELY(MY)
           GO TO 260
      240  MCT = IGEOM
           GO TO 260
      245  ICHK = IMGLSD
           MCT = IGEOM
           DO 250 MX = 1,NCARDX
               KD(MX) = IX(MX)
      250      DELK(MX) = DELX(MX)
           GO TO 260
      255  MCT = JGEOM
      260  DO 370 NCT = 1,MCT
               GO TO (265,270,275,280),NSIG
      265      SDUMA = STARTY(NCT)
               SDUMB = STOPY(NCT)
               GO TO 285
      270      SDUMA = BEGY(NCT)
               SDUMB = ENDY(NCT)
               GO TO 285
      275      SDUMA = STARTX(NCT)
               SDUMB = STOPX(NCT)
               GO TO 285
      280      SDUMA = BEGX(NCT)
               SDUMB = ENDX(NCT)
      285      KSIG = 0
               SUMY = 0.
               I = 2
               IF (ICHK.EQ.1) GO TO 290
               J = 1
               GO TO 295
      290      J = 2
      295      IDUM = (SDUMA + .00005) * 1000.
               IF (IDUM.EQ.0) GO TO 305
      300      SUMY = SUMY + DELK(I-1)
               IDUMA = (SUMY + .00005) * 1000.
               J = J + 1
```

```
                IF (J.GE.KD(I)) I = I + 1
                IF (KSIG.NE.0) GO TO 310
                IF (IDUMA.LT.IDUM) GO TO 300
  305           KSTART(NCT) = J
                IDUM = (SDUMB + .00005) * 1000.
                KSIG = 1
                GO TO 300
  310           IF (IDUMA-IDUM) 300,315,320
  315           KEND(NCT) = J
                GO TO 325
  320           KEND(NCT) = J - 1
  325           GO TO (330,340,350,360),NSIG
  330           DO 335 NSUBY = 1,JGEOM
                      JSTART(NSUBY) = KSTART(NSUBY)
  335                 JSTOP(NSUBY) = KEND(NSUBY)
                GO TO 370
  340           DO 345 NSUBX = 1,IGEOM
                      JBEG(NSUBX) = KSTART(NSUBX)
  345                 JEND(NSUBX) = KEND(NSUBX)
                GO TO 370
  350           DO 355 NSUBX = 1,IGEOM
                      ISTART(NSUBX) = KSTART(NSUBX)
  355                 ISTOP(NSUBX) = KEND(NSUBX)
                GO TO 370
  360           DO 365 NSUBY = 1,JGEOM
                      IBEG(NSUBY) = KSTART(NSUBY)
  365                 IEND(NSUBY) = KEND(NSUBY)
  370           CONTINUE
            NSIG = NSIG + 1
            IF (NSIG.LT.5) GO TO 225
            IF (KAREAD.NE.1) GO TO 395
            IF (IFILE.NE.0) GO TO 385
            READ (5,375) ((PHED(I,J),I = 1,MCOL),J = 1,MROW)
  375 FORMAT (6D13.6)
  380 MSIG = 0
            IF(MCHNGE.EQ.0) GO TO 550
            GO TO 420
C
C***** POSSIBLE MODIFICATION TYPE M5.
C
  385 DO 390 ICT = 1,IFILE
  390       READ(9) PHED
            GO TO 380
C
C*** INITIALIZE PHED ARRAY.**********
C
  395 IF(INISIG.EQ.0)GO TO 410
            ZHED = -ELEV
            MY = 1
            J = 1
  400 DO 405 I = 1, MCOL
  405       PHED(I,J) = ZHED
            J = J + 1
            IF (J.GT.MROW) GO TO 420
            ZHED = ZHED + DELY(MY)
            IF (J .EQ.JY(MY+1)) MY = MY + 1
            GO TO 400
  410 DO 415 J = 1,MROW
              DO 415 I = 1,MCOL
  415             PHED(I,J) = PHEDS
  420 MSIG = 0
            GO TO 830
C
C*** SET H-SPECIFIED BOUNDARY CONDITIONS AT ENDS OF ROWS.
C     PRIOR TO 1ST ITERATION ONLY.**********
C
  425 DO 445 NSUBY = 1,JGEOM
              IJK = JSTART(NSUBY)
              KSTOP = JSTOP(NSUBY)
              I = IBEG(NSUBY)
              LSTOP = IEND(NSUBY)
              IF (JBETA(NSUBY)-1) 430,432,426
```

```
  426       MY = 2
  427       IF (IJK.LE.JY(MY)) GO TO 428
            MY = MY + 1
            GO TO 427
  428       DO 429 J = IJK,KSTOP
               PHED(I-1,J) = BCL(NSUBY)
               IF (J.EQ.JY(MY)) MY = MY + 1
  429          BCL(NSUBY) = BCL(NSUBY) + DELY(MY-1)
            GO TO 432
  430       DO 431 J = IJK,KSTOP
  431          PHED(I-1,J) = BCLJ(NSUBY)
  432       IF (JETA(NSUBY)-1) 437,445,433
  433       MY = 2
  434       IF (IJK.LE.JY(MY)) GO TO 435
            MY = MY + 1
            GO TO 434
  435       DO 436 J = IJK,KSTOP
               PHED(LSTOP+1,J) = BCR(NSUBY)
               IF (J.EQ.JY(MY)) MY = MY + 1
  436          BCR(NSUBY) = BCR(NSUBY) + DELY(MY-1)
            GO TO 445
  437       DO 440 J = IJK,KSTOP
  440          PHED(LSTOP+1,J) = BCRJ(NSUBY)
  445       CONTINUE
C
C*** SET H-SPECIFIED BOUNDARY CONDITIONS AT ENDS OF COLUMNS.
C     PRIOR TO 1ST ITERATION ONLY.**********
C*** SET FLUX AND IMPERMEABLE BOUNDARY CONDITIONS AT TOPS OF COLUMNS.
C     AFTER EACH ITERATION.**********
C
  450 DO 545 NSUBX = 1,IGECM
            IJK = ISTART(NSUBX)
            KSTOP = ISTOP(NSUBX)
            J = JBEG(NSUBX)
            LSTOP = JEND(NSUBX)
            IF (IBETA(NSUBX).EQ.0) GO TO 490
            MY = 2
  460       IF (J-JY(MY)) 470,475,465
  465       MY = MY + 1
            GO TO 460
  470       DEL = 2. * DELY(MY-1) * COSAL
            GO TO 480
  475       DEL = (DELY(MY-1) + DELY(MY)) * COSAL
  480       DO 485 I = IJK,KSTOP
               KAVE = (HCON(I,J-1) + HCON(I,J) + HCON(I,J+1))/3.
  485          PHED(I,J-1) = PHED(I,J+1) - DEL * (1. + FLUX(NSUBX)/KAVE)
            GO TO 500
  490       IF (MSIG.NE.0) GO TO 545
            DO 495 I = IJK,KSTOP
  495          PHED(I,J-1) = BCUI(NSUBX)
  500       IF (MSIG.NE.0) GC TO 545
            IF (IETA(NSUBX).EQ.0) GO TO 535
            J = LSTOP
            MY = 2
  505       IF (J-JY(MY)) 520,515,510
  510       MY = MY + 1
            GO TO 505
  515       DEL = (DELY(MY-1) + DELY(MY)) * COSAL
            GO TO 525
  520       DEL = 2. * DELY(MY-1) * COSAL
  525       DO 530 I = IJK,KSTOP
  530          PHED(I,J+1) = PHED(I,J-1) + DEL
            GO TO 545
  535       DO 540 I = IJK,KSTOP
  540          PHED(I,LSTOP+1) = BCBI(NSUBX)
  545       CONTINUE
        IF (MSIG.EQ.1) GO TO 1085
C
C*** PRINT HEADING AND INITIALIZATION DATA.**********
C
  550 MSIG = 1
        WRITE (6,555) CCMENT
```

```
C
C***** POSSIBLE MODIFICATION TYPE M3.
C
  555 FORMAT(1H1.20A4/(20A4))
      WRITE (6.560)
  560 FORMAT (1H0.6HESTIME 5X.6HKAREAD 5X.6HKARPCH 5X.4HITER 5X.
     1    5HIFILE 5X.5HIPSIG 5X.5HILSIG 5X.6HKTABLE 5X.6HMCHNGE    )
      WRITE (6.565) ESTIME.KAREAD.KARPCH.ITER.IFILE.IPSIG.
     1  ILSIG.KTABLE.MCHNGE
  565 FORMAT (1H F6.0.I9.2I11.I8.3I10.I12)
      WRITE (6.570)
  570 FORMAT (1H0 3X.4HLGTH 5X.5HDEPTH 5X.5HSLOPE 5X.6HIMGTOP 5X.6HIMGBO
     1T 5X.6HIMGLSD 5X.6HIMGRSD 5X.6HINISIG 5X.5HPHEDS 5X.4HELEV    )
      WRITE (6.575) LGTH.DEPTH.SLOPE.IMGTOP.IMGBOT.IMGLSD.IMGRSD.
     1  INISIG.PHEDS.ELEVS
  575 FORMAT (1H F8.2.F10.2.F8.2.I10.4I11.F13.2.F9.2)
      WRITE (6.580)
  580 FORMAT (1H0 5HITMAX 5X.6HINTPRT 5X.5HOMEGA 5X.6HNOMEGA 5X.6HNNODES
     1 5X.5HIDBLE 5X.6HNCARDY 5X.6HNCARDX 5X.5HJGEOM 5X.5HIGEOM    )
      WRITE (6.585) ITMAX.INTPRT.OMEGA.NOMEGA.NNODES.IDBLE.NYCRD.
     1  NXCRD.JGEOM.IGEOM
  585 FORMAT (1H I4.I10.F11.2.2I10.3I11.2I10)
      WRITE (6.590)
  590 FORMAT (1H0.37HGEOMETRY AND BOUNDARY CONDITION DATA    )
      WRITE (6.595)
  595 FORMAT (1H0 4HMCOL 5X.4HMROW    )
      WRITE (6.600) MCOL.MROW
  600 FORMAT (1H I4.I9)
      WRITE (6.605)
  605 FORMAT (1H0.20HFOR ROW SUBSECTIONS    )
      WRITE (6.610)
  610 FORMAT (1H 5HNSUBY 5X.6HSTARTY 5X.5HSTOPY 5X.4HBEGX 5X.4HENDX 5X.
     15HJBETA 5X.4HJETA 5X.4HBCLJ 6X.4HBCRJ    )
      WRITE (6.615) (N.STARTY(N).STOPY(N).BEGX(N).ENDX(N).JBETA(N).
     1  JETA(N).BCLJ(N).BCRJ(N).N = 1.JGEOM)
  615 FORMAT (1H I3.F13.2.F10.2.2F9.2.I8.I10.2F10.2)
      WRITE (6.620)
  620 FORMAT (1H 5HNSUBY 3X.6HJSTART 3X.5HJSTOP 3X.4HIBEG 3X.4HIEND    )
      WRITE (6.625)(NSUBY. JSTART(NSUBY).JSTOP(NSUBY).IBEG(NSUBY).
     1  IEND(NSUBY).NSUBY = 1.JGEOM)
  625 FORMAT (1H I3.I9.I8.2I8)
      WRITE (6.630)
  630 FORMAT (1H0.23HFOR COLUMN SUBSECTIONS    )
      WRITE (6.635)
  635 FORMAT (1H 5HNSUBX 5X.6HSTARTX 5X.5HSTOPX 5X.4HBEGY 5X.4HENDY 5X.
     15HIBETA 5X.4HIETA 5X.4HBCUI 6X.4HBCBI 5X.4HFLUX    )
      WRITE (6.640) (N.STARTX(N).STOPX(N).BEGY(N).ENDY(N).IBETA(N).
     1IETA(N).BCUI(N).BCBI(N).FLUX(N).N=1.IGEOM)
  640 FORMAT (1H I3.F13.2.F10.2.2F9.2.I8.I10.2F10.2.E10.2)
      WRITE (6.645)
  645 FORMAT (1H 5HNSUBX 3X.6HISTART 3X.5HISTOP 3X.4HJBEG 3X.4HJEND    )
      WRITE (6.625) (NSUBX.ISTART(NSUBX).ISTOP(NSUBX).JBEG(NSUBX).
     1  JEND(NSUBX).NSUBX = 1.IGEOM)
      WRITE (6.650)
  650 FORMAT (1H0 55HVARIABLE MESH INCREMENT DATA FOLLOW IN TRIPLETS AS
     1XYZ    )
      WRITE (6.655)
  655 FORMAT (1H 5X.57HWHERE X = MEASURED DISTANCE FROM AXIS (DXLGTH OR
     1DYLGTH)    )
      WRITE (6.660)
  660 FORMAT (1H 11X.36HY = ROW OR COLUMN NUMBER (JY OR IX)    )
      WRITE (6.665)
  665 FORMAT (1H 11X.36HZ = INCREMENT LENGTH (DELY OR DELX)    )
      WRITE (6.670)
  670 FORMAT (1H  9HVERTICAL    )
      WRITE (6.675) (DYLGTH(N).JY(N).DELY(N).N = 1.NCARDY)
  675 FORMAT (1H 4(F8.2.I5.F8.3.8X))
      WRITE (6.680)
  680 FORMAT (1H 11HHORIZONTAL    )
      WRITE (6.675) (DXLGTH(N).IX(N).DELX(N).N = 1.NCARDX)
      DO 755 NS = 1.LUNITS
         IF (NUMLIN(NS).EQ.999) GO TO 750
         WRITE (6.685) NS
```

```
  685 FORMAT(1H0,30X,43HHYDRAULIC CONDUCTIVITY TABLE FOR SOIL UNIT    I3)
          WRITE (6,690)
  690 FORMAT (1H  79HBOTTOM CF UNIT  LIES ALONG STRAIGHT LINES CONNECTIN
     1G THE FOLLOWING COORDINATES    )
          WRITE (6,695)
  695     FORMAT(1H 28HAS (X,Y) MEASURED FROM AXES    )
          IDUM = NUMBRK(NS)
          WRITE (6,700) (XBRK(N,NS),YBRK(N,NS), N = 1,IDUM)
  700     FORMAT (1H 5(2F9.2,5X))
          WRITE (6,705)
  705     FORMAT (1H 31HAS COLUMN AND ROW NUMBER (I,J)     )
          WRITE (6,710) (IBRK(N,NS),JBRK(N,NS), N = 1,IDUM)
  710     FORMAT (1H 7(2I5,5X))
          WRITE(6,715)
  715     FORMAT (1H0,12X,1HP  12X,1HK3(14X,1HP12X,1HK))
          NUM = NUMLIN(NS)/4
          NUMA = NUM*4
          IF (NUMA.EQ.NUMLIN(NS)) GO TO 720
          LNUM = 0
          NUM = NUM + 1
          NUMA = NUM*3
          NUMA = NUMLIN(NS) - NUMA
          GO TO 725
  720     LNUM = 1
          NUMA = NUM
  725     IT = 1
  730     WRITE (6,735) PTAB(IT,NS),KTAB(IT,NS),PTAB(IT+NUM,NS),KTAB
     1      (IT+NUM,NS),PTAB(IT+2*NUM,NS),KTAB(IT+2*NUM,NS),PTAB(IT+
     2      3*NUM,NS),KTAB(IT+3*NUM,NS)
  735     FORMAT (1H ,4(5X,E10.3,3X,E10.3))
          IT = IT + 1
          IF (IT.LE.NUMA) GO TO 730
          IF (LNUM.EQ.1) GO TO 755
  740     WRITE (6,745) PTAB(IT,NS),KTAB(IT,NS),PTAB(IT+NUM,NS),KTAB
     1      (IT+NUM,NS),PTAB(IT+2*NUM,NS),KTAB(IT+2*NUM,NS)
  745     FORMAT (1H ,3(5X,E10.3,3X,E10.3))
          IT = IT + 1
          IF (IT.LE.NUM) GO TO 740
          GO TO 755
  750     WRITE (6,760) NS
  755     CONTINUE
  760 FORMAT (1H0 32HHYDRAULIC CONDUCTIVITY FOR UNIT    I5, 53H IS OBTAIN
     1ED FROM AN EQUATION.  SEE PROGRAM LISTING.  )
        IF (KHPRNT.NE.1) GO TO 775
        WRITE (6,765)
  765 FORMAT (1H0 48X,23HINITIAL K-DISTRIBUTION    )
      DO 770 J = 1,MROW
  770     WRITE(6,1190) J,(HCON(I,J),I=1,MCOL)
  775 IF (IPSIG.NE.1) GO TO 785
      WRITE (6,780)
  780 FORMAT (1H0 ,40X,39HSTARTING DISTRIBUTION OF PRESSURE HEAD    )
      LSIG = 2
      GO TO 1180
  785 IF (NNODES.EQ.0) GO TO 825
      WRITE (6,790)
  790 FORMAT(1H0 ,70HITERATION NO. AND PRESSURE HEAD AT SELECTED NODES A
     1S IDENTIFIED BELOW    )
      WRITE (6,795)
  795 FORMAT (1H 40HCOORDINATES AS MEASURED FROM AXES (X,Y)    )
      WRITE (6,800) (COORDI(K),COORDJ(K),K = 1,NNODES)
  800 FORMAT(1H 5(F8.2,1H,,F8.2,4X))
      WRITE (6,805)
  805 FORMAT (1H  43HCOORDINATES AS ROW AND COLUMN NUMBER (I,J)    )
      WRITE (6,810) (INODE(K),JNODE(K),K=1,NNODES)
  810 FORMAT (1H  ,4HITER,1X,I4,1H,,I4,8(4X,I4,1H,,I4    )  )
      WRITE (6,815)
  815 FORMAT (1H     )
      WRITE (6,820) ITER,(PHED(INODE(K),JNODE(K)),K=1,NNODES)
  820 FORMAT (1H ,I5,9D13.5)
C
C*** BEGIN AN ITERATION.**********
C
  825 ITER = ITER + 1
```

```
C
C*** TABLE LOOKUP FOR HYDRAULIC CONDUCTIVITY.**********
C
  830 YDIST = 0.
      IF (IMGTOP.EQ.1) YDIST = -DELY(1)
      MY = 2
      J = 1
  835 I = 1
      MX = 2
      XDIST = 0.
      IF (IMGLSD.EQ.1) XDIST = - DELX(1)
      NS = 1
  840 N = 2
  845 IF (I.LE.IBRK(N,NS)) GO TO 860
  850 N = N + 1
      IF (N.LE.NUMBRK(NS)) GO TO 845
  855 NS = NS + 1
      IF (NS.LE.LUNITS) GO TO 840
      GO TO 870
  860 GRAD = (YBRK(N,NS) - YBRK(N-1,NS)) / (XBRK(N,NS) - XBRK(N-1,NS))
      ELIJ = YBRK(N-1,NS) + (XDIST - XBRK(N-1,NS)) * GRAD
      IF (YDIST.LE.ELIJ) GO TO 880
  865 NS = NS + 1
      IF (NS.LE.LUNITS) GO TO 840
  870 WRITE (6,875)
  875 FORMAT(1H0 27H SOIL UNIT INPUT DATA ERROR    )
      LSIG = 1
      GO TO 1152
  880 IF (NUMLIN(NS).NE.999) GO TO 885
C
C   *********INSERT PROGRAMMING FOR CALCULATING K FROM EQUATIONS, IF
C                   REQUIRED.
C
      IF (MSIG.EQ.0) GO TO 425
      GO TO 980
  885 IF (PHED(I,J).GE.0.) GO TO 895
      IT = NUMLIN(NS)/2
      IF (PHED(I,J) - PTAB(IT,NS)) 890,930,900
  890 L = 5
      GO TO 905
  895 HCON(I,J) = KTAB(1,NS)
      GO TO 935
  900 L = 1
  905 IT = L*NUMLIN(NS)/8
      IF (PHED(I,J) - PTAB(IT,NS)) 910,930,925
  910 L = L + 1
      IF (L.LE.8) GO TO 905
  915 WRITE (6,920)ITER,I,J
  920 FORMAT (1H0,39HK TABLE LIMITS EXCEEDED. ITERATION NO.   I5,5H I =
     1   I5,5H J =  I5)
      LSIG = 1
      GO TO 1180
  925 IT = IT - 1
      IF (PHED(I,J).GT.PTAB(IT,NS))GO TO 925
  930 FACTOR = (PHED(I,J) - PTAB(IT,NS)) / (PTAB(IT+1,NS) - PTAB(IT,NS))
      HCON(I,J) = KTAB(IT,NS) + FACTOR * (KTAB(IT+1,NS) - KTAB(IT,NS))
  935 I = I + 1
      IF (I-MCOL) 950,940,945
  940 IF (IMGRSD.EQ.1) GO TO 955
      GO TO 950
  945 J = J + 1
      IF (J.GT.MROW) GO TO 975
      IF (J.EQ.MROW.AND.IMGBCT.EQ.1) GO TO 835
      YDIST = YDIST + DELY(MY-1)
      IF (J.GE.JY(MY)) MY = MY + 1
      GO TO 835
  950 XDIST = XDIST + DELX(MX-1)
      IF (I.GE.IX(MX)) MX = MX + 1
      IF (I.GT.IBRK(N,NS)) GO TO 850
      ELIJ = YBRK(N-1,NS) + (XDIST - XBRK(N-1,NS)) * GRAD
      IF (YDIST.GT.ELIJ) GO TO 865
  955 IF (PHED(I,J).GE.0.) GO TO 895
      IF (PHED(I,J) - PTAB(IT,NS))960,930,970
```

```
  960 IT = IT + 1
      IF (IT.GT.NUMLIN(NS))GO TO 915
      IF (PHED(I,J) - PTAB(IT,NS)) 960,930,965
  965 IT = IT - 1
      GO TO 930
  970 IT = IT - 1
      IF (PHED(I,J).GT.PTAB(IT,NS)) GO TO 970
      GO TO 930
  975 IF (MSIG.EQ.0) GO TO 425
C
C*** BEGIN EQUIVALENT OF "DO-LOOP" SOLVING FINITE DIFFERENCE
C     EQUATION FOR EACH NODE OF MESH.**********
C
  980 DO 1080 NSUBY = 1,JGEOM
C
C*** BEGIN A ROW SUBSECTION.**********
C
            J = JSTART(NSUBY)
            KSTOP = JSTOP(NSUBY)
            IJK = IBEG(NSUBY)
            LSTOP = IEND(NSUBY)
C
C*** LOCATE DELTA X AND DELTA Y AT TOP AND LEFT SIDE OF THE
C     ROW SUBSECTION.**********
C
            MY = 2
  985       IF (J.LE.JY(MY)) GO TO 990
            MY = MY + 1
            GO TO 985
  990       MX = 2
  995       IF (IJK.LE.IX(MX)) GO TO 1000
            MX = MX + 1
            GO TO 995
 1000       MXMYST = MX
C
C*** START A ROW WITHIN THE SUBSECTION.**********
C
 1005       I = IJK
            KBETA = JBETA(NSUBY)
            KETA = 0
            MX = MXMYST
            DELYM = DELY(MY-1)
            IF (J.LT.JY(MY)) GO TO 1010
            DELYP = DELY(MY)
            MY = MY + 1
            GO TO 1015
 1010       DELYP = DELY(MY-1)
 1015       IF (J.NE.KSTOP) GO TO 1045
C
C*** SET IMPERMEABLE BOUNDARY CONDITION, IF REQUIRED, AT BOTTOM
C     OF COLUMNS PRIOR TO SWEEPING LAST ROW OF SUBSECTION.**********
C
            DO 1040 NSUBX = 1,IGEOM
 1020          IF (J.NE.JEND(NSUBX)) GO TO 1040
               IF (IETA(NSUBX).EQ.0) GO TO 1040
               IF (J.EQ.JY(MY-1)) GO TO 1025
               DEL = 2. * DELY(MY-1) * COSAL
               GO TO 1030
 1025          DEL = (DELY(MY-2) + DELY(MY-1)) * COSAL
 1030          IDUMA = ISTART(NSUBX)
               IDUM = ISTOP(NSUBX)
               DO 1035 K = IDUMA,IDUM
 1035             PHED(K,J+1) = PHED(K,J-1) + DEL
 1040       CONTINUE
C
C*** CALCULATIONS FOR INDIVIDUAL NODE WITHIN A ROW STARTS HERE**********
C
 1045       DELXM = DELX(MX-1)
            IF (I.LT.IX(MX))  GO TO 1050
            DELXP = DELX(MX)
            MX = MX + 1
            GO TO 1055
```

```
 1050     DELXP = DELX(MX-1)
C
C*** PREPARE PARAMETERS FOR FINITE DIFFERENCE EQUATION.**********
C
 1055     YA = .5 * (HCON(I,J) + HCON(I,J-1))
          YC = .5 * (HCON(I,J) + HCON(I,J+1))
          XA = .5 * (HCON(I,J) + HCON(I-1,J))
          XC = .5 * (HCON(I,J) + HCON(I+1,J))
          HEDA = PHED(I-1,J)
          HEDB = PHED(I+1,J)
          IF (KBETA.EQ.0) GO TO 1060
          KBETA = 0
          XA = XC
          HEDA = HEDB + (DELXM + DELXP) * SINAL
          GO TO 1065
 1060     IF (KETA.EQ.0) GO TO 1065
          XC = XA
          HEDB = HEDA - (DELXM + DELXP) * SINAL
 1065     AY = YA / DELYM
          CY = YC / DELYP
          AX = XA / DELXM
          CX = XC / DELXP
          YB = (DELYP * YA + DELYM * YC) / (DELYM * DELYP)
          XB = (DELXP * XA + DELXM * XC) / (DELXM * DELXP)
          EY = 2. / (DELYM + DELYP)
          EX = 2. / (DELXM + DELXP)
          DELTA = ((HCON(I,J-1) - HCON(I,J+1)) / (DELYM + DELYP)) *
     1      COSAL + ((HCON(I-1,J) - HCON(I+1,J)) / (DELXM + DELXP)) *
     2      SINAL
C
C*** THE FINITE DIFFERENCE EQUATION.**********
C
          PHED(I,J) = (1.- OMEGA) * PHED(I,J) + OMEGA *
     1      (EX * (AX * HEDA + CX * HEDB) +
     2      EY * (AY * PHED(I,J-1) + CY * PHED(I,J+1)) + DELTA) /
     3      (EX * XB + EY * YB)
          I = I + 1
          IF (I-LSTOP) 1045,1070,1075
 1070     KETA = JETA(NSUBY)
          GO TO 1045
 1075     J = J + 1
          IF (J.LE.KSTOP) GO TO 1005
 1080     CONTINUE
      GO TO 450
C
C*** THIS COMPLETES AN ITERATION**********
C*** CHECK WHETHER TO PRINT, ON CPU TIME, AND ON NUMBER OF
C     ITERATIONS**********
C
 1085 IPRINT = ITER/INTPRT
      IPRINT = INTPRT * IPRINT
      IF (ITER.NE.IPRINT) GO TO 825
      IF (NNODES.EQ.0) GO TO 1090
      WRITE (6,820) ITER,(PHED(INODE(K),JNODE(K)),K=1,NNODES)
C
C***** POSSIBLE MODIFICATION TYPE M2 -- NEXT 3 STATEMENTS.
C
 1090 CALL TASKTM(CHKTM)
      TTIME = TTIME + CHKTM
      TIME = TTIME / 1000.
      IF (TIME.GE.ESTIME) GO TO 1100
      IF (ITER.LT.ITMAX) GO TO 825
      IF (ILSIG.NE.1) GO TO 1110
      WRITE (6,1095) ITER
 1095 FORMAT (1H0,33X,33HPRESSURE HEAD DISTRIBUTION AFTER   I5,
     112H ITERATIONS    )
      LSIG = 3
      GO TO 1180
 1100 LSIG = 4
      WRITE (6,1105)
 1105 FORMAT (1H0 16HESTIME EXCEEDED    )
 1110 IF (KARPCH.NE.1) GO TO 1145
```

```
C
C*** PUNCH ON CARDS OR WRITE ON TAPE RESTART**********
C
       IF (IFILE.NE.0) GO TO 1130
C
C***** POSSIBLE MODIFICATION TYPE M4.
C
      WRITE (7,375) ((PHED(I,J),I = 1,MCOL),J = 1,MROW)
      WRITE (7,1120)
 1120 FORMAT (16H*****END OF FILE)
      WRITE (6,1125)
 1125 FORMAT (1H0,16HRESTART PUNCHED    )
      GO TO 1140
C
C***** POSSIBLE MODIFICATION TYPE M5.
C
 1130 WRITE(10)PHED
      WRITE (6,1135)
 1135 FORMAT (1H0,24HRESTART WRITTEN ON TAPE    )
 1140 IF (IDBLE.NE.1.OR.TIME.GE.ESTIME) GO TO 1145
      IDBLE = 0
      ITMAX = ITMAX + 15
      GO TO 825
 1145 WRITE (6,1150) TIME
 1150 FORMAT (1H0,17HTOTAL CASE TIME =,F10.6,10H SECONDS.    )
 1152 IF (NOMEGA.NE.0) GO TO 1205
      IF (LSIG.EQ.1) GO TO 5
C
C*** CONVERT PRESSURE HEAD TO HYDRAULIC HEAD.**********
C
      J = 1
      MY = 2
 1155 A = ELEV * COSAL
      I = 0
      MX = 2
      XDIST = 0
      IF (IMGLSD.EQ.1) XDIST = -DELX(I)
 1160 B = XDIST * SINAL
      I = I + 1
      IF (I.GT.MCOL) GO TO 1165
      XDIST = XDIST + DELX(MX-1)
      IF(IX(MX).EQ.I) MX= MX + 1
      HEAD(I,J) = PHED(I,J) + A + B
      GO TO 1160
 1165 J = J + 1
      IF (J.GT.MROW) GO TO 1170
      ELEV = ELEV - DELY(MY-1)
      IF (JY(MY).EQ.J) MY= MY + 1
      GO TO 1155
 1170 WRITE (6,1175)
 1175 FORMAT (1H1,46X,28HHYDRAULIC HEAD DISTRIBUTION    )
      LSIG = 4
C
C*** OBTAIN PRINTOUT OF "PHED" ARRAY OR "HEAD" ARRAY.**********
C
 1180 DO 1185 J = 1,MROW
 1185     WRITE (6,1190) J,(PHED(I,J),I = 1,MCOL)
 1190 FORMAT (1H I3,2X,10D12.5/(6X,10D12.5))
      IF (LSIG -2) 1152,785,1200
 1200 IF (LSIG.EQ.4) GO TO 5
      GO TO 1110
C
C*** READ IN OMEGA VALUE FOR NEXT SEGMENT OF SEGMENTED RUN.**********
C
 1205 READ (5,1210) OMEGA,ITMAX
 1210 FORMAT (F5.2,I5)
      IF (ITMAX.EQ.0) GO TO 5
      IF (LSIG.EQ.1) GO TO 1205
      WRITE (6,1215) OMEGA,ITMAX
 1215 FORMAT (1H0 22HCONTINUE WITH OMEGA =    F 5.2,10H, ITMAX =    I5)
      GO TO 825
      END
```

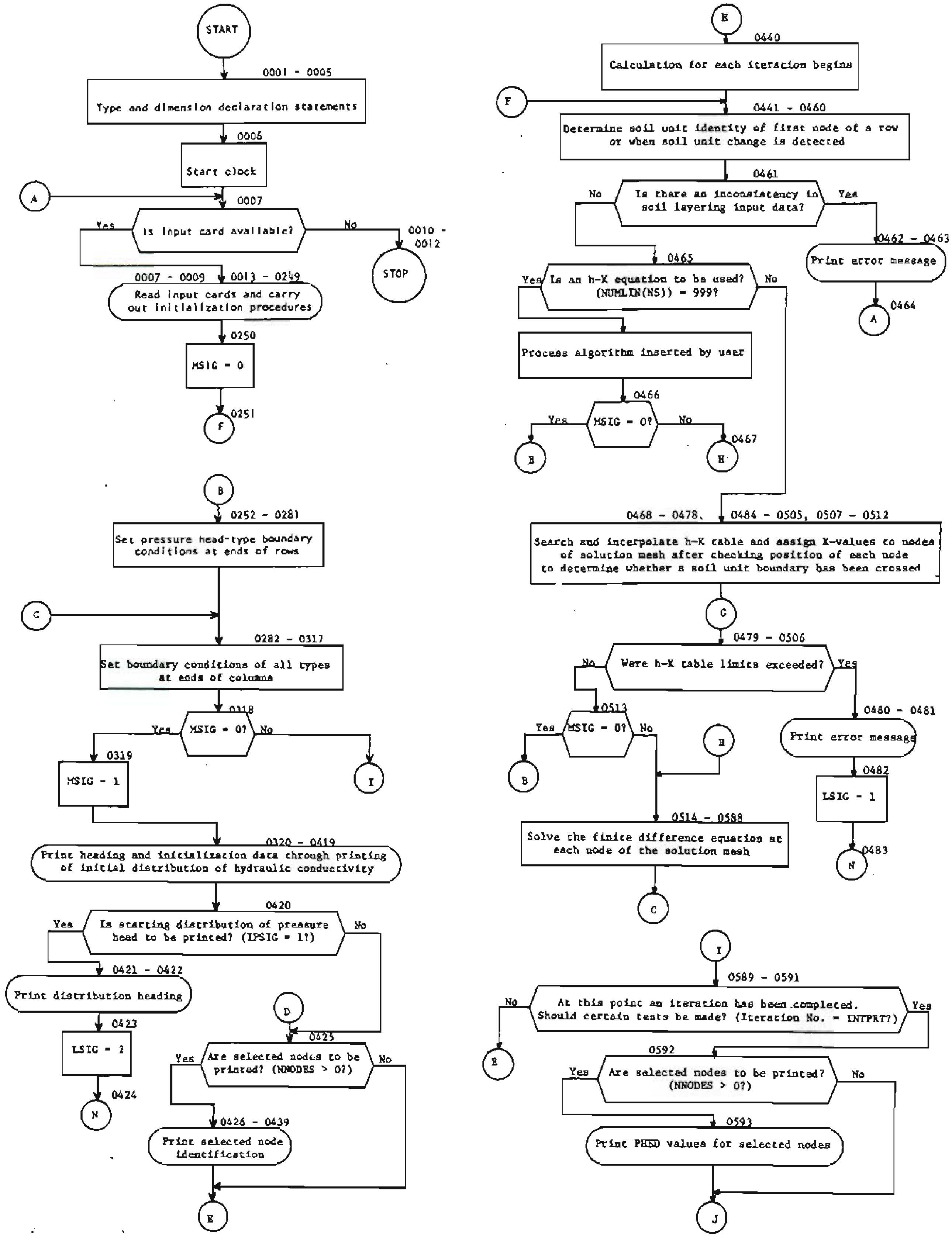
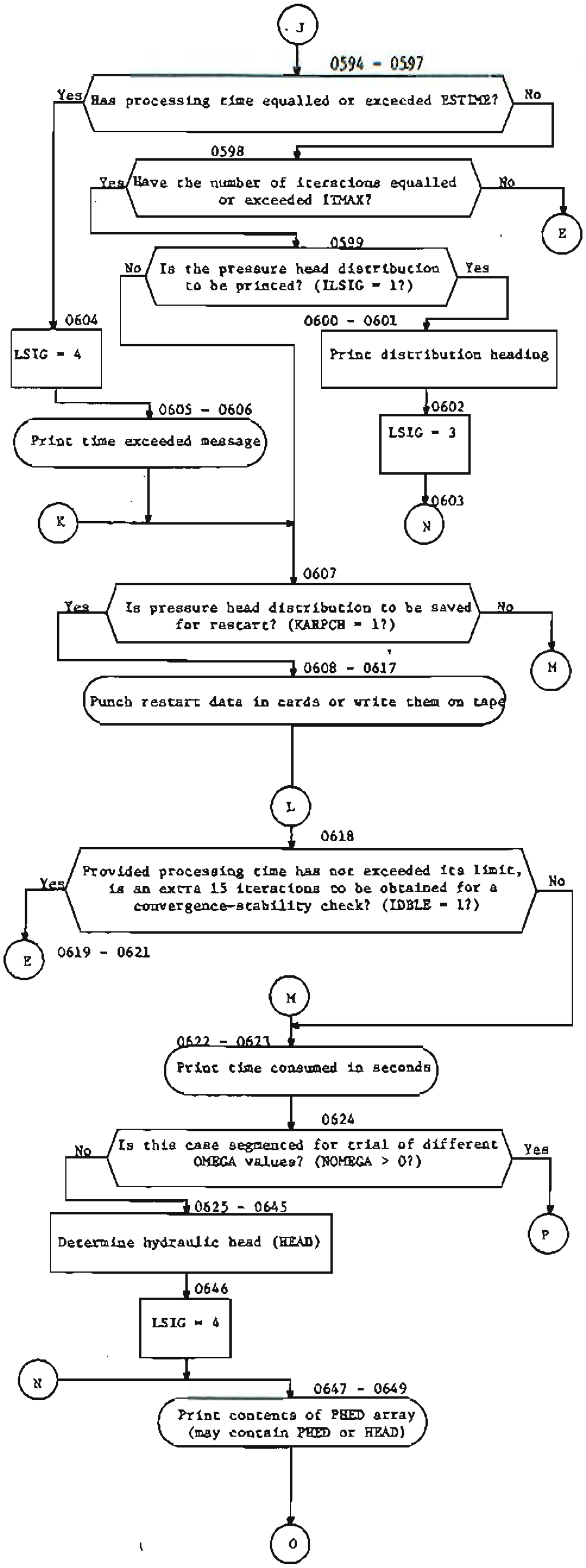


Figure 12.--Flow chart for STDY2.



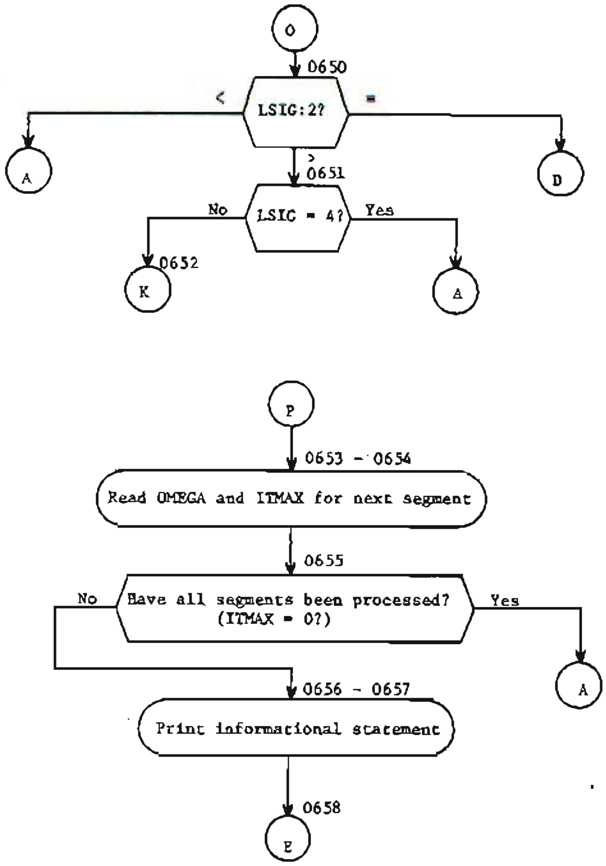


Figure 12.--Continued.

Flow chart

The flow chart (fig. 12) contains the major branching points in the STDY2 program listing. A major branch, in this context, significantly shifts the flow of the program from one part of the listing to another.

The numbers over each box of the flow chart key the operation(s) described in that box to statements in the program listing. When the number key over a box consists of the end points of a range of numbers, the program processes the included statements in the order of their appearance as modified by local branching to nearby statements.

Glossary of input variables

Input variables are defined in the following glossary in the order in which they must appear in the input data deck. As noted in their explanations, not all card groups are needed for every case.

Figure 13 shows punchcard layouts for the input data. The format number at the left end of each card serves to identify it with an identically numbered format statement in the STDY2 program listing. Each layout image

Text continues on page 45.

Figure 13.--Input data layout for STDY2.

USDA-FOREST SERVICE

MULTIPLE-CARD LAYOUT FORM

PROGRAM NAME:	FORMAT DESIGNED BY:	DATE:	PROGRAM NUMBER:	PAGE:
STDY2		3/74		1 OF 4

Format 10 Card Group 1

Card columns	1–5	6–10	11–15	16–20	21–25	26–30	31–35	36–40	41–45	46–80
Variable	ESTIME	KAREAD	KARPCH	ITER	IFILE	IPSIG	ILSIG	KTABLE	MCHNGE	Blank
Format	F5.0	I5	I5	I5	I5	I5	I5	I5	I5	
Value	15	0	1	0	0	1	1	0	0	

Format 30 Card Group 2

Card columns	1–10	11–20	21–25	26–30	31–35	36–40	41–45	46–50	51–60	61–70	71–80
Variable	LGTH	DEPTH	SLOPE	IMGTOP	IMGBOT	IMGLSD	IMGRSD	INISIG	PHEDS	ELEV	Blank
Format	F10.2	F10.2	F5.2	I5	I5	I5	I5	I5	F10.2	F10.2	
Value	6.00	7.00	0.00	1	0	1	1	1	0.00	7.00	

Format 35 Card Group 3

Card columns	1–5	6–10	11–20	21–25	26–30	31–35	36–40	41–45	46–50	51–55	56–80
Variable	ITMAX	INTPRT	OMEGA	NOMEGA	NNODES	IDBLE	NCARDY	NCARDX	JGEOM	IGEOM	Blank
Format	I5	I5	F10.2	I5	I5	I5	I5	I5	I5	I5	
Value	30	1	1.00	1	5	0	3	3	4	4	

Format 40 Card Group 4

Card columns	1–10	11–20	21–30	31–40	41–50	51–60	61–70	71–80
Variable	COORDI(1)	COORDJ(1)	COORDI(2)	COORDJ(2)	COORDI(3)	COORDJ(3)	COORDI(4)	COORDJ(4)
Format	F10.2	F10.2	F10.2	F10.2	F10.2	F10.2	F10.2	F10.2

GPO 916-667 6600-1 (11/66)

Figure 13.--Continued.

USDA-FOREST SERVICE	PROGRAM NAME: STDY2		PROGRAM NUMBER:
MULTIPLE-CARD LAYOUT FORM	FORMAT DESIGNED BY:	DATE: 3/74	PAGE: 2 OF 4

CARD IDENT.

Format 45, Card Group 5

Columns 1-80
COMENT 20A4
STDY2 2D SOR STEADY POROUS MEDIA FLOW MODEL

Format 50, Card Group 6

Columns 1-5	Columns 6-10	Columns 11-80
LUNITS	KHPRNT	Blank
I5	I5	
1	0	

Format 55, Card Group 7

1-5	6-10	11-15	16-20	21-25	26-30	31-35	36-40	41-45	46-50	51-80
NUMLIN (1)	NUMBRK (1)	NUMLIN (2)	NUMBRK (2)	NUMLIN (3)	NUMBRK (3)	NUMLIN (4)	NUMBRK (4)	NUMLIN (5)	NUMBRK (5)	Blank
I5	I5	I5	I5	I5	I5	I5	I5	I5	I5	
37	2									

Format 65, Card Group 8a

1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80
XBRK(1,1)	YBRK(1,1)	XBRK(2,1)	YBRK(2,1)	XBRK(3,1)	YBRK(3,1)	XBRK(4,1)	YBRK(4,1)
F10.2	F10.2	F10.2	F10.2	F10.2	F10.2	F10.2	F10.2
0.00	7.00	6.00	7.00				

GPO 916-687 6600-1 (11/66)

Figure 13.--Continued.

USDA-FOREST SERVICE **MULTIPLE-CARD LAYOUT FORM**	PROGRAM NAME: STDY2		PROGRAM NUMBER:
	FORMAT DESIGNED BY:	DATE: 3/74	PAGE: 3 OF 4

CARD IDENT.								
Format 70 Card Group 8b	PTAB(1,1) E10.3 .000+00	PTAB(2,1) E10.3 -.300+01	PTAB(3,1) E10.3 -.700+01	PTAB(4,1) E10.3 -.800+01	PTAB(5,1) E10.3 -.900+01	PTAB(6,1) E10.3 -.100+02	PTAB(7,1) E10.3 -.120+02	PTAB(8,1) E10.3 -.130+02
Format 70 Card Group 8c	KTAB(1,1) E10.3 .926-03	KTAB(2,1) E10.3 .532-03	KTAB(3,1) E10.3 .231-03	KTAB(4,1) E10.3 .197-03	KTAB(5,1) E10.3 .185-03	KTAB(6,1) E10.3 .174-03	KTAB(7,1) E10.3 .162-03	KTAB(8,1) E10.3 .150-03
Format 75 Card Group 9	DYLGTH(1) F10.2 0.00	DELY(1) F10.3 1.000	DYLGTH(2) F10.2 1.00	DELY(2) F10.3 0.500	DYLGTH(3) F10.2 5.00	DELY(3) F10.3 1.000	DYLGTH(4) F10.2	DELY(4) F10.3
Format 75 Card Group 10	DXLGTH(1) F10.2 0.00	DELX(1) F10.3 1.000	DXLGTH(2) F10.2 2.00	DELX(2) F10.3 0.500	DXLGTH(3) F10.2 5.00	DELX(3) F10.3 1.000	DXLGTH(4) F10.2	DELX(4) F10.3

GPO 916-687 6600-1 (11/66)

Figure 13.--Continued.

USDA-FOREST SERVICE	PROGRAM NAME:		PROGRAM NUMBER:
MULTIPLE-CARD LAYOUT FORM	STDY2		
	FORMAT DESIGNED BY:	DATE: 3/74	PAGE: 4 OF 4

Format 215, Card Group 11

Columns	1–10	11–20	21–30	31–40	41–45	46–50	51–60	61–70	71–80
Field	STARTY(1)	STOPY(1)	BEGX(1)	ENDX(1)	JBETA (1)	JETA (1)	BCLJ(1)	BCRJ(1)	Blank
Format	F10.2	F10.2	F10.2	F10.2	I5	I5	F10.2	F10.2	
Value	0.00	2.00	0.00	6.00	1	1	0.00	0.00	

Format 220, Card Group 12

Columns	1–10	11–20	21–30	31–40	41–45	46–50	51–60	61–70	71–80
Field	STARTX(1)	STOPX(1)	BEGY(1)	ENDY(1)	IBETA (1)	IETA (1)	BCUI(1)	BCBI(1)	FLUX(1)
Format	F10.2	F10.2	F10.2	F10.2	I5	I5	F10.2	F10.2	F10.2
Value	0.00	2.90	0.00	2.00	1	1	0.00	0.00	0.00

Format 375, Card Group 13

Columns	1–13	14–26	27–39	40–52	53–65	66–78	79–80
Field	PHED(1,1)	PHED(2,1)	PHED(3,1)	PHED(4,1)	PHED(5,1)	PHED(6,1)	Blank
Format	D13.6	D13.6	D13.6	D13.6	D13.6	D13.6	
Value	-0.750000D+01	-0.214859D+02	-0.216063D+02	-0.218078D+02	-0.220783D+02	-0.223953D+02	

Format 1210, Card Group 14

Columns	1–5	6–10	11–80
Field	OMEGA	ITMAX	Blank
Format	F5.2	I5	
Value	1.20	60	

GPO 916-687 6600-1 (11/66)

represents a group of one or more cards. They are in the order of their appearance in the input deck, which is also the order in which they are discussed in the glossary. Some of these variables were also discussed in the text and their use illustrated in the sample problem. Besides variable names, Fortran formats are given in the figure and the glossary. A sample entry of each variable is shown in figure 13.

The input deck for any given run may include the data for as many cases as the user desires. Simply add one case behind another, that is, card group 1 for case m follows immediately behind the last card group of case m-1.

Card Group 1 - A single card. Must be present in input deck for processing each case. Format (F5.0,8I5).

ESTIME - Time in seconds that user allots for the processing of a case within a computer's central processing unit. It obtains restart data before a run stops when it exceeds the time limit given on the job control card. This latter limit must be greater than ESTIME by at least the amount of time necessary to compile the run. CAUTION: The frequency of time checks depends on INTPRT--see card group 3. When several cases follow each other in a single run, the job control card time should exceed the sum of the several ESTIME values. CAUTION: When a run enters an endless loop because of an input data error, ESTIME cannot be checked. See the discussion of ITMAX, card group 3, for the procedure to follow when initializing a case--or when making a major change in input data.

KAREAD = 1 means restart data are read in.
≠ 1 means case starts from one of program-generated initial PHED-arrays.

KARPCH = 1 means restart data are punched or written on magnetic tape at the end of case processing.
≠ 1 means restart data are not saved in punched card or magnetic tape form.

ITER - Iteration number. Its input value should be 0 when starting to solve a new case. Otherwise, it should have the value of the number of the iteration at the end of which the restart data were produced. The value of ITER is not critical to the solution, but the number of iterations processed by a run is ITMAX-ITER. So, if one increases ITMAX without increasing ITER, he will get more iterations than he wants or else ESTIME will be exceeded.

IFILE = 0 for card readin, card punchout of restart PHED data.
≠ 0 for tape readin, tape readout of restart PHED data.
A given input tape may have several files or PHED-arrays.[5]
The value assigned IFILE for reading that tape may be determined as follows:
IFILE = R - S
where R = the number representing the position on the tape of the desired file.
S = the number representing the position on the tape of the file read in by the preceding case (has the value zero for the first case of a run).
For example, if the first case of a new run should start from the PHED-array of the first file, then IFILE = 1 - 0 = 1. If the second case must then use the fourth file, IFILE = 4 - 1 = 3. Again, if the third case should use the sixth file, IFILE = 6 - 4 = 2. When the initial PHED-array is set up under control of INISIG, card group 2, IFILE must either be 0 to produce punchcard restart or any number other than 0 to produce tape restart.

IPSIG = 1 to print PHED-distribution with initialization data.
≠ 1 to suppress print of PHED-distribution during initialization.

ILSIG = 1 to print PHED-distribution at end of case.
≠ 1 to suppress print of PHED-distribution at end of case.

KTABLE = 1 to retain h-K table from the immediately preceding case of the same run for use in processing a new case.
≠ 1 to read new h-K table before processing a case.
KTABLE must be 0 or some value other than 1 for first case of run.

MCHNGE = 0 means that restart data have not been modified in any way after they were punched at the end of the preceding run, so that boundary conditions in the deck are compatible with the pressure head distribution of the deck.
≠ 0 means that restart data to be read in have been processed through program CARRY or modified in some other way such that boundary conditions must be set during initialization of the new run. MCHNGE may also be given a value other than 0 for a normally restarting run when the user wants to print the starting distribution of K-values. See KHPRNT, card group 6.
MCHNGE has no meaning for a run not starting with a restart PHED-array.

Card Group 2 - A single card. Must be present in input deck for processing each case. Format (2F10.2,F5.2,5I5,2F10.2)

LGTH - Perpendicular distance from y-axis to the rightmost boundary of the flow system. See page 6 Solution Mesh and the Cartesian Coordinate System.

DEPTH - Perpendicular distance from x-axis to the lowest boundary of the flow system.

SLOPE - Tangent of the angle between the x-axis and the horizontal.

IMGTOP = 1 if any part of top cross section boundary coincident with x-axis is impermeable or is subject to a non-zero flux.
≠ 1 if all parts of the top boundary are subject to a specified pressure head.

IMGBOT = 1 if lower cross section boundary (or its lowest segment, if complex) is impermeable.
≠ 1 otherwise.

IMGLSD = 1 if any part of left cross section boundary coincident with the y-axis is impermeable.
≠ 1 otherwise.

IMGRSD = 1 if right cross section boundary (or its rightmost segment, if complex) is impermeable.
≠ 1 otherwise.

INISIG = 0 to generate initial PHED-array in which PHED = PHEDS at every node in solution mesh, except those on h-specified boundaries.
≠ 0 to generate initial PHED-array which is smoothly distributed in the y-direction beginning with -ELEV on the top boundary and decreasing by increments of Δy from row to row.

[5] These are program-defined files and are not to be confused with logical tape files. A given computer run may produce several program-defined files in one logical file, depending on the number of cases processed, the number of segments in a case, and whether IDBLE = 1 for one or more cases. Logical files used by a given run are identified in job control cards. At the end of a run, a logical file is terminated with an end-of-file mark on the tape. The program-defined files are not so terminated, but their limits are defined by the DIMENSION statement.

PHEDS - The beginning PHED-value for each node in the solution mesh except those on h-specified boundaries when beginning the processing of a new case. Has no meaning for restarts. Has no meaning if INISIG = 1.

ELEV - Elevation above a datum of the origin of coordinates of the Cartesian coordinate system. It is often convenient to set the datum at the lower left-hand corner of the cross section, but its position may be completely arbitrary. See INISIG.

Card Group 3 - A single card. Must be present in input deck for processing each case. Format (2I5,F10.2,7I5)

ITMAX - ITMAX-ITER is the number of iterations to be processed during a given run. When this number is reached, if KARPCH = 1 (card group 1), restart data are obtained in cards or on tape. This is the best way to stop the processing of a case. Also see the definition for IDBLE below. If the case is segmented to try different values of OMEGA, this is the number of iterations allotted to the first OMEGA value. The value given ITMAX should be a multiple of that given INTPRT. When starting a new case or when making such major changes in input data as changing Δx and Δy, ITMAX should be given a small value, say from 1 to 3, to obtain just enough iterations to make sure no errors have been made in the input data. For such a run, the time limit on the job control card should be set at only 5 seconds or so to assure that an input error that throws the program into an endless loop does not result in the use of excessive computer time.

INTPRT - The number of iterations between printouts of pressure heads for selected nodes. Time checks (for comparison with ESTIME, card group 1) and iteration checks (for comparison with ITMAX) take place only after such printouts. Even if no nodes are selected for printing (see NNODES), INTPRT must be given a value to control the frequency of checking ESTIME and ITMAX.

OMEGA - The overrelaxation factor (ω). If the case is segmented and several ω-values tested, this is the initial ω-value.

NOMEGA = 0 for normal, unsegmented case.
$\neq$ 0 for case segmented for changing relaxation factor (ω).
When KARPCH = 1, NOMEGA $\neq$ 0 results in a program-defined file of PHED data in cards or magnetic tape for each value of ω.

NNODES - The number of nodes selected for printing under control of INTPRT. May range from 0 to 8.

IDBLE = 1 means that two sets of restart data 15 iterations apart are wanted for convergence checking by program COMPAR. In this event, 15 iterations in excess of the given ITMAX are processed. KARPCH must have the value 1.
$\neq$ 1 means that case stops at the end of ITMAX iterations with only one restart data set (the latter is obtained only if KARPCH = 1).

NCARDY - The maximum value of MY, that is, the number of pairs of DYLGTH(MY), DELY(MY). See card group 9.

NCARDX - The maximum value of MX, that is, the number of pairs of DXLGTH(MX), DELX(MX). See card group 10.

JGEOM - The number of row subsections into which a flow region has been divided. See card group 11.

IGEOM - The number of column subsections into which a flow region has been divided. See card group 12.

Card Group 4 - An optional card group which must be included if NNODES (card group 3) is different from 0. Consists of one or two cards, depending on the number of nodes selected for printing under control of INTPRT. Format (8F10.2)—up to four pairs of the following variables per card. Unneeded fields may be blank.

COORDI(K) - The x-coordinate of the Kth node selected for printing under control of INTPRT.

COORDJ(K) - The y-coordinate (positive down) of the Kth node selected for printing under control of INTPRT.

NOTE: For use in the model, the above must be converted to I,J coordinates. If a point identified by its x,y coordinates as given by the above variables does not coincide with a node of the solution mesh, the nearest node to the right and below the x,y position will be used.

Card Group 5 - Five cards, even if some or all of them are blank, must be in input deck for processing each case. Format (20A4)

COMENT - Variable representing the string of alphanumeric characters that serve as an identification and heading for a case's printout.

Card Group 6 - A single card that must be present in the input deck for processing each case. Format (2I5)

LUNITS - The number of soil units present in the cross section. May be any number up to and including 5.

KHPRNT = 1 if the starting K-distribution is printed.
$\neq$ 1 otherwise.
NOTE: The facility for printing the K-distribution is provided as a useful way to check whether soil unit boundaries have been assigned correctly in the solution mesh. The user should set INISIG = 0 and PHEDS $\geq$ 0.00 (both in card group 2). This will specify a starting condition of $h \geq 0$ at all nodes so that the printed K-distribution will contain only saturated hydraulic conductivities. With such a distribution, it is relatively easy to correlate nodes and soil units. A single iteration is all that is needed to make this check. Then the case may be started over again with whatever initial PHED-distribution is most appropriate to the case. The initial K-distribution will contain meaningless data if KAREAD = 1 (if the case is being restarted), unless MCHNGE has some value other than 0.

Card Group 7 - A single card which must be present in the input deck for processing each case. Format (10I5)—up to five pairs of the following variables. Unneeded fields may be blank.

NUMLIN(NS) - Number of pairs of values of pressure head vs. hydraulic conductivity appearing in the NSth h-K table.
= 999 indicates that h-K relation is given by equation rather than by table.

NUMBRK(NS) - Number of x,y coordinate pairs needed to describe the lower boundary of the NSth soil unit. See XBRK(K), YBRK(K), card group 8. NUMBRK(NS) has a minimum value of 2 for each soil unit.

Card Group 8 - A multiple card group. The input deck must contain a separate card group 8 for each soil unit in the modeled cross section.

Subset a - One or two cards depending upon the number of x,y coordinate pairs needed to describe the lower boundary of a soil unit. Format (8F10.2)—up to four pairs per card of the following variables. Unneeded fields may be blank.

XBRK(N,NS) - The x-coordinate of the Nth breakpoint in the bottom boundary of the NSth soil unit. Such a boundary, which may be curving or complex in shape, may be approximated by a series of up to seven straight lines, the meeting point between two consecutive segments of different slope being called a breakpoint. An intersection of the lower boundary of a soil unit with a boundary of the modeled cross section is also considered a breakpoint. So, each soil unit has a minimum of two lower-boundary breakpoints.

YBRK(N,NS) - The y-coordinate (positive down) of the Nth breakpoint in the bottom boundary of the NSth soil unit.

NOTE: The note appended to card group 4 applies to this subset also.

Subset b - One or more cards depending on NUMLIN(NS) (card group 7). This subset is absent from the input deck if NUMLIN(NS) = 999. It is also omitted if KTABLE = 1 (card group 1). Format (8E10.3)—eight values of the following variable per card—unneeded fields in the last card may be blank.

PTAB(IT,NS) - The array of pressure heads in the table of pressure head versus hydraulic conductivity for soil unit NS. Subscript IT identifies the particular line of the table, that is, $1 \leq$ IT $\leq$ NUMLIN(NS). The order of entry should be from high pressure to low (in the order of increasing suction).

Subset c - One or more cards depending on NUMLIN(NS) (card group 7). This subset is absent from the input deck if NUMLIN(NS) = 999. It is also omitted if KTABLE = 1 (card group 1). Format (8E10.3) - eight values of the following variable per card--unneeded fields in the last card may be blank.

KTAB(IT,NS) - The array of hydraulic conductivities in the table of pressure head versus hydraulic conductivity for soil unit NS. See discussion of subset b. There should be the same number of KTAB entries as there are PTAB entries. KTAB(1,NS) should be the saturated value of K, corresponding to h = 0 = PTAB(1,NS). KTAB(2,NS) should be the K-value corresponding to PTAB(2,NS) and so on.

Card Group 9 - One to three cards depending on the number of changes in Δy in the solution mesh. Must be present in input deck for processing each case. Format (4(F10.2,F10.3))—up to four pairs of the following variables per card—unneeded fields in the last card may be blank.

DYLGTH(MY) - The distance from the x-axis to the MYth boundary between regions of different Δy in the solution mesh. The x-axis itself is the first such boundary, so DYLGTH(1) = 0.00. If Δy is constant throughout the solution mesh, only DYLGTH(1) is needed. One does not need to measure DYLGTH values precisely. If DYLGTH(MY) does not correspond exactly to the J-value of some row of the solution mesh, then Δy in that mesh will change at the row immediately above the indicated position.

DELY(MY) - The MYth value of Δy.

Card Group 10 - One to three cards depending on the number of changes in Δx in the solution mesh. Must be present in input deck for processing each case. Format (4(F10.2,F10.3))—up to four pairs of the following variables per card—unneeded fields in the last card may be blank.

DXLGTH(MX) - The distance from the y-axis to the MXth boundary between regions of different Δx in the solution mesh. The y-axis itself is the first such boundary, so DXLGTH(1) = 0. If Δx is constant throughout the solution mesh, only DXLGTH(1) is needed. One does not need to measure DXLGTH values exactly. If DXLGTH(MX) does not correspond exactly to the I-value of some column of the solution mesh, then Δx in that mesh will change at the column immediately to the left of the indicated position.

DELX(MX) - The MXth value of Δx.

Card Group 11 - One card for each row subsection. Must be present in input deck for processing each case. Format (4F10.2,2I5,2F10.2)—one set of the following variables per card.

STARTY(NSUBY) - The distance from the x-axis to the top boundary row of row subsection NSUBY. NSUBY is an index variable taking the values 1, 2, . . . , JGEOM. See page 8, Subdivision of Flow Cross Section and the note below for details of dividing the cross section into subsections.

STOPY(NSUBY) - The distance from the x-axis to the bottom boundary row of row subsection NSUBY.

BEGX(NSUBY) - The distance from the y-axis to the left-hand boundary column of row subsection NSUBY.

ENDX(NSUBY) - The distance from the y-axis to the right-hand boundary column of row subsection NSUBY.

JBETA(NSUBY) - Signals type of boundary condition at the beginning of rows in subsection NSUBY.
- = 0 where h is known and has the same value everywhere on the boundary. An exception to the condition of h being equal everywhere on the boundary is discussed on page 25.
- = 1 for impervious boundary.
- = 2 where the h-distribution along the boundary is hydrostatic.

JETA(NSUBY) - Signals type of boundary condition at the ends of rows in subsection NSUBY.
- = 0 where h is known and has the same value everywhere on the boundary. See JBETA(NSUBY) for an exception.
- = 1 for impervious boundary.
- = 2 where the h-distribution along the boundary is hydrostatic.

BCLJ(NSUBY) - The pressure head for an h-specified boundary at the left end of the rows in subsection NSUBY. If h is distributed hydrostatically, this is the value of h at the left end of the top row of subsection NSUBY. Has no meaning for an impervious boundary or for the pressure head boundary discussed on page 25.

BCRJ(NSUBY) - The pressure head for an h-specified boundary at the right end of the rows in subsection NSUBY. If h is distributed hydrostatically, this is the value of h at the right end of the top row of subsection NSUBY. Has no meaning for an impervious boundary or for the pressure head boundary discussed on page 25.

NOTE: Before being used for program control in a computer, the first two variables on this card are converted to the J-value of the top and bottom boundary rows of subsection NSUBY. The next pair of variables is converted to the I-value of the left and right boundary columns of the same subsection.

Consider two row subsections, one of which lies immediately above the other. At the point where the two subsections connect, there is either a geometrical change or a boundary condition change. In either event, one usually selects Δy-values so that the point of change coincides with a row of nodes. Such a row is in a fixed position relative to the x-axis, though its J-value will change if Δy between the x-axis and the row changes. Other rows in the vicinity are subject to change both in position and in J-value when Δy changes. The row of fixed position will either be the bottom boundary of the upper subsection or the top boundary of the lower. Thus, the floating row immediately below it or above it, respectively, will be a boundary of the other subsection. These floating subsection boundaries are also encountered next to boundaries on which pressure heads are specified because such boundaries are fixed but are not included in cross sections.

Because it depends on Δy or Δx, the exact position of a floating column or boundary may be tedious to determine and may also change if mesh increments are changed during the course of solving a given case. Determining them exactly is not necessary, however, if one follows certain precautions. In general, these rules should guide the specification of subsection boundary positions:

1. The position of fixed boundaries should be specified exactly as a distance x or y.
2. For a floating boundary next to a fixed row or column:
   a. If nearer the principal axis than the fixed line, measure to the latter and subtract some quantity that is smaller than the smallest mesh increment likely to be used.

b. If farther from the principal axis than the fixed line, measure to the latter and add some quantity that is smaller than the smallest mesh increment likely to be used.

The small quantities mentioned in 2a and 2b above should not be smaller than 0.001.

If h at one or both ends of a subsection is distributed hydrostatically, one must specify its value(s) for the top row of the subsection. Thus, if that row is a floating boundary, one has to locate it precisely to specify h. STARTY(NSUBX) may be given either precisely or according to the method given in the preceding paragraph, but h must be given its exact value.

Card Group 12 - One card for each column subsection. Must be present in input deck for processing each case. Format (4F10.2,2I5,2F10.2,E10.2)—one set of the following variables per card.

STARTX(NSUBX) - The distance from the y-axis to the left-hand boundary column of column subsection NSUBX. NSUBX is an index variable taking the values 1, 2, . . ., IGEOM. See page 8, Subdivision of Flow Cross Section, and the note at the end of card group 11 for detailed discussion of dividing a cross section into subsections.

STOPX(NSUBX) - The distance from the y-axis to the right-hand boundary column of column subsection NSUBX.

BEGY(NSUBX) - The distance from the x-axis to the top boundary row of column subsection NSUBX.

ENDY(NSUBX) - The distance from the x-axis to the bottom boundary row of column subsection NSUBX.

IBETA(NSUBX) - Signals type of boundary condition at the top of the columns in subsection NSUBX.
- = 0 for known h boundary.
- ≠ 0 for impervious or flux boundary.

IETA(NSUBX) - Signals type of boundary condition at the bottom of the columns in subsection NSUBX.
- = 0 for known h boundary.
- ≠ 0 for impervious boundary.

BCUI(NSUBX) - The pressure head for an h-specified boundary at the tops of columns in subsection NSUBX. Has no meaning for an impervious or flux boundary.

BCBI(NSUBX) - The pressure head for an h-specified boundary at the bottoms of columns in subsection NSUBX. Has no meaning for an impervious boundary.

FLUX(NSUBX) - The flux of water perpendicular to the upper, horizontal surface of column subsection NSUBX. Units should be the same as hydraulic conductivity units. For an impervious surface, FLUX = 0.0. Has no meaning for an h-specified boundary.

Card Group 13 - A multiple card group produced by a previous run for restart purposes. May also be a keypunched initialization deck when a user has some way to closely approximate the solution PHED-array. For a new run for which the user cannot give an approximate PHED-distribution, there is no card group 13. When a magnetic tape is used for restart data, there is no card group 13. Format (6D13.6)--six values of the following variable per card--unneeded fields in the last card may be blank.

PHED(I,J) - The pressure head value at node I,J as it was at the end of the last iteration of the previous run.

NOTE: See discussion of DOUBLE PRECISION mentioned previously under Model Dimensions, page 26. When magnetic tape is used for input/output and when it is anticipated that CARRY might subsequently be used to refine the mesh size, then PHED should be dimensioned for the most refined mesh expected, that is, so that I,J in PHED(I,J) for the DOUBLE PRECISION statement have the largest values they are ever expected to have for the case at hand.

NOTE: For cross sections containing a large number of nodes, using a magnetic tape in place of card group 13 is faster, cheaper, and easier. This calls for the inclusion of tape assignment cards among the job control cards given ahead of the source deck. It also calls for giving IFILE (card group 1) some value besides 0. KAREAD (card group 1), however, must have the value 1 for tape readin as well as for card readin. See the discussion of IFILE, card group 1.

Card Group 14 - A group consisting of two or more cards. Present in the input deck only if the processing of a case is segmented to try different overrelaxation factors (NOMEGA = 1). Format (F5.2,I5)—one set of the following variables per card.

OMEGA - See same variable in card group 2. When given in this card group, OMEGA is the 2d, 3d, 4th, . . . value of ω to be tried while processing the case at hand.

ITMAX - See same variable in card group 2. When given in this card group, ITMAX is the value of the iteration number (accumulating) at which processing using the associated ω-value will stop. ITMAX should have a value that is a multiple of INTPRT.
- = 0 on final card of this group.

Glossary of noninput variables

A - Used in calculation of HEAD.

ALPHA - Represents the angle whose tangent is the value SLOPE (input card group 2).

AX - A term in the finite difference equation.

$$= \frac{K_{i-\frac{1}{2},j}}{\Delta x_-}$$

AY - A term in the finite difference equation.

$$= \frac{K_{i,j-\frac{1}{2}}}{\Delta y_-}$$

B - Used in calculation of HEAD.

BCL(NSUBY) = BCLJ(NSUBY) and used to set hydrostatic boundary condition.

BCR(NSUBY) = BCRJ(NSUBY) and used to set hydrostatic boundary condition.

CHKTM - To set or reset computer clock and to read elapsed time. This variable might not be necessary at other facilities.

COSAL - Cosine of the angle ALPHA.

CX - A term in the finite difference equation.

$$= \frac{K_{i+\frac{1}{2},j}}{\Delta x_+}$$

CY - A term in the finite difference equation.

$$= \frac{K_{i,j+\frac{1}{2}}}{\Delta y_+}$$

DEL - Used in setting impermeable and flux boundary conditions at ends of columns and is the elevation difference between an imaginary node and its real counterpart immediately inside the boundary.

DELK(MY) = DELX(MX) or DELY(MY) (input card groups 10 and 9) introduced so that either could be used in a single algorithm.

DELTA - A term in the finite difference equation.

$$= \left(\frac{K_{i,j-1} - K_{i,j+1}}{\Delta y_- + \Delta y_+}\right) \cos \alpha + \left(\frac{K_{i-1,j} - K_{i+1,j}}{\Delta x_- + \Delta x_+}\right) \sin \alpha$$

DELXM = Δx_-, that is, Δx to the left of a node.

DELXP = Δx_+, that is, Δx to the right of a node.

DELYM = Δy_-, that is, Δy above a node.

DELYP = Δy_+, that is, Δy below a node.

ELEVS - Readin value of ELEV, saved because the latter is modified twice in the program.

ELIJ - Elevation of node I,J; used in determining which soil unit applies at node I,J.

EX - A term in the finite difference equation.

$$= \frac{2}{\Delta x_{-} + \Delta x_{+}}$$

EY - A term in the finite difference equation.

$$= \frac{2}{\Delta y_{-} + \Delta y_{+}}$$

FACTOR - Interpolation factor for calculating hydraulic conductivity when the corresponding pressure head lies between two entries in the table of PTAB versus KTAB.

GRAD - Represents the slope of a straight-line segment in the lower boundary of a soil unit. Used in determining which soil unit applies at each node.

HCON(I,J) - Conductivity for node I,J.

HEAD(I,J) - Hydraulic head at node I,J.

HEDA - Dummy variable that represents PHED(I-1,J) in finite difference equation. Provides vehicle for substituting

$$PHED(I+1,J) + (\Delta x_{-} + \Delta x_{+}) \sin \alpha$$

in equation for left boundary node when that boundary is impermeable.

HEDB - Dummy variable that represents PHED(I+1,J) in finite difference equation. Provides vehicle for substituting

$$PHED(I-1,J) - (\Delta x_{-} + \Delta x_{+}) \sin \alpha$$

in equation for right boundary node when that boundary is impermeable.

I - Column number, I, in the finite difference equation. Also used as a DO loop index.

IBEG(NSUBY) - The I-value of the left-hand boundary column of row subsection NSUBY.

IBRK(N,NS) - The I-coordinate of the Nth breakpoint in the bottom boundary of the NSth soil unit.

ICHK - Represents IMGTOP or IMGLSD (input card group 2) so that either can appear in a single algorithm.

ICT - Counter used in reading restart tape. Enables program to skip over unwanted files of data on a tape produced by a multicase run.

IDUM - Dummy variable that represents other variables where the latter, because of subscripting or because they are of the REAL type, cannot be used in USASI Fortran.

IDUMA - Dummy variable used in the same way as IDUM.

IEND(NSUBY) - The I-value of the right-hand boundary column of row subsection NSUBY.

IJK - Dummy variable that represents a given value of I or J as the starting index of a DO loop, using I or J as an index.

INODE(K) - The I-coordinate of the Kth node selected for printing under control of INTPRT.

IPRINT - Used with INTPRT (input card group 3) in controlling frequency of printing PHED-values for selected nodes and also frequency of checking elapsed time and number of iterations processed.

ISTART(NSUBX) - The I-value of the left-hand boundary column of column subsection NSUBX.

ISTOP(NSUBX) - The I-value of the right-hand boundary column of column subsection NSUBX.

ITMAXS - A storage variable for ITMAX. When changing OMEGA values in a run, each ω is used a number of iterations equal to ITMAX. ITMAXS is set equal to ITMAX at the beginning and increments ITMAX each time a new ω is read.

IX(MX) - The I-value of the MXth column at which Δx changes in value.

J - Row number, J, in the finite difference equation.

JBEG(NSUBX) - The J-value of the top boundary row of column subsection NSUBX.

JBRK(N,NS) - The J-coordinate of the Nth breakpoint in the bottom boundary of the NSth soil unit.

JEND(NSUBX) - The J-value of the bottom boundary row of column subsection NSUBX.

JNODE(K) - The J-coordinate of the Kth node selected for printing control of INTPRT.

JSTART(NSUBY) - The J-value of the top row of row subsection NSUBY.

JSTOP(NSUBY) - The I-value of the bottom row of row subsection NSUBY.

JY(MY) - The J-value of the MYth row at which Δy changes in value.

K - A subscripting index for DO loops.

KAVE - An average value of hydraulic conductivity at the surface of the soil. Used in setting non-saturated flux boundary condition.

KBETA - Unsubscripted representation of JBETA(NSUBY), whose value can be changed during execution. Used in setting boundary condition at left end of row.

KD(MY) - Represents IX or JY so that either may appear in a single algorithm.

KEND(NCT) - Represents ISTOP(NSUBX), JSTOP(NSUBY), IEND(NSUBY), JEND(NSUBX) so that any one of them may appear in a single equation.

KETA - Unsubscripted representation of JETA(NSUBY), whose value can be changed during execution. Sets boundary condition at right end of row.

KSIG - A signal variable; controls flow of program.

KSTART(NCT) - Represents ISTART(NSUBX), JSTART(NSUBY), IBEG(NSUBY), JBEG(NSUBX) so that any one of them may appear in a single equation.

KSTOP - The I-value or J-value of the last column or row, respectively, in a subsection. Replaces subscripted variables as an index in DO loops.

L - Takes the values 1-8 and is used to break the h-K table into eighths for the rapid lookup of HCON(I,J).

LNUM - Signal variable; directs flow of program while printing h-K table.

LSIG - Signal variable; directs flow of program after writing PHED- or HEAD-array.

LSTOP - The I- or J-value of the final (boundary) node at the end of a row or column. Replaces subscripted variables as an index in DO loops.

MCOL - The number of columns, including imaginary columns, in the solution mesh.

MCT - Represents JGEOM or IGEOM so that either may appear in a single algorithm.

MROW - The number of rows, including imaginary rows, in the solution mesh.

MSIG - Signal variable; directs flow of program after setting boundary conditions.

MXMYST - Stores the starting value of MX for a given subsection. Resets MX when starting new rows within the subsection.

NCT - An index; controls certain program loops.

NSIG - A signal variable; controls flow of program.

NUM - Separates the printed pressure head-hydraulic conductivity table into four columns in which PTAB increases down first row first, then down second, and so forth.

NUMA - Used in printing PTAB-KTAB table. Allows changing format when blanks occur in fourth segment of table.

NXCRD - Initial value of NCARDX.

NYCRD - Initial value of NCARDY.

SDUMA - A dummy variable that represents STARTJ(NSUBY), BEGJ(NSUBY), STARTI(NSUBX), and BEGI(NSUBX), so that any one of them may appear in a single algorithm.

SDUMB - A dummy variable used to represent STOPJ(NSUBY), ENDI(NSUBY), STOPI(NSUBX), and ENDJ(NSUBX), so that any one of them may appear in a single algorithm.

SINAL - The sine of ALPHA.

SUMX - Determines the I- and J-values associated with various x and y input measurements.

SUMY - Used in same way as SUMX.

TIME - Accumulated CPU time in seconds. Its value is updated periodically by the internal timing routine (TASKTM) and compared against ESTIME. When TIME exceeds ESTIME, the run is stopped.

TTIME - Converts time obtained from TASKTM to seconds. This variable might not be necessary at some computer facilities.

XA - An average hydraulic conductivity

$$\left(\frac{HCON(I,J) + HCON(I-1,J)}{2}\right)$$

for preparing terms for the finite difference equation.

XB - A term in the finite difference equation.

$$= \frac{\Delta x_+ (K_{i-\frac{1}{2},j}) + \Delta x_- (K_{i+\frac{1}{2},j})}{\Delta x_- \cdot \Delta x_+}$$

XC - An average hydraulic conductivity

$$\left(\frac{HCON(I,J) + HCON(I+1,J)}{2}\right)$$

for preparing terms for the finite difference equation.

XDIST - Represents distance in x-direction from y-axis.

YA - An average hydraulic conductivity

$$\left(\frac{HCON(I,J) + HCON(I,J-1)}{2}\right)$$

for preparing terms for the finite difference equation.

YB - A term in the finite difference equation.

$$= \frac{\Delta y_+ (K_{i,j-\frac{1}{2}}) + \Delta y_- (K_{i,j+\frac{1}{2}})}{\Delta y_- \cdot \Delta y_+}$$

YC - An average hydraulic conductivity

$$\left(\frac{HCON(I,J) + HCON(I,J+1)}{2}\right)$$

for preparing terms for the finite difference equation.

YDIST - Represents distance in y-direction from x-axis.

ZHED - Elevation head used in calculating total hydraulic head and in setting the drained-to-equilibrium initial PHED-array.

Appendix B:

CARRY — To Facilitate Changing Finite Difference Mesh Spacing

Operation of such a finite difference model as STDY2 produces an array of values of the dependent variable, each value being associated with a node of the solution mesh superimposed over the region of interest. The accuracy with which these values represent the true values of the dependent variable at these points depends in large measure upon the mesh spacing chosen before running the model.

After beginning or even finishing a solution for a particular mesh spacing, one may wish to refine this spacing in part or all of the solution mesh and continue running the model for an improved estimate. The results of the previous run provide a good estimated distribution from which to start the improvement run, but a refined mesh will usually contain nodes at positions where values of the dependent variable have not been estimated and may eliminate some nodes of the original mesh. Considerable time and effort would be required to effect the necessary changes by hand.

In terms of the dependent variable of STDY2, the purpose of CARRY is to convert a given PHED-array into another of different mesh spacing. Linear interpolation provides PHED-values for nodes whose positions do not correspond with those of nodes in the original mesh.

Processing imaginary rows and columns on the outer limits of the solution mesh is unnecessary and, for operational reasons, undesirable in CARRY; but providing these rows and columns in CARRY's output deck is necessary for later input to STDY2. To reserve their positions in the card deck or on magnetic tape, a value of 0 is applied at each imaginary node on the outer limits.

For economy of operation, such imaginary and unused nodes within the solution mesh as those within the trench or notch of figures 5 and 6 are processed in the same way as all other nodes. Their new values are of no consequence to STDY2.

This appendix contains (1) a program listing, (2) a glossary of input variables, (3) a glossary of noninput variables, and (4) a sample problem.

A program listing of CARRY follows. Modifications which might be necessary before running the program on other computers are flagged. Their numbering and explanations are the same as for STDY2, appendix A. In general, the variables in the DOUBLE PRECISION and DIMENSION statements should have the same dimensions as in STDY2. For magnetic tape input/output, it is particularly necessary that PHED and PHEDN have the same dimensions as PHED in STDY2.

The logic of this program is straightforward and is readily apparent from inspection of the listing. Therefore, a flow chart is not included.

Glossary of input variables

Input variables are defined in the order of their appearance in the input data deck. Figure 14 shows punchcard layouts for the input data. The manner of presentation is the same as in appendix A. As with STDY2, several cases may be processed during a single computer run.

Card Group 1 - Five cards, even if some are blank. Must be in the input deck for processing each case. Format (20A4)

COMENT - Alphanumeric identification printed at head of output.

Card Group 2 - A single card. Must be present in input deck for processing each case. Format (2F10.2,6I5)

LGTH - Same value as variable of same name in input card group 2 of STDY2.

DEPTH - Same value as variable of same name in input card group 2 of STDY2.

Text continues on page 56.

```
C  CARRY - GIVEN A TWO DIMENSIONAL ARRAY OF VALUES OF SOME QUANTITY AT A
C     GIVEN, NOT NECESSARILY UNIFORM, GRID SPACING. THIS PROGRAM PRODUCES
C     A SECOND ARRAY OF THE SAME QUANTITY DISTRIBUTED OVER THE SAME SIZE
C     AREA AS THE ORIGINAL ARRAY, BUT WITH DIFFERENT, NOT NECESSARILY
C     UNIFORM, GRID SPACING.  OLD AND NEW GRID SPACINGS ARE COMPLETELY
C     INDEPENDENT.
C
C     3/18/74.
C
C
C***** POSSIBLE MODIFICATION TYPE M1.
C
       DOUBLE PRECISION PHED(60.70)
       REAL LGTH
C
C***** POSSIBLE MODIFICATION TYPE M3.
C
       DIMENSION PHEDN(60.70).DELX(12).DELXN(12).DELY(12).DELYN(12).
      1IX(12).IXN(12).JY(12).JYN(12).COMENT(100).DXLGTH(12).DXLGTN(12).
      2DYLGTH(12).DYLGTN(12)
C
C***** POSSIBLE MODIFICATION TYPE M4
C
     1 READ (5.5.END=15) COMENT
C
C***** POSSIBLE MODIFICATION TYPE M3.
C
     5 FORMAT (20A4)
C
C***** POSSIBLE MODIFICATION TYPE M4.
C
       WRITE (6.10) COMENT
C
C***** POSSIBLE MODIFICATION TYPE M3.
C
    10 FORMAT (1H1.20A4/(20A4))
       GO TO 20
    15 STOP
    20 READ (5.25) LGTH.DEPTH.IFILE.ITAPE.NCARDY.NCARDX.NCRDYN.NCRDXN
    25 FORMAT (2F10.2.6I5)
       READ (5.30)IMGTOP.IMGBOT.IMGLSD.IMGRSD
    30 FORMAT (4I5)
       READ (5.35) (DYLGTH(MY).DELY(MY).MY=1.NCARDY)
    35 FORMAT (4(F10.2.F10.3))
       READ (5.35) (DXLGTH(MX).DELX(MX).MX=1.NCARDX)
       READ (5.35) (DYLGTN(MYN).DELYN(MYN).MYN=1.NCRDYN)
       READ (5.35) (DXLGTN(MXN).DELXN(MXN).MXN=1.NCRDXN)
C
C*** CONVERT X.Y COORDINATE DATA TO I J COORDINATES.**********
C
       NYCRD = NCARDY
       NXCRD = NCARDX
       NYCRDN = NCRDYN
       NXCRDN = NCRDXN
       NCARDY = NCARDY +1
       NCARDX = NCARDX + 1
       NCRDYN = NCRDYN + 1
       NCRDXN = NCRDXN + 1
       DYLGTH(NCARDY) = DEPTH
       DYLGTN(NCRDYN) = DEPTH
       DXLGTH(NCARDX) = LGTH
       DXLGTN(NCRDXN) = LGTH
       DELY(NCARDY) = DELY(NCARDY-1)
       DELX(NCARDX) = DELX(NCARDX-1)
       DELYN(NCRDYN) = DELYN(NCRDYN-1)
       DELXN(NCRDXN) = DELXN(NCRDXN-1)
       IF (IMGTOP.EQ.1) GO TO 40
       JY(1) = 1
       JYN(1) = 1
       GO TO 45
    40 JY(1) = 2
       JYN(1) = 2
    45 DO 50 MY = 2.NCARDY
          IDUM = (DYLGTH(MY) -DYLGTH(MY-1)) / DELY(MY-1)
    50    JY(MY) = JY(MY-1) + IDUM
       DO 55 MYN = 2.NCRDYN
          IDUM = (DYLGTN(MYN) - DYLGTN(MYN-1)) /DELYN(MYN-1)
    55    JYN(MYN) = JYN(MYN-1) + IDUM
       IF (IMGBOT.NE.1) GO TO 60
       JY(NCARDY) = JY(NCARDY) + 1
       JYN(NCRDYN) = JYN(NCRDYN) + 1
    60 MROW = JY(NCARDY)
       MROWN = JYN(NCRDYN)
       IF (IMGLSD.EQ.1) GO TO 65
       IX(1) = 1
       IXN(1) = 1
       GO TO 70
    65 IX(1) = 2
       IXN(1) = 2
    70 DO 75 MX = 2.NCARDX
          IDUM = (DXLGTH(MX) - DXLGTH(MX-1)) / DELX(MX-1)
    75    IX(MX) = IX(MX-1) + IDUM
```

```
      DO 80 MXN = 2.NCRDXN
          IDUM = (DXLGTN(MXN) - DXLGTN(MXN-1)) / DELXN(MXN-1)
   80     IXN(MXN) = IXN(MXN-1) + IDUM
      IF(INGRSD.NE.1) GO TO 85
      IX(NCARDX) = IX(NCARDX) + 1
      IXN(NCRDXN) = IXN(NCRDXN) + 1
   85 MCOL = IX(NCARDX)
      MCOLN = IXN(NCRDXN)
C
C*** PRINT HEADING AND INPUT DATA. *********
C
      WRITE (6.90)
   90 FORMAT (1H0 3X.4HLGTH 3X.5HDEPTH 3X.5HIFILE 3X.5HITAPE 3X.6HNCARDY
     1 3X.6HNCARDX 3X.6HNCRDYN 3X.6HNCRDXN    )
      WRITE (6.95) LGTH.DEPTH.IFILE.ITAPE.NYCRD.NXCRD.NYCRDN.NXCRDN
   95 FORMAT (1H 2F8.2.I8.I8.4I9)
      WRITE (6.100)
  100 FORMAT (1H0.4HMCOL. 3X.4HMROW 3X.5HMCOLN 3X.5HMROWN    )
      WRITE (6.105)MCOL.MROW.MCOLN.MROWN
  105 FORMAT (1H 14.17.218)
      WRITE (6.110)
  110 FORMAT (1H0 6HIMGTOP 3X.6HIMGBOT 3X.6HIMGLSD 3X.6HINGRSD    )
      WRITE (6.115)IMGTOP.IMGBOT.IMGLSD.IMGRSD
  115 FORMAT (1H 14.319)
      WRITE (6.120)
  120 FORMAT (1H0 (55HVARIABLE MESH INCREMENT DATA FOLLOW IN TRIPLETS AS
     1 XYZ    )/(5X.73HWHERE X = MEASURED DISTANCE FROM AXIS (DXLGTH. DYL
     2GTH. DXLGTN OR DYLGTN)    )/(11X.46HY = ROW OR COLUMN NUMBER (IX. J
     3Y. IXN OR JYN)    )/(11X.50HZ = INCREMENT LENGTH (DELX. DELY. DELXN
     4 OR DELYN)    )/(13H OLD VERTICAL    ))
      WRITE (6.125) (DYLGTH(K).JY(K).DELY(K).K=1.NCARDY)
  125 FORMAT (1H 4(F8.2.I5.F8.3.8X))
      WRITE (6.130)
  130 FORMAT (1H 15HOLD HORIZONTAL    )
      WRITE (6.125)(DXLGTH(K).IX(K).DELX(K).K=1.NCARDX)
      WRITE (6.135)
  135 FORMAT (1H 13HNEW VERTICAL    )
      WRITE (6.125)(DYLGTN(K).JYN(K).DELYN(K).K=1.NCRDYN)
      WRITE (6.140)
  140 FORMAT (1H 15HNEW HORIZONTAL    )
      WRITE (6.125)(DXLGTN(K).IXN(K).DELXN(K).K=1.NCRDXN)
C
C*** READ AND PRINT OLD (INPUT) ARRAY. **********
C
      IF (IFILE.EQ.0) GO TO 150
C***** POSSIBLE MODIFICATION TYPE M5.
      DO 145 ICT = 1.IFILE
  145     READ(9) PHED
      GO TO 160
  150 READ (5.155) ((PHED(I.J).I=1.MCOL).J=1.MROW)
  155 FORMAT (6D13.6)
  160 WRITE (6.165)
  165 FORMAT (1H0 49X.23HOLD PRESSURE HEAD ARRAY    )
      DO 170 J = 1.MROW
  170     WRITE (6.175) J.(PHED(I.J).I=1.MCOL)
  175 FORMAT (1H I3.2X.10D12.5/(6X.10D12.5))
      DO 180 J = 1.MROW
  180     PHED(MCOL+1.J) = 0.
      DO 185 I = 1.MCOLN
  185     PHEON(I.MROW+1) = 0.
C
C*** SWEEP ROWS. INTERPOLATING NEW COLUMNS OR COPYING OR DELETING OLD
C           COLUMNS AS REQUIRED. **********
C
      IF (IMGTOP.EQ.1) GO TO 190
      IJK = 1
      GO TO 195
  190 IJK = 2
  195 DO 230 J = IJK.MROW
        MX = 1
        IDUMA = 2
        MXN = 1
        IDUM = 2
        IF (IMGLSD.EQ.1) GO TO 200
        I = 1
        IN = 1
        GO TO 205
  200   I = 2
        IN = 2
  205   XPOSA = 0.
        XPOSN = 0.
        XPOSB = XPOSA + DELX(1)
  210   PHEON(IN.J) = PHED(I.J) + ((XPOSN - XPOSA) / (XPOSB - XPOSA)) *
     1  (PHED(I+1.J) - PHED(I.J))
        IN = IN + 1
        IF (IN.LE.IXN(IDUM)) GO TO 215
        MXN = MXN + 1
        IDUM = MXN + 1
        IF (MXN.GE.NCRDXN) GO TO 230
  215   XPOSN = XPOSN + DELXN(MXN)
        IXPOSB = (XPOSB + .000005) * 10000
        IXPOSN = (XPOSN + .000005) * 10000
        IF (IXPOSN.LT.IXPOSB) GO TO 210
  220   I = I + 1
```

```
          IF ((.LT.IX(IDUMA)) GO TO 225
          MX = MX + 1
          IDUMA = MX + 1
    225   XPOSA = XPOSB
          XPOSB = XPOSB + DELX(MX)
          IXPOSB = (XPOSB + .000005) * 10000
          IF (IXPOSN.GE.IXPOSB) GO TO 220
          GO TO 210
    230     CONTINUE
C
C*** SWEEP COLUMNS INTERPOLATING NEW ROWS OR COPYING OR DELETING OLD
C     ROWS AS REQUIRED.**********
C
        IF (IMGLSD.EQ.1) GO TO 235
        IJK = 1
        K = MCOLN
        GO TO 240
    235 IJK = 2
    240 DO 275 I = IJK.MCOLN
          MY = 1
          IDUMA = 2
          MYN = 1
          IDUM = 2
          IF (IMGTOP.EQ.1) GO TO 245
          J = 1
          JN = 1
          GO TO 250
    245   J = 2
          JN = 2
    250   YPOSA = 0.
          YPOSN = 0.
          YPOSB = YPOSA + DELY(1)
    255   PHED(I.JN) = PHEON(I.J) + ((YPOSN - YPOSA) / (YPOSB - YPOSA)) *
       1  (PHEON(I.J+1) - PHEDN(I.J))
C
C*** SET IMAGINARY NODES. IF ANY. ON OUTERMOST ROWS AND COLUMNS TO ZERO.
C     **********.
C
          JN = JN + 1
          IF (JN.LE.JYN(IDUM)) GO TO 260
          MYN = MYN + 1
          IDUM = MYN + 1
          IF (MYN.GE.NCRDYN) GO TO 275
    260   YPOSN = YPOSN + DELYN(MYN)
          IYPOSB = (YPOSB + .000005) * 10000
          IYPOSN = (YPOSN + .000005) * 10000
          IF (IYPOSN.LT.IYPOSB) GO TO 255
    265   J = J + 1
          IF (J.LT.JY(IDUMA)) GO TO 270
          MY = MY + 1
          IDUMA = MY + 1
    270   YPOSA = YPOSB
          YPOSB = YPOSB + DELY(MY)
          IYPOSB = (YPOSB + .000005) * 10000
          IF (IYPOSN.GE.IYPOSB) GO TO 265
          GO TO 255
    275     CONTINUE
        IF (IMGRSD.NE.1) GO TO 285
        DO 280 J = 1.MROWN
    280   PHED(MCOLN.J) = 0.
    285 IF (IMGBOT.NE.1) GO TO 295
        DO 290 I = 1.MCCLN
    290   PHED(I.MROWN) = 0.
    295 IF (IMGLSD.NE.1) GO TO 305
        DO 300 J = 1.MROWN
    300   PHED(1,J) = 0.
    305 IF (IMGTOP.NE.1) GO TO 315
        DO 310 I = 1.MCCLN
    310   PHED(I.1) = 0.
C
C*** PRINT AND PUNCH (OR WRITE ON TAPE) THE NEW (OUTPUT) ARRAY.**********
C
    315 WRITE (6.320)
    320 FORMAT (1H0 49X.23HNEW PRESSURE HEAD ARRAY    )
        DO 325 J = 1.MROWN
    325     WRITE (6.176) J.(PHED(I.J).I=1.MCOLN)
        IF (ITAPE.EQ.0) GO TO 335
C
C***** POSSIBLE MODIFICATION TYPE M5.
C
        WRITE (10) PHED
        WRITE (6.330)
    330 FORMAT (1H0 13HTAPE WRITTEN.    )
        GO TO 1
C
C***** POSSIBLE MODIFICATION TYPE M4.
C
    335 WRITE (7.155) ((PHED(I.J).I=1,MCOLN).J=1.MROWN)
        WRITE (7.340)
    340 FORMAT (16H*****END OF FILE    )
        WRITE (6.345)
    345 FORMAT (1H0 14HCARDS PUNCHED.    )
        GO TO 1
        END
```

Figure 14.--Input data layout for CARRY.

USDA-FOREST SERVICE **MULTIPLE-CARD LAYOUT FORM**	PROGRAM NAME: CARRY		PROGRAM NUMBER:
	FORMAT DESIGNED BY:	DATE: 3/74	PAGE: 1 OF 2

CARD IDENT.

Format 5 Card Group 1

COMENT (columns 1–80)
20A4
CARRY CONVERTING SMALL-SCALE SEPTIC TANK (6 x 7 CM) RESTART

Format 25 Card Group 2

LGTH (1–10)	DEPTH (11–20)	IFILE (21–25)	ITAPE (26–30)	NCARDY (31–35)	NCARDX (36–40)	NCRDYN (41–45)	NCRDXN (46–50)	Blank (51–80)
F10.2	F10.2	I5	I5	I5	I5	I5	I5	
6.00	7.00	0	1	1	1	3	3	

Format 30 Card Group 3

IMGTOP (1–5)	IMGBOT (6–10)	IMGLSD (11–15)	IMGRSD (16–20)	Blank (21–80)
I5	I5	I5	I5	
1	0	1	1	

Format 35 Card Group 4

DYLGTH(1) (1–10)	DELY(1) (11–20)	DYLGTH(2) (21–30)	DELY(2) (31–40)	DYLGTH(3) (41–50)	DELY(3) (51–60)	DYLGTH(4) (61–70)	DELY(4) (71–80)
F10.2	F10.3	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3
0.00	1.000						

GPO 916-687 6600-1 (11/66)

Figure 14.--Continued.

USDA-FOREST SERVICE MULTIPLE-CARD LAYOUT FORM	PROGRAM NAME: CARRY		PROGRAM NUMBER:
	FORMAT DESIGNED BY:	DATE: 3/74	PAGE: 2 OF 2

CARD IDENT.	Columns 1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80
Format 35 Card Group 5	DXLGTH(1)	DELX(1)	DXLGTH(2)	DELX(2)	DXLGTH(3)	DELX(3)	DXLGTH(4)	DELX(4)
	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3
	0.00	1.000						
Format 35 Card Group 6	DYLGTN(1)	DELYN(1)	DYLGTN(2)	DELYN(2)	DYLGTN(3)	DELYN(3)	DYLGTN(4)	DELYN(4)
	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3
	0.00	1.000	1.00	0.500	5.00	1.000		
Format 35 Card Group 7	DXLGTN(1)	DELXN(1)	DXLGTN(2)	DELXN(2)	DXLGTN(3)	DELXN(3)	DXLGTN(4)	DELXN(4)
	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3
	0.00	1.000	2.00	0.500	5.00	1.000		

CARD IDENT.	Columns 1-13	14-26	27-39	40-52	53-65	66-78	79-80
Format 155 Card Group 8	PHED(1,1)	PHED(2,1)	PHED(3,1)	PHED(4,1)	PHED(5,1)	PHED(6,1)	Blank
	D13.6	D13.6	D13.6	D13.6	D13.6	D13.6	
	-0.800000D+01	-0.271098D+02	-0.271040D+02	-0.270635D+02	-0.268273D+02	-0.254916D+02	

GPO 916-687

6600-1 (11/66)

IFILE = 0 when input array is in card form and output array is in card form.
> 0 when input is read from tape and output is written on tape. The value indicates the position of the input file on a multifile tape. See explanation of same variable in STDY2, card group 1, for instructions regarding determination of position.

ITAPE = 0 when output is on cards.
≠ 0 when output is on tape.

NCARDY - Same value as variable of same name in card group 3 of STDY2.

NCARDX - Same value as variable of same name in card group 3 of STDY2.

NCRDYN - The number of DYLGTN-DELYN pairs in card group 6 of this program.

NCRDXN - The number of DXLGTN-DELXN pairs in card group 7 of this program.

Card Group 3 - A single card. Must be present in input deck for processing each case. Format (4I5)

IMGTOP - Same value as variable of same name in input card group 2 of STDY2.

IMGBOT - Same value as variable of same name in input card group 2 of STDY2.

IMGLSD - Same value as variable of same name in input card group 2 of STDY2.

IMGRSD - Same value as variable of same name in input card group 2 of STDY2.

Card Group 4 - One to three cards depending on the number of changes of Δy in the old (input) solution mesh. Must be present in input deck for processing each case. Format (4(F10.2,F10.3))

For a given case, this card group is identical both in variable names and in values to card group 9 of the input deck for STDY2.

Card Group 5 - One to three cards depending on the number of changes in Δx in the old (input) solution mesh. Must be present in input deck for processing each case. Format (4(F10.2,F10.3))

For a given case, this card group is identical both in variable names and in values to card group 10 of the input deck for STDY2.

Card Group 6 - One to three cards depending on the number of changes in Δy in the new (output) solution mesh. Must be present in input deck for processing each case. Format (4(F10.2,F10.3))—four pairs of the following variables per card—unneeded fields in the last card may be blank.

DYLGTN(MYN) - The distance from the x-axis to the MYNth boundary between regions of different Δy in the new (output) solution mesh.

DELYN(MYN) - The MYNth value of Δy in the new (output) mesh.

NOTE: When the new (output) deck has been obtained from CARRY, this card group may be substituted directly into STDY2's input deck as card group 9. The values in the cards are then equated to DYLGTH(MY) and DELY(MY).

Card Group 7 - One to three cards depending on the number of changes in Δx in the new (output) solution mesh. Must be present in input deck for processing each case. Format (4(F10.2,F10.3))—four pairs of the following variables per card—unneeded fields in the last card may be blank.

DXLGTN(MXN) - The distance from the y-axis to the MXNth boundary between regions of different Δx in the new (output) solution mesh.

DELXN(MXN) - The MXNth value of Δx in the new (output) mesh.

NOTE: When the new (output) deck has been obtained from CARRY, this card group may be substituted directly into STDY2's input deck as card group 10. The values in the cards are then equated to DXLGTH(MX) and DELX(MX).

Card Group 8 - Group of several cards, the number being dependent upon the number of nodes in the solution mesh. When magnetic tape is used for input, there is no card group 8. Format (6D13.6)—six values of the following variable per card—unneeded fields in the last card may be blank.

PHED(I,J) - The value of pressure head at the node at the intersection of the Ith column and Jth row of the old (input) mesh.

See PHEDN(I,J) in glossary of noninput variables for use of PHED-array after input is complete.

NOTE: When magnetic tape is used for input/output, be sure that PHED(I,J) and PHEDN(I,J) in CARRY are dimensioned exactly as PHED(I,J) is in STDY2. When punch cards are used, PHED(I,J) and PHEDN(I,J) must have dimensions at least as large as those needed for the new (output) mesh. Dimensioning is specified in the DOUBLE PRECISION and DIMENSION statements.

Glossary of noninput variables

I - Column number, i, in the solution mesh.

ICT - Counter used in reading restart tape. Enables program to skip over unwanted files of data on a tape produced by a multicase run.

IDUM - Dummy variable; represents other variables where the latter, because of subscripting or because they are of the REAL type, cannot be used.

IDUMA - Dummy variable used in same way as IDUM.

IJK - Dummy variable representing the I- or J-value with which to start a DO loop.

IN - Index variable used in place of I to represent column position when setting up the output array.

IX(MX) - The I-value of the MXth column in the old (input) mesh at which Δx changes in value.

IXN(MXN) - The I-value of the MXNth column in the new (output) mesh at which Δx changes in value.

IXPOSB, IXPOSN - Integer representations of XPOSB and XPOSN after adding .000001 and multiplying by 10,000. Necessary for comparing equality because real number comparisons are not reliable.

IYPOSB, IYPOSN - Integer representations of YPOSB and YPOSN. See IXPOSB and IXPOSN.

J - Row number, j, in the solution mesh.

JN - Index variable used in place of J to represent row position when setting up the output array.

JY(MY) - The J-value of the MYth row in the old (input) mesh at which Δy changes in value.

JYN(MYN) - The J-value of the MYNth row in the new (output) mesh at which Δy changes in value.

K - A DO loop index.

MCOL - The number of columns, including imaginary columns, in the old (input) solution mesh.

MCOLN - The number of columns, including imaginary columns, in the new (output) solution mesh.

MROW - The number of rows, including imaginary rows, in the old (input) solution mesh.

MROWN - The number of rows, including imaginary rows, in the new (output) solution mesh.

NXCRD - Storage variable representing the input variable NCARDX (card group 2).

NXCRDN - Storage variable representing the input variable NCRDXN (card group 2).

NYCRD - Storage variable representing the input variable NCARDY (card group 2).

NYCRDN - Storage variable representing the input variable NCRDYN (card group 2).

PHEDN(I,J)—The value of pressure head at the (I,J)th node in an intermediate mesh. Each row of PHED(I,J) is swept from left to right so that PHED-values at nodes at common distances from the left boundary are copied into PHEDN. Values at inserted nodes are interpolated linearly and also entered into PHEDN at the proper distance from the left boundary. When all rows are swept, PHEDN contains an array with the number of columns to be contained in the output array but with the number of rows contained in the input array.

The columns in PHEDN are then swept from top to bottom, reading values at common distances from the top back into PHED and interpolating values for inserted nodes. When column sweeping is finished, PHED contains the output array with the desired number of columns and rows.

XPOSA, XPOSN, XPOSB - Distances from the y-axis.

A node in the new array (PHEDN) with coordinates (IN,J) may fall between two nodes in the old (input) array with coordinates (I,J) and (I+1,J).

XPOSA gives the distance to (I,J)
XPOSN gives the distance to (IN,J)
XPOSB gives the distance to (I+1,J)

These form the basis for interpolating PHEDN(IN,J) between PHED(I,J) and PHED(I+1,J).

YPOSA, YPOSN, YPOSB - Distances from the x-axis.

See XPOSA, XPOSN, XPOSB description.

YPOSA gives the distance to (I,J)
YPOSN gives the distance to (I,JN)
YPOSB gives the distance to (I,J+1)

These form the basis for interpolating PHED(I,JN) between PHEDN(I,J) and PHEDN(I,J+1).

Sample problem

Figure 15 shows the cross section of figure 4 with a superimposed uniform mesh of 1-cm spacing. Figure 16 shows sample data for converting the PHED-array yielded by a STDY2 solution for that spacing to an array with the mesh spacing of figures 8 and 9. Figure 17 shows the printout for the sample run, including the new array of PHED-values.

Discussion of the STDY2 sample problem in the text noted that node (2,2) seemed to converge most slowly. Taking its converged value, −23.407 cm, as an index of comparison, the STDY2 sample problem converged to within 99 percent of that value in about 80 iterations with ω = 1.60. Conceivably, an investigator might have approached the septic tank problem first with the coarser mesh and then might have wanted to refine it. Instead of starting anew, as in the STDY2 sample problem, the converged PHED-array for the coarser mesh might have been used as the basis of the initial guess for the refined mesh. The data for the CARRY printout were given STDY2 in the form of a restart deck, and convergence to the same solution as that achieved in the STDY2 sample problem was reached in 45 iterations, a saving of about 43 percent.

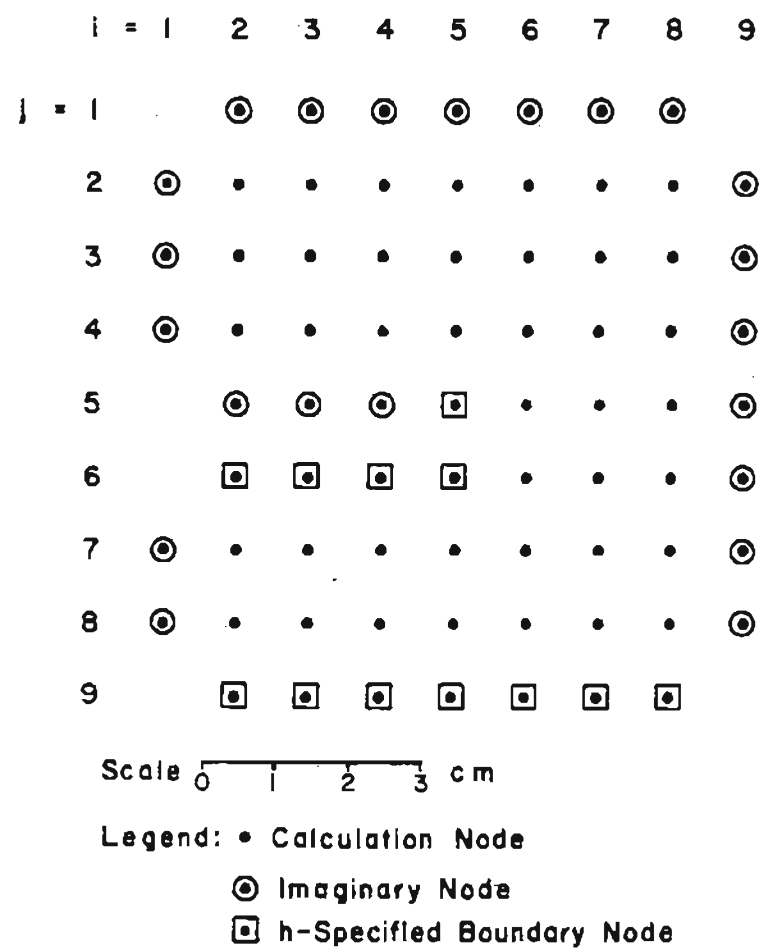


Figure 15.--Uniform 1-cm mesh superimposed on cross section of figure 4.

Figure 16.--Input data for CARRY sample problem.

DATA SHEET CARRY | Date 3/74 | Page 1 of 3

Case Small-Scale Septic Tank | Name

Card Grp 1										
Variable	COMENT									
Format	(20A4)									
Value Cd 1	CARRY--CONVERTING SMALL-SCALE SEPTIC TANK (6 x 7 CM) RESTART									
Cd 2	OLD: DELTA X = DELTA Y = 1 CM									
Cd 3	NEW: FINE MESH SIZE NEAR NOTCH									
Cd 4	BLANK									
Cd 5	BLANK									
Card Grp 2										
Variable	LGTH	DEPTH	IFILE	ITAPE	NCARDY	NCARDX	NCRDYN	NCRDXN		
Format	F10.2	F10.2	I5	←		→		I5		
Value	6.00	7.00	0	0	1	1	3	3		
Card Grp 3										
Variable	IMGTOP	IMGBOT	IMGLSD	IMGRSD						
Format	I5	←	→	I5						
Value	1	0	1	1						

DATA SHEET CARRY | Date 3/74 | Page 2 of 3

Case Small-Scale Septic Tank | Name

Card Grp 4										
Variable	DYLGTH	DELY	DYLGTH	DELY	DYLGTH	DELY	DYLGTH	DELY		
Format	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3		
Value	0.00	1.000								
Card Grp 5										
Variable	DXLGTH	DELX	DXLGTH	DELX	DXLGTH	DELX	DXLGTH	DELX		
Format	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3		
Value	0.00	1.000								
Card Grp 6										
Variable	DYLGTN	DELYN	DYLGTN	DELYN	DYLGTN	DELYN	DYLGTN	DELYN		
Format	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3		
Value	0.00	1.000	1.00	0.500	5.00	1.000				
Card Grp 7										
Variable	DXLGTN	DELXN	DXLGTN	DELXN	DXLGTN	DELXN	DXLGTN	DELXN		
Format	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3	F10.2	F10.3		
Value	0.00	1.000	2.00	0.500	5.00	1.000				

DATA SHEET CARRY   Date 3/74   Page 3 of 3

Case Small-Scale Septic Tank   Name

Card Grp 8										
Variable-	PHED	PHED	PHED	PHED	PHED	PHED				
Format	D13.6	←			→	D13.6				
Value	USE RESTART DECK PRODUCED BY STDY2 RUN									

Figure 16.--Continued.

```
CARRY - CONVERTING SMALL SCALE SEPTIC TANK (6X7 CM) RESTART
OLD: DELTA X = DELTA Y = 1 CM
NEW:  FINE MESH SIZE NEAR NOTCH


  LGTH   DEPTH   IFILE   ITAPE   NCARDY   NCARDX   NCRDYN   NCRDXN
  6.00    7.00     0       0        1        1        3        3

NCOL   NROW   NCOLN   NROWN
  9      9      12      13

INGTOP   INGBOT   INGLSD   INGRSD
  1        0        1        1

VARIABLE MESH INCREMENT DATA FOLLOW IN TRIPLETS AS XYZ
    WHERE X = MEASURED DISTANCE FROM AXIS (DXLGTH, DYLGTH, DXLGTN OR DYLGTN)
          Y = ROW OR COLUMN NUMBER (IX, JY, IXN OR JYN)
          Z = INCREMENT LENGTH (DELX, DELY, DELXN OR DELYN)
OLD VERTICAL
    0.0     2   1.000           7.00    9   1.000
OLD HORIZONTAL
    0.0     2   1.000           6.00    9   1.000
NEW VERTICAL
    0.0     2   1.000           1.00    3   0.500           5.00   11   1.000           7.00   13   1.000
NEW HORIZONTAL
    0.0     2   1.000           2.00    4   0.500           5.00   10   1.000           6.00   12   1.000

                                   OLD PRESSURE HEAD ARRAY
  1  -0.80000D+01-0.27110D+02-0.27104D+02-0.27064D+02-0.26827D+02-0.25492D+02-0.24248D+02-0.23775D+02-0.80000D+01
  2  -0.70000D+01-0.26082D+02-0.26053D+02-0.25922D+02-0.25512D+02-0.24496D+02-0.23528D+02-0.23149D+02-0.70000D+01
  3  -0.60000D+01-0.25110D+02-0.25104D+02-0.25064D+02-0.24827D+02-0.23492D+02-0.22248D+02-0.21775D+02-0.60000D+01
  4  -0.50000D+01-0.24148D+02-0.24187D+02-0.24395D+02-0.25288D+02-0.22462D+02-0.20265D+02-0.19510D+02-0.50000D+01
  5  -0.40000D+01-0.23110D+02-0.23104D+02-0.23064D+02-0.30000D+02-0.21038D+02-0.17040D+02-0.15915D+02-0.40000D+01
  6  -0.30000D+01-0.30000D+02-0.30000D+02-0.30000D+02-0.30000D+02-0.16228D+02-0.12041D+02-0.11016D+02-0.30000D+01
  7  -0.20000D+01-0.11665D+02-0.11603D+02-0.11336D+02-0.10487D+02-0.78147D+01-0.65181D+01-0.61444D+01-0.20000D+01
  8  -0.10000D+01-0.39735D+01-0.39467D+01-0.38398D+01-0.35661D+01-0.30356D+01-0.27201D+01-0.26204D+01-0.10000D+01
  9   0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0

                                   NEW PRESSURE HEAD ARRAY
  1   0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
      0.0         0.0
  2   0.0        -0.26082D+02-0.26053D+02-0.25922D+02-0.25717D+02-0.25512D+02-0.25004D+02-0.24496D+02-0.24012D+02-0.23628D+02
     -0.23149D+02 0.0
  3   0.0        -0.25110D+02-0.25104D+02-0.25063D+02-0.24945D+02-0.24827D+02-0.24159D+02-0.23492D+02-0.22870D+02-0.22248D+02
     -0.21775D+02 0.0
  4   0.0        -0.24629D+02-0.24646D+02-0.24729D+02-0.24894D+02-0.25058D+02-0.24017D+02-0.22977D+02-0.22117D+02-0.21256D+02
     -0.20642D+02 0.0
  5   0.0        -0.24148D+02-0.24187D+02-0.24395D+02-0.24842D+02-0.25288D+02-0.23875D+02-0.22462D+02-0.21364D+02-0.20265D+02
     -0.19510D+02 0.0
  6   0.0        -0.23629D+02-0.23646D+02-0.23729D+02-0.25687D+02-0.27644D+02-0.24697D+02-0.21750D+02-0.20201D+02-0.18652D+02
     -0.17713D+02 0.0
  7   0.0        -0.23110D+02-0.23104D+02-0.23063D+02-0.26532D+02-0.30000D+02-0.25519D+02-0.21038D+02-0.19039D+02-0.17040D+02
     -0.15915D+02 0.0
  8   0.0        -0.26555D+02-0.26552D+02-0.26532D+02-0.28266D+02-0.30000D+02-0.24317D+02-0.18633D+02-0.16587D+02-0.14540D+02
     -0.13465D+02 0.0
  9   0.0        -0.30000D+02-0.30000D+02-0.30000D+02-0.30000D+02-0.30000D+02-0.23114D+02-0.16228D+02-0.14134D+02-0.12041D+02
     -0.11016D+02 0.0
 10   0.0        -0.20832D+02-0.20801D+02-0.20668D+02-0.20456D+02-0.20243D+02-0.16132D+02-0.12021D+02-0.10650D+02-0.92797D+01
     -0.85800D+01 0.0
 11   0.0        -0.11665D+02-0.11603D+02-0.11336D+02-0.10911D+02-0.10487D+02-0.91506D+01-0.78147D+01-0.71664D+01-0.65181D+01
     -0.61444D+01 0.0
 12   0.0        -0.39735D+01-0.39467D+01-0.38398D+01-0.37030D+01-0.35661D+01-0.33009D+01-0.30356D+01-0.28779D+01-0.27201D+01
     -0.26204D+01 0.0
 13   0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
      0.0         0.0

CARDS PUNCHED.
```

Figure 17.--Printout of CARRY sample problem.

Appendix C:
COMPAR — For Comparing Two-Dimensional Data Arrays

This program was developed specifically for comparing PHED-arrays obtained from program STDY2. But it can be used for comparing any pair of two-dimensional data arrays, provided they have the same numbers of rows and columns.

STDY2 iteratively solves a system of finite difference equations to change an arbitrary array of PHED-values to one that satisfies some particular set of boundary conditions. This solution array is approached asymptotically, so that one may decide that a solution is acceptable if, at each node in the solution mesh, PHED-values from successive iterations do not differ by more than some small amount.

COMPAR receives two PHED-arrays as input. At each node of the solution mesh, it obtains the ratio of the difference in the two PHED-values to the value of one of them. If the absolute value of this ratio is larger than a value specified by the user, information is printed that identifies the location of the node and gives the two PHED-values and the ratio.

A program listing of COMPAR is given below. Modifications that might be necessary before running the program on other computers are flagged. Their numbering and explanations are the same as for STDY2, appendix A. For magnetic tape input, PHED1(I,J) and PHED2(I,J) must be given the same dimensions as PHED(I,J) in STDY2. This dimensioning is done by means of the DOUBLE PRECISION statement in COMPAR.

The logic of this program is straightforward and is readily deduced from inspection of the listing. No flow chart is given.

A user may compare more than one pair of arrays in one COMPAR run by simply submitting an input data deck for each case compared.

Glossary of input variables

Input variables are defined in the order of their appearance in the input data deck. Figure 18 shows punchcard layouts for these data. The manner of presentation is the same as in appendix A.

Card Group 1 - Five cards, even if some are blank. Must be in the input deck for processing each case. Format (20A4)

COMENT - Alphanumeric identification printed at head of output.

Card Group 2 - A single card. Must be present in input deck for processing each case. Format (D13.6,3I5)

DLIMIT - If RATIO (defined in the glossary of noninput variables) exceeds the value given for DLIMIT, a line of printout identifies the offending node. Because the main purpose of COMPAR is convergence checking, one usually selects as a value for DLIMIT the maximum value of RATIO he is willing to accept in what he considers a converged solution.

MROW - The number of rows in the solution mesh. This value is obtained from the printout of the STDY2 case producing the arrays compared.

MCOL - The number of columns in the solution mesh. This value is obtained from the printout of the STDY2 case producing the arrays compared.

IFILE = 0 if input array is in card form.
- > 0 if input is read from tape. The value is the position of the input file on a multifile tape. See explanation of same variable in STDY2, glossary of input variables, card group 1. This program reads two files at once, however, so that the definition of R and S in the equation IFILE = R - S must be modified:
  - R = the number representing the position of the PHED1 file on the tape
  - S = the number representing the position on the tape of the second file (PHED2) read in by the preceding case of the same run (has the value zero for the first case of the run)

Card Group 3 - A multiple card group produced by a STDY2 run. If STDY2 wrote magnetic tape, there is no card group 3. Format (6D13.6)—six values of the following variable per card--unneeded fields in the last card may be left blank.

PHED1(I,J) - The pressure head value at node (I,J) of the solution mesh. See PHED2(I,J), card group 4.

Card Group 4 - A multiple card group produced by a STDY2 run. If STDY2 wrote magnetic tape, there is no card group 4. Format (6D13.6)—six values of the following variable per card--unneeded fields in the last card may be blank.

PHED2(I,J) - The pressure head value at node (I,J) of the solution mesh. The arrays containing PHED1 and PHED2 are obtained from STDY2 one or more iterations apart. Through the use of IDBLE, STDY2 card group 3, these arrays are 15 iterations apart. When using cards, however, the user may restart a case and obtain PHED2 after any number of iterations, considering the cards used in the input data deck for restart as PHED1 if he wishes.

NOTE: If magnetic tape is used for input, be sure that PHED1(I,J) and PHED2(I,J) are dimensioned exactly the same as PHED(I,J) in STDY2. If punch cards are used, PHED1(I,J) and PHED2(I,J) must have dimensions at least as large as those exhibited by the input data. Dimensioning is specified in the DOUBLE PRECISION and DIMENSION statements.

Glossary of noninput variables

DIFF - The absolute difference between PHED1 and PHED2.

I - Column number, I, in the arrays being compared.

ICT - A counter used in selecting the wanted input file from a multifile tape.

J - Row number, J, in the arrays being compared.

RATIO - DIFF divided by PHED1.

Figure 18.--Input data layout for COMPAR.

USDA-FOREST SERVICE **MULTIPLE-CARD LAYOUT FORM**	PROGRAM NAME: COMPAR		PROGRAM NUMBER:
	FORMAT DESIGNED BY:	DATE: 3/74	PAGE: 1 OF 1

CARD IDENT.	Columns 1–80
Format 10 Card Group 1	COMENT 20A4 COMPAR TESTING FOR STDY2 CONVERGENCE

CARD IDENT.	1–13	14–18	19–23	24–28	29–80
Format 20 Card Group 2	DLIMIT D13.6 0.100000-03	MROW I5 13	MCOL I5 12	IFILE I5 0	Blank

CARD IDENT.	1–13	14–26	27–39	40–52	53–65	66–78	79–80
Format 40 Card Group 3	PHED1(1,1) D13.6 -0.800000D+01	PHED1(2,1) D13.6 -0.271098D+02	PHED1(3,1) D13.6 -0.271040D+02	PHED1(4,1) D13.6 -0.270635D+02	PHED1(5,1) D13.6 -0.268273D+02	PHED1(6,1) D13.6 -0.254916D+02	Blank
Format 40 Card Group 4	PHED2(1,1) D13.6 -0.800000D+01	PHED2(2,1) D13.6 -0.271098D+02	PHED2(3,1) D13.6 -0.271040D+02	PHED2(4,1) D13.6 -0.270635D+02	PHED2(5,1) D13.6 -0.268273D+02	PHED2(6,1) D13.6 -0.254916D+02	Blank

6600-1 (11/66)

Program listing

```
C
C    COMPAR --- COMPARES DATA IN TWO ARRAYS, PHED1(I,J)
C       AND PHED2(I,J), OF EQUAL DIMENSIONS.
C
C    3/18/74
C
C***** POSSIBLE MODIFICATION TYPE M1.
C
      DOUBLE PRECISION PHED1(60,70),PHED2(60,70),RATIO,DIFF,DLIMIT
C
C***** POSSIBLE MODIFICATION TYPE M3.
C
      DIMENSION COMENT (100)
C
C***** POSSIBLE MODIFICATION TYPE M4.
C
    5 READ (5,10,END=70)COMENT
   10 FORMAT (20A4)
      WRITE (6,15) COMENT
   15 FORMAT (1H1,20A4/(20A4))
      READ (5,20) DLIMIT,MROW,MCOL,IFILE
   20 FORMAT (D13.6,3I5)
      WRITE (6,25)
   25 FORMAT (1H0,4X,6HDLIMIT 6X,4HMROW 3X,4HMCOL 2X,5HIFILE    )
      WRITE (6,30) DLIMIT,MROW,MCOL,IFILE
   30 FORMAT (1H D13.6,I6,I7,I6)
      WRITE (6,35)
   35 FORMAT (1H0 2X,1HI 2X,1HJ 5X,5HRATIO 5X,10HPHED1(I,J) 3X,
     110HPHED2(I,J)   //)
      IF (IFILE.NE.0) GO TO 45
      READ (5,40) ((PHED1(I,J),I=1,MCOL),J=1,MROW)
   40 FORMAT (6D13.6)
      READ (5,40) ((PHED2(I,J),I=1,MCOL),J=1,MROW)
      GO TO 55
C
C***** POSSIBLE MODIFICATION TYPE M5.
C
   45 DO 50 ICT = 1,IFILE
      READ (9) PHED1
   50 READ (9) PHED2
C
C*** COMPARE ARRAYS NODE BY NODE **********
C
   55 DO 65 J = 1,MROW
          DO 65 I = 1,MCOL
           IF (PHED1(I,J).EQ.0.OR.PHED2(I,J).EQ.0) GO TO 65
           DIFF = DABS(PHED1(I,J) - PHED2(I,J))
           RATIO = DABS (DIFF / PHED1(I,J))
           IF (RATIO.LT.DLIMIT) GO TO 65
           WRITE (6,60) I,J,RATIO,PHED1(I,J),PHED2(I,J)
   60      FORMAT (1X,2I3,3D13.6)
   65      CONTINUE
      GO TO 5
   70 STOP
      END
```

Appendix D: List of Non-Fortran Symbols

h - Soil water pressure head (L).
H - Hydraulic head (L).
i - Column number in solution mesh.
j - Row number in solution mesh.
K - Hydraulic conductivity (LT^{-1}).
m - Iteration number in the finite differencing scheme.
v - Flux rate (LT^{-1}).
x - Distance parallel to x-axis of the Cartesian coordinate system, positive to the right (L).
y - Distance parallel to the y-axis. For purposes of presentation of the model equations, y is positive upward. The significance of this is that infiltration is a negative flux and upward evaporation is a positive flux. For purposes of measurement between the x-axis and rows of nodes in the solution mesh, however, y is positive downward (L).
z - Elevation above a datum (L).
α - The tangent of the angle α is the slope of the cross section.
Δx_- - Length of mesh increment to left of node i,j in the solution mesh (L).
Δx_+ - Length of mesh increment to right of node i,j (L).
Δy_- - Length of mesh increment above node i,j (L).
Δy_+ - Length of mesh increment below node i,j (L).
ω - Overrelaxation factor.

Appendix E: Program Updating

Although the program has been run for a number of different cases, there will undoubtedly be reason to alter it in the future--either to correct as yet undetected errors, to modify output formats, or to improve efficiency.

Notification of updating will be by mimeographed reports. Users who wish to receive update notices should ask to be placed on the update mailing list by writing to the author:

C. R. Amerman
USDA ARS
207 Business Loop 70 East
Columbia, MO 65201

GPO 812-158