-24-87

JB

(2)

I-31922

DR 0309-3

(5)

ORNL-6386

ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

3DFEMWATER: A Three-Dimensional Finite Element Model of WATER Flow Through Saturated-Unsaturated Media

G. T. Yeh

Environmental Sciences Division
Publication No. 2904

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A15 Microfiche A01

ORNL--6386

DE87 014955

3DFEMWATER: A Three-Dimensional Finite Element Model of WATER Flow through Saturated-Unsaturated Media

G. T. Yeh

Environmental Sciences Division
Publication No. 2904

NUCLEAR AND CHEMICAL WASTE PROGRAMS
(Activity No. AR 05 10 05 K, ONL-WN17)

Date of Issue - August 1987

Prepared for the
Office of Defense Waste and Transportation Management

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC
for the
U. S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-84OR21400

MASTER

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

YEH, G. T. 1987. 3DFEMWATER: A Three-Dimensional Finite Element Model of WATER Flow through Saturated-Unsaturated Media. ORNL-6386. Oak Ridge National Laboratory, Oak Ridge, TN. 304 pp.

This report presents the development and verification of a three-dimensional finite element model of water flow through saturated-unsaturated media (3DFEMWATER). The special features of 3DFEMWATER are its flexibility and versatility in modeling as wide a range of real-world problems as possible. The model is designed to: (1) treat heterogeneous and anisotropic media consisting of as many geologic formations as desired, (2) consider both distributed and point sources/sinks that are spatially and temporally dependent, (3) accept the prescribed initial conditions or obtain them by simulating a steady state version of the system under consideration, (4) deal with a transient head distributed over the Dirichlet boundary, (5) handle time-dependent fluxes due to pressure gradient varying along the Neumann boundary, (6) treat time-dependent total fluxes distributed over the Cauchy boundary, (7) automatically determine variable boundary conditions of evaporation, infiltration, or seepage on the soil-air interface, (8) include the off-diagonal hydraulic conductivity components in the modified Richards equation for dealing with cases when the coordinate system does not coincide with the principal directions of the hydraulic conductivity tensor, (9) give three options for esmating the nonlinear matrix, (10) include two options (successive subregion block iterations and successive point iterations) for solving the linearized matrix equations, (11) automatically reset time step size when boundary conditions or source/sinks change abruptly, and (12) check the mass balance computation over the entire region for every time step. The model is verified with analytical solutions or other numerical models for three examples.

1. INTRODUCTION

To study the transport of dissolved constituents in a subsurface flow system, the velocity field, pressure distribution, and moisture content must be determined. Numerous models by finite differences and finite elements have been reported in the literature for simulating the flow variables in saturated-unsaturated porous media (Freeze 1972; Neuman et al. 1974; Reeves and Duguid 1975; Segol 1976; van Genuchten et al. 1977; Narasimhan and Witherspoon 1977; van Genuchten and Pinder 1978; Yeh and Ward 1980; Reisenauer et al. 1982; Huyakorn 1986). While the Richards equation (Richards 1931) or its modified form has been the common basis for the development of practically all numerical models, the variations among these models lie mainly in the number of space dimensions, the choice of numerical methods, the treatment of initial and boundary conditions, and the consideration of source/sink.

The vast majority of subsurface flow models in saturated-unsaturated media are limited to two-dimensional and/or one-dimensional problems. Only a few three-dimensional models have appeared in the report and open literature (Segol 1976; Reisenauer, et al. 1982; Huyakorn 1986). In nature, water movement in the surface system occurs in all directions (three-dimensional problems) under saturated-unsaturated flow conditions. This report presents the development and verification of a three-dimensional model of water flow through saturated-unsaturated media to deal with naturally occurring or artificially disturbed flow systems. The model can handle completely saturated, completely unsaturated, or partially saturated and partially unsaturated flows. It can be applied to confined or unconfined aquifer regimes. The most difficult task in modeling unconfined aquifer flows is locating the moving phreatic surfaces under transient conditions. This difficulty can be alleviated by treating the media as being under combined saturated-unsaturated conditions. The surface of the zero pressure head, which is a function of time, can then be considered as the moving phreatic surface.

The most common numerical methods used to solve the Richards equations are the finite difference methods (FDMs) and finite element methods (FEMs) (Forsythe and Wasow 1960; Huebner 1975; Lapidus and Pinder 1982). The fundamental distinction between FEM and FDM is that the former is based directly upon approximating the function, whereas the latter is based on approximating the derivatives. Thus, the FEM provides spatially continuous solutions, whereas the FDM yields solutions only at discrete points. In addition, FEM has several advantages over FDM: (1) a better description of irregular boundaries, without the need for special formulas, (2) the ease of employing irregular grid to provide different levels of spatial discretization in different regions of the aquifers, (3) easy handling of aquifer heterogeneity and anisotropy, (4) need (sometimes) for fewer node points to represent the aquifer to the same level of accuracy, resulting in savings of computational time and computer storage, and (5) the natural result of flux types of boundary conditions from the integral formulation. Hence the FEM is used in the 3DFEMWATER code because it is believed that the FEM is the best method, especially when it is used in conjunction with point or block iteration (Yeh 1985; Yeh et al. 1985) for the solution of matrix equations, as far as the Richards equation is concerned.

Most of the existing models assume that initial conditions are known and hence are the inputs to the model (Freeze 1972; Neumann et al 1974; Segol 1976; Reisenauer et al. 1982). A few models, however, do provide the option of obtaining the initial conditions by simulating the steady-state version of the Richards equation subject to time-invariant boundary conditions (Reeves and Duguid 1975; Yeh and Ward 1980).

As far as boundary conditions are concerned, four types may be prescribed for the most general cases: (1) Dirichlet, (2) Neumann, (3) Cauchy, and (4) variable conditions. For the Dirichlet conditions, the pressure head is prescribed on the boundary. These conditions are normally applied to the soil-water interfaces such as streams, rivers, lakes, artificial impoundments, and coastal lines. For the Neumann conditions, the pressure gradient is prescribed on the boundary. These conditions do not occur or may not be specified very often in real-world problems. They may be applied to the bottom of the media where natural drainage takes place. For the Cauchy conditions, the total normal flux on the boundary due to pressure gradient and gravity is prescribed. These conditions are normally applied to surface water bodies with known infiltration rates through their bottom layers of sediments or liners into subsurface media. On the variable boundary, either Cauchy or Dirichlet conditions prevail depending on the conductivity of the media, the availability of water such as rainfall rates, and the potential evapotranspiration. These conditions normally occur at the soil-air interface. Most of the existing large computational codes have not included all four types of boundary conditions. Many considered only two or, at the most, three types. Furthermore, most of these were not designed to treat boundary values that are time dependent either gradually or abruptly and that vary along the boundary. Concerning the treatment of source/sinks, existing computational codes fare much better since quite a few include transient, point, as well as distributed source/sink terms (Neumann et al. 1974).

In real-world problems, the recharge and pumping due to rainfall and artificial withdrawals are likely to be time dependent and distributed over the region of interest. The given boundary pressure heads, with rare exception, vary with time and along Dirichlet segments at the surface water bodies, which inevitably have their water levels fluctuating along the soil-water interfaces. The prescribed pressure gradients along Neumann boundary segments are, in general, time dependent. The known total water fluxes on Cauchy boundary segments are transient in nature;

for example, the infiltration rate from the bottom of surface water bodies into subsurface media depends on the depth of water, which, of course, varies with time. Furthermore, on the air-soil interface, the rainfall is intermittent and its intensity is not constant with time, resulting in transient total fluxes if the soil is capable of taking all the throughfall. Thus, for field applications, a computational model must have the capability to deal with these aspects simultaneously. This report presents the development of such a model to achieve these purposes.

In addition to its versatility and flexibility to treat transient source/sinks and boundary conditions and to provide options to obtain initial conditions, 3DFEMWATER also includes several special features. By including the off-diagonal hydraulic conductivity components, the model can be applied to anisotropic and nonuniform media. For an anisotropic uniform medium, one can imbed the coordinate system to coincide with the principal directions of the hydraulic conductivity tensor. However, for anisotropic and heterogeneous media, the inclusion of off-diagonal conductivities becomes necessary since the principal directions of the hydraulic conductivity tensor vary with space, prohibiting the imbedance of a unique coordinate system.

The Richards equation is nonlinear and its numerical solution exhibits fluctuations in iterations for many situations. Underrelaxation is thus provided in 3DFEMWATER to reduce the fluctuations. When certain sets of system parameters such as hydraulic retention and hydraulic conductivity are variable, the rate of numerical convergence is very slow. Hence, an overrelaxation technique is given as an option to speed up the convergence.

In solving the linearized matrix equation to obtain the iterates, practically all finite element models use the direct elimination methods to solve the system of algebraic equations simultaneously. When the region of interest requires a large number of nodes for discretization, the use of a direct elimination solution to the matrix equation becomes impractical because of excessive requirements on the central process unit (CPU) memory. For example, an 80 x 40 x 20 node discretization in a three-dimensional problem would need a CPU memory of about 960 megabytes for the banded coefficient matrix alone. These plus other overhead CPU memory are beyond the capacity of any of the mainframe computers today. To alleviate this problem, two options are provided in solving the matrix equation which will greatly reduce the CPU memory requirement: successive subregional block iteration and successive point iteration.

Finally, for abrupt change of external conditions, small time step size is needed to yield accurate and stable solutions. Automatic resetting of the time step size is built into 3DFEMWATER to facilitate the computations. This alleviates the troublesome external manipulations.

2. MATHEMATICAL STATEMENTS

2.1 Derivation of the Governing Equations

The mathematical formulation of the conservation of mass and Darcy's law extended to three-dimensional flow have been described in detail elsewhere (Bear 1972). Using (1) the continuity of fluid, (2) the continuity of solid, (4) the motion of fluid (Darcy's law), (4) equation of the state, and (5) the law of consolidation of media, one can derive a governing equation for the distribution of pressure head in the saturated-unsaturated subsurface media. The derivation of this combined equation is described below.

The continuity of fluid is expressed in integral form as:

$$\frac{D}{Dt}\int_{\nu} S n_e \rho_f \, d\nu = -\int_{\Gamma} \mathbf{n}\cdot(\rho_f \mathbf{V}_{fs}) \, d\Gamma + \int_{\nu} \rho_f q \, d\nu \, , \qquad (1)$$

where

t = time (T),

$D(\ )/Dt$ = total derivative of ( ) with respect to time,

n_e = effective porosity (L^3/L^3),

S = degree of saturation (dimensionless),

ρ_f = fluid density (M/L^3),

$\mathbf{n}$ = outward unit vector normal to the surface,

V_{fs} = Darcy's velocity relative to the solid matrix (L^3),

ν = volume of material containing constant amount of solids (Fig. 1) (L^3),

q - internal source/sink $[(L^3/T)/L^3]$,

Γ - surface enclosing the volume ν (Fig. 1) (L^2).

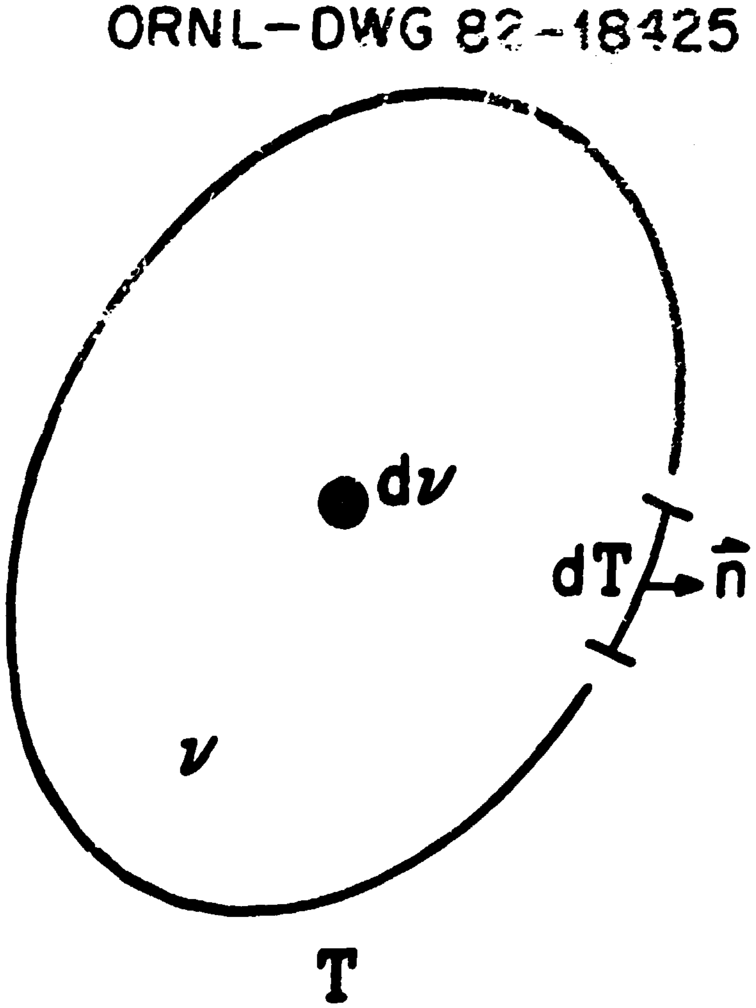

Fig. 1. An element volume containing constant amount of solids

By the Reynolds transport theorem (Owczarek 1964), Eq. (1) can be written as

$$\int_{\nu} \frac{\partial \rho_f n_e S}{\partial t} \, d\nu + \int_{\Gamma} n \cdot (S \rho_f n_e V_s) d\Gamma + \int_{\Gamma} n \cdot (\rho_f V_{fs}) d\Gamma = \int_{\nu} \rho_f q \, d\nu \ , \tag{2}$$

where

V_s = velocity of the deformable surface Γ due to consolidation (L/T).

Applying the Gaussian divergence theorem to Eq. (2) and using the fact that ν is arbitrary, one can obtain the following continuity equation of fluid in differential form:

$$\frac{\partial \rho_f n_e S}{\partial t} + \nabla \cdot (\rho_f n_e S V_s) + \nabla \cdot (\rho_f V_{fs}) = \rho_f q \ . \tag{3}$$

The granular skeleton of the media is considered compressible under pressure, but the grains themselves are considered incompressible. The continuity statement of incompressible solids but a compressible skeleton is

$$\frac{D}{Dt} \int_{\nu} \rho_s (1 - n_e) \, d\nu = 0 \ , \tag{4}$$

where

ρ_s = density of solid grains (M/L^3).

Again applying the Reynolds transport theorem to Eq. (4), we obtain:

$$\int_{\nu} \frac{\partial(1 - n_e)\rho_s}{\partial t} d\nu + \int_{\Gamma} n \cdot [(1 - n_e)\rho_s V_s] d\Gamma = 0 . \qquad (5)$$

Using the Gaussian divergence theorem and the fact that ξ is arbitrary, we can write Eq. (5) in the differential form as:

$$\frac{\partial(1 - n_e)\rho_s}{\partial t} + \nabla \cdot [(1 - n_e)\rho_s V_s] = 0 . \qquad (6)$$

However, since the solid grains are incompressible, we have

$$\frac{\partial \rho_s}{\partial t} + V_s \cdot \nabla \rho_s = 0 . \qquad (7)$$

Combining Eqs. (6) and (7), we have:

$$\frac{\partial(1 - n_e)}{\partial t} + \nabla \cdot [(1 - n_e)V_s] = 0 . \qquad (8)$$

Equations (3) and (8) are derived based on the continuity law of fluid and solid grains, with the assumptions that the grains are incompressible. These two equations involve five state variables, ρ_f, n_e, V_{fs}, V_s, and S. Up to this point, no empiricism has been introduced in the derivation of the governing equations [Eqs. (3) and (8)]. However, the number of state variables exceeds the number of equations. Thus, constitutive relationships must be empirically or theoretically established among n_e, ρ_f, V_{fs}, V_s, and S.

The first constitutive relationship is the empirical Darcy's law, stating that the relative Darcy's velocity V_{fs} is proportional to the gradient of hydraulic head. The proportionality constant is termed the hydraulic conductivity. This empirical law, in fact, can be derived theoretically from the continuity of momentum with the assumptions of neglecting the inertial forces and linearizing the frictional force (Polubarinova-Kochina 1962). The extension of Darcy's law to anisotropic media may be written in the form of

$$V_{fs} = - K \cdot \nabla H \ , \tag{9}$$

$$H = h + z \ , \tag{10}$$

where

H = total head (L),

h = pressure head (L),

z = potential head (L),

K = hydraulic conductivity tensor (L/T).

The hydraulic conductivity tensor is a function of fluid and medium properties and is defined as

$$K = \frac{\rho_f g}{\mu_f} P \ , \tag{11}$$

where

P = intrinsic permeability tensor of the media (L^2),

μ_f = dynamic viscosity of the fluid $(ML/T^2/L_2/T)$,
g = acceleration of gravity (L/T^2).

A new state variable h in Eqs. (9) and (10) has been introduced. Thus, we will need a constitutive relationship to take care of the new state variable h. Thus, the second constitutive relationship will be the empirical, thermodynamic equation of state. We will assume that the fluid density ρ_f is either constant or a function of the pressure p only; that is,

$$\rho_f = \rho(p) \ , \tag{12}$$

where

p = pressure $[(ML/T^2)/L^2]$.

With the assumption of Eq. (12), we can now relate the pressure head to the pressure as

$$h = \frac{p - p_o}{\rho_f g} = \frac{\sigma}{\rho_f g} \ , \tag{13}$$

where

p_o = the datum pressure $[(ML/T^2)/L^2]$,
σ = the incremental pressure of the fluid.

The third constitutive relationship will be derived based on the consolidation law of the media to relate the solid velocity V_s to the pressure head h in this paragraph. The three-dimensional consolidation equation developed by Biot (1940) is

$$(\lambda_s + 2\mu_s)\nabla^2 e = \nabla^2 \sigma \ , \tag{14}$$

where

λ_s = Lamé first constant $[(ML/T^2)/L^2]$,

μ_s = Lamé second constant $[(ML/T^2)/L^2]$,

e = dilatation of the media (dimensionless).

The dilatation e and the solid velocity V_s are defined by:

$$e = \nabla \cdot U \qquad \text{and} \qquad V_s = \frac{\partial U}{\partial t} \ , \tag{15}$$

where

U = displacement of the media (L).

Taking the divergence of V_s, and from Eq. (15), we have

$$\nabla \cdot V_s = \frac{\partial}{\partial t}(\nabla \cdot U) = \frac{\partial e}{\partial t} \ . \tag{16}$$

Equations (13), (14), and (16) have implicitly established the constitutive relationship between V_s and the pressure head h. To explicitly express this relationship, we integrate Eq. (14) to yield

$$(\lambda_s + 2\mu_s)e = \sigma + f \ . \tag{17}$$

where

f = integration function.

The integration function f must satisfy the Laplace's equation for all time. To simplify the matter further, we will only consider vertical consolidation. Under this condition, it has been shown (Verruijt 1969) that the integration function f is equal to 0. It then follows from Eq. (17) that

$$\frac{\partial e}{\partial t} = \alpha \frac{\partial \sigma}{\partial t} , \tag{18}$$

where

$$\alpha = \frac{1}{(\lambda_s + 2\mu_s)} \tag{19}$$

in which

α = the coefficient of consolidation of the media.

Substituting Eqs. (13) and (18) into Eq. (16), we obtain

$$\nabla \cdot V_s = \frac{\partial e}{\partial t} = \alpha \frac{\partial \sigma}{\partial t} = \alpha \rho_f g \frac{\partial h}{\partial t} = \alpha' \frac{\partial h}{\partial t} , \tag{20}$$

where α' is the modified compressibility of the media.

To complete the final, fourth constitutive relationship, we shall assume that the moisture content is a unique function of the pressure head. This function is termed the retention curve in the soil physics literature and expressed mathematically as:

$$\theta = n_e S = \theta(h) \ , \qquad (21)$$

where θ is the moisture content. When the media are under saturated conditions, θ is equal to the effective porosity, n_e, and S is equal to 1.

Equations (3), (8), (9), (13), (20), and (21) (representing continuity of fluid, continuity of solids, Darcy's law, the equation of state, vertical consolidation of the solid matrix, and retention characteristics of the media, respectively) contain six variables ρ_f, n_e, V_{fs}, h, V_s, and S. Hence the number of equations is equal to the number of unknowns. The system is complete, and a mathematical statement is posed. However, we can combine these six equations into a single one to simplify the problem. The simplification is demonstrated below.

Expanding Eqs. (3) and (8), we have

$$S\rho_f\left[\frac{\partial n_e}{\partial t} + \nabla\cdot(n_e V_s)\right] + n_e S \frac{\partial \rho_f}{\partial t} + n_e \rho_f \frac{\partial S}{\partial t} =$$

$$- (n_e V_s)\cdot\nabla(S\rho_f) - \nabla\cdot(\rho_f V_{fs}) + \rho_f q \ , \qquad (22)$$

and

$$\frac{\partial n_e}{\partial t} = -\nabla\cdot(n_e V_s) + \nabla\cdot V_s \ , \qquad (23)$$

respectively. Neglecting the second-order term, $(n_e V_s)\cdot\nabla(S\rho_f)$, and substituting Eq. (23) into (22), we obtain

$$n_e \rho_f \frac{\partial S}{\partial t} + n_e S \frac{\partial \rho_f}{\partial t} + S \rho_f \nabla \cdot V_s = - \nabla \cdot (\rho_f V_{fs}) + \rho_f q \ . \qquad (24)$$

From Eq. (12), a compressibility of the fluid, β, is defined by

$$\beta = \frac{1}{\rho_f} \frac{d\rho_f}{dp} \ . \qquad (25)$$

Rewriting Eq. (25) in the following form,

$$d\rho_f = \beta \rho_f dp \ , \qquad (26)$$

and using Eq. (13), we have

$$\frac{\partial \rho_f}{\partial t} = \rho_f \beta \frac{\partial p}{\partial t} = \rho_f \beta' \frac{\partial h}{\partial t} \ , \qquad (27)$$

where

$$\beta' = \rho_f g \beta \qquad (28)$$

is the modified compressibility of water. From Eq. (21), we can define a specific moisture capacity as

$$\frac{d\theta}{dh} = n_e \frac{dS}{dh} \ , \qquad (29)$$

from which

$$\frac{d\theta}{dh}\frac{\partial h}{\partial t} = n_e \frac{\partial S}{\partial t} \quad . \tag{30}$$

Substituting Eqs. (9), (20), (24), and (30) into (24), we obtain:

$$\rho_f \frac{d\theta}{dh}\frac{\partial h}{\partial t} + \rho_f \theta \beta' \frac{\partial h}{\partial t} + S\rho_f \alpha' \frac{\partial h}{\partial t} = \nabla\cdot(\rho_f K\cdot\nabla H) + \rho_f q \ . \tag{31}$$

Expanding the fourth term of Eq. (31) and neglecting the second-order term $[(K\cdot\nabla H)\cdot(\nabla\rho_f)]$, we finally have the following governing equation for saturated-unsaturated media:

$$F\frac{\partial h}{\partial t} = \nabla\cdot[K\cdot(\nabla h + \nabla z)] + q \ , \tag{32}$$

in which

$$F = \alpha' \frac{\theta}{n_e} + \beta'\theta + \frac{d\theta}{dh} \quad . \tag{33}$$

The first two terms in Eq. (33) make Eq. (32) a modified form of the Richards equation.

2.2 Initial and Boundary Conditions

To completely define the problem, Eq. (32) must be constrained by initial and boundary conditions. It is assumed that an initial distribution of pressure head can be described in the region of interest, R (Fig. 2):

$$h = h_I(x,y,z) \qquad \text{in } R, \tag{34}$$

where h_I is the prescribed initial condition, which can be defined or obtained by solving the steady state version of Eq. (32).

The specification of boundary conditions is the most difficult and intricate task in groundwater flow modeling. From the dynamic point of view, a segment of boundary may be classified as flow-through or impervious. From the physical point of view, it may be considered as a soil-air interface, a soil-soil interface, or a soil-water interface. From the mathematical point of view, it may be treated as a Dirichlet boundary on which the functional values are prescribed, a Neumann boundary on which the gradients of the function are known, or a Cauchy boundary on which the total fluxes are given. An even more difficult mathematical boundary is the variable conditions on which the boundary conditions are not known a priori but are themselves the solution to be sought. In other words, on the mathematically variable boundary, either Dirichlet or Cauchy conditions may prevail and change with time. As to which condition prevails at a particular time can only be determined in the cyclic process of solving the governing equations (Freeze 1972; Reeves and Duguid 1975; Yeh and Ward 1980).

Whatever point of view is taken, all boundary conditions must be transformed into mathematical equations for quantitative simulation.

Thus, we will specify the boundary condition from the mathematical point of view in concert with dynamic and physical considerations. The boundary conditions imposed on any segment of the boundary are taken to be either Dirichlet, Neumann, Cauchy, or variable. Thus, the boundary may be split into four parts, B_D, B_N, B_C, and B_V (Fig. 2), denoting Dirichlet, Neumann,

ORNL-DWG 87M-9564

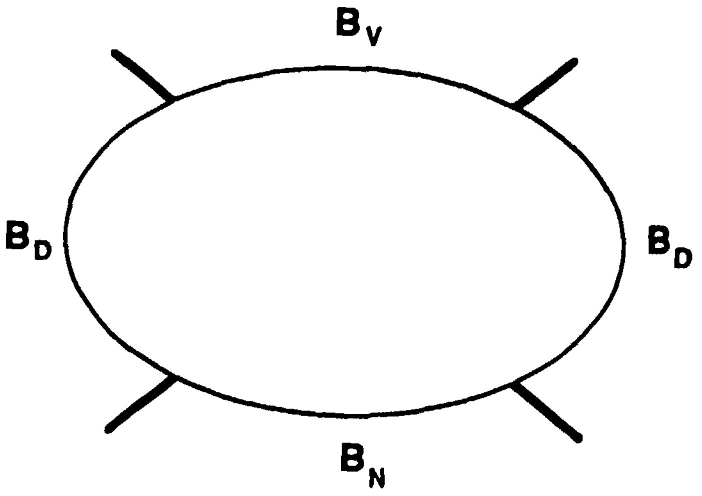

Fig. 2. Region of interest and its boundaries

Cauchy, and variable boundaries, respectively. The conditions imposed on the first three types of boundaries are given as follows:

$$h = h_D(x_b, y_b, z_b, t) \qquad \text{on} \quad B_D \;, \tag{35}$$

$$-\mathbf{n} \cdot \mathbf{K} \cdot \nabla h = q_N(x_b, y_b, z_b, t) \qquad \text{on} \quad B_N, \tag{36}$$

and

$$-\mathbf{n} \cdot (\mathbf{K} \cdot \nabla h + \mathbf{K} \cdot \nabla z) = q_C(x_b, y_b, z_b, t) \qquad \text{on} \quad B_C, \tag{37}$$

where (x_b, y_b, z_b) is the spatial coordinate on the boundary; and h_D, q_N, and q_C are the prescribed Dirichlet functional value, Neumann flux, and Cauchy flux, respectively.

The conditions imposed on the variable-type boundary, which is normally the soil-air or soil-water interface, are treated separately for precipitation and nonprecipitation periods. During precipitation periods, we impose

$$h = h_p(x_b, y_b, z_b, t) \qquad \text{on} \quad B_V, \tag{38}$$

or

$$-\mathbf{n} \cdot (\mathbf{K} \cdot \nabla h + \mathbf{K} \cdot \nabla z) = q_p(x_b, y_b, z_b, t) \qquad \text{on} \quad B_V, \tag{39}$$

where h_p is the allowed ponding depth and q_p is the throughfall of precipitation. Either Eq. (38) or Eq. (39) is applied to the boundary B_V

when the exact boundary conditions cannot in general be predicted a priori. Such a case would arise at the ground surface where either Dirichlet (ponding) or Cauchy (infiltration) conditions could prevail. The changeover from Dirichlet conditions specified by Eq. (38) to Cauchy conditions specified by Eq. (39) or vice versa is determined in the cyclic process of solving Eq. (32). Numerical implementation of this type of boundary conditions is treated in Sect. 3.3.

During nonprecipitation periods, we imposed

$$h = h_p(x_b,y_b,z_b,t) \qquad \text{on } B_V, \tag{40}$$

or

$$h = h_m(x_b,y_b,z_b,t) \qquad \text{on } B_V, \tag{41}$$

or

$$-\mathbf{n}\cdot\mathbf{K}\cdot(\nabla h + \nabla z) = q_e(x_b,y_b,z_b,t) \qquad \text{on } B_V, \tag{42}$$

where h_m is the allowed minimum pressure on the air-soil interface and q_e is the allowed maximum evaporation rate, which is the potential evaporation. Again, only one of Eqs. (40) through (42) is used at any point on the variable boundary.

Equations (32) and (35) through (42) constitute a general mathematical statement of physical problems of flow in saturated-unsaturated subsurface media. Analytical solutions for this general system do not exist. Numerical algorithms have to be devised to solve the problem. The finite element method rather than the finite difference method will be used in this report because of its ability to treat compound regions with complex boundaries and its simplicity in handling flux-type boundary conditions, whereas the FDM has to use a regular grid system and to interpolate the flux-type boundary conditions.

3. NUMERICAL APPROXIMATIONS

Equations. (32) and (35) through (42) constitute an initial-boundary value problem governing the hydrological flow problems in subsurface systems. Analytical solution to the system in general is beyond the capability of present day applied mathematics. Numerical methods are the only tools that can be used to achieve a solution. A large number of numerical approximations can be used to reduce the partial differential equation governing the subsurface flow to a system of algebraic equations. The most common numerical methods used to approximate Eq. (32) are finite difference methods (FDMs) and finite element methods (FEMs) (Forsythe and Wasow 1960; Huebner 1975; Lapidus and Pinder 1982). Many other numerical techniques such as the integrated finite difference method (IFDM) (Narasimhan and Witherspoon 1977) or the integrated compartment method (ICM) (Yeh and Luxmoore 1983), have been employed to deal with special cases of the Richards equation. Only finite differences and finite elements can be applied to the most generalized form of the transport equations. The advantages of the FEM are its inherent ability to discretize complex boundaries, ease in dealing with flux-type boundary conditions, and flexibility in including cross-derivative terms. The disadvantages of the FEM are the requirements of CPU time to obtain element matrices, and the inflexibility of using iteration methods to solve the resulting matrix equation. On the other hand, the FDM offers great economy because of simple interpolation for the derivatives, and it provides flexibility in solving the resulting matrix equation with various iteration methods. However, it suffers from the problems that the regular rectangular grid system must be used, the flux-type boundary conditions must be extrapolated, and the cross-derivative terms cannot be consistently approximated. The most severe limitations of the IFDM are its inability to treat anisotropic media and its usage of the Jacobian iteration method, in which the rate of convergency is extremely slow, but it offers even more flexibility than the FEM in discretizing the complex boundaries, and the physical representation of the method is most clearly understood. The ICM, while retaining the advantage of the IFDM, can deal with anisotropic media by defining new variables but at the expense of having to solve a large number of simultaneous field equations (Yeh and Luxmoore 1983). In addition, ICM provides options of using the direct elimination method and iteration methods with the Gauss-Seidel or successive overrelaxation schemes to solve the matrix equation (Yeh and Luxmoore 1983). In light of these discussions and the fact that great progress has been made in using the iteration methods to solve finite element equations (Yeh 1985; Yeh 1986) and that influence coefficient methods have been proposed to analytically and thus economically compute the element matrices (Huyakorn and Thomas 1984), the FEM is used in this report to approximate the Richards equation.

3.1 Finite Element Approximations in Space

Equation (32) will be integrated in the spatial dimensions by the weighted residual method in conjunction with finite elements. Because the formulation and use of the finite element method has been well documented (Lapidus and Pinder 1982), the theoretical basis will not presented here. Only the numerical procedures are summarized in the following. The region of interest is subdivided into an assemblage of smaller subdomains called elements, which are interconnected by nodes that may be either on the vertices or the boundaries of the elements. Following the procedure of the finite element, weighted-residual method, approximate formulation of the distribution of the pressure head in Eq. (32) will be obtained. Thus, let the variable h be approximated by

$$h = \sum_{j=1}^{N} h_j(t)N_j(x,y,z) \ , \tag{43}$$

where

N_j = the basis function of the spatial coordinate for the j-th node,

h_j = the value of h at node j,

N = number of nodes in the region.

Upon substituting Eq. (43) into Eq. (32) and applying the principle of weighted-residual, we obtain the following matrix equation:

$$[M]\left\{\frac{dh}{dt}\right\} + [S]\{h\} = \{G\} + \{Q\} + \{B\} \ , \tag{44}$$

where {dh/dt} and {h} are the column vectors containing the values of dh/dt and h, respectively, at all nodes; [M] is the mass matrix resulting from the storage term; [S] is the stiff matrix resulting from the action of conductivity; and {G}, {Q}, and {B} are the load vectors from the gravity force, internal source/sink, and boundary conditions, respectively. The matrices [M] and [S] are given by

$$M_{ij} = \sum_{e \epsilon M_e} \int_{R_e} N_\alpha^e F N_\beta^e \, dR \ , \tag{45}$$

$$S_{ij} = \sum_{e \epsilon M_e} \int_{R_e} [(\nabla N_\alpha^e) \cdot K \cdot (\nabla N_\beta^e) \, dR \ , \tag{46}$$

where

R_e = the region of element e,

M_e = the set of elements that have a local side α-β coinciding with the global side i-j,

N_α^e = the α-th local basis function of element e.

Similarly, the load vectors {G}, {Q} and {B} are given by

$$G_i = - \sum_{e \epsilon M_e} \int_{R_e} (\nabla N_\alpha^e) \cdot K \cdot \nabla z \, dR \ , \tag{47}$$

$$Q_i = \sum_{e \epsilon M_e} \int_{R_e} N_\alpha^e q \, dR \ , \tag{48}$$

and

$$B_i = - \sum_{e \in N_{se}} \int_{B_e} N_\alpha^e . (-K \cdot \nabla H) \, dB \ . \qquad (49)$$

where

B_e = the length of boundary segment e,

N_{se} = the set of boundary segments that have a local node α coinciding with the global node i.

It should be noted that in applying the weighted-residual finite element method to Eq. (32), we have used the Galerkin finite element method, in which the set of basis functions is chosen as the set of weighting functions for all terms.

The reduction of the partial differential equation (PDE), Eq. (32), to the set of ordinary differential equations (ODE), Eq. (44), simplifies to the evaluation of integrals on the right-hand side of Eqs. (45) through (49) for every element or boundary segment e. The major tasks that remain are the specification of basis and weighting functions and the performance of integration to yield the element matrices. Linear hexahedron elements are employed in this report.

For a hexahedron element with eight corner nodes, a trilinear polynomial base function for the α-th node may be written in terms of local normalized coordinate as

$$N_\alpha^e = (1 + \xi_\alpha \xi)(1 + \eta_\alpha \eta)(1 + \zeta_\alpha \zeta)/8 \ , \quad \alpha = 1, 2, \ldots, 8, \qquad (50)$$

where ξ_α, η_α, and ζ_α are the local coordinates of the corner nodes, which are numbered 1 to 4 progressing around the bottom face of the element in a counterclockwise direction and 5 to 8 around the top face of the element also counterclockwise direction, as shown in Fig. 3. The transformation from local coordinate (ξ,η,ζ) to the global coordinate (x,y,z) is achieved by

ORNL-DWG 87-9884

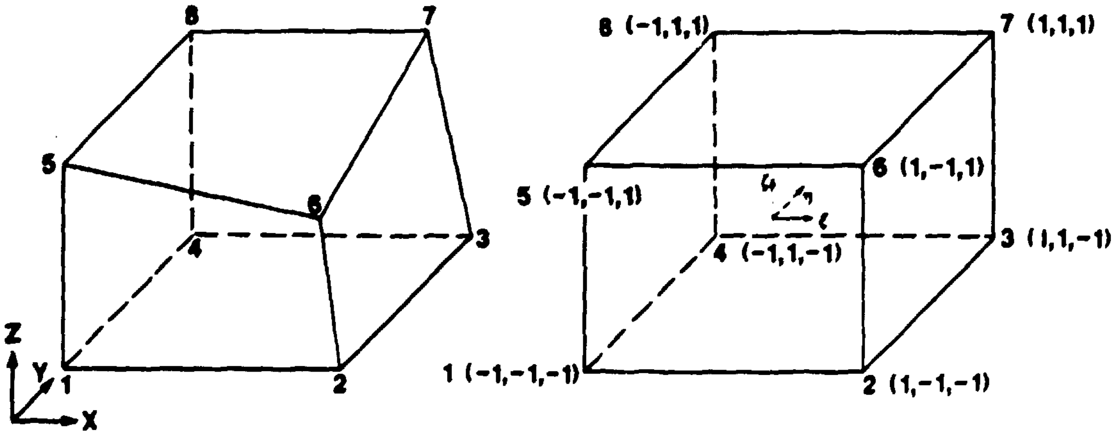

Fig. 3. Global coordinate vs local coordinate for a typical element

$$x = \sum_{\alpha=1}^{8} x_\alpha N_\alpha^e(\xi,\eta,\zeta) \ , \qquad y = \sum_{\alpha=1}^{8} y_\alpha N_\alpha^e(\xi,\eta,\zeta) \ , \quad \text{and}$$

$$z = \sum_{\alpha=1}^{8} z_\alpha N_\alpha^e(\xi,\eta,\zeta) \ . \tag{51}$$

To complete the reduction of the PDE [Eq. (32)] to the ODE [Eq. (44)], one has to evaluate the integrals on the right-hand sides of Eqs. (45) through (49) for every element to yield the element mass matrix $[M^e]$ and stiff matrix $[S^e]$ as well as the element gravity column vector $\{G^e\}$, source/sink column vector $\{Q^e\}$, and boundary column vector $\{B^e\}$ as

$$M_{\alpha\beta}^e = \int_{R_e} N_\alpha^e F N_\beta^e \, dR \ , \tag{52}$$

$$S_{\alpha\beta}^e = \int_{R_e} (\nabla N_\alpha^e)\cdot K\cdot(\nabla N_\beta^e) \, dR \ , \tag{53}$$

$$G_\alpha^e = - \int_{R_e} (\nabla N_\alpha^e)\cdot K\cdot\nabla z \, dR \ , \tag{54}$$

$$Q_\alpha^e = \int_{R_e} N_\alpha^e q \, dR \ , \tag{55}$$

and

$$B_\alpha^e = - \int_{B_e} N_\alpha^e n\cdot(-K\cdot\nabla H) \, dB \ , \tag{56}$$

where the superscript or subscript e denotes the element and α, $\beta = 1$, 2, 3, 4, 5, 6, 7, or 8.

For a hexahedron element, Eqs. (52) through (55) are computed by Gaussian quadrature (Conte 1965) because it is not easy to solve Eq. (51) for ξ, η, and ζ in terms of x, y, and z. The computation of Eqs. (52) through (55) with the Gaussian quadrature is straightforward. However, the computation of Eq. (56) will need further elaboration, which will be described in Sect. 3.3.

3.2 Finite Difference Approximation in Time

An important advantage of finite element approximation over finite difference approximation is its inherent ability to handle complex boundaries and obtain the normal derivatives therein. In the time dimension, such advantages are not evident. Thus, finite difference methods are typically used in the approximation of the time derivative. Using a time weighting factor w, we obtain from Eq. (44) the following matrix equation:

$$[C]\{h\}_{t+\Delta t} = \{L\} + \{B\} \ , \tag{57}$$

where $\{h\}_{t+\Delta t}$ is the column vector representing the values of $\{h\}$ at time $(t + \Delta t)$ with Δt being the time step size, and the stiff matrix $[C]$ and the load vector $\{L\}$ are given as

$$[C] = [M]/\Delta t + w[S] \ , \qquad (58)$$

and

$$\{L\} = \{D\} + \{Q\} + ([M]/\Delta t - (1 - w)[S])\{h\}_t \ , \qquad (59)$$

in which w is the time weighting factor and $\{h\}_t$ is the values of $\{h\}$ at time t. When w = 0, the time integration is explicit. When w = 0.5, it is the Crank-Nicolson central difference. For the implicit (or backward) difference, w is equal to 1.0.

3.3 Numerical Implementation of Boundary Conditions

To incorporate the boundary conditions, we have to evaluate Eq. (56) for every boundary segment e to yield the load vector $\{B^e\}$. For the Cauchy boundary condition given by Eq. (37), we simply substitute Eq. (37) into Eq. (56) to yield a boundary-element column vector $\{B_c^e\}$ for a Cauchy segment:

$$\{B_c^e\} = \{q_c^e\} \ , \qquad (60)$$

where $\{q_c^e\}$ is the Cauchy boundary flux vector given by

$$q_{c\alpha}^e = - \int_{B_e} N_\alpha^e q_C \, dB, \qquad \alpha = 1, 2, 3, 4. \qquad (61)$$

This Cauchy boundary flux vector represents the normal fluxes through the two nodal points of the segment B_e on B_C.

The surface integration of Eq. (61) in three-dimensional space is not as straightforward as the line integration in two-dimensional space. This integration requires further elaboration. Any surface integral of a continuous function F(x,y,z) specified on the surface S (Fig. 4) can be reduced to the area integration. Let I represent the surface integral:

ORNL-DWG 87-9581

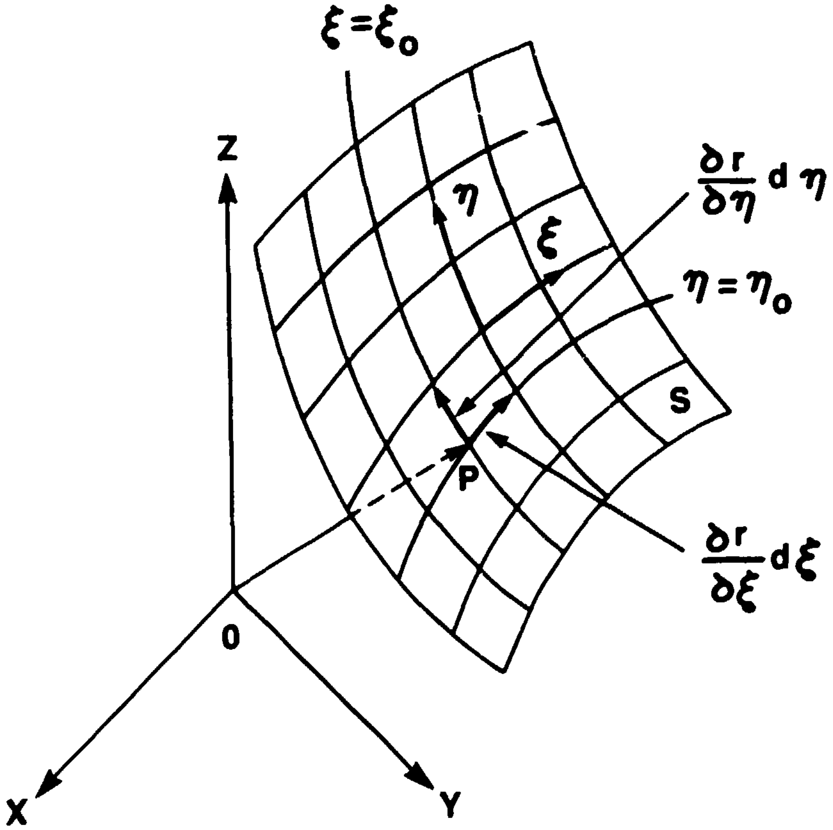

Fig. 4. A surface area and its imbedded local coordinate

$$I = \int_S F(x,y,z)dS \quad , \tag{62}$$

where

$$z = f(x,y) \tag{63}$$

is the equation for the surface S. Also, let P be any point on the surface S with coordinates (x,y,z) or (ξ,η) (Fig. 4). Then the vector **r** from O to P is given by

$$\mathbf{r} = x\,\mathbf{i} + y\,\mathbf{j} + z\,\mathbf{k} \quad . \tag{64}$$

The tangent vectors to the coordinate curves $\xi = \xi_o$ and $\eta = \eta_o$ on the surface S are $\partial\mathbf{r}/\partial\eta$ and $\partial\mathbf{r}/\partial\xi$, respectively (Fig. 4). The area dS of Fig. 4 is given by

$$dS = \left| \frac{\partial \mathbf{r}}{\partial \xi} \;X\; \frac{\partial \mathbf{r}}{\partial \eta} \right| d\xi\, d\eta \quad , \tag{65}$$

where X represents vector multiplication. But

$$\frac{\partial \mathbf{r}}{\partial \xi} \;X\; \frac{\partial \mathbf{r}}{\partial \eta} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{vmatrix} \tag{66}$$

so that

$$dS = \quad J_x^2 + J_y^2 + J_z^2 \;\; d\xi d\eta \quad , \tag{67}$$

where

$$J_z = \begin{vmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{vmatrix} \quad , \qquad (68a)$$

$$J_y = \begin{vmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial z}{\partial \xi} \\ \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial z}{\partial \eta} \end{vmatrix} \quad , \qquad (68b)$$

$$J_x = \begin{vmatrix} \dfrac{\partial z}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \\ \dfrac{\partial z}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{vmatrix} \quad . \qquad (68c)$$

Substituting Eq. (67) into Eq. (62), we obtain

$$\int_S F(x,y,z)\ dS = \int_S \Phi(\xi,\eta)\ \ J_x^2 + J_y^2 + J_z^2 \ \ d\xi d\eta \ , \qquad (69)$$

where

$$\Phi(\xi,\eta) = F[x(\xi,\eta),y(\xi,\eta),z(\xi,\eta)] \ . \qquad (70)$$

Using Eq. (69), we can easily evaluate Eq. (61) with the Gaussian quadrature to yield the numerical value of q_c^e .

For the Neumann boundary condition given by Eq. (36), we substitute Eq. (36) into Eq. (56) to yield a boundary-element column vector $\{B_n^e\}$ for a Neumann segment:

$$\{B_n^e\} = \{q_n^e\} \ , \tag{71}$$

where $\{q_n^e\}$ is the Neumann boundary flux vector given by

$$q_{n\alpha}^e = \int_{B_e} (N_\alpha^e \mathbf{n}\cdot K\cdot\nabla z \ - \ N_\alpha^e q_N)\, dB \ ; \quad \alpha = 1,\ 2,\ 3,\ 4, \tag{72}$$

which is independent of the pressure head. Again using Eq. (69), we can easily compute the integral in Eq. (72) to obtain the Neumann boundary flux $q_{n\alpha}^e$.

The implementation of a variable-type boundary condition is more involved. It normally occurs at the air-soil interface. During a precipitation period, we will assume that only seepage or infiltration can occur for any point on the air-soil interface. No evapotranspiration is allowed since it is not likely to happen. If seepage occurs, the Dirichlet boundary condition, Eq. (38), must be imposed. On the other hand, if infiltration occurs, either the Dirichlet boundary condition, Eq. (38), or the Cauchy boundary condition, Eq. (39), may be specified, depending on the soil properties and throughfall rate q_p in Eq. (39). The problem is which equation, Eq. (38) or Eq. (39), should be prescribed for a point on the boundary. This problem is settled by iteration. The procedure adopted is as follows. At each iteration, we examine the solution at each node along the variable boundary and test whether the existing boundary condition is still consistent. Specifically, if the existing condition is Eq. (39) (Cauchy boundary condition), we compute the pressure head at the boundary node. If the head is greater than the allowed ponding depth h_p in Eq. (38), too much water has been forced into the region through the node. In other words, the throughfall rate is greater than that which the medium can absorb. To account for this, the boundary condition is changed to Eq. (38), which in practice should result in infiltration at a rate less than q_p in Eq. (39) or result in seepage. If the computed head is less than the ponding depth, the medium is capable of absorbing all throughfall and no change of boundary condition is required. On the other hand, if the existing boundary condition is Eq. (38) (Dirichlet boundary condition), we compute Darcy's flux at the node. If the computed Darcy flux is going out of the region (seepage) or into the region (infiltration) but its magnitude is less than q_p in Eq. (39), no change of boundary condition is needed. However, if the computed Darcy flux is directed into the region (infiltration) with a rate greater than the throughfall rate q_p, a change of boundary condition to Eq. (39) is required since Eq. (38) would force more water than available into the region. By changing the boundary condition to Eq. (39), it should in practice result in a pressure head less than h_p. The iteration outlined above is discontinued when no changeover of boundary condition is encountered along the entire boundary.

Similarly, during a nonprecipitation period, we will assume that only evapotranspiration or seepage can occur and no infiltration is allowed. If seepage actually occurs at a node, Eq. (40) (Dirichlet boundary condition) must specified at the node. On the other hand, if evapotranspiration happens, either Eq. (41) (Dirichlet boundary condition) or Eq. (42) (Cauchy boundary condition) may be imposed at the node. The problem is again to determine which of the three equations should be used as boundary conditions. Iteration procedure is used to solve the problem. If the existing boundary condition is Eq. (40), we calculate Darcy's flux. When the computed Darcy flux is going out of the region, the existing boundary condition is consistent and no change on boundary condition is necessary. When the Darcy flux is directed into the region (remember no infiltration is allowed), the application of Eq. (40) implies the infiltration and prohibits evapotranspiration. Hence, the boundary condition is changed to Eq. (42), which in practice would generate evapotranspiration and would result in a pressure head lower than the ponding depth in Eq. (40). If the existing boundary condition is Eq. (41), we compute the Darcy flux. Since the minimum pressure is prescribed on the boundary, it is unlikely that this computed Darcy flux will be directed into the region. Thus, when the computed outgoing Darcy flux is less than q_e in Eq. (42), the existing boundary condition is consistent and no change on boundary condition is needed. When the computed Darcy flux is greater then q_e in Eq. (42), the application of Eq. (41) implies the imposition of too much suction at the node. Hence, the boundary condition is changed to Eq. (42), which in practice should result in a pressure head greater than h_m in Eq. (41). If the existing boundary condition is Eq. (42), we calculate pressure head at the node. If this computed pressure head is not lower than h_m in Eq. (41), the boundary condition is consistent and no change is required. However, if the computed head is lower than h_m in Eq. (41), the application of Eq. (42) implies that too much water is removed through the node yielding too low a pressure head. Hence, the boundary condition is changed to Eq. (41), which should yield an evapotranspiration rate less then q_e in Eq. (42). This iteration process is completed only when consistent boundary conditions have been applied to all nodes on the variable boundary.

During the iteration of boundary conditions on the variable boundary, one of the Eqs. (38) through (42) is used at a node. If either Eq. (39) or (42) is used, we substitute it into Eq. (56) to yield a boundary element column vector $\{B_v^e\}$ for a variable boundary segment:

$$\{B_v^e\} = \{q_v^e\} , \tag{73}$$

where $\{q_v^e\}$ is the variable boundary flux given by

$$q_{v\alpha}^e = - \int_{B_e} N_\alpha^e q_p \, dB, \quad \text{or} \quad q_{v\alpha} = - \int_{B_e} N_\alpha^e q_e \, dB; \quad \alpha = 1, 2, 3, 4 . \tag{74}$$

Assembling over all Neumann, Cauchy, and variable boundary segments, we obtain the global boundary column vector $\{B\}$ as

$$\{B\} = \{q\} , \tag{75}$$

in which

$$\{q\} = \sum_{e \epsilon N_{ne}} \{q_n^e\} + \sum_{e \epsilon N_{ce}} \{q_c^e\} + \sum_{e \epsilon N_{ve}} \{q_v^e\} , \tag{76}$$

where N_{ne}, N_{ce}, and N_{ve} are the number of Neumann boundary segments, Cauchy boundary segment, and variable boundary segments with flux conditions imposed on them, respectively.

Substituting Eq. (75) into Eq. (57) and dropping the subscript $(t + \Delta t)$ to simplify the notation, we obtain

$$[C]\{h\} = \{R\} , \tag{77}$$

where $\{R\}$ is given by

$$\{R\} = \{L\} + \{q\} . \tag{78}$$

At nodes where Dirichlet boundary conditions are applied, an identity equation is generated for each node and included in the matrices of Eq. (77). The detailed method of applying this type of boundary condition can be found elsewhere (Wang and Connor, 1975). The Dirichlet nodes include the nodes on the Dirichlet boundary and the nodes on the variable boundary to which either Eq. (38), (40), or (41) is applied.

3.4 Solution of the Matrix Equations

The matrix equation, Eq. (77), is linear only when the entire region of interest is under saturated conditions. If part of the region is unsaturated, the governing equation, Eq. (32), is nonlinear, hence Eq. (77) is a nonlinear matrix equation. To solve it, some type of iterative procedure is required. The approach taken here is to make an initial estimate of the unknown $\{h^k\}$. Using this estimate, we then compute the coefficient matrix [C] and solve the linearized matrix equation by the method of linear algebra. The new estimate is now obtained by the weighted average of the new solution and the previous estimate:

$$\{h^{(k+1)}\} = \{h\} + (1 - \omega)\{h^k\} , \tag{79}$$

where $\{h^{(k+1)}\}$ is the new estimate, $\{h^k\}$ is the previous estimate, $\{h\}$ is the new solution, and ω is the iteration parameter. The procedure is repeated until the new solution $\{h\}$ is within a tolerance error. When the iteration parameter is greater than or equal to 0 but is less than 1, the iteration is termed under-relaxation. If $\omega = 1$, the method is the exact relaxation. For the cases when ω is greater than 1 but less than or equal to 2, the iteration is termed overrelaxation. The underrelaxation option should be used to overcome cases when nonconvergency or the slow convergence rate is due to fluctuation rather than due to "blowup" computations. Overrelaxation should be used to speed up the convergence rate when the rate of convergency decreases monotonically.

Two options are employed to solve the linearized matrix equation: one is the direct elimination method and the other is pointwise iteration. When the direct elimination method is used to solve the matrix equation, a single iteration loop is used to iterate the nonlinearity. However, when pointwise iterations are used, a double loop is required: the inner loop to solve the linearized equation and the outer loop to iterate the nonlinearity. Three options have been employed when the pointwise iteration method is used to solve the linearized matrix equation. These are the successive under-relaxation (SUR), Gauss-Seidel (G-S) iteration, and successive over-relaxation (SOR). These three methods are unified by a relaxation parameter, Ω. When the relaxation parameter is less than 1 but greater than or equal to 0, the method is termed SUR iteration. When Ω is equal to 1, the method is termed G-S iteration. If the relaxation parameter is greater than 1 but less than or equal to 2, the method is termed SOR iteration.

There are six options in 3DFEMWATER to solve the finite element equations. These are the combination of two ways to solve a given set of linearized algebraic equations and three ways of estimating the coefficient matrix.

3.5 Computation of Mass Balance

One of the most important aspects in numerical modeling of subsurface flow is to check the mass balance over the whole region. The error in mass balance provides a crude index on the accuracy and convergence of numerical computations. The mass balance over a region R enclosed by the boundary $B(x,y,z) = 0$ can be obtained by integrating Eq. (32):

$$F_V = \int_R \left(F \frac{\partial h}{\partial t} - q\right) dR \quad , \tag{80}$$

and

$$F_B = \int_B F_n dB \quad , \tag{81}$$

where F_V represents the net volumetric increasing rate of water in the region, F_B is the net volumetric flow rate through the entire boundary out from the region, and F_n is the outward normal flux. In fact, F_n can be defined as

$$F_n = - \mathbf{n} \cdot \mathbf{K} \cdot \nabla H \tag{82}$$

Having obtained the pressure head field by solving Eq. (77), one can integrate Eqs. (81) and (81) independently. If the solution for h is free of error, one would expect the sum of two integrals equal to zero. In the present report, the integral of Eq. (82) is broken into two parts:

$$F_P = \int_R F \frac{\partial h}{\partial t} dR \quad , \tag{83a}$$

and

$$F_s = \int_R q \, dR \; , \qquad (83b)$$

where F_P and F_s represent the volumetric rates due to head change and artificial sources, respectively. Similarly, the integral in Eq. (82) is broken into six parts:

$$F_D = \int_{B_D} F_n \, dB \; , \qquad (84a)$$

$$F_C = \int_{B_C} F_n dB \; , \qquad (84b)$$

$$F_N = \int_{B_N} F_n dB \; , \qquad (84c)$$

$$F_{VO} = \int_{B_{VO}} F_n dB \; , \qquad (84d)$$

$$F_{VI} = \int_{B_{VI}} F_n dB \; , \qquad (84e)$$

and

$$F_L = \int_{B-B_D-B_C-B_N-B_V} F_n dB \; , \qquad (84f)$$

in which F_D, F_C, F_N, F_{VO}, F_{VI}, and F_L represent fluxes through the

Dirichlet boundary B_D, the Cauchy boundary B_C, the Neumann boundary B_N, the part of the variable boundary B_{VO} with flow going out from the region, the part of the variable boundary B_{VI} with flow going into the region, and the unspecified boundary $B-B_D-F_C-B_N-B_V$.

For an exact solution, the sum of the net outgoing flux F_B across the entire boundary and the total volumetric increase rate F_V should be equal to zero. In addition, F_L should theoretically be equal to zero. However, in any practical simulation, F_B plus F_V will not be equal to zero, and F_L will be nonzero. Nevertheless, the mass balance computation should provide a means to check the numerical scheme and the consistency in the computer code.

4. DESCRIPTION OF THE COMPUTER PROGRAM

4.1 Purpose of 3DFEMWATER

There are two source codes for 3DFEMWATER. Both of these two codes are designed to solve the initial and boundary value problems given by Eqs. (32) and (34) through (42). We have used the subregional block iteration methods (Yeh et al. 1985) in one code (referred to as code BLI), which is listed in Appendix A. For the other code (referred to as code PTI), listed in Appendix B, we have used the point iteration method. The differences between BLI and PTI are: (1) in BLI two subroutines BLKITR and SOLVE are used to solve the matrix equation, whereas in PTI the subroutine PISS is used to solve the matrix equation; (2) for BLI five pointer arrays are generated in subroutine PAGEN, whereas for PTI only one pointer array is generated in subroutine PAGEN; (3) the controlling subroutine GW3D for BLI contains more variables than for PTI; and (4) the subroutine DATAIN for BLI needs to read additional data about subregional information, whereas that for PTI does not need to read those data.

To refresh our memory, we restate the governing equation and initial and boundary conditions that 3DFEMWATER is designed to solve:

Governing Equation

$$F \frac{\partial h}{\partial t} = \nabla \cdot [K_s K_r \cdot (\nabla h + \nabla z)] + q \ , \qquad (85)$$

where h is the pressure head, t is time, K_s is the saturated hydraulic conductivity tensor, K_r is the relative hydraulic conductivity or relative permeability, z is the potential head, q is the source and/or sink, and F is the water capacity given by

$$F = d\theta/dh \qquad (86)$$

after we have neglected the compressibility of the water and media. In Eq. (86) θ is the moisture content.

Initial Conditions

$$h = h_i(x,y,z) \qquad \text{in } R, \qquad (87)$$

where R is the region of interest and h_i is the prescribed initial condition, which can be defined or obtained by solving the steady state version of Eq. (85).

Boundary Conditions

Dirichlet Conditions:

$$h = h_d(x_b,y_b,z_b,t) \qquad \text{on } B_d . \qquad (88)$$

Neumann Conditions:

$$- \mathbf{n} \cdot K_s K_r \cdot \nabla h = q_n(x_b, y_b, z_b, t) \qquad \text{on } B_r \ . \tag{89}$$

Cauchy Conditions:

$$-\mathbf{n} \cdot (K_s K_r \cdot \nabla h + K_s K_r \cdot \nabla z) = q_c(x_b, y_b, z_b, t) \qquad \text{on } B_c \ . \tag{90}$$

Variable Conditions - During Precipitation Period:

$$h = h_p(x_b, y_b, z_b, t) \qquad \text{on } B_v \ , \tag{91a}$$

or

$$-\mathbf{n} \cdot (K_s K_r \cdot \nabla h + K_s K_r \cdot \nabla z) = q_p(x_b, y_b, z_b, t) \qquad \text{on } B_v \ . \tag{91b}$$

Variable Conditions - During Nonprecipitation Period:

$$h = h_p(x_b, y_b, z_b, t) \qquad \text{on } B_v \ , \tag{91c}$$

or

$$h = h_m(x_b, y_b, z_b, t) \qquad \text{on } B_v \ , \tag{91d}$$

or

$$-\mathbf{n} \cdot K_s K_r \cdot (\nabla h + \nabla z) = q_e(x_b, y_b, z_b, t) \qquad \text{on } B_v \ . \tag{91e}$$

where (x_b, y_b, z_b) is the spatial coordinate on the boundary; $\mathbf{n}$ is an outward unit vector normal to the boundary; h_d, q_n, and q_c are the prescribed Dirichlet functional value, Neumann flux, and Cauchy flux, respectively; B_d, B_n, and B_c are the Dirichlet, Neumann, and Cauchy boundary, respectively; B_v is the variable boundary; h_p is the allowed ponding depth and q_p is the throughfall of precipitation, respectively, on the variable boundary; h_m is the allowed minimum pressure on the variable boundary; and q_e is the allowed maximum evaporation rate on the variable boundary, which is the potential evaporation. Only one of Eqs. (91a) through (91e) is used at any point on the variable boundary at any one time.

4.2 Program Structure

The description in this section is for the computer code that employs the subregional block iteration method (BLI). Whenever the description is not valid for the code that uses the point iteration method (PTI), it is noted.

3DFEMWATER consists of a MAIN program, a section of BLOCK DATA for specifying the size of array dimensions, and 22 subroutines for Code BLI or 21 subroutines for Code PTI. The MAIN is used to specify the sizes of all arrays. The control and coordinate activity are performed by the subroutine GW3D. Figure 5 shows the structure of the program. The functions of these subroutines are described below.

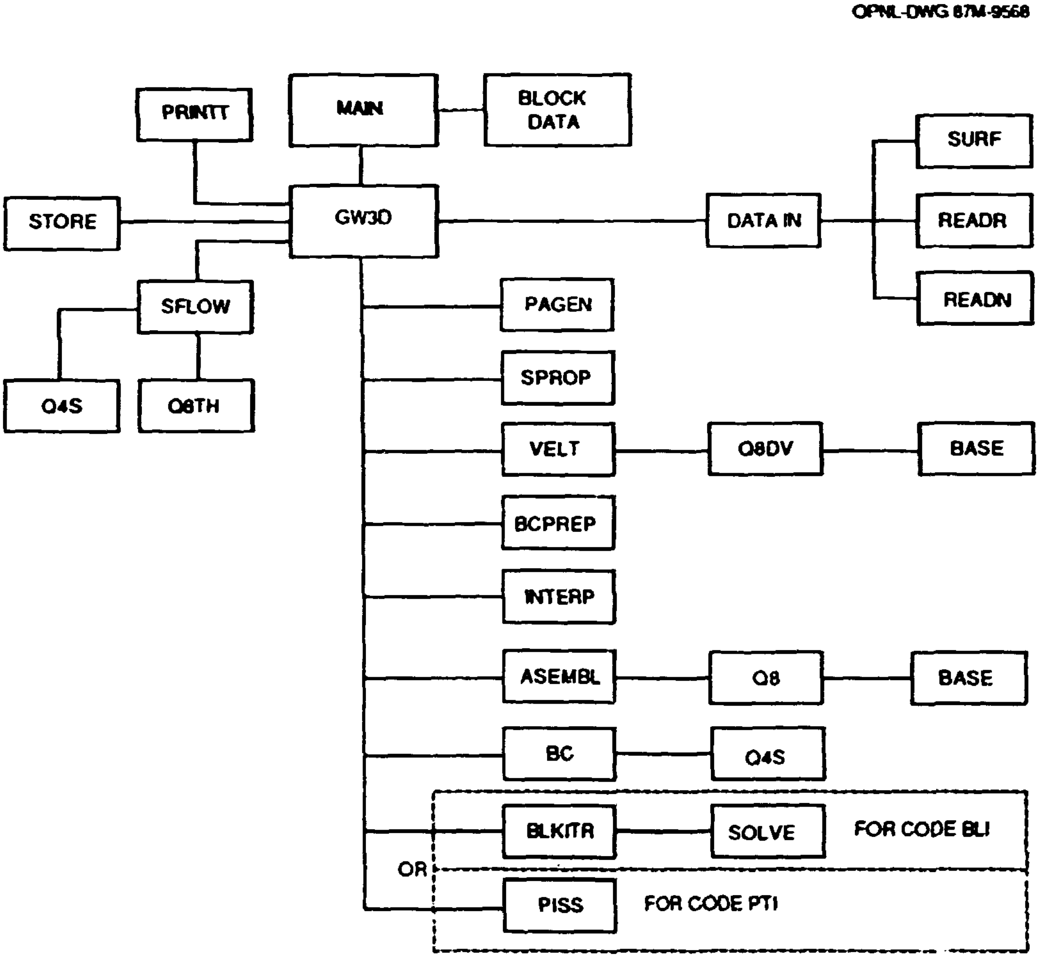

Fig. ?. Program structure of 3DFEMWATER

Subroutine GW3D

The subroutine GW3D controls the entire sequence of operations, a function generally performed by the MAIN program. It is, however, preferable to keep a short MAIN and several subroutines with variable storage allocation. This makes it possible to place most of the FORTRAN deck on a permanent file and to deal with a site-specific problem without making changes in array dimensions throughout all subroutines.

The subroutine GW3D will perform either the steady state computation alone (KSS = 0 and NTI = 0), or a transient state computation using the steady-state solution as the initial conditions (KSS = 0, NTI > 0), or a transient computation using user-supplied initial conditions (KSS = 1, NTI > 0). The flow chart of this subroutine is given in Fig. 6.

GW3D calls to subroutine DATAIN to read and print input data; subroutine PAGEN to generate pointer arrays; subroutine INTERP to obtain sources/sinks and boundary values; subroutine SPROP to obtain the relative hydraulic conductivity, water capacity, and moisture content from the pressure head; subroutine VELT to compute Darcy's velocity; subroutine BCPREP to determine if a change of boundary conditions is required; subroutine ASEMBL to assemble the element matrices over all elements; subroutine BC to implement the boundary conditions; subroutine BLKITR to form and solve the subregional block matrix equations for BLI or subroutine PISS to solve the matrix equation for PTI; subroutine SFLOW to

ORNL-DWG 87M-9567

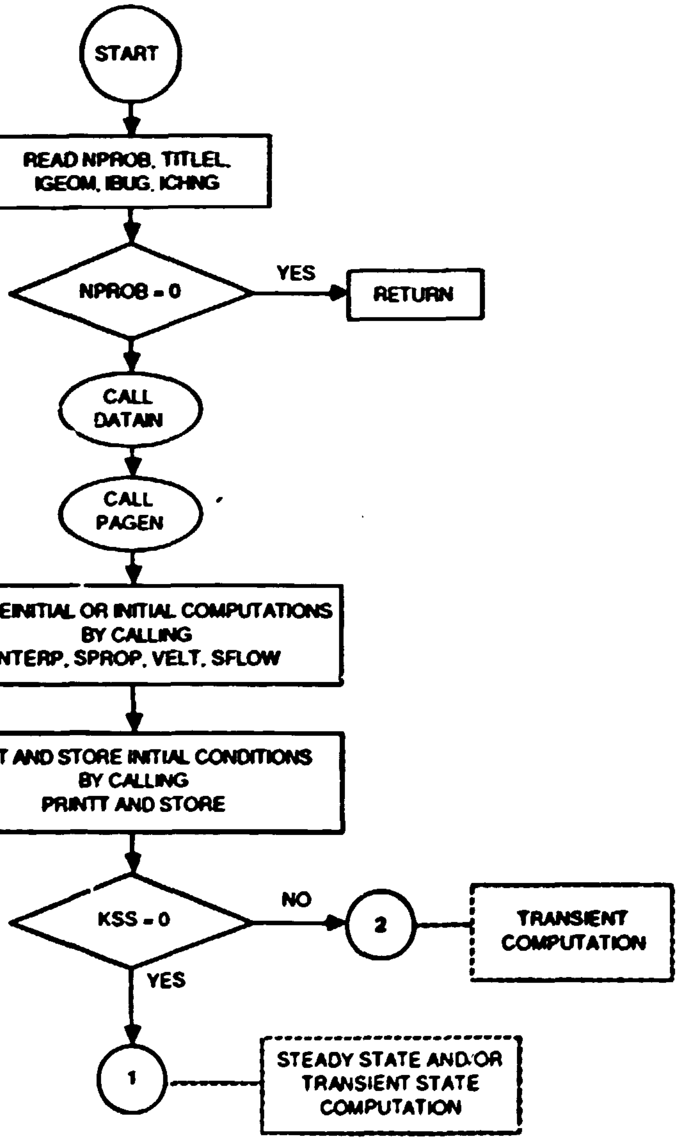

Fig. 6. Flow chart of subroutine GW3D (the first of three parts).

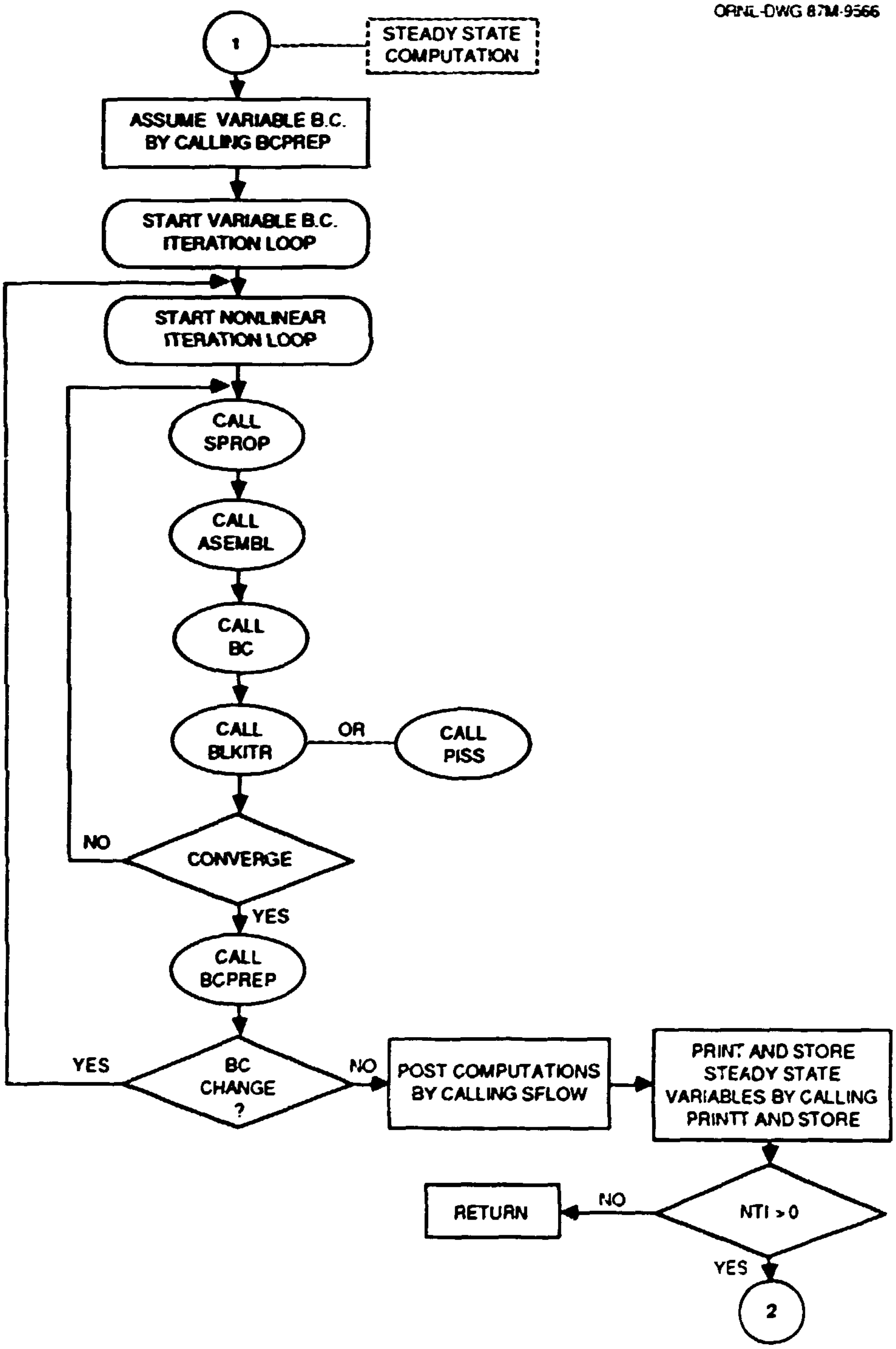

Fig. 6. Flow chart of subroutine GW3D (the second of three parts).

ORNL-DWG 87M-9565

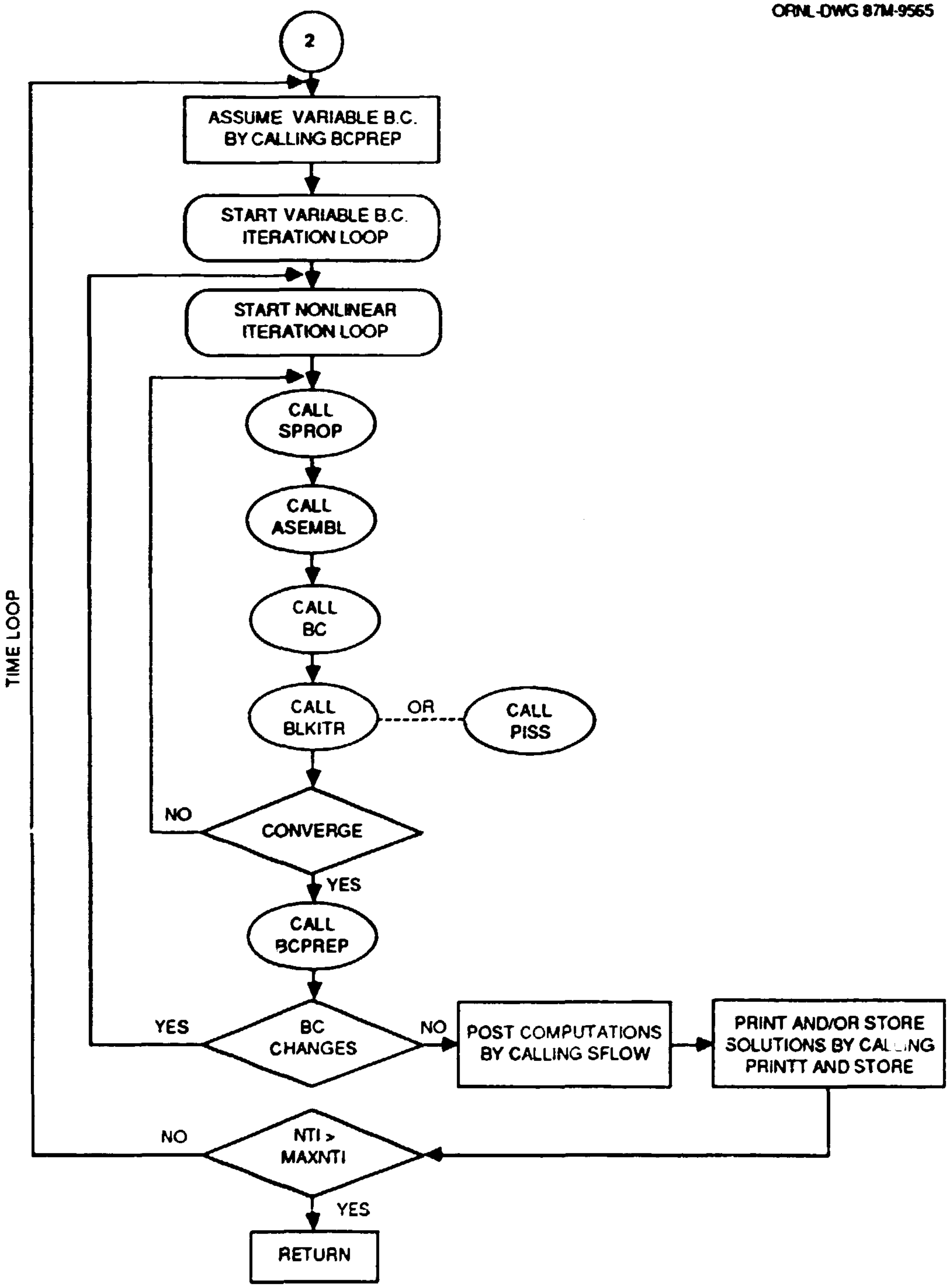

Fig. 6. Flow chart of subroutine GW3D (3 of 3) (the third of three parts).

calculate flux through all types of boundaries and water accumulated in the media; subroutine PRINTT to print out the results; and subroutine STORE to store the flow variables for input to a 3DFEMWASTE or for plotting (Fig. 5).

Subroutine DATAIN

Subroutine DATAIN reads all data input described in Sect. 5.3 except the card group 1. It also prints all the input information, and calls subroutine SURF to identify the surface elements and boundary nodes and subroutines READR and READN, respectively, to automatically generate real and integer numbers.

Subroutine SURF

Subroutine SURF identifies the boundary sides, sequences the boundary nodes, and computes the directional cosine of the surface sides. The mappings from boundary nodes to global nodes are stored in NPBB(I) (where NPBB(I) is the global node number of i-th boundary node). The boundary node numbers of the four nodes for each boundary side are stored in ISB(I,J) (where ISB(I,J) is the boundary node number of I-th node of J-th side, I = 1 to 4). There are six sides for each element. Which of these six sides is the boundary side is determined automatically in the subroutine SURF and is stored in ISB(5,J). The global element number, to which the J-th boundary side belongs, is also preprocessed in the subroutine SURF and is stored in ISB(6,J). The directional cosines of the J-th boundary side are computed and stored in DCOSB(I,J) (where DCOSB(I,J) is the directional cosine of the J-th surface with I-th coordinate, I = 1 to 3). The information contained in NPBB, ISB, and DOSB, along with the number of boundary nodes and the number of boundary sides is returned to subroutine DATAIN for other users.

Subroutine READR

This subroutine is called by the subroutine DATAIN to generate real numbers for data sets 7, 14(e), and 14(f) of Sect. 5.3. Automatic generation of regularly patterned data is built into this subroutine.

Subroutine READN

This subroutine is also called by the subroutine DATAIN to generate integer data sets 8, 9, 12(c), 12(f), 14(b) through 14(d), 15(c), 16(b), 16(c), 17(b), and 17(c) of Sect. 5.3.

Subroutine PAGEN

This subroutine is called by the controlling subroutine GW3D to preprocess pointer arrays that are needed to store the global matrix in compressed form and to construct the subregional block matrices. The pointer arrays automatically generated in this subroutine include the global node connectivity (stencil) GNOJCN(J,N), regional node connectivity LNOJCN(J,I,K), total node number for each subregion NTNPLR(K), bandwidth indicator for each subregion LMAXDF(K), and partial fill-up for the mapping array between global node number and local subregion node number GNPLR(I,k) with I = NNPLR(K) + 1 to NTNPLR(K). Here GNOJCN(J,N) is the global node number of J-th node connected to the global node N; LNOJN(J,I,K) is the local node number of J-th node connected to the local node I in K-th subregion; NTNPLR(K) is the total number of nodes in the K-th subregion, including the interior nodes, the global boundary nodes, and intra-boundary nodes; LMAXDF(K) is the maximum difference between any two nodes of any element in K-th subregion; and GNPLR(I,K) is the global node number of I-th local-region node in the K-th subregion. These pointer arrays are generated based on the element connectivity IE(M,J), the number of node for each subregion NNPLR(K), and the mapping between global node and local-region node GNLR(I,K) with I = 1, NNPLR(K). Here IE(M,J) is the global node number of J-th node of element M; NNPLR(K) is the number of nodes in the K-th subregion including the interior nodes and the global boundary nodes but not the intraboundary nodes.

Subroutine INTERP

This subroutine is called by the subroutine GW3D to compute the functional values (such as the Dirichlet total head, element source/sink, point source/sink, Neumann fluxes, Cauchy fluxes, and rainfall) at a particular time for all profiles. It uses the linear interpolation of the tabular data.

Subroutine SPROP

This subroutine calculates the values of moisture content, relative hydraulic conductivity, and the water capacity. Analytical functions are used. Thus, the users must supply the functional form.

Subroutine VELT

This subroutine calls Q8DV to evaluate the element matrices and the derivatives of the total head. It then sums over all element matrices to form a matrix equation governing the velocity components at all nodal points. To save computational time, the matrix is diagonalized by lumping. The velocity components can thus be solved point by point. The computed velocity field is then returned to GW3D through the argument. This velocity field is also passed to subroutine BCPREP to evaluate the Darcy flux across the seepage-infiltration-evapotranspiration surfaces.

Subroutine Q8DV

Subroutine Q8DV is called by the subroutine VELT to compute the element matrices given by

$$QB(I,J) = \int_{R_e} N_i^e N_j^e \, dR \quad , \tag{92a}$$

where N_i^e and N_j^e are the basis functions for nodal point i and j of element e, respectively. Subroutine Q8DV also evaluates the element load vector:

$$QRX(I) = - \int_{R_e} N_i^e \mathbf{i} \cdot K_s K_r \cdot (\nabla N_j^e) H_j \, dR \quad , \tag{92b}$$

$$QRY(I) = -\int_{R_e} N_i^e \mathbf{j}\cdot K_s K_r\cdot(\nabla N_j^e)H_j dR \ , \tag{92c}$$

$$QRZ(I) = -\int_{R_e} N_i^e \mathbf{k}\cdot K_s K_r\cdot(\nabla N_j^e)H_j dR \ , \tag{92d}$$

where

H_j = the total head at nodal point j,

$\mathbf{i}$ = the unit vector along the x-coordinate,

$\mathbf{j}$ = the unit vector along the y-coordinate,

$\mathbf{k}$ = the unit vector along the z-coordinate,

K_s = the saturated hydraulic conductivity tensor,

K_r = the relative hydraulic conductivity.

Subroutine BCPREP

This subroutine is called by GW3D to prepare the infiltration-seepage boundary conditions during a rainfall period or the seepage-evapotranspiration boundary conditions during non-rainfall periods. It decides the number of nodal points on the variable boundary to be considered as Dirichlet or Cauchy points. It computes the number of points that change boundary conditions from ponding depth (Dirichlet types) to infiltration (Cauchy types), or from infiltration to ponding depth, or from minimum pressure (Dirichlet types) to infiltration during rainfall periods. It also computes the number of points that change boundary conditions from potential evapotranspiration (Cauchy types) to minimum pressure, or from ponding depth to potential evapotranspiration, or from minimum pressure to potential evapotranspiration during non-rainfall periods. Upon completion, this subroutine returns the Darcy flux (DCYFLX), infiltration/potential evapotranspiration rate (FLX), the ponding depth nodal index (NPCON), the flux-type nodal index (NPFLX), the minimum pressure nodal index (NPMIN), and the number of nodal points (NCHG) that have changed boundary conditions.

Subroutine ASEMBL

This subroutine calls Q8 to evaluate the element matrices. It then sums over all element matrices to form a global matrix equation governing the pressure head at all nodes.

Subroutine Q8

This subroutine is called by the subroutine ASEMBL to compute the element matrix given by

$$QA(I,J) = \int_{R_e} N_i^e F \, N_j^e dR \, , \tag{93a}$$

$$QB(I,J) = \int_{R_e} (\nabla N_i^e) \cdot K_s K_r \cdot (\nabla N_j^e) \; dR \, , \tag{93b}$$

where F is the soil property function. Subroutine Q8 also calculates the element load vector given by

$$RQ(I) = \int_{R_e} [(\nabla N_i^e)\cdot K_s K_r \cdot (\nabla z) \; - \; N_i^e q] \, dR \quad , \qquad (93c)$$

where q is the source/sink.

Subroutine BASE

This subroutine is called by subroutines Q8DV and Q8 to evaluate the value of the base function at a Gaussian point. The computation is straightforward.

Subroutine BC

This subroutine incorporates Dirichlet, Cauchy, Neumann, and variable boundary conditions. For a Dirichlet boundary condition, an identity algebraic equation is generated for each Dirichlet nodal point. Any other equation having this nodal variable is modified accordingly to simplify the computation. For a Cauchy surface, the integration of the surface source is obtained by calling the subroutine Q4S, and the result is added to the load vector. For a Neumann surface, the integrations of both the gradient and gravity fluxes are obtained by calling the subroutine Q4S. These fluxes are added to the load vector. The subroutine BC also implements the variable boundary conditions. First, it checks all infiltration-evapotranspiration-seepage points, identifying any of them that are Dirichlet points. If there are Dirichlet points, the method of incorporating Dirichlet boundary conditions mentioned above is used. If a given point is not the Dirichlet point, the point is bypassed. Second, it checks all rainfall-evaporation-seepage points again to see if any of them is a Cauchy point. If it is a Cauchy point, then the computed flux by infiltration or potential evapotranspiration is added to the load vector. If a given point is not a Cauchy point, it is bypassed. Because the infiltration-evaporation-seepage points are either Dirichlet or Cauchy points, all points are taken care of in this manner.

Subroutine Q4S

This subroutine is called by the subroutines BC and SFLOW to compute the surface node flux of the type

$$RQ(I) = \int_{R_e} N_i^e q \, dB \ , \tag{94}$$

where q is either the Cauchy flux, Neumann flux, or gravity flux.

Subroutine BLKITR

This subroutine is called by the subroutine GW3D to solve the matrix equation with block iteration methods. For each subregion, a block matrix equation is constructed based on the global matrix equation and two pointer arrays GNPLR and LNOJCN (see subroutine PAGEN), and the resulting block matrix equation is solved with the direct band matrix solver by calling subroutine SOLVE. This is done for all subregions for each iteration until a convergent solution is obtained. This subroutine and the subroutine SOLVE, to be described in the next paragraph, are needed only for the code BLI.

Subroutine SOLVE

This subroutine is called by the subroutine BLKITR to solve for the matrix equation of the type

$$[C]\{x\} = \{y\} \, , \qquad (95)$$

where [C] is the coefficient matrix and $\{x\}$ and $\{y\}$ are two vectors. $\{x\}$ is the unknown to be solved, and $\{y\}$ is the known load vector. The computer returns the solution $\{y\}$ and stores it in $\{y\}$. The computation is a standard banded Gaussian direct elimination procedure.

Subroutine PISS

This subroutine is called by the subroutine GW3D to solve the matrix equation with point iteration methods. This subroutine is needed only for the code PTI.

Subroutine PRINTT

This subroutine is used to line-print the flow variables. These include the fluxes through variable boundary surfaces, the pressure head, total head, moisture content, and Darcy's velocity components.

Subroutine STORE

This subroutine is used to store the flow variables on Logical Unit 1. It is intended for use with a subsequent computer model 3DFEMWASTE, or for plotting. The information stored includes region geometry, subregion data, and hydrological variables such as pressure head, tal head, moisture content, and Darcy's velocity components.

Subroutine SFLOW

This subroutine is used to compute the fluxes through various types of boundaries and the increasing rate of water content in the region of interest. The function of FRATE(7) is to store the flux through the whole boundary enclosing the region of interest. It is given by

$$\text{FRATE}(7) = \int_B (V_x n_x + V_y n_y + V_z n_z)\, dB \ , \tag{96}$$

where B is the global boundary of the region of interest; V_x, V_y, and V_z are Darcy's velocity components; and n_x, n_y, and n_z are the directional cosines of the outward unit vector normal to the boundary B. FRATE(1) through FRATE(5) store the flux through Dirichlet boundary B_D, Cauchy boundary B_C, Neumann boundary B_N, the seepage/evapotranspiration boundary B_S, and infiltration boundary B_R, respectively, and are given by

$$\text{FRATE}(1) = \int_{B_D} (V_x n_x + V_y n_y + V_z n_z)\, dB \ , \tag{97a}$$

$$\text{FRATE}(2) = \int_{B_C} (V_x n_x + V_y n_y + V_z n_z)\, dB \ , \tag{97b}$$

$$\text{FRATE}(3) = \int_{B_N} (V_x n_x + V_y n_y + V_z n_z)\, dB \ , \tag{97c}$$

$$\text{FRATE}(4) = \int_{B_S} (V_x n_x + V_y n_y + V_z n_z)\ dB\ , \tag{97d}$$

$$\text{FRATE}(5) = \int_{B_R} (V_x n_x + V_y n_y + V_z n_z)\ dB\ . \tag{97e}$$

FRATE(6), which is related to the numerical loss, is given by

$$\text{FRATE}(6) = \text{FRATE}(7) - \sum_{I=1}^{5} \text{FRATE}(I)\ . \tag{98}$$

FRATE(8) and FRATE(9) are used to store the source/sink and increased rate of water within the media, respectively:

$$\text{FRATE}(8) = -\int_R q\ dR\ , \tag{99}$$

and

$$\text{FRATE}(9) = \int_R F\,\frac{\partial h}{\partial t}\ dR\ . \tag{100}$$

If there is no numerical error in the computation, the following equation should be satisfied:

$$\text{FRATE}(9) = -\ [\text{FRATE}(7) + \text{FRATE}(8)]\ , \tag{101}$$

and FRATE(6) should be equal to zero. Equation (98) simply states that the negative rate of water going out from the region through the entire boundary and due to a source/sink is equal to the rate of water accumulated in the region.

Subroutine Q8TH

This subroutine is used to compute the contribution of the increasing rate of the water content from an element e

$$QTHP = \int_{R_e} \frac{d\theta}{dh} \frac{\partial h}{\partial t} \, dR \ . \tag{102}$$

The computation of the above integration is straightforward.

5. USER'S MANUAL

The following describes what one should do about 3DFEMWATER code for each site-specific application and how data input should be prepared.

5.1 Specifications of Array Dimensions

The subscripted variable arrays have to be dimensioned for each site-specific problem in the MAIN program (see example problems). Seventy-eight (78) array variables contained between lines MAIN 130 and MAIN 305 in the source code (Appendix A) should be dimensioned according to the indicated subscripts (a total of 35). The listing and definitions of these parameters and their associated subscripts (maximum-control numbers) are given below:

Arrays for Node Coordinates and Element Connectivity

X(MAXNP), Y(MAXNP), Z(MAXNP), IE(MAXEL,9);

MAXEL - maximum number of elements,
MAXNP - maximum number of nodes;

X(NP) - x-coordinate of NP-th nodal point,
Y(NP) - y-coordinate of NP-th nodal point,
Z(NP) - z-coordinate of NP-th nodal point,
IE(M,I) - global node number of the I-th node of the M-th element if I is between 1 and 8, integer to indicate the material type of the M-th element if I is equal to 9.

Arrays to Store the Global Matrix, Global Stencil, and Load Vector

CMATRX(MAXNP,JBAND), GNOJCN(JBAND,MAXNP), RLD(MAXNP), RI(MAXNP), RL(MAXNP);

JBAND - maximum number of nonzero elements in any row of the matrix;

CMATRIX(NP,I) - an array to store the assembled global matrix,

GNOJCN(I,NP) - global node number of I-th node connected to node NP,
RLD(NP) - an array to store the assembled global load vector,
RI(NP) - pressure head iterate in subroutine BLKITR,
RL(NP) - a working array to contain final solution for the pressure head in subroutine BLKITR.

Arrays for Subregions (These arrays are not needed for Code PTI)

NTNPLR(MXREGN), LMAXDF(MXREGN), NNPLR(MXREGN),
GNLR(LTMXNP,MXREGN), LNOJCN(JBAND,LMXNP,MXREGN),
CMTRXL(LMXNP,LMXBW), RLDL(LMXNP);

MXREGN - maximum number of subregions,
LTMXNP - maximum number of total nodal points in any subregion, including interior nodes, global boundary nodes, and intraboundary nodes,
LMXBW - maximum number of the bandwidth in any subregion;

NTNPLR(K) - total number of nodes for K-th subregion including interior and global boundary and intra-boundary nodes,
LMAXDF(K) - maximum difference between any two nodes of any element in K-th subregion,
NNPLR(K) - number of interior nodes for K-th subregion, including global boundary nodes,
GNLR(I,K) - global nodal number of I-th local nodal number for K-th subregion,
LNOJCN(J,I,K) - local node number of J-th connecting node surrounding I-th local node for K-th subregion,
CMTRXL(NP,I) - assembled matrix for a subregion,
RLDL(NP) - assembled load vector for a subregion.

Arrays for Hydrological Variables and Nonconvergent Nodes

H(MAXNP), HP(MAXNP), HW(MAXNP), HT(MAXNP),
VX(MAXNP), VY(MAXNP), VZ(MAXNP),
TH(8,MAXEL), DTH(8,MAXEL), AKR(8,MAXEL)
NPCNV(MAXNP);

H(NP) - pressure head at NP-th node,
HP(NP) - pressure head of NP-th node at previous time,
HW(NP) - nonlinear pressure head iterate of NP-th node,
HT(NP) - total head of NP-th node,
VX(NP) - x-component velocity at NP-th node,
VY(NP) - y-component velocity at NP-th node,
VZ(NP) - z-component velocity at NP-th node,
TH(I,M) - water content at I-th gaussian point of M-th element,
DTH(I,M) - dTH/dH - water capacity of I-th gaussian point of M-th element,
AKR(I,M) - relative hydraulic conductivity at I-th gaussian point of M-th element,
NPCNV(NP) - nodal point of NP-th nonconvergent node.

Arrays for Boundary Surfaces

DCOSB(3,MAXBES), ISB(6,MAXBES), NPBB(MAXBNP),
BFLX(MAXBNP), BFLXP(MAXBNP);

MAXBES - maximum number of boundary-element surfaces,
MAXBNP - maximum number of boundary nodal points;

DCOSB(1,I) - x-directional cosine of I-th boundary element surface,
DCOSB(2,I) - y-directional cosine of I-th boundary element surface,
DCOSB(3,I) - z-directional cosine of I-th boundary element surface,
ISB(1,I) - boundary node number of the first node of I-th boundary element surface,
ISB(2,I) - boundary node number of the second node of I-th boundary element surface,
ISB(3,I) - boundary node number of the third node of I-th boundary element surface,
ISB(4,I) - boundary node number of the fourth node of I-th boundary element surface,
ISB(5,I) - element side index of I-th boundary element surface,
1 - left side, 2 - front side, 3 - right side,
4 - back side, 5 - bottom side, 6 - top side,
ISB(6,I) - element number to which the I-th boundary element-surface belongs,
NPBB(I) - global node number of I-th boundary node,
BFLX(I) - boundary flux at I-th boundary node,
BFLXP(I) - boundary flux at previous time of I-th boundary node.

Arrays for Element (Distributed) and Well (Point) Sources/Sinks

SOS(MXSPR), SOSF(MXSDP,MXSPR), TSOSF(MXSDP,MXSPR),
ISTYP(MXSEL), MSEL(MXSEL),
WSS(MXWPR), WSSF(MXWDP,MXWPR), TWSSF(MXWDP,MXWPR),
IWTYP(MXWNP), NPW(MXWNP);

MXSEL - maximum number of source elements,
MXSPR - maximum number of source profiles,
MXSDP - maximum number of data points on each well source/sink profile,
MXWNP - maximum number of well nodal points,
MXWPR - maximum number of well source/sink profiles,
MXWDP - maximum number of data points on each well source/sink profile;

SOS(I) - value of I-th element source/sink profile,
SOSF(I,J) - source/sink strength of I-th data point in J-th element source/sink strength vs time profile,
TSOSF(I,J) - time of I-th data point in J-th element source/sink strength vs time profile,
MSEL(M) - element number of M-th source/sink element,
ISTYP(M) - source/sink type assigned to M-th source/sink element,
WSS(I) - value of I-th well source/sink profile at present time,
WSSF(I,J) - source/sink strength of I-th data point in J-th

well source/sink strength vs time profile,
TWSSF(I,J) - time of I-th data point in J-th well source/sink strength vs time profile,
NPW(N) - global node number of N-th source/sink well,
IWTYP(N) - source/sink type assigned to N-th source/sink well.

Arrays for Cauchy Boundary Conditions

QCB(MXCPR), QCBF(MXCDP,MXCPR), TQCBF(MXCDP,MXCPR), ICTYP(MXCES), ISC(5,MXCES), NPCB(MXCNP);

MXCNP - maximum number of Cauchy nodal points,
MXCES - maximum number of Cauchy element surfaces,
MXCPR - maximum number of Cauchy-flux profiles,
MXCDP - maximum number of data points on each Cauchy-flux profile;

QCB(I) - value of I-th Cauchy flux profile at the present time,
QCBF(I,J) - flux of I-th data point in J-th Cauchy flux vs time profile,
TQCBF(I,J) - time of I-th data point in J-th Cauchy flux vs time profile,
ICTYP(MP) - type of Cauchy flux vs time profile assigned to MP-th Cauchy boundary surface,
NPCB(NP) - global node number of NP-th Cauchy node for inputting, then is changed to contain boundary node number,
ISC(1,MP) - global node number of the first node of MP-th Cauchy boundary surface,
ISC(2,MP) - global node number of the second node of MP-th Cauchy boundary surface,
ISC(3,MP) - global node number of the third node of MP-th Cauchy boundary surface,
ISC(4,MP) - global node number of the fourth node of MP-th Cauchy boundary surface,
ISC(5,MP) - boundary side number of MP-th Cauchy boundary surface,

Arrays for Neumann Boundary Conditions

QNB(MXNPR), QNBF(MXNDP,MXNPR), TQNBF(MXNDP,MXNPR), INTYP(MXNES), ISN(5,MXNES), NPNB(MXNNP);

MXNNP - maximum number of Neumann nodal points,
MXNES - maximum number of Neumann element surfaces,
MXNPR - maximum number of Neumann-flux profiles,
MXNDP - maximum number of data points on each Neumann-flux profile;

QNB(I) - value of I-th Neumann flux profile at present time,
QNBF(I,J) - flux of I-th data point in J-th Neumann flux vs time profile,
TQNBF(I,J) - time of I-th data point in J-th Neumann flux vs time profile,

INTYP(MP) - type of Neumann flux vs time profile assigned to MP-th Neumann boundary surface,

NPNB(NP) - global node number of NP-th Neumann node for inputting, then is changed to contain boundary node number,

ISN(1,MP) - global node number of the first node of MP-th Neumann boundary surface,

ISN(2,MP) - global node number of the second node of MP-th Neumann boundary surface,

ISN(3,MP) - global node number of the third node of MP-th Neumann boundary surface,

ISN(4,MP) - global node number of the fourth node of MP-th Neumann boundary surface,

ISN(5,MP) - boundary side number of MP-th Neumann boundary surface.

Arrays for Variable Boundary Conditions

RFALL(MXRPR), RF(MXRDP,MXRPR), TRF(MXRDP,MXRPR),
IRTYP(MXVES), ISV(5,MXVES), NPVB(MXVNP),
DCYFLX(MXVNP), FLX(MXVNP), HCON(MXVNP), HMIN(MXVNP),
NPFLX(MXVNP), NPCON(MXVNP), NPMIN(MXVNP);

MXVES - maximum number of variable element surfaces,

MXVNP - maximum number of variable nodal points,

MXRPR - maximum number of rainfall profiles,

MXRDP - maximum number of data point on each rainfall profile;

RFALL(I) - value of I-th rainfall via time profile at present time,

TRF(I,J) - time of I-th data point on J-th rainfall vs time profile

RF(I,J) - rainfall value of I-th data point on J-th rainfall vs time profile,

IRTYP(MP) - type of rainfall profile assigned to MP-th variable boundary surface,

ISV(1,MP) - global node number of the first node of the MP-th variable element surface for inputting, then is changed to contain compressed variable boundary node number,

ISV(2,MP) - global node number of the second node of the MP-th variable element surface for inputting, then is changed to contain compressed variable boundary node number,

ISV(3,MP) - global node number of the third node of the MP-th variable element surface for inputting, then is changed to contain compressed variable boundary node number,

ISV(4,MP) - global node number of the fourth node of the MP-th variable element surface for inputting, then is changed to contain compressed variable boundary node number,

ISV(5,MP) - boundary side number of the MP-th variable element surface,

NPVB(NP) - global node number of the NP-th variable boundary node for inputting, ther is changed to contain boundary node number,
DCYFLX(NP) - Darcy flux through the NP-th variable boundary node,
FLX(NP) - rainfall rate through the NP-th variable boundary node,
HCON(NP) - ponding depth allowed for the NP-th variable boundary node,
HMIN(NP) - minimum pressure head allowed for the NP-th variable boundary node,
NPFLX(NP) - Cauchy boundary condition indicator for the NP-th variable node,
NPCON(NP) - ponding depth Dirichlet boundary condition indicator for the NP-th variable node,
NPMIN(NP) - minimum pressure head Dirichlet boundary condition indicator for the NP-th variable boundary node.

Arrays for Dirichlet Boundary Conditions

HDB(MXDPR), HDBF(MXDDP,MXDPR), THDBF(MXDDP,MXDPR), IDTYP(MXDNP), NPDB(MXDNP);

MXDNP - maximum number of Dirichlet nodal points,
MXDPR - maximum number of Dirichlet total head profiles,
MXDDP - maximum number of data points on each Dirichlet profile;

HDB(I) - value of total head at the present time of I-th total head vs time profile for Dirichlet boundary,
HDBF(I,J) - total head of I-th data point in J-th Dirichlet total head vs time profile,
THDBF(I,J) - time of I-th data point in J-th Dirichlet total head vs time profile,
IDTYP(NP) - type of Dirichlet total head vs time profile assigned to NP-th Dirichlet node,
NPDB(NP) - global node number of NP-th Dirichlet node for inputting, then is changed to contain boundary node number.

Arrays for Material Property Array and Material Characteristics

PROP(MXMPPM,MAXMAT), THPROP(MXSPPM,MAXMAT), AKPROP(MXSPPM,MAXMAT);

MAXMAT - maximum number of material types (medium formations),
MXSPPM - maximum number of soil parameters per material to describe soil characteristic curves,
MXMPPM - maximum number of material properties per material;

PROP(J,I) - J-th material property of I-th materials,
J = 1 - saturated x-hydraulic conductivity or permeability,
J = 2 - saturated y-hydraulic conductivity or permeability,
J = 3 - saturated z-hydraulic conductivity or permeability,

J = 4 - saturated xy-hydraulic conductivity or permeability,
J = 5 - saturated xz-hydraulic conductivity or permeability,
J = 6 - saturated yz-hydraulic conductivity or permeability,

THPROP(J,I) - J-th parameter to describe the moisture content as function of pressure head for I-th material,
AKPROP(J,I) - J-th parameter to describe the relative hydraulic conductivity as a function of pressure head for I-th material.

Arrays for Output Control and Time Step Size Change

KPR(MAXNTI), KDSK(MAXNTI), TDTCH(MXNDTC);

MAXNTI - maximum number of time steps,
MXNDTC - maximum number of DELT changes;

KPR(I) - line print control for I-th time step,
KDSK(I) - disk storage control for I-th time step,
TDTCH(I) - time of I-th time to reset time step size - DELTO.

The central process unit (CPU) memory requirements in words for all arrays can be computed as

$$\begin{aligned}\text{Real Storage} = {} & [(13 + \mathrm{JBAND})*\mathrm{MAXNP}] + (24*\mathrm{MAXEL}) \\ & + (3*\mathrm{MAXBES} + 2*\mathrm{MAXBNP}) + [(1 + \mathrm{LMXBW})*\mathrm{LMXNP}] \\ & + [(1 + 2*\mathrm{MXSDP})*\mathrm{MXSPR}] + [(1 + 2*\mathrm{MXWDP})*\mathrm{MXWPR}] \\ & + [(1 + 2*\mathrm{MXCDP})*\mathrm{MXCPR}] + [(1 + 2*\mathrm{MXNDP})*\mathrm{MXNPR}] \\ & + [(1 + 2*\mathrm{MXRDP})*\mathrm{MXRPR} + 4*\mathrm{MXVNP}] + [(1 + 2*\mathrm{MXDDP})*\mathrm{MXDPR}] \\ & + [(\mathrm{MXMPPM} + 2*\mathrm{MXSPPM})*\mathrm{MAXMAT}] \ , \qquad (103)\end{aligned}$$

$$\begin{aligned}\text{Integer Storage} = {} & [(1 + \mathrm{JBAND})*\mathrm{MAXNP}] + (9*\mathrm{MAXEL}) + (6*\mathrm{MAXBES} + \mathrm{MAXBNP}) \\ & + [(\mathrm{JBAND}*\mathrm{LMXNP} + \mathrm{LTMXNP}) + 3]*\mathrm{MXREGN} \\ & + [(2*\mathrm{MXSEL}) + (2*\mathrm{MXWNP})] + (6*\mathrm{MXCES} + \mathrm{MXCNP}) \\ & + (6*\mathrm{MXNES} + \mathrm{MXNNP}) + (6*\mathrm{MXVES} + 4*\mathrm{MXVNP}) \\ & + (2*\mathrm{MXDNP}) + (2*\mathrm{MAXNTI} + \mathrm{MXNDTC}) \ . \qquad (104)\end{aligned}$$

The subscripts (maximum-control numbers) of the above 78 arrays should be specified in the BLOCK DATA in the source program. One should always assign correct numbers to these subscripts so that the program can function properly. If the program fails to function, it is most likely because either the arrays mentioned above are not properly dimensioned or the numbers assigned in the BLOCK DATA to these specification numbers of the arrays are not correct. In the following, we demonstrate how to dimension the above 78 arrays in the MAIN and how to specify the maximum-control numbers in the BLOCK DATA with an example.

Let us assume that a region of interest is discretized by 30 x 20 x 10 nodes and 29 x 19 x 9 elements. In other words, we are discretizing the region with 30 nodes along the longitudinal or x-direction, 20 nodes along the lateral or y-direction, and 10 nodes along the vertical or z-direction. Since we have a total of 30 x 20 x 10 - 60,00 nodes, the maximum number of nodes is MAXNP - 6000. The total number of elements is 29 x 19 x 9 - 4,959, i.e, MAXEL - 4959. For this simple discretization problem, the maximum connecting number to any of the 6,000 nodes in the region of interest is 27, i.e., JBAND - 27. There will be 29 x 19 - 551 element surfaces each on the bottom and top faces of the region, 29 x 9 - 261 element-surfaces each on the front and back faces of the region, and 19 x 9 - 171 element-surfaces each on the left and right faces of the region. Thus, there will be a total of 1966 element-surfaces, i.e., MAXBES - 1966. Similarly, we can compute the surface-boundary nodes to be 1968, i.e., MAXBNP - 1968. With these maximum-control numbers specified, the arrays related to the global region, global boundary, and hydrologic variables can be dimensioned as

```
DIMENSION X(6000),Y(6000),Z(6000),IE(4959,9)
DIMENSION CMATRX(6000,27),GNOJCN(27,60000),RLD(6000),
DIMENSION RI(6000),RL(6000)
DIMENSION H(6000),HP(6000),HW(6000),HT(6000),
DIMENSION VX(6000),VY(6000),VZ(6000)
DIMENSION TH(8,4959),DTH(8,4959),AKR(8,4959),NPCNV(6000)
DIMENSION DCOSB(3,1966),ISB(6,1966),NPBB(1968),
DIMENSION BFLX(1968),BFLXP(1968)
```

In order to dimension the pointer arrays and the subregion data, we have to know how the region of interest is subdivided into subregions. Let us assume we have subdivided the region of interest into 20 subregions, each subregion has 30 x 10 nodes. It is seen, in fact, we are taking a vertical slice as a subregion. For this subregionalization, we have MXREGN = 20. Each subregion has 30 x 10 = 300 nodes, resulting LMXNP = 300. It is also seen that there will be 600 intraboundary nodes, 300 nodes each on the two neighboring slices of a subregion. Thus, we have LTMXNP = 900. For each subregion, the maximum bandwidth can be computed as LMXBW = 23 if the nodes are labelled along the z-directions consecutively. From these discussions, we can dimension the following arrays as

```
DIMENSION NTNPLR(20),NNPLR(20),LMAXDF(20)
DIMENSION GNLR(900,20),LNOJCN(27,300,20)
DIMENSION CMTRXL(300,23),RLDL(300)
```

We will assume that there will be a maximum of 11 elements that have the distributed sources/sinks (i.e., MXSEL - 11) and a maximum of 10 nodal points that can be considered as well sources/sinks (i.e. MXWNP - 10). We will also assume that there will be three different distributed source/sink profiles and five distinct point source/sink profiles. Then we will have MXSPR - 3 and MXWPR - 5. Let us further assume that four data points are needed to describe the distributed source/sink profiles as a function of time and that 8 data points are required to described point source/sink profiles (i.e., MXSDP - 4 and MXWDP - 8). With these assumptions, we can now dimension the arrays related to source/sink conditions

```
      DIMENSION SOS(3),SOSF(4,3),TSOSF(4,3),ISTYP(11),MSEL(11)
      DIMENSION WSS(5),WSSF(8,5),TWSSF(8,5),IWTYP(10),NPW(10)
```

To specify arrays for boundary conditions, let us assume that the top face is a variable boundary (i.e., on the air-soil interface, either ponding, infiltration, or evapotranspiration may take place). On the left face, fluxes from the adjacent aquifer are known. On the right face, the total head is assumed known. On the bottom face, natural drainage is assumed to occur (i.e., the gradient of the pressure head can be assumed zero).

There are 20 x 10 - 200 nodes on the left face and 19 x 9 - 171 element surfaces; thus MXCNP - 200 and MXCES - 171. It is further assumed that there two different fluxes going into the region through the left face and that each flux can be described by four data points as a function of time (i.e., MXCPR - 2, and MXCDP - 4). The Cauchy boundary condition arrays can be dimensioned as

```
DIMENSION QCB(2),QCBF(4,2),TQCBF(4,2)
DIMENSION ICTYP(171),ISC(5,171), NPCB(200)
```

On the bottom surface, there are 30 x 20 - 600 nodes and 29 x 19 - 551 surface elements. Since the gradient of pressure head on the bottom surface is zero, there is only one Neumann flux profile, and two data points, one at zero time and the other at infinite time, are sufficient to describe the constant value of zero. Hence, we have MXNNP - 600, MXNES - 551, MXNPR - 1, and MXNDP - 2; and the Neumann boundary condition arrays can be dimensioned as

```
DIMENSION QNB(1),QNBF(2,1),TQNBF(2,1)
DIMENSION INTYP(551),ISN(5,551),NPNB(600)
```

On the top face, there will be 30 x 20 - 600 nodes and 29 x 19 - 551 surface elements. Let us assume that there are three different rainfall intensities that might be fall on the air-soil interface, and that each rainfall intensity is a function of time and can be described by 24 data points. With these descriptions, we have MXVNP - 600, MXVES - 551, MXRPR - 3, and MXRDP - 24; and the variable boundary condition arrays can be specified as

```
DIMENSION RFALL(3),RF(24,3),TRF(24,3)
DIMENSION IRTYP(551),ISV(5,551),NPVB(600)
DIMENSION DCYFLX(600),FLX(600),HCON(600),HMIN(600)
DIMENSION NPFLX(600),NPCON(600),NPMIN(600)
```

On the right face, there are 20 x 10 - 200 nodes. Let us assume that there are twenty different values of the total head, one each on a vertical line of the right face. We further assume that each of these twenty total head can be described by 8 data points as function of time. We then have MXDNP = 200, MXDPR = 20, and MXDDP = 8; and the Dirichlet boundary condition arrays can be dimensioned as

```
DIMENSION HDB(20),HDBF(8,20),THDBF(8,20),IDTYP(200),NPDB(200)
```

In this report, we have six material properties (six saturated hydraulic conductivity components) per material. We will assume that the whole region of interest is composed of three different kinds of materials. The characteristic curves of each material are assumed to be described by four parameters. We then have MAXMAT = 3, MXMPPM = 6, and MXSPPM = 4; and the material property arrays can be dimensioned as

```
DIMENSION PROP(6,3)
DIMENSION THPROP(4,3),AKPROP(4,3)
```

If we assume that we will make a 500 time step simulation and we will reinitiate the change on the time step size for 20 times during our simulation, then we have MAXNTI = 500 and MXNDTC = 20. With this assumption, arrays related to time can be dimensioned as

```
DIMENSION KPR(500),KDSK(500),TDTCH(20)
```

Corresponding to the above dimension specifications, the following data statements must be made in BLOCK DATA to specify the maximum-control integers:

```
      DATA MAXEL,MAXNP,MAXBES,MAXBNP,JBAND/4959,6000,1966,1968,27/
      DATA MAXNTI,MXNDTC/500,20/
      DATA LTMXNP,LMXNP,LMXBW,MXREGN/900,300,23,20/
      DATA MXSEL,MXSPR,MXSDP,MXWNP,MXWPR,MXWDP/11,3,4,10,5,8/
      DATA MXCNP,MXCES,MXCPR,MXCDP/200,171,2,4/
      DATA MXNNP,MXNES,MXNPR,MXNDP/600,551,1,2/
      DATA MXVES,MXVNP,MXRPR,MXRDP/551,600,3,24/
      DATA MXDNP,MXDPR,MXDDP/200,20,8/
      DATA MAXMAT,MXSPPM,MXMPPM/3,4,6/
```

5.2 Soil Function Specifications

Analytical functions are used to describe the functional relationships of water content, water capacity, and relative hydraulic conductivity with pressure head. Therefore, the user must supply three functions to compute the water content, water capacity, and relative hydraulic conductivity based on the current value of pressure head. The parameters needed to specify the functional form are read and stored in THPROP and AKPROP. One example is shown in the subroutine SPROP between lines SPRO 405 and SPRO 535 in Appendix A. In this example, the water content, water capacity, and relative hydraulic conductivity are given by (van Genuchten 1980):

$$\theta = \theta_r + \frac{\theta_s - \theta_r}{[1 + (\alpha h)^n]^m} , \tag{105}$$

$$\frac{d\theta}{dh} = \alpha(n - 1) [1 - f(\theta)]^m [f(\theta)] (\theta_s - \theta_r) , \tag{106}$$

$$K_r = [(\theta - \theta_s)/(\theta_s - \theta_r)]^{1/2} (1 - [1 - f(\theta)]^m)^2 , \tag{107}$$

in which

$$f(\theta) = [(\theta - \theta_r)/(\theta_s - \theta_r)]^{1/m} \quad , \tag{108}$$

and

$$m = 1 - n \quad , \tag{109}$$

where α, m, and n are the parameters.

To further demonstrate how we should modify the subroutine SPROP in Appendix A to accommodate the material property functions that are different from those given by Eqs. (105) through (109), let us assume that the following Fermi types of functions are used to represent the unsaturated hydraulic properties (Yeh 1987):

$$\theta = \theta_r + (\theta_s - \theta_r)/\{1 + \exp[-\alpha(h - h_\theta)]\} \quad , \tag{110}$$

$$d\theta/dh = \alpha(\theta_s - \theta_r)\exp[-\alpha(h - h_\theta)]/\{1 + \exp[-\alpha(h - h_\theta)]\}^2 , \tag{111}$$

and

$$\log_{10}(K_r) = \epsilon/\{1 + \exp[-\beta(h - h_k)]\} - \epsilon \quad , \tag{112}$$

where θ_s, θ_r, α, and h_θ are the parameters for computing the water content and water capacity; and β, ϵ, and h_k are the parameters for computing the relative hydraulic conductivity. Lines between SPRO 405 and SPRO 535 in the subroutine SPROP in Appendix A may be changed, for this example, to

```
WCR=THPROP(1,MTYP)
WCS=THPROP(2,MTYP)
ALPHA=THPROP(3,MTYP)
HTHETA=THPROP(4,MTYP)
EPS=AKPROP(1,MTYP)
BETA=AKPROP(2,MTYP)
HSUBK=AKPROP(3,MTYP)
```

```
      DO 800 KG=1,8
C
C ------- SATURATED CONDITION
C
      IF(HNP.GT.0.0) GO TO 700
      TH(KG,M)=WCS
      DTH(KG,M)=0.0D0
      AKR(KG,M)=1.0D0
      GO TO 800
C
C ------ UNSATURATED CASE
C
  700 EXPAH=DEXP(-ALPHA*(HNP-HTHETA))
     TH(KG,M)=WCR+(WCS-WCR)/(1.0D0+EXPAH)
     DTH(KG,M)=ALPHA*(WCS-WCR)*EXPAH/(1.0D0+EXPAH)**2
     AKRLOG=EPS/(1.0D0+DEXP(-BETA*(HNP-HSUBK))) - EPS
     AKR(KG,M)=10.0D0**AKRLOG
```

5.3 Data Input Guide

The input format for each data card is specified in the following. The number under the rule is the last column of the field. In general, an integer has a field of 5 and should be right-justified. On the other hand, a real number has a field of 10 and can be placed anywhere within the field if F-specification is used. If E- or D- format is used, then the exponential should be right-justified.

1. TITLE
   One card per problem is required.

   FORMAT(I5,9A8,3I1)

NPROB	TITLE	IGEOM	IBUG	ICHNG
5	77	78	79	80

NPROB - Problem number.

TITLE - Array for the title of the problem. It may contain up to 72 characters from column 6 to column 77.

IGEOM - Integer indicating if (1) the geometry, boundary, and pointer arrays are to be printed; (2) the boundary and pointer arrays are to be computed or read via logical units. If to be computed, they should be written on logical units. If IGEOM is even number, (1) will not be printed. If IGEOM is odd number, (1) will be printed. If IGEOM is less than or equal to 1, boundary arrays will be computed and written on unit 3. If IGEOM is greater than 1, boundary arrays will be read via unit 3. If IGEOM is less than or equal to 3, pointer arrays will be computed and written on unit 4. If IGEOM is greater than 3, pointer arrays will be read via unit 4.

IBUG - Integer indicating if the diagnostic output is desired. 0 - No, 1 - Yes.

ICHNG - Integer control number indicating if the cyclic change of rainfall-seepage nodes is to be printed. 0 - No, 1 - Yes.

2. BASIC INTEGER PARAMETERS

One card per problem is required.

FORMAT(16I5)

NNP	NEL	NMAT	NCM	NTI	KSS	NSPPM	NMPPM
5	10	15	20	25	30	35	40

KSTR	KCP	KGRAV	NITER	NCYL	NDTCHG	NPITER	NREGN
45	50	55	60	65	70	75	80

NNP - Number of nodal points.

NEL - Number of elements.

NMAT - Number of material types.

NCM - Number of elements with material property correction.

NTI - Number of time steps or time increments.

KSS - Steady state control.
0 - steady state solution desired,
1 - transient state or transient solutions.

NSPPM - Number of parameters to specify analytical soil functions per material.

NMPPM - Number of material properties per material.
NMPPM - 6 for the present model.

KSTR - Auxiliary storage output control;
0 - no storage,
1 - output stored in Logical Unit 1.

KCP - Permeability input control;
0 - input saturated hydraulic conductivity,
1 - input saturated permeability.

KGRAV - Gravity term control;
0 - no gravity term,
1 - with gravity term.

NITER - Number of iterations allowed for solving the nonlinear equation.

NCYL - Number of cycles permitted for iterating rainfall-seepage boundary conditions per time step.

NDTCHG - Number of times to reset time step size to initial time step size.

NPITER - Number of iterations for block or pointwise solution.

NREGN - Number of subregions.

NOTE: NTI can be computed by NTI - I1 + 1 + I2 + 1,
where I1 - largest integer not exceeding
Log(DELMAX/DELT)/Log(1+CHNG),
I2 - largest integer not exceeding
{RTIME-DELT*[(1+CHNG)**(I1+1)-1]/CHNG}/DELMAX,
RTIME - Real simulation time,
DELMAX,DELT,and CHNG are defined in Data Set 3.

3. BASIC REAL PARAMETERS

Two cards per problem. Use of an E-, D-, or another F-type field specification in the input card overrides any of the D10.3 field.

Card 1 - FORMAT(8D10.3).

DELT	CHNG	DELMAX	TMAX	TOLA	TOLB	RHO	GRAV
10	20	30	40	50	60	70	80

DELT - Initial time step size (T)

CHNG - Percentage of change in time step size in each of the subsequent time increment, (dimensionless in decimal point).

DELMAX - Maximum value of DELT (T).

TMAX - Maximum simulation time (T).

TOLA - Steady-state convergence criteria (L).

TOLB - Transient-state convergence criteria (L).

RHO - Density of water (M/L^3).

GRAV - Acceleration of gravity (L/T^2).

Card 2 - FORMAT(8D10.3).

VISC	W	OME	OMI		
10	20	30	40	45	80

VISC - Dynamic viscosity of water [(M/L)/T].

W - Time derivative weighting factor;
0.5 - Crank-Nicolson central and/or mid-difference,
1.0 - backward difference.

OME - Iteration parameter for solving the nonlinear matrix equation;
0.0 -- 1.0 - underrelaxation,
1.0 -- 1.0 - exact relaxation,
1.0 -- 2.0 - over-relaxation.

OMI - Relaxation parameter for solving the linearized matrix equation pointwise;
0.0 -- 1.0 - underrelaxation,
1.0 -- 1.0 - exact relaxation,
1.0 -- 2.0 - overrelaxation.

4. PRINTER AND DISK STORE CONTROL AND TIMES FOR STEP SIZE RESETTING
The number of cards here depends on the number of time increments NTI and the times of resetting step size NDTCHG. The number of cards is [(NTI/80+1)*2 + (NDTCHG/8+1)]. (NTI/80+1) cards are for printer output control, (NTI/80+1) cards for storage control, and (NDTCHG/8+1) cards for time-step-size resetting.

Card 1 to Card [(NTI/80+1)] -- FORMAT(80I1)

KPRO	KPR(1)	KPR(2)	---	KPR(I)	---	KPR(NTI)
1	2	3				80

KPRO - Printer control for steady state and initial conditions;
0 - print nothing,
1 - print FLOW, FRATE, and TFLOW,
2 - print above (1) plus pressure head H,
3 - print above (2) plus total head,
4 - print above (3) plus moisture content,
5 - print above (4) plus Darcy's velocity.

KPR(I) - Printer control for I-th time step similar to KPRO.

Card [(NTI/80+1)+1] to Card [(NTI/80+1)*2] -- FORMAT(80I1)

KDSKO	KDSK(1)	KDSK(2)	---	KDSK(I)	---	KDSK(NTI)
1	2	3				80

KDSKO - Auxiliary storage control for steady state and initial condition;
0 - no storage,
1 - store on Logical Unit 1.

KDSK(I) - Auxiliary storage control for I-th time step similar to KDSKO.

Card [(NTI/80+1)*2 + 1] to card [(NTI/80+1)*2 + (NDTCHG/8+1)] -- FORMAT(8D10.3).

TDTCH(1)	TDTCH(2)	---	TDTCH(I)	---	TDTCH(NDTCHG)
10	20				80

TDTCH(I) - Time when I-th step size resetting is needed.

5. MATERIAL PROPERTIES

A total of NMAT cards are required for this data set.

FORMAT(8D10.3)

PROP(1,1)	PROP(2,1)	----	PROP(6,1)	
10	20		60	80

.
.
.

PROP(1,I)	PROP(2,I)	----	PROP(6,I)	
10	20		60	80

.
.
.

PROP(1,NMAT)	PROP(2,NMAT)	----	PROP(6,NMAT)	
10	20		60	80

PROP(1,I) - Saturated xx- hydraulic conductivity or permeability of medium I (L/T) or (L^2).

PROP(2,I) - Saturated yy- hydraulic conductivity or permeability of medium I (L/T) or (L^2).

PROP(3,I) - Saturated zz- hydraulic conductivity or permeability of medium I (L/T) or (L^2).

PROP(4,I) - Saturated xy- hydraulic conductivity or permeabiltiy of medium I (L/T) or (L^2).

PROP(5,I) - Saturated xz- hydraulic conductivity or permeability of medium I (L/T) or (L^2).

PROP(6,I) - Saturated yz- hydraulic conductivity or permeability of medium I (L/T) or (L^2).

6. ANALYTICAL SOIL PARAMETERS

Two sets of cards per material -- one for moisture-content parameters and the other for conductivity (permeability) parameters. The number of cards per set is determined by the number of parameters used to specify the soil properties per material, NSPPM, and by the number of materials, NMAT:

FORMAT(8D10.3)

THPROP(1,1)	THPROP(2,1)	... THPROP(NSPPM,1)
10	20	80

.

.

THPROP(1,NMAT)	THPROP(2,NMAT)	... THPROP(NSPPM,NMAT)
10	20	80

AKPROP(1,1)	AKPROP(2,1)	... AKPROP(NSPPM,1)
10	20	80

.

.

AKPROP(1,NMAT)	AKPROP(2,NMAT)	... AKPROP(NSPPM,NMAT)
10	20	80

THPROP(J,I) - Analytical moisture-content parameter J of material I.

AKPROP(J,I) - Analytical relative conductivity parameter J of material I.

**** NOTE: THPROP(J,I) is also used to compute dTH/dH, the water capacity. One should derive the dTH/dH function by himself.

7. **NODAL POINT COORDINATE**
Usually a total of NNP cards are required. However, if a group of subsequent nodes appears in regular pattern, automatic generation can be made.

FORMAT(3I5,5X,6D10.3)

NI	NSEQ	NAD		XNI	YNI	ZNI	XIAD	YIAD	ZIAD
5	10	15	20	30	40	50	60	70	80

NI - Node number of the first node in the sequence.

NSEQ - NSEQ subsequent nodes will be automatically generated.

NAD - Increment of node number for each of the NSEQ subsequent nodes.

XNI - x-coordinate of node NI (L).

YNI - y-coordinate of node NI (L).

ZNI - z-coordinate of node NI (L).

XAD - Increment of x-coordinate for each of the NSEQ subsequent nodes (L).

YAD - Increment of y-coordinate for each of the NSEQ subsequent nodes (L).

ZAD - Increment of z-coordinate for each of the NSEQ subsequent nodes (L).

**** NOTE: A blank card must be used to signal the end of this data set.

8. SUBREGION DATA

(a) Number of Nodes for each Subregion: Normally NREGN cards are required. However, if regular pattern appears, automatic generation can be made.

FORMAT(16I5)

NK	NSEQ	NKAD	NODES	NODAD	
5	10	15	20	25	80

NK - Subregion number of the first region in a sequence.

NSEQ - NSEQ subsequent subregions will have their number of nodes automatically generated.

NKAD - Increment of NK in each of the NSEQ subsequent subregions.

NODES - Number of nodes for the subregion NK.

NODAD - Increment of NODES in each of the NSEQ subsequent subregions.

**** NOTE: A blank card must be used to end the input of this subdata set.

(b) Mapping between Global nodes and Subregion Nodes: This subdata set should be repeated NREGN times, one for each subregion.

FORMAT(16I5)

LI	NSEQ	LIAD	NI	NIAD	
5	10	15	20	25	80

LI - Local node number of the first node in a sequence.

NSEQ - NSEQ subsequent local nodes will have their corresponding global node number generated automatically.

LIAD - Increment of LI for each of the NSEQ subsequent nodes.

NI - Global node number of local node LI.

NIAD - Increment of NI for each of the NSEQ subsequent nodes.

**** NOTE: A blank card must be used to signal the end of this subdata set.

9. ELEMENT INCIDENCES

Usually a total of NEL cards are needed. However, if a group of elements appears in regular pattern, automatic generation is made.

FORMAT(16I5)

MI	NSEQ	MIAD	IE(MI,1)	---	IE(MI,8)	IEMAD
5	10	15	50			80

MI - Global element number of the first element in a sequence.

NSEQ - NSEQ subsequent elements will have their connectivity automatically generated.

MIAD - Increment of MI for each of the NSEQ subsequent elements.

IE(MI,1) - Global node number of the first node of element MI.

IE(MI,2) - Global node number of the second node of element MI.

IE(MI,3) - Global node number of the third node of element MI.

IE(MI,4) - Global node number of the fourth node of element MI.

IE(MI,5) - Global node number of the fifth node of element MI.

IE(MI,6) - Global node number of the sixth node of element MI.

IE(MI,7) - Global node number of the seventh node of element MI.

IE(MI,8) - Global node number of the eighth node of element MI.

IEMAD - Increment of IE(MI,1) through IE(MI,8) for each of the NSEQ subsequent elements.

IE(MI,1) through IE(MI,8) are numbered according the convention shown in Fig. 7. The first four nodes start from the front, lower, left corner and progress around the bottom element surface in a counterclockwise direction. The other four nodes begin from the front, upper, left corner and progress around the top element surface in a counterclockwise direction.

ORNL-DWG 87-9588

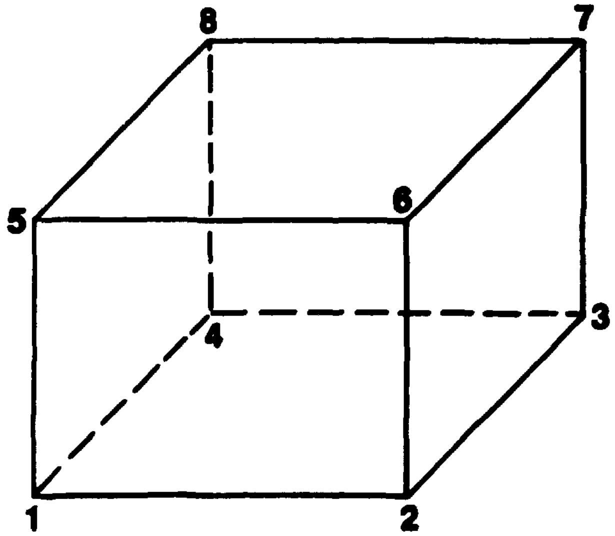

Fig. 7. Numbering of element incidences.

10. MATERIAL TYPE CORRECTION

This data set is required only if NCM > 0. Normally, NCM cards are required. However, if a group of elements appears in regular pattern, automatic generation may be made.

FORMAT(4I5)

MI	NSEQ	MAD	MITYP	MTYPAD	
5	10	15	20	25	80

MI - Global element number of the first element in the sequence.

NSEQ - NSEQ subsequent elements will be generated automatically.

MAD - Increment of element number for each of the NSEQ subsequent elements.

MITYP - Type of material correction for element MI.

MTYPAD - Increment of the type of material correction for each of the NSEQ subsequent elements.

**** NOTE: A blank card must be used to signal the end of this data set.

11. CARD INPUT FOR INITIAL OR PRE-INITIAL CONDITIONS

NNP cards, one card for each node, are needed. However, if a group of nodes appears in regular pattern, auto-generation is made.

FORMAT(3I5,5X,4D10.3)

NI	NSEQ	NAD		HNI	HAD	
5	10	15	20	30	40	80

NI - Global node number of the first node in the sequence.

NSEQ - NSEQ subsequent nodes will be generated automatically.

NAD - Increment of node number for each of the NSEQ nodes.

HNI - Initial or pre-initial pressure head of node NI (L).

HAD - Increment of initial or pre-initial head for each of the NSEQ nodes (L).

**** NOTE: A blank card must be used to signal the end of this data set.

** NOTE ON INITIAL CONDITIONS: The initial condition for a transient calculation may be obtained in two different ways: from card input or steady-state calculation using time-invariant boundary conditions that are different from those for transient computation. In the latter case, a card input of the pre-initial conditions is required as the zeroth order iterate of the steady state solution.

** NOTE ON AUXILIARY STORAGE UNITS: Logical Unit 1 is used to store output if KSTR > 0. Proper identification of this logical unit must be made in the job control language (JCL).

** NOTE ON STEADY STATE INPUT: Steady state option may be used to provide either the final state of a system under study or the initial conditions for a transient state calculation. In former case, KSS = 0 and NTI = 0, and in the latter case KSS = 0 and NTI > 0. If KSS > 0, there will be no steady state calculation.

12. INTEGER PARAMETERS FOR TRANSIENT SOURCE and BOUNDARY CONDITIONS
Two cards per problem are required.

FORMAT(16I5)

NSEL	NSPR	NSDP	NWNP	NWPR	NWDP	NCES	NCNP
5	10	15	20	25	30	35	40

NCPR	NCDP	NNES	NNNP	NNPR	NNDP	NVES	NVNP
45	50	55	60	65	70	75	80

NVPR	NVDP	NDNP	NDPR	NDDP			
5	10	15	20	25			80

NSEL - Number of source/sink elements.

NSPR - Number of source/sink profiles, should be .GE. 1.

NSDP - Number of data points in each of the NSPR. source/sink profiles, should be .GE. 2.

NWNP - Number of well or point source/sink nodal points.

NWPR - Number of well or point source/sink strength profiles.

NWDP - Number of data points in each of the NWPR profiles.

NCES - Number of Cauchy boundary element sides.

NCNP - Number of Cauchy boundary nodes.

NCPR - Number of Cauchy flux profiles.

NCDP - Number of data points in each of the NCPR Cauchy flux profiles.

NNES - Number of Neumann boundary element sides.

NNNP - Number of Neuamnn nodal points.

NNPR - Number of Neumann flux profiles.

NNDP - Number of data points in each of the NNPR Neumann flux profiles.

NVES - Number of variable boundary element sides.

NVNP - Number of variable boundary nodes.

NVPR - Number of variable boundary profile.

NVDP - Number of data points in each of the NVPR variable boundary profiles.

NDNP - Number of Dirichlet nodal points, should be .GE. 1.

NDPR - Number of total Dirichlet-head profiles, should be .GE. 1.

NDDP - Number of data points in each of the NDNP total Dirichlet-head profiles, should be .GE. 1.

13. DISTRIBUTED AND POINT SOURCES/SINKS

Two sets of cards are needed: one for the distributed source/sink, and the other for the point source/sink.

(1) Distributed Source/sink Data: This subdata set is needed if and only if NSEL > 0. When NSEL is greater than zero, three groups of cards are needed: one for the source profile, one for the source element number, and the other for the source type.

(a) Source/sink Profiles: The number of cards depends on NSPR and NSDP. This sub-data set is read in NSPR-wisely. Each card contains four data points.

FORMAT(8D10.3)

irst profile

TSOSF(1,1)	SOSF(1,1)	TSOSF(2,1)	SOSF(2,1) ---	
10	20	30	40	80

.
.

---	TSOSF(NSDP,1)	SOSF(NSDP,1)	
			80

Second profile

TSOSF(1,2)	SOSF(1,2)	TSOSF(2,2)	SOSF(2,2) ---	
10	20	30	40	80

.
.

---	TSOSF(NSDP,2)	SOSF(NSDP,2)	
			80

.
.
.

I-th profile

TSOSF(1,I)	SOSF(1,I)	TSOSF(2,I)	SOSF(2,I) ---	
10	20	30	40	80

---	TSOSF(NSDP,I)	SOSF(NSDP,I)	
			80

.
.
.

NSPR-th profile

TSOSF(1,NSPR)	SOSF(1,NSPR)	TSOSF(2,NSPR)	SOSF(2,NSPR)	.
10	20	30	40	80

.
.

--- TSOSF(NSDP,NSPR)	SOSF(NSDP,NSPR)	
		80

TSOSF(J,I) = Time of J-th data point in I-th profile (T).

SOSF(J,I) = Source/sink value of J-th data point in I-th profile $[(L^3/T)/L^3]$.

(b) Global Element Number of Compressed Distributed Source/Sink Element: The number of cards required in this subdata set depends on NSEL. Each card contains 16 nodes.

FORMAT(16I5).

MSEL(1)	MSEL(2)	MSEL(3)	--- MSEL(I)	--- MSEL(NSEL)
5	10	15		80

MSEL(I) = Global element number of I-th compressed distributed source/sink element.

(c) Source Type in Each Element: Usually one card per element is required; however, automatic generation can be made.

FORMAT(5I5)

MI	NSEQ	MAD	MITYP	MTYPAD	
5	10	15	20	25	80

MI = Global element number of the first element in the sequence.

NSEQ = NSEQ elements will contain the source types that will be automatically generated.

MAD - Increment of element number for each of the NSEQ elements.

MITYP - Source type in element MI.

MTYPAD - Increment of the source type for each of the NSEQ subsequent elements.

**** NOTE: A blank card must be used to signal the end of this data set.

(2) Point Source/Sink Data: This subdata set is needed if and only if NWNP > 0. When NWNP is greater than zero, this subdata set is read in similarly to subdata set 12 (1).

(a) Well or Point Source/Sink Profiles: The number of cards depends on NWPR and NWDP. This sub-data set is read in similarly to sub data set 12(1)(a).

FORMAT(8D10.3)

First Profile

TWSSF(1,1)	WSSF(1,1)	TWSSF(2,1)	WSSF(2,1)	---
10	20	30	40	80

.

.

--- TWSSF(NWDP,1)	WSSF(NWDP,1)	
		80

Second Profile

TWSSF(1,2)	WSSF(1,2)	TWSSF(2,2)	WSSF(2,2)	---
10	20	30	40	80

.

.

--- TWSSF(NWDP,2)	WSSF(NWDP,2)	
		80

.

.

I-th Profile

```
TWSSF(1,I)   WSSF(1,I)   TWSSF(2,I)   WSSF(2,I)    ---
-------------------------------------------------------
        10          20           30          40      80
```

.

.

```
--- TWSSF(NWDP,I)   WSSF(NWDP,I)
-------------------------------------------------------
                                                     80
```

.

.

.

NWPR-th Profile

```
TWSSF(1,NWPR) WSSF(1,NWPR) TWSSF(2,NWPR) WSSF(2,NWPR) ---
---------------------------------------------------------
          10           20            30          40  80
```

.

.

```
--- TWSSF(NWDP,NWPR)   WSSF(NWDP,NWPR)
---------------------------------------------------------
                                                       80
```

TWSSF(J,I) - Time of J-th data point in I-th well or point source/sink profile (T).

WSSF(J,I) - Strength of point source/sink of J-th data point in I-th profile (L^3/T).

(b) Global Nodal Number of Compressed Well Source/Sink Nodes: The number of cards required in this subdata set depends on NWNP. Each card contains 16 nodes.

FORMAT(16I5)

```
NPW(1)  NPW(2)  NPW(3)  ---  NPW(I)  ---  NPW(NWNP)
----------------------------------------------------
     5      10      15                            80
```

NPW(I) - Global node number of I-th compressed well source/sink node.

(c) Types of Well Source/Sink Nodes: Normally, one card per well node is needed (i.e. a total of NWNP cards). However, if the well nodes appear in regular pattern, automatic generation may be made.

FORMAT(5I5)

NI	NSEQ	NIAD	NITYP	NITYPA	
5	10	15	20	25	80

NI - Compressed well node number of the first node in a sequence.

NSEQ - NSEQ subsequent well nodes will be generated automatically.

NIAD - Increment of compressed well node number for each of the NSEQ subsequent nodes.

NITYP - Type of well/sink source profile assigned to node NI.

NITYPA - Increment of NITYP for each of the NSEQ subsequent nodes.

*** NOTE: A blank card must be used to signal the end of this subdata set.

14. RAINFALL/EVAPORATION-SEEPAGE (VARIABLE) BOUNDARY CONDITIONS

Six groups of cards are required for this data set if and only if NVES > 0. The first group is used to specify the rainfall/evaporation profile, the second group is used to assign the type of rainfall/evaporation profile to each of the NVES rainfall/evaporation-seepage sides, the third group is used to read four global nodal numbers of all NVES rainfall/evaporation-seepage sides, the fourth group is used to read the global nodal number of all the NVNP nodes, the fifth group is used to read in the allowed ponding depth for each of the NVNP nodes, and the sixth group is used to the allowed minimum pressure for each of the NVNP nodes.

(a) Rainfall/Potential Evaporation Profiles: This subdata set is read in similarly to that of subdata set 12(a).

FORMAT(8D10.3)

First profile

TRF(1,1)	RF(1,1)	TRF(2,1)	RF(2,1) ...	
10	20	30	40	80

.

.

...	TRF(NVDP,1)	RF(NVDP,1)	
			80

Second profile

TRF(1,2)	RF(1,2)	TRF(2,2)	RF(2,2) ...	
10	20	30	40	80

.

.

...	TRF(NVDP,2)	RF(NVDP,2)	
			80

.

.

.

I-th profile

TRF(1,I)	RF(1,I)	TRF(2,I)	RF(2,I) ...	
10	20	30	40	80

.

.

...	TRF(NVDP,I)	RF(NVDP,I)	
			80

.

.

.

NVPR-th profile

TRF(1,NVPR)	RF(1,NVPR)	TRF(2,NVPR)	RF(2,NVPR) ...	
10	20	30	40	80

... TRF(NVDP,NVPR)	RF(NVDP,NVPR)	
		80

TRF(J,I) - Time of the J-th data point on I-th rainfall/evaporation-vs-time profile (T).

RF(J,I) - Rainfall/evaporation rate of the J-th data point on I-th profile (L/T). For rainfall, input positive value, for potential evaporation, input negative value.

(b) Rainfall/Evaporation Type Assigned to Each of All NVES Surfaces: This subdata set is read in similarly to that in subdata set 12(1)(c).

FORMAT(5I5)

NI	NSEQ	NIAD	NITYP	NTYPAD	
5	10	15	20	25	80

NI - Compressed variable boundary (rainfall/seepage) element side of the first side in a sequence.

NSEQ - NSEQ subsequent sides will be generated automatically.

NIAD - Increment of the compressed NI for each of the NSEQ subsequent sides.

NITYP - Type of rainfall/evaporation profile assigned to side NI.

NTYPAD - Increment of the type of rainfall/evaporation profile for each of the NSEQ subsequent sides.

*** NOTE: A blank card must be used to signal the end of this subdata set.

(c) Specification of Rainfall/evaporation-seepage (RES) Sides: Normally, NVES cards are required, one each for a RES boundary element side. However, if a group of RES element sides appears in a regular pattern, automatic generation may be made.

FORMAT(16I5)

MI	NSEQ	MIAD	I1	I2	I3	I4	I1AD	I2AD	I3AD	I4AD	
5	10	15	20	25	30	35	40	45	50	55	80

MI - Compressed RES element side number of the first element side in a sequence.

NSEQ - NSEQ subsequent RES element sides will be generated automatically.

MIAD - Increment of MI for each of the NSEQ subsequent RES element sides.

I1 - Global node number of the first node of element side MI.

I2 - Global node number of the second node of element side MI.

I3 - Global node number of the third node of element side MI.

I4 - Global node number of the fourth node of element side MI.

I1AD - Increment of I1 for each of the NSEQ subsequent element sides.

I2AD - Increment of I2 for each of the NSEQ subsequent element sides.

I3AD - Increment of I3 for each of the NSEQ subsequent element sides.

I4AD - Increment of I4 for each of the NSEQ subsequent element sides.

**** NOTE: A blank card must be used to signal the end of this subdata set.

(d) Global Nodal Number of All Rainfall/Evaporation-Seepage (RES) Nodes: This subdata set is read in similarly to that in subdata set 12(1)(c).

FORMAT(5I5)

NI	NSEQ	NIAD	NODE	NODEAD	
5	10	15	20	25	80

NI - Compressed variable boundary node number of the first node in a sequence.

NSEQ - NSEQ subsequent nodes will be generated automatically.

NIAD - Increment for NI for each of the NSEQ nodes.

NODE - Global nodal number of the node NI.

NODEAD - Increment of the global nodal number for each of the NSEQ subsequent nodes.

**** NOTE: A blank card must used to signal the end of this subdata set.

(e) Ponding depth: Normally, NVNP cards are needed, one for each of the NVNP variable boundary (VB) nodes. However, if a group of VB nodes has a regular pattern of ponding depth, automatic generation is made.

FORMAT(3I5,5X,2D10.3)

NI	NSEQ	NIAD		HCONNI	HCONAD	
5	10	15	20	30	40	80

NI - Compressed VB nodal number of the first node in a sequence.

NSEQ - NSEQ subsequent nodes will be generated automatically.

NIAD - Increment of the compressed VB nodal number for each of the NSEQ subsequent nodes.

HCONNI - Ponding depth of node NI (L).

HCONAD - Increment of ponding depth for each of the NSEQ subsequent nodes (L).

**** NOTE: A blank card must be used to signal the end of this subdata set.

(f) Minimum Pressure Head Allowed in Each VB Nodes: This subdata set is read in similarly to the above subdata set.

NI	NSEQ	NIAD		HMININI	HMINAD	
5	10	15	20	30	40	80

NI - Compressed VB node number of the first node in a sequence.

NSEQ - NSEQ subsequent VB nodes will be generated automatically.

NIAD - Increment of the compressed VB node number for each of the NSEQ subsequent nodes.

HMINNI - Minimum pressure head of node NI (L).

HMINAD - Increment of minimum pressure head for each of the NSEQ subsequent nodes (L).

**** NOTE: A blank card must be used to signal the end of this subdata set.

15. DIRICHLET BOUNDARY CONDITIONS
This data set is required if NDNP > 0.

(a) Dirichlet-Head Profiles: This subdata set is read in similarly to that in subdata set 12(1)(a).

FORMAT(8D10.3)

First profile

THDBF(1,1)	HDBF(1,1)	HDBF(2,1)	HDBF(2,1) ---	
10	20	30	40	80

--- THDBF(NDDP,1) HDBF(NDDP,1)
80

Second profile

```
    THDBF(1,2)    HDBF(1,2)    THDBF(2,2)    HDBF(2,2) ---
-----------------------------------------------------------
             10           20            30          40  80
                                    .
                                    .
-----------------------------------------------------------
 ---   THDBF(NDDP,2)     HDBF(NDDP,2)
-----------------------------------------------------------
                                                        80
                             .
                             .
                             .
```

I-th profile

```
    THDBF(1,I)    HDBF(1,I)    THDBF(2,I)    HDBF(2,I) ---
-----------------------------------------------------------
             10           20            30          40  80
                                    .
                                    .
-----------------------------------------------------------
 ---   THDBF(NDDP,I)    HDBF(NDDP,I)
-----------------------------------------------------------
                                                        80
                             .
                             .
                             .
```

NDPR-th profile

```
 THDBF(1,NDPR) HDBF(1,NDPR) THDBF(2,NDPR) HDBF(2,NDPR) ...
-----------------------------------------------------------
             10             20              30        40  80
                                    .
                                    .
-----------------------------------------------------------
   ... THDBF(NDDP,NDPR)  HDBF(NDDP,NDPR)
-----------------------------------------------------------
                                                        80
```

THDBF(J,I) - Time of J-th data point in I-th Dirichlet head profile (T).

HDBF(J,I) - Total head of J-th data point in I-th Dirichlet head profile (L).

(b) Dirichlet Nodes: The number of cards in this subdata set depends on NDNP. Each card contains 16 nodes.

FORMAT(16I5)

NPDB(1)	NPDB(2)	... NPDB(I) ... NPDB(NDNP)	
5	10		80

NPDB(I) - Global node number of I-th Dirichlet node.

(c) Type of Dirichlet Node: Normally, one card per Dirichlet node is required (i. e. a total of NDNP cards). However, if the Dirichlet nodes appear in regular pattern, automatic generation may be made.

FORMAT(5I5)

NI	NSEQ	NAD	NITYP	NTYPAD	
5	10	15	20	25	80

NI - Compressed Dirichlet node number of the first node in the sequence.

NSEQ - NSEQ subsequent Dirichlet nodes will be generated automatically.

NAD - Increment of compressed Dirichlet node number for each of the NSEQ nodes.

NITYP - Type of Dirichlet-head profile for node NI and the NSEQ subsequent nodes.

NTYPAD - Increment of NITYP for each of the NSEQ subsequent nodes.

*** NOTE: A blank card must be used to signal the end of this subdata set.

16. CAUCHY BOUNDARY CONDITIONS

This data set is required if and only if NCES > 0. Four groups of cards are required. One group is used to read Cauchy flux profiles, the second group is used to read the type of Cauchy flux profile assigned to each of the NCES Cauchy sides, the third group is to read Cauchy boundary element sides, and the fourth group is to read the Cauchy boundary nodal points.

(a) Prescribed Cauchy flux Profiles: This subdata set is read in similarly to that in subdata set 12(a).

FORMAT(8D10.3)

First profile

TQCBF(1,1)	QCBF(1,1)	TQCBF(2,1)	QCBF(2,1)	...
10	20	30	40	80

.

.

...	TQCBF(NCDP,1)	QCBF(NCDP,1)	
			80

Second profile

TQCBF(1,2)	QCBF(1,2)	TQCBF(2,2)	QCBF(2,2)	...
10	20	30	40	80

.

.

...	TQCBF(NCDP,2)	QCBF(NCDP,2)	
			80

.

.

.

I-th profile

TQCBF(1,I)	QCBF(1,I)	TQCBF(2,I)	QCBF(2,I)	...
10	20	30	40	80

.

.

TQCBF(NCDP,I)	QCBF(NCDP,I)	
		80

.

.

.

NCPR-th profile

TQCBF(1,NCPR)	QCBF(1,NCPR)	TQCBF(2,NCPR)	QCBF(2,NCPR) ...	
10	20	30	40	80

... TQCBF(NCDP,NCPR)	QCBF(NCDP,NCPR)	
		80

TQCBF(J,I) - Time of the J-th data point on I-th Cauchy-flux profile (T).

QCBF(J,I) - Normal Cauchy flux of the J-th data point on I-th profile $[(L^3/T)/L^2]$. QCBF is positive if directed out from the region, it is negative if directed into the region.

(b) Type of Cauchy Flux Profile Assigned to Cauchy Sides: Normally, one card per Cauchy node (i. e. a total of NCES cards) is required. However, if a group of cards appears in regular pattern, automatic generation may be made.

NI	NSEQ	NAD	NITYP	NTYPAD	
5	10	15	20	25	80

NI - Compressed Cauchy side number of the first side in a sequence.

NSEQ - NSEQ subsequent Cauchy sides will be generated automatically.

NAD - Increment of NI for each of the NSEQ subsequent sides.

NITYP - Type of Cauchy flux profile assigned to side NI.

NTYPAD - Increment of NITYP for each of the NSEQ subsequent sides.

*** NOTE: A blank card is used to signal the end of this data set.

(c) Cauchy Boundary Element Sides: Normally, one card per Cauchy boundary-element side (i.e., a total of NCES cards) is required. However, if a group of Cauchy boundary element sides appears in regular pattern, automatic generation may be made.

FORMAT(16I5)

MI	NSEQ	MIAD	I1	I2	I3	I4	I1AD	I2AD	I3AD	I4AD
5	10	15	20	25	30	35	40	45	50	55

MI = Compressed Cauchy element side number of the first element side in a sequence.

NSEQ = NSEQ subsequent Cauchy element sides will be generated automatically.

MIAD - Increment of MI for each of the NSEQ subsequent element sides.

I1 - Global nodal number of the first node on the MI-th Cauchy side.

I2 - Global nodal number of the third node on the MI-th Cauchy side.

I3 - Global nodal number of the fourth node on the MI-th Cauchy side.

I4 = Global nodal number of the fourth node on the MI-th Cauchy side.

I1AD - Increment of I1 for each of the NSEQ subsequent element sides.

I2AD - Increment of I2 for each of the NSEQ subsequent element sides.

I3AD - Increment of I3 for each of the NSEQ subsequent element sides.

I4AD - Increment of I4 for each of the NSEQ subsequent element sides.

*** NOTE: A blank card is used to end this data set input.

(d) Cauchy Nodal point: The number of cards required is (NCNP/16+1). Each card contain 16 points.

FORMAT(16I5)

NPCB(1)	NPCB(2)	... NPCB(I) ... NPCB(NCNP)
5	10	80

NPCB(I) - Global nodal number of I-th compressed Cauchy node.

17. NEUMANN BOUNDARY CONDITIONS

This data set is required if and only if NNES > 0. Four groups of cards are required. One group is used to read Neumann flux profiles, the second group is used to read the type of Neumann flux profile assigned to each of the NNES Neumann sides, the third group is to read Neumann boundary element sides, and the fourth group is to read Neumann boundary nodal points.

(a) Prescribed Neumann Flux Profiles: This subdata set is read in similarly to that in subdata set 15(a).

FORMAT(8D10.3)

First profile

TQNBF(1,1)	QNBF(1,1)	TQNBF(2,1)	QNBF(2,1)	---
10	20	30	40	80

.

.

---	TQNBF(NNDP,1)	QNBF(NNDP,1)	
			80

Second profile

QNBF(1,2)	QNBF(1,2)	TQNBF(2,2)	QNBF(2,2)	---
10	20	30	40	80

.

.

---	TQNBF(NNDP,2)	QNBF(NNDP,2)	
			80

⋮

I-th profile

TQNBF(1,I)	QNBF(1,I)	TQNBF(2,I)	QNBF(2,I) ---
			80

⋮

--- TQNBF(NNDP,I)	QNBF(NNDP,I)	
		80

⋮

NNPR-th profile

TQNBF(1,NNPR)	QNBF(1,NNPR)	TQNBF(2,NNPR)	QNBF(2,NNPR) ...	
10	20	30	40	80

⋮

... TQNBF(NNDP,NNPR)	QNBF(NNDP,NNPR)	
		80

TQNBF(J,I) - Time of J-th data point in I-th Neumann flux profile (T).

QNBF(J,I) - Value of Neumann flux of J-th data point in I-th Neumann flux profile $[(L^3/T)/L^2]$.

(b) Type of Neumann Flux Profile Assigned to Neumann Sides: This group of cards is read in similarly to subdata set 15(b).

FORMAT(16I5)

NI	NSEQ	NAD	NITYP	NTYPAD	
5	10	15	20	25	80

NI - Compressed Neumann side number of the first side in the sequence.

NSEQ - NSEQ subsequent Neumann sides will be assigned a Neumann flux of type NITYP.

NAD - Increment of compressed Neumann side number in each of the NSEQ nodes.

NITYP - Type of Neumann flux profile for side NI and its NSEQ subsequent sides.

NTYPAD - Increment of NITYP for each of the NSEQ subsequent sides.

*** NOTE: A blank card must be used to signal the end of this subdata set.

(c) Neumann Boundary Element Sides: This group of data is read in similarly to subdata set 15(c).

FORMAT(16I5)

MI	NSEQ	MIAD	I1	I2	I3	I4	I1AD	I2AD	I3AD	I4AD	
5	10	15	20	25	30	35	40	45	50	55	80

MI - Compressed element side number of the first element side in the sequence.

NSEQ - NSEQ subsequent element-sides will be generated automatically.

MIAD - Increment of MI for each of the NSEQ subsequent element sides.

IS1 - Global nodal number of the first node on the MI-th Neumann side.

IS2 - Global nodal number of the second node on the MI-th Neumann side.

IS3 - Global nodal number of the third node on the MI-th Neumann side.

IS4 - Global nodal number of the fourth node on the MI-th Neumann side.

I1AD - Increment of I1 for each of the NSEQ subsequent Neumann sides.

I2AD - Increment of I2 for each of the NSEQ subsequent Neumann sides.

I3AD - Increment of I3 for each of the NSEQ subsequent Neumann sides.

I4AD - Increment of I4 for each of the NSEQ subsequent Neumann sides.

*** NOTE: A blank card must be used to signal the end of this subdata set.

(d) Neumann Nodes: The number of cards needed in this subdata set depend on NNNP. Each card contains 16 nodes.

FORMAT(16I5)

NPNB(1)	NPNB(2)	...	NPNB(I)	...	NPNB(NNNP)
5	10				80

NPNB(I) = global nodal point number of I-th Neumann node.

18. END OF JOB
If another problem is to be run, then input begins again with input data set 1. If termination of the job is desired, a blank card must be inserted at the end of the data set.

5.4 Input and Output Devices

Five logical units are needed to execute 3DFEMWATER. Units 5 and 6 are standard card input and line printer devices, respectively. Unit 1 must be specified to store the simulation results, which can be used for input to 3DFEMWASTE or for plotting purposes. Unit 3 is used to store the boundary arrays for later uses, if these arrays are computed for the present job. Unit 4 is used to store pointer arrays for later uses, if these arrays are generated for the present job. For large problems, our experience has indicated that it would take too much time to process the boundary arrays and to generate pointer arrays. Hence, it is advisable that for multijob executions, these boundary and pointer arrays should be computed only once and written on units 3 and 4, respectively. Once they are stored on units 3 and 4, the JCL should be properly identified for the new job so they can be read via units 3 and 4, respectively.

6. SAMPLE PROBLEMS

To verify 3DFEMWATER, three illustrative examples reported elsewhere (Huyakorn 1986) are presented. These represent the one-, two-, and three-dimensional problems, respectively.

6.1 One-Dimensional Column Problem

This example is selected to represent the simulation of a one-dimensional problem with 3DFEMWATER. The column is 200 cm long and 50 by 50 cm in crosssection (Fig. 8). The column is assumed to contain the soil with a saturated hydraulic conductivity of 10 cm/d, a porosity of 0.45 and a field capacity of 0.1. The unsaturated characteristic hydraulic properties of the soil in the column are given as

$$\theta = \theta_s - (\theta_s - \theta_r)\left(\frac{1}{h_b - h_a} \right) , \tag{113}$$

and

$$K_r = \frac{\theta - \theta_r}{\theta_s - \theta_r} , \tag{114}$$

where h_b and h_a are the parameters used to compute the water content and the relative hydraulic conductivity, respectively.

The initial conditions assumed are a pressure head of -90 cm imposed on the top surface of the column, 0 cm on the bottom surface of the column, and -97 cm elsewhere. The boundary conditions are given as: no

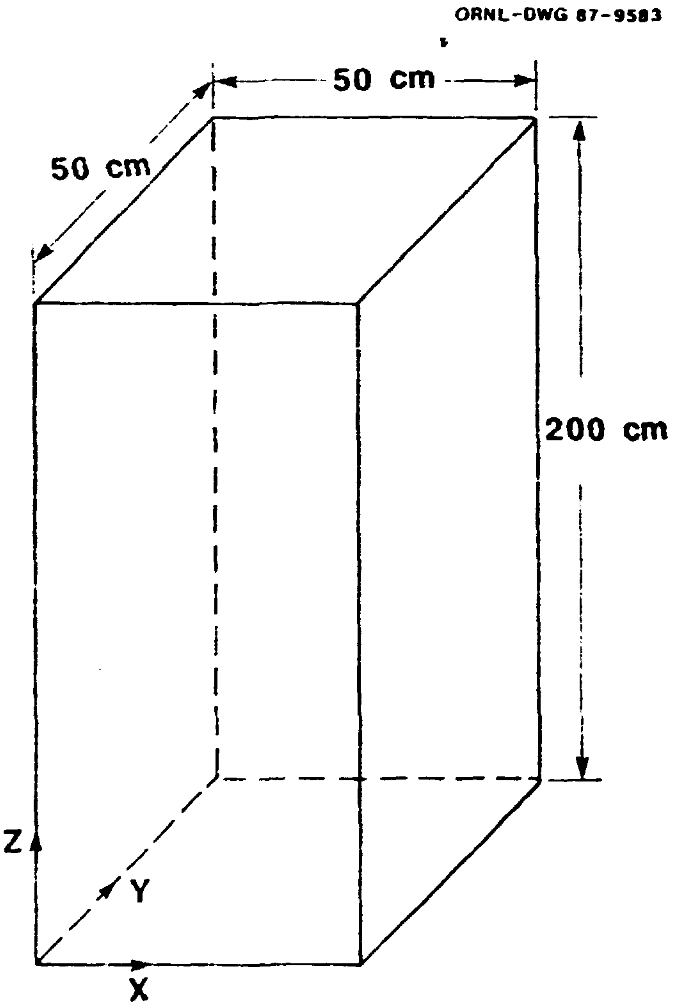

Fig. 8. Problem definition for the one-dimensional transient flow in a soil column

flux is imposed on the left, front, right, and back surfaces of the column; pressure head is held at 0 cm on the bottom surface; and variable condition is used on the top surface of the column with a ponding depth of zero, minimum pressure of -90 cm, and a rainfall of 5 cm/d for the first ten days and a potential evaporation of 5 cm/d for the second 10 days.

The region of interest, that is, the whole column, will be discretized with 1 x 1 x 40 = 40 elements with element size = 50 x 50 x 5 cm, resulting in 2 x 2 x 41 = 164 node points. For 3DFEMWATER simulation, each of the four vertical lines will be considered a subregion. Thus, the a total of four subregions, each with 41 node points, is used for the subregional block iteration simulation.

A variable time step size is used. The initial time step size is 0.05 days, and each subsequent time step size is increased by 0.2 times with a maximum time step size not greater than 1.0 d. Because there is an abrupt change in the flux value from 5 cm/d (infiltration) to -5 cm/d (evaporation) imposed on the top surface at day 10, the time step size is automatically reset to 0.05 d on the tenth day. A 20-day simulation will be made with 3DFEMWATER. With the time step size described above, 44 time steps are needed.

The pressure head tolerance is $2 \cdot 10^{-2}$ cm for nonlinear iteration and is $1 \cdot 10^{-2}$ cm for block iteration. The relaxation factors for both the nonlinear iteration and block iteration are set equal to 0.5.

With the above descriptions, the input data can be prepared according to the instructions given in Sect. 5.3. The input data are given in Table 1.

To execute the problem, the arrays in the main program should be dimensioned as

```
DIMENSION X(164),Y(164),Z(164),IE(40,9)
DIMENSION CMATRX(164,27),GNOJCN(27,164),RLD(164),RI(164),RL(164)
DIMENSION NTNPLR(4),NNPLR(4),LMAXDF(4)
DIMENSION GNLR(164,4),LNOJCN(27,41,4)
DIMENSION CMTRXL(41,3),RLD(41)
DIMENSION H(164),HP(164),HW(164),HT(164),VX(164),VY(164),VZ(164)
DIMENSION TH(8,40),DTH(8,40),AKR(8,40), NPCNV(164)
DIMENSION DCOSB(3,162),ISB(6,162),NPBB(164),BFLX(164),BFLXP(164)
DIMENSION SOS(1),SOSF(1,1),TSOSF(1,1),ISTYP(1),MSEL(1)
DIMENSION WSS(1),WSSF(1,1),TWSSF(1,1),IWTYP(1),NPW(1)
DIMENSION QCB(1),QCBF(1,1),TQCBF(1,1)
DIMENSION ICTYP(1),ISC(5,1), NPCB(1)
DIMENSION QNB(1),QNBF(1,1),TQNBF(1,1)
DIMENSION INTYP(1),ISN(5,1), NPNB(1)
DIMENSION RFALL(1),RF(4,1),TRF(4,1)
DIMENSION IRTYP(1),ISV(5,1), NPVB(4)
DIMENSION DCYFLX(4),FLX(4),HCON(4),HMIN(4)
DIMENSION NPFLX(4),NPCON(4),NPMIN(4)
DIMENSION HDB(1),HDBF(2,1),THDBF(2,1),IDTYP(4),NPDB(4)
DIMENSION PROP(6,1)
DIMENSION THPROP(4,1),AKPROP(4,1)
DIMENSION KPR(44),KDSK(44),TDTCH(3)
```

Table 1. Input data for the one-dimensional column problem

```
  1  SIMULATION OF ONE-D COLUMN INFILTRATION-EVAPORATION; L-CM, T-DAY, M-G 111
 164   40    1    0   44    1    4    6    1    0    1   50   20    3  100    4
   0.05D0     0.2D0     1.0D0    20.0D0    2.0D-2    2.0D-2     1.0D0  7.316D12
  1.1232D2    1.0D0     0.5D0     0.5D0
55515151115111511151111555151511151115111511115
11101010001000100100001110101000100010010000l
   1.0D01  2.0000D1    1.0D38
    0.0D0     0.0D0    10.0D0     0.0D0     0.0D0     0.0D0
  0.150D0   0.450D0   0.00D0    -1.0D2                                THPROP
  0.000D0   0.000D0   0.00D0     0.0D0                                AKPROP
   1   40    1          0.0D0    50.0D0     0.0D0     0.0D0     0.0D0     5.0D0
  42   40    1          0.0D0     0.0D0     0.0D0     0.0D0     0.0D0     5.0D0
  83   40    1         50.0D0     0.0D0     0.0D0     0.0D0     0.0D0     5.0D0
 124   40    1         50.0D0    50.0D0     0.0D0     0.0D0     0.0D0     5.0D0

   1    3    1   41    0
                                                            END OF NNPLR(K)
   1   40    1    1    1
                                                            END OF GNLR(I,1)
   1   40    1   42    1
                                                             END OF GNLR(I,2)
   1   40    1   83    1
                                                            END OF GNLR(I,3)
   1   40    1  124    1
                                                            END OF GNLR(I,4)
   1   39    1   42   83  124    1   43   84  125    2    1
                                                                   END OF IE
   1    3   41           0.0D0     0.0D0
   2   38    1         -9.70D1     0.0D0
  43   38    1         -9.70D1     0.0D0
  84   38    1         -9.70D1     0.0D0
 125   38    1         -9.70D1     0.0D0
  41    3   41         -9.00D1     0.0D0
                                                                     END OF IC
```

Table 1. (Continued)

```
 0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    4
 1    4    4    1    2
  0.0D0     5.0D0    10.0D0      5.0D0   10.001D0    -5.0D0    1.0D38    -5.0D0
 1    0    0    1    0
                                                          END OF IRTYP
 1    0    0   82  123  164   41    0    0    0    0
                                                    END OF ISV(J,I) J-1,4
 1    3    1   41   41
                                                           END OF NPVB
 1    3    1          0.0D0      0.0D0
                                                        END OF HCON
 1    3    1        -90.0D0      0.0D0
                                                        END OF HMIN
  0.0D0     0.0D0     1.0D38     0.0D0
 1   42   83  124
 1    3    1    1    0
                                                                END OF IDTYP
                     END OF JOB
```

Corresponding to the above dimension specification, we must make the following data statements in BLOCK DATA to specify the maximum-control integers:

```
    DATA MAXEL,MAXNP,MAXBES,MAXBNP,JBAND/40,164,162,164,27/
    DATA MAXNTI,MXNDTC/44,3/
    DATA LTMXNP,LMXNP,LMXBW,MXREGN/164,41,3,4/
    DATA MXSEL,MXSPR,MXSDP,MXWNP,MXWPR,MXWDP/1,1,1,1,1,1/
    DATA MXCNP,MXCES,MXCPR,MXCDP/1,1,1,1/
    DATA MXNNP,MXNES,MXNPR,MXNDP/1,1,1,1/
    DATA MXVES,MXVNP,MXVPR,MXVDP/1,4,1,4/
    DATA MXDNP,MXDPR,MXDDP/4,1,2/
    DATA MAXMAT,MXSPPM,MXMPPM/1,4,6/
```

To reflect the soil property function given by Eqs. (113) and (114), we have to modify the subroutine SPROP given in Appendix A. Lines between SPRO 405 and SPRO 535 in the subroutine SPROP must be modified as follows:

```
      WCR=THPROP(1,MTYP)
      WCS=THPROP(2,MTYP)
      HAA=THPROP(3,MTYP)
      HAB=THPROP(4,MTYP)
      DO 800 KG=1,8
      HNP=HKG(KG)
      HNP=-HNP
C
C ------- SATURATED CONDITION
C
      IF(HNP.GT.0) GO TO 700
      TH(KG,M)=WCS
      AKR(KG,M)=1.0D0
      DTH(KG,M)=0.0D0
      GO TO 800
C
C ------- UNSATURATED CASE
C
  700 THMKG=WCS-(WCS-WCR)*(-HNP-HAA)/(HAB-HAA)
      TH(KG,M)=THMKG
      AKR(KG,M)=(THMKG-WCR)/(WCS-WCR)
      DTH(KG,M)=-(WCS-WCR)/(HAB-HAA)
C
```

The pressure head simulated with 3DFEMWATER is plotted in Fig. 9. As expected, during the rainfall period the pressure head is gradually built up, starting from the top surface of the column. In the subsequent evaporation period, the pressure head is gradually reduced. The minimum allowable pressure head of -90 cm was first reached at the top surface of the column and eventually propagated into the soil column. These numerical results are almost identical to those given by Huyakorn (1986).

Thus, the consistency of the computer code is partially verified.

ORNL-DWG 87C-9550

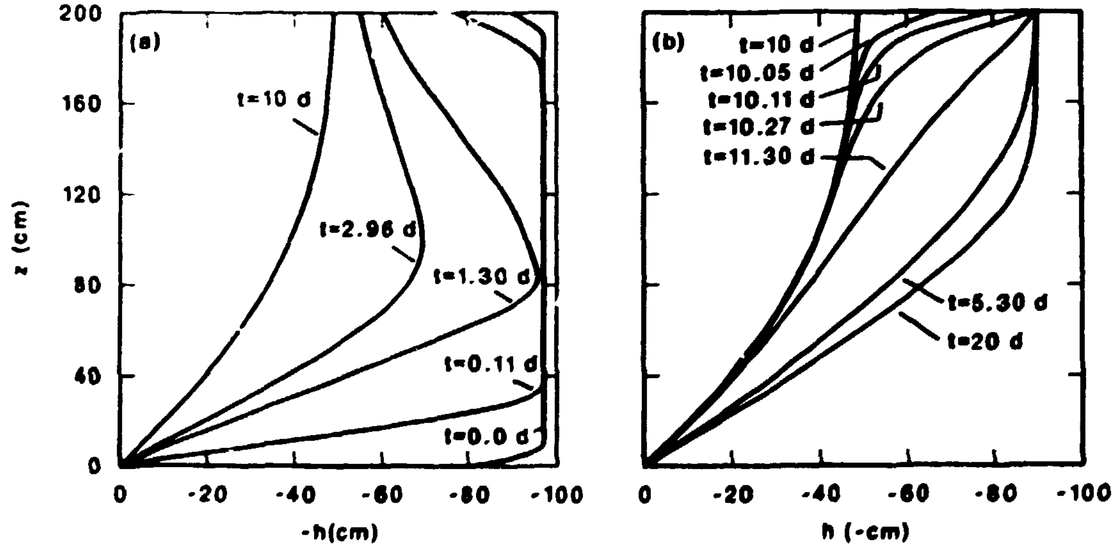

Fig. 9. Simulated pressure head profiles at various times during (a) infiltration and (b) evaporation for the one-dimensional column problem.

6.2 Two-dimensional Drainage Problem

This example is selected to represent the simulation of a two-dimensional problem with 3DFEMWATER. The region of interest is bounded on the left and right by parallel drains fully penetrating the medium, on the bottom by an impervious aquifuge, and on the top by an air-soil interface (Fig. 10). The distance between the two drains is 20 m apart (Fig. 10). The medium is assumed to have a saturated hydraulic conductivity of 0.01 m/d, a porosity of 0.25, and a field capacity of 0.05. The unsaturated characteristic hydraulic properties of the medium are given as

$$\theta = \theta_s + (\theta_s - \theta_r) \frac{A}{A + |h-h_a|^B} , \qquad (115)$$

and

$$K_r = (\frac{\theta - \theta_r}{\theta_s - \theta_r})^n , \qquad (116)$$

where h_a, A, and B are the parameters used to compute the water content and n is the parameter to compute the relative hydraulic conductivity.

Because of the symmetry, the region for numerical simulation will be taken as $0 < x < 10$ m and $0 < z < 10$ m, and 10 m wide along the y-direction will be assumed. The boundary conditions are given as: no flux is imposed on the left ($x = 0$), front ($y = 0$), back ($y = 10$), and bottom ($z = 0$) sides of the region; pressure head is assumed to vary from zero at the water surface ($z = 2$) to 2 m at the bottom ($z = 0$) on the right side ($x = 10$); and variable conditions are used elsewhere. Ponding depth is

ORNL-DWG 87-9866

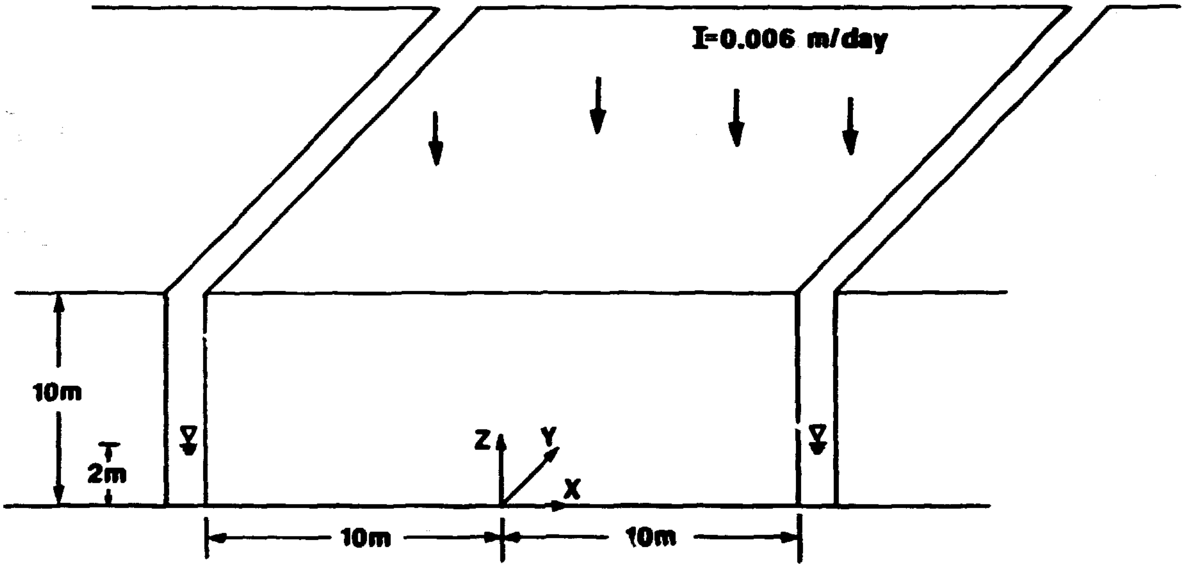

Fig. 10. Problem definition for two-dimensional steady state flow to parallel drains.

assumed to be zero meter on the whole variable boundary. Fluxes on the top side of the variable boundary are assumed equal to 0.006 m/d and on the right side above the water surface are equal to zero. A steady state solution will be sought. A pre-initial condition is set as h = 10 - z.

The region of interest is discretized with 10 x 1 x 10 = 100 elements with element size = 1 x 10 x 1 cm, resulting in 11 x 2 x 11 = 242 node points. For 3DFEMWATER simulation, each of the two vertical planes will be considered a subregion. Thus, the total of two subregions, each with 121 node points, is used for the subregional block iteration simulation.

The pressure head tolerance is $2 \cdot 10^{-3}$ m for nonlinear iteration and is 10^{-3} m for block iteration. The relaxation factors for both the nonlinear iteration and block iteration are set equal to 0.5.

With the above descriptions, the input data can be prepared according to the instructions given in Sect. 5.3. The input data are given in Table 2.

To execute the problem, the arrays in the main program should be dimensioned as

```
      DIMENSION X(242),Y(242),Z(242),IE(100,9)
      DIMENSION CMATRX(242,27),GNOJCN(27,242),RLD(242),RI(242),RL(242)
      DIMENSION NTNPLR(2),NNPLR(2),LMAXDF(2)
```

Table 2. Input data for two-dimensional drainage problem

```
    2   SIMULATION OF TWO-D STEADY DRAINAGE; L-M, T-DAY, M-KG                  111
  242  100    1    0    0    0    5    6    1    0    1   50   20    1  100    2
    0.05D0     0.2D0     1.0D0    20.0D0    2.0D-3    2.0D-3     1.0D0  7.316D10
  1.1232D4     1.0D0     0.5D0     0.5D0
55
11
    1.0D38
    0.01D0     0.0D0    0.01D0     0.0D0     0.0D0     0.0D0
   0.050D0   0.250D0    0.00D0    10.0D0     4.0D0                THPROP
   4.000D0   0.000D0    0.00D0     0.0D0                          AKPROP
    1   10   11          0.0D0     0.0D0     0.0D0     1.0D0     0.0D0     0.0D0
    2   10   11          0.0D0     0.0D0     1.0D0     1.0D0     0.0D0     0.0D0
    3   10   11          0.0D0     0.0D0     2.0D0     1.0D0     0.0D0     0.0D0
    4   10   11          0.0D0     0.0D0     3.0D0     1.0D0     0.0D0     0.0D0
    5   10   11          0.0D0     0.0D0     4.0D0     1.0D0     0.0D0     0.0D0
    6   10   11          0.0D0     0.0D0     5.0D0     1.0D0     0.0D0     0.0D0
    7   10   11          0.0D0     0.0D0     6.0D0     1.0D0     0.0D0     0.0D0
    8   10   11          0.0D0     0.0D0     7.0D0     1.0D0     0.0D0     0.0D0
    9   10   11          0.0D0     0.0D0     8.0D0     1.0D0     0.0D0     0.0D0
   10   10   11          0.0D0     0.0D0     9.0D0     1.0D0     0.0D0     0.0D0
   11   10   11          0.0D0     0.0D0    10.0D0     1.0D0     0.0D0     0.0D0
  122   10   11          0.0D0    10.0D0     0.0D0     1.0D0     0.0D0     0.0D0
  123   10   11          0.0D0    10.0D0     1.0D0     1.0D0     0.0D0     0.0D0
  124   10   11          0.0D0    10.0D0     2.0D0     1.0D0     0.0D0     0.0D0
  125   10   11          0.0D0    10.0D0     3.0D0     1.0D0     0.0D0     0.0D0
  126   10   11          0.0D0    10.0D0     4.0D0     1.0D0     0.0D0     0.0D0
  127   10   11          0.0D0    10.0D0     5.0D0     1.0D0     0.0D0     0.0D0
  128   10   11          0.0D0    10.0D0     6.0D0     1.0D0     0.0D0     0.0D0
  129   10   11          0.0D0    10.0D0     7.0D0     1.0D0     0.0D0     0.0D0
  130   10   11          0.0D0    10.0D0     8.0D0     1.0D0     0.0D0     0.0D0
  131   10   11          0.0D0    10.0D0     9.0D0     1.0D0     0.0D0     0.0D0
  132   10   11          0.0D0    10.0D0    10.0D0     1.0D0     0.0D0     0.0D0

    1    1    1  121    0
                                                            END OF NNPLR(K)
    1  120    1    1    1
                                                            END OF GNLR(I,1)
    1  120    1  122    1
                                                            END OF GNLR(I,2)
    1    9    1    1   12  133  122    2   13  134  123    1
   11    9    1   12   23  144  133   13   24  145  134    1
   21    9    1   23   34  155  144   24   35  156  145    1
   31    9    1   34   45  166  155   35   46  167  156    1
   41    9    1   45   56  177  166   46   57  178  167    1
   51    9    1   56   67  188  177   57   68  189  178    1
```

Table 2 (Continued)

```
  61    9    1   67   78  199  188   68   79  200  189    1
  71    9    1   78   89  210  199   79   90  211  200    1
  81    9    1   89  100  221  210   90  101  222  211    1
  91    9    1  100  111  232  221  101  112  233  222    1
                                                              END OF IE
   1   10   11          10.0D0      0.0D0
   2   10   11           9.0D0      0.0D0
   3   10   11           8.0D0      0.0D0
   4   10   11           7.0D0      0.0D0
   5   10   11           6.0D0      0.0D0
   6   10   11           5.0D0      0.0D0
   7   10   11           4.0D0      0.0D0
   8   10   11           3.0D0      0.0D0
   9   10   11           2.0D0      0.0D0
  10   10   11           1.0D0      0.0D0
  11   10   11           0.0D0      0.0D0
 122   10   11          10.0D0      0.0D0
 123   10   11           9.0D0      0.0D0
 124   10   11           8.0D0      0.0D0
 125   10   11           7.0D0      0.0D0
 126   10   11           6.0D0      0.0D0
 127   10   11           5.0D0      0.0D0
 128   10   11           4.0D0      0.0D0
 129   10   11           3.0D0      0.0D0
 130   10   11           2.0D0      0.0D0
 131   10   11           1.0D0      0.0D0
 132   10   11           0.0D0      0.0D0
                                                          END OF IC
   0    0    0    0    0    0    0    0    0    0    0    0    0    0   18   38

   2    2    6    1    2
    0.0D0    6.0D-3    1.0D38    6.0D-3
    0.0D0    0.0D00    1.0D38    0.0D00
   1    9    1    1    0
  11    7    1    2    0
                                                          END OF IRTYP
   1    9    1   11   22  143  132   11   11   11   11
  11    7    1  120  241  242  121   -1   -1   -1   -1
                                                      END OF ISV(J,I) J=1,4
   1   10    1   11   11
  12    7    1  120   -1
  20   10    1  132   11
  31    7    1  241   -1
                                                            END OF NPVB
   1   37    1           0.0D0      0.0D0
                                                          END OF HCON
```

Table 2 (Continued)

```
    1    37     1          -90.0D2      0.0D0
                                                                END OF HMIN
     0.0D0      2.0D0       1.0D38      2.0D0
  111   112   113   232   233   234
    1     5     1     1     0
                                                                       END OF IDTYP

                        END OF JOB
```

```
DIMENSION GNLR(242,2),LNOJCN(27,121,2)
DIMENSION CMTRXL(121,25),RLD(121)
DIMENSION H(242),HP(242),HW(242),HT(242),VX(242),VY(242),VZ(242)
DIMENSION TH(8,100),DTH(8,100),AKR(8,100), NPCNV(242)
DIMENSION DCOSB(3,240),ISB(6,240),NPBB(242),BFLX(242),BFLXP(242)
DIMENSION SOS(1),SOSF(1,1),TSOSF(1,1),ISTYP(1),MSEL(1)
DIMENSION WSS(1),WSSF(1,1),TWSSF(1,1),IWTYP(1),NPW(1)
DIMENSION QCB(1),QCBF(1,1),TQCBF(1,1)
DIMENSION ICTYP(1),ISC(5,1), NPCB(1)
DIMENSION QNB(1),QNBF(1,1),TQNBF(1,1)
DIMENSION INTYP(1),ISN(5,1), NPNB(1)
DIMENSION RFALL(2),RF(2,2),TRF(2,2)
DIMENSION IRTYP(18),ISV(5,18), NPVB(38)
DIMENSION DCYFLX(38),FLX(38),HCON(38),HMIN(38)
DIMENSION NPFLX(38),NPCON(38),NPMIN(38)
DIMENSION HDB(1),HDBF(2,1),THDBF(2,1),IDTYP(6),NPDB(6)
DIMENSION PROP(6,1)
DIMENSION THPROP(5,1),AKPROP(5,1)
DIMENSION KPR(1),KDSK(1),TDTCH(1)
```

Corresponding to the above dimension specification, we must make the following data statements in BLOCK DATA to specify the maximum-control integers:

```
    DATA MAXEL,MAXNP,MAXBES,MAXBNP,JBAND/100,242,240,242,27/
    DATA MAXNTI,MXNDTC/1,1/
    DATA LTMXNP,LMXNP,LMXBW,MXREGN/242,121,25,2/
    DATA MXSEL,MXSPR,MXSDP,MXWNP,MXWPR,MXWDP/1,1,1,1,1,1/
    DATA MXCNP,MXCES,MXCPR,MXCDP/1,1,1,1/
    DATA MXNNP,MXNES,MXNPR,MXNDP/1,1,1,1/
    DATA MXVES,MXVNP,MXVPR,MXVDP/18,38,2,2/
    DATA MXDNP,MXDPR,MXDDP/6,1,2/
    DATA MAXMAT,MXSPPM,MXMPPM/1,5,6/
```

To reflect the soil property function given by Eqs. (115) and (116), we have to modify the subroutine SPROP given in Appendix A. Lines between SPRO 405 and SPRO 535 in the subroutine SPROP must be modified according to

```
      WCR=THPROP(1,MTYP)
      WCS=THPROP(2,MTYP)
      HAA=THPROP(3,MTYP)
      THAA=THPROP(4,MTYP)
      THBB=THPROP(5,MTYP)
      POWER=AKPROP(1,MTYP)
      DO 800 KG=1,8
      HNP=HKG(KG)
      HNP=-HNP
C
C ------- SATURATED CONDITION
C
      IF(HNP.GT.0) GO TO 700
      TH(KG,M)=WCS
      AKR(KG,M)=1.0D0
```

```
      DTH(KG,M)=0.0D0
      GO TO 800
C
C ------- UNSATURATED CASE
C
  700 THMKG=WCS+(WCS-WCR)*THAA/(THAA+(DABS(-HNP-HAA))**THBB)
      TH(KG,M)=THMKG
      AKR(KG,M)=((THMKG-WCR)/(WCS-WCR))**POWER
      DNOM=THAA+(DABS(-HNP-HAA))**THBB
      DTH(KG,M)=(WCS-WCR)*THAA*(DABS(-HNP-THAA))**(THBB-1.0D0)/DNOM**2
C
```

The pressure head simulated with 3DFEMWATER is plotted in Fig. 11. Numerical predictions of these pressure heads are in agreement with those given by Huyakorn (1986). This verifies the one-step steady state solution algorithm provided in 3DFEMWATER. The slight difference between 3DFEMWATER and that of Huyakorn results from the implementation of the variable boundary conditions. While 3DFEMWATER allows the whole amount of flux to infiltrate if the media are capable to do so; the model given by Huyakorn only allows a fraction of flux to infiltrate as long as the ponding depth condition is not violated.

ORNL-DWG 87-9587

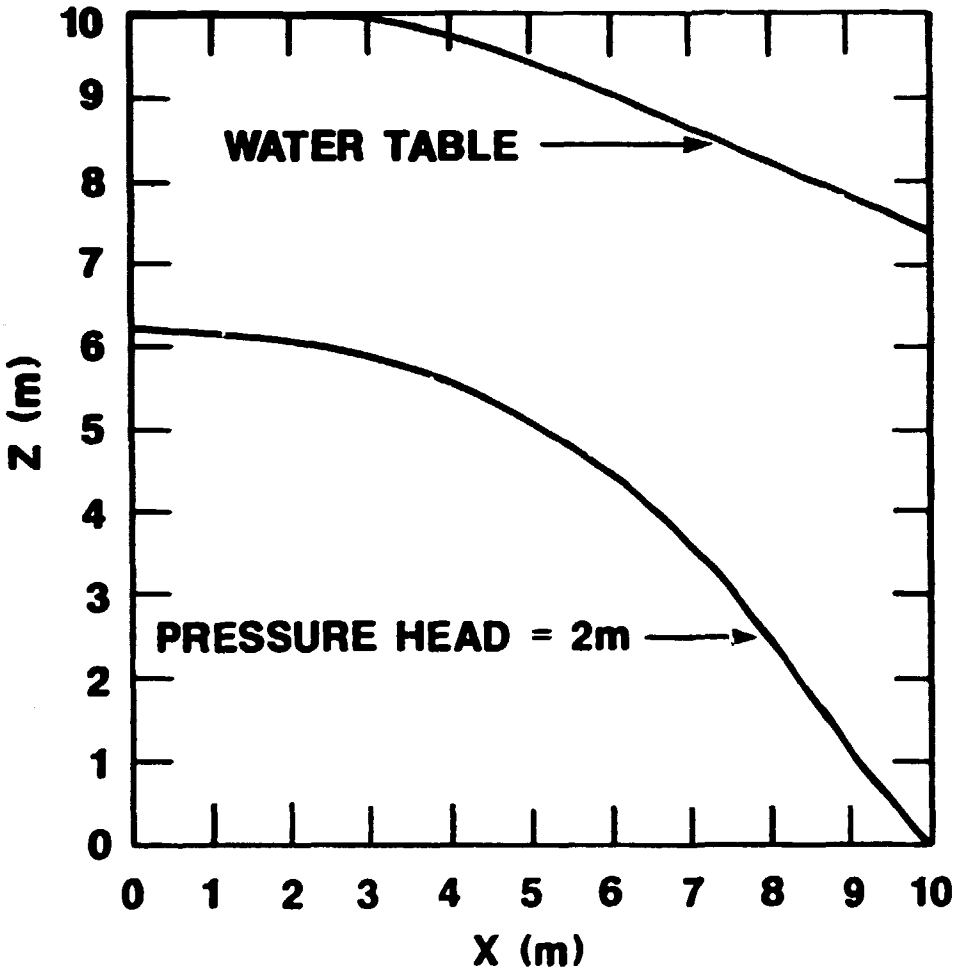

Fig. 11. Simulated steady state profiles of water table and base pressure head for the two-dimensional drainage problem.

6.3 Three-Dimensional Pumping Problem

This example is selected to represent the simulation of a three-dimensional problem with 3DFEMWATER. The problem involves the steady state flow to a pumping well. The region of interest is bounded on the left and right by hydraulically connected rivers; on the front, back, and bottom by impervious aquifuges; and on the top by an air-soil interface (Fig. 12). A pumping well is located at (x,y) = (540,400) (Fig. 12). Initially, the water table is assumed to be horizontal and is 60 m above the bottom of the aquifer. The water level at the well is then lowered to a height of 30 m. This height is held until a steady state condition is reached. The medium in the region is assumed to be anisotropic and have saturated hydraulic conductivity components K_{xx} = 5 m/d, K_{yy} = 0.5 m/d, and K_{zz} = 2 m/d. The porosity of the medium is 0.25 and the field capacity is 0.0125. The unsaturated characteristic hydraulic properties of the medium are given as

$$\theta = \theta_s + (\theta_s - \theta_r) \frac{1}{1 + (\alpha|h_a - h|)^\beta} , \qquad (117)$$

and

$$K_r = \left(\frac{\theta - \theta_r}{\theta_s - \theta_r}\right)^2 , \qquad (118)$$

where h_a, α, and β are the parameters used to compute the water content and the relative hydraulic conductivity.

ORNL-DWG 87-9580

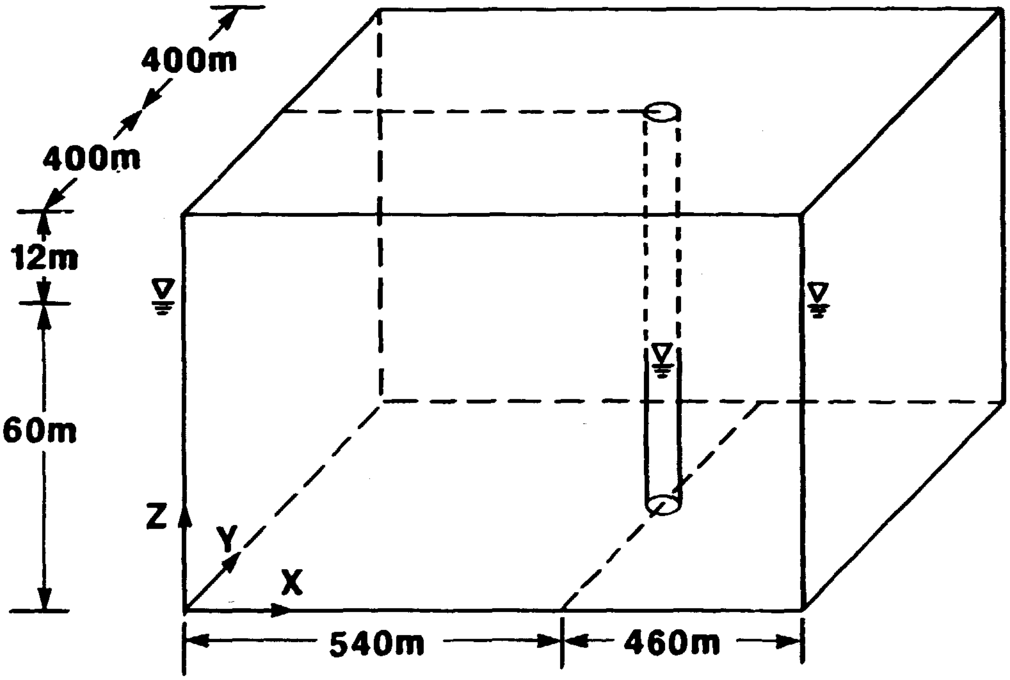

Fig. 12. Problem definition for the three-dimensional steady state flow from pumping.

Because of the symmetry, the region for numerical simulation will be taken as $0 < x < 1000$ m, $0 < y < 400$ m, and $0 < z < 72$ m. The boundary conditions are given as: pressure head is assumed hydrostatic on two vertical planes located at $x = 0$ and $0 < z < 60$, and $x = 1000$ and $0 < z < 60$, respectively; no flux is imposed on all other boundaries of the flow regime. A steady state solution will be sought. A pre-initial condition is set as $h = 60 - z$.

The region of interest is discretized with 20 x 8 x 10 = 1600 elements resulting in 21 x 9 x 11 = 2079 node points. The nodes are located at x = 0, 70, 120, 160, 200, 275, 350, 400, 450, 500, 540, 570, 600, 650, 750, 800, 850, 900, 950, and 1000 in the x-direction, and at z = 0, 15, 30, 35, 40, 45, 50, 55, 60, 66, and 72 m in the z-direction as reported by Huyakorn et at (1986). In the y-direction, nodes are spaced evenly at $\Delta z = 50$ m. For 3DFEMWATER simulation, each of the nine vertical planes perpendicular to the y-axis will be considered a subregion. Thus, a total of 9 subregions, each with 231 node points, is used for the subregional block iteration simulation.

The pressure head tolerance is 10^{-2} m for nonlinear iteration and is $5 \cdot 10^{-3}$ m for block iteration. The relaxation factors for nonlinear iteration and block iteration are set equal to 1.0 and 1.5, respectively.

With the above descriptions, the input data can be prepared according to the instructions given in Sect. 5.3. The input data are given in Table 3.

To execute the problem, the arrays in the main program should be dimensioned as

```
DIMENSION X(2079),Y(2079),Z(2079),IE(1600,9)
DIMENSION CMATRX(2079,27),GNOJCN(27,2079),RLD(2079),RI(2079),RL(2079)
DIMENSION NTNPLR(9),NNPLR(9),LMAXDF(9)
DIMENSION GNLR(693,9),LNOJCN(27,231,9)
DIMENSION CMTRXL(231,25),RLD(231)
DIMENSION H(2079),HP(2079),HW(2079),HT(2079),VX(2079),VY(2079),VZ(2079)
DIMENSION TH(8,1600),DTH(8,1600),AKR(8,1600), NPCNV(2079)
DIMENSION DCOSB(3,880),ISB(6,880),NPBB(882),BFLX(882),BFLXP(882)
DIMENSION SOS(1),SOSF(1,1),TSOSF(1,1),ISTYP(1),MSEL(1)
DIMENSION WSS(1),WSSF(1,1),TWSSF(1,1),IWTYP(1),NPW(1)
DIMENSION QCB(1),QCBF(1,1),TQCBF(1,1)
DIMENSION ICTYP(1),ISC(5,1), NPCB(1)
DIMENSION QNB(1),QNBF(1,1),TQNBF(1,1)
DIMENSION INTYP(1),ISN(5,1), NPNB(1)
DIMENSION RFALL(1),RF(1,1),TRF(1,1)
DIMENSION IRTYP(1),ISV(5,1), NPVB(1)
DIMENSION DCYFLX(1),FLX(1),HCON(1),HMIN(1)
DIMENSION NPFLX(1),NPCON(1),NPMIN(1)
DIMENSION HDB(2),HDBF(2,2),THDBF(2,2),IDTYP(165),NPDB(165)
DIMENSION PROP(6,1)
DIMENSION THPROP(5,1),AKPROP(5,1)
DIMENSION KPR(1),KDSK(1),TDTCH(1)
```

Corresponding to the above dimension specification, we must make the

Table 3. Input data for the three-dimensional pumping problem

```
    3  SIMULATION OF THREE-D PUMPING WELL; L - M, T - DAY, M - KG                 11
 2079 1600     1    0     0     0     5     6     1     0     1    50    20     3  100     9
    0.05D0      0.0D0      1.0D0     20.0D0      1.0D-2      1.0D-2      1.0D0   7.316D10
  1.1232D4      1.0D0      1.0D0      1.5D0
55
11
    1.0D38
     5.0D0      0.5D0      2.0D0      0.0D0       0.0D0       0.0D0
  0.0125D0    0.250D0     0.00D0      0.5D0       2.0D0                     THPROP
   0.000D0    0.000D0     0.00D0      0.0D0       0.0D0                     AKPROP
    1    8   231       0.00D+00   0.00D+00   0.00D+00   0.00D+00   0.50D+02   0.00D+00
    2    8   231       0.00D+00   0.00D+00   0.15D+02   0.00D+00   0.50D+02   0.00D+00
    3    8   231       0.00D+00   0.00D+00   0.30D+02   0.00D+00   0.50D+02   0.00D+00
    4    8   231       0.00D+00   0.00D+00   0.35D+02   0.00D+00   0.50D+02   0.00D+00
    5    8   231       0.00D+00   0.00D+00   0.40D+02   0.00D+00   0.50D+02   0.00D+00
    6    8   231       0.00D+00   0.00D+00   0.45D+02   0.00D+00   0.50D+02   0.00D+00
    7    8   231       0.00D+00   0.00D+00   0.50D+02   0.00D+00   0.50D+02   0.00D+00
    8    8   231       0.00D+00   0.00D+00   0.55D+02   0.00D+00   0.50D+02   0.00D+00
    9    8   231       0.00D+00   0.00D+00   0.60D+02   0.00D+00   0.50D+02   0.00D+00
   10    8   231       0.00D+00   0.00D+00   0.66D+02   0.00D+00   0.50D+02   0.00D+00
   11    8   231       0.00D+00   0.00D+00   0.72D+02   0.00D+00   0.50D+02   0.00D+00
   12    8   231       0.70D+02   0.00D+00   0.00D+00   0.00D+00   0.50D+02   0.00D+00
   13    8   231       0.70D+02   0.00D+00   0.15D+02   0.00D+00   0.50D+02   0.00D+00
   14    8   231       0.70D+02   0.00D+00   0.30D+02   0.00D+00   0.50D+02   0.00D+00
   15    8   231       0.70D+02   0.00D+00   0.35D+02   0.00D+00   0.50D+02   0.00D+00
   16    8   231       0.70D+02   0.00D+00   0.40D+02   0.00D+00   0.50D+02   0.00D+00
   17    8   231       0.70D+02   0.00D+00   0.45D+02   0.00D+00   0.50D+02   0.00D+00
   18    8   231       0.70D+02   0.00D+00   0.50D+02   0.00D+00   0.50D+02   0.00D+00
   19    8   231       0.70D+02   0.00D+00   0.55D+02   0.00D+00   0.50D+02   0.00D+00
   20    8   231       0.70D+02   0.00D+00   0.60D+02   0.00D+00   0.50D+02   0.00D+00
   21    8   231       0.70D+02   0.00D+00   0.66D+02   0.00D+00   0.50D+02   0.00D+00
   22    8   231       0.70D+02   0.00D+00   0.72D+02   0.00D+00   0.50D+02   0.00D+00
   23    8   231       0.12D+03   0.00D+00   0.00D+00   0.00D+00   0.50D+02   0.00D+00
   24    8   231       0.12D+03   0.00D+00   0.15D+02   0.00D+00   0.50D+02   0.00D+00
   25    8   231       0.12D+03   0.00D+00   0.30D+02   0.00D+00   0.50D+02   0.00D+00
   26    8   231       0.12D+03   0.00D+00   0.35D+02   0.00D+00   0.50D+02   0.00D+00
   27    8   231       0.12D+03   0.00D+00   0.40D+02   0.00D+00   0.50D+02   0.00D+00
   28    8   231       0.12D+03   0.00D+00   0.45D+02   0.00D+00   0.50D+02   0.00D+00
   29    8   231       0.12D+03   0.00D+00   0.50D+02   0.00D+00   0.50D+02   0.00D+00
   30    8   231       0.12D+03   0.00D+00   0.55D+02   0.00D+00   0.50D+02   0.00D+00
   31    8   231       0.12D+03   0.00D+00   0.60D+02   0.00D+00   0.50D+02   0.00D+00
   32    8   231       0.12D+03   0.00D+00   0.66D+02   0.00D+00   0.50D+02   0.00D+00
   33    8   231       0.12D+03   0.00D+00   0.72D+02   0.00D+00   0.50D+02   0.00D+00
   34    8   231       0.16D+03   0.00D+00   0.00D+00   0.00D+00   0.50D+02   0.00D+00
   35    8   231       0.16D+03   0.00D+00   0.15D+02   0.00D+00   0.50D+02   0.00D+00
   36    8   231       0.16D+03   0.00D+00   0.30D+02   0.00D+00   0.50D+02   0.00D+00
   37    8   231       0.16D+03   0.00D+00   0.35D+02   0.00D+00   0.50D+02   0.00D+00
```

Table 3 (Continued)

38	8	231	0.16D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
39	8	231	0.16D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
40	8	231	0.16D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
41	8	231	0.16D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
42	8	231	0.16D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
43	8	231	0.16D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
44	8	231	0.16D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
45	8	231	0.20D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
46	8	231	0.20D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
47	8	231	0.20D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
48	8	231	0.20D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
49	8	231	0.20D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
50	8	231	0.20D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
51	8	231	0.20D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
52	8	231	0.20D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
53	8	231	0.20D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
54	8	231	0.20D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
55	8	231	0.20D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
56	8	231	0.28D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
57	8	231	0.28D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
58	8	231	0.28D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
59	8	231	0.28D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
60	8	231	0.28D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
61	8	231	0.28D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
62	8	231	0.28D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
63	8	231	0.28D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
64	8	231	0.28D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
65	8	231	0.28D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
66	8	231	0.28D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
67	8	231	0.35D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
68	8	231	0.35D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
69	8	231	0.35D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
70	8	231	0.35D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
71	8	231	0.35D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
72	8	231	0.35D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
73	8	231	0.35D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
74	8	231	0.35D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
75	8	231	0.35D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
76	8	231	0.35D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
77	8	231	0.35D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
78	8	231	0.40D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
79	8	231	0.40D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
80	8	231	0.40D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
81	8	231	0.40D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
82	8	231	0.40D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00

Table 3 (Continued)

83	8	231	0.40D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
84	8	231	0.40D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
85	8	231	0.40D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
86	8	231	0.40D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
87	8	231	0.40D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
88	8	231	0.40D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
89	8	231	0.45D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
90	8	231	0.45D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
91	8	231	0.45D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
92	8	231	0.45D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
93	8	231	0.45D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
94	8	231	0.45D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
95	8	231	0.45D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
96	8	231	0.45D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
97	8	231	0.45D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
98	8	231	0.45D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
99	8	231	0.45D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
100	8	231	0.50D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
101	8	231	0.50D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
102	8	231	0.50D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
103	8	231	0.50D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
104	8	231	0.50D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
105	8	231	0.50D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
106	8	231	0.50D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
107	8	231	0.50D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
108	8	231	0.50D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
109	8	231	0.50D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
110	8	231	0.50D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
111	8	231	0.54D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
112	8	231	0.54D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
113	8	231	0.54D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
114	8	231	0.54D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
115	8	231	0.54D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
116	8	231	0.54D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
117	8	231	0.54D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
118	8	231	0.54D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
119	8	231	0.54D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
120	8	231	0.54D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
121	8	231	0.54D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
122	8	231	0.57D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
123	8	231	0.57D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
124	8	231	0.57D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
125	8	231	0.57D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
126	8	231	0.57D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
127	8	231	0.57D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00

Table 3 (Continued)

128	8	231	0.57D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
129	8	231	0.57D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
130	8	231	0.57D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
131	8	231	0.57D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
132	8	231	0.57D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
133	8	231	0.60D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
134	8	231	0.60D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
135	8	231	0.60D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
136	8	231	0.60D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
137	8	231	0.60D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
138	8	231	0.60D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
139	8	231	0.60D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
140	8	231	0.60D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
141	8	231	0.60D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
142	8	231	0.60D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
143	8	231	0.60D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
144	8	231	0.65D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
145	8	231	0.65D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
146	8	231	0.65D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
147	8	231	0.65D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
148	8	231	0.65D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
149	8	231	0.65D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
150	8	231	0.65D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
151	8	231	0.65D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
152	8	231	0.65D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
153	8	231	0.65D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
154	8	231	0.65D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
155	8	231	0.70D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
156	8	231	0.70D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
157	8	231	0.70D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
158	8	231	0.70D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
159	8	231	0.70D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
160	8	231	0.70D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
161	8	231	0.70D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
162	8	231	0.70D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
163	8	231	0.70D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
164	8	231	0.70D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
165	8	231	0.70D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
166	8	231	0.75D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
167	8	231	0.75D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
168	8	231	0.75D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
169	8	231	0.75D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
170	8	231	0.75D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
171	8	231	0.75D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
172	8	231	0.75D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00

Table 3 (Continued)

173	8	231	0.75D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
174	8	231	0.75D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
175	8	231	0.75D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
176	8	231	0.75D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
177	8	231	0.80D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
178	8	231	0.80D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
179	8	231	0.80D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
180	8	231	0.80D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
181	8	231	0.80D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
182	8	231	0.80D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
183	8	231	0.80D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
184	8	231	0.80D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
185	8	231	0.80D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
186	8	231	0.80D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
187	8	231	0.80D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
188	8	231	0.85D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
189	8	231	0.85D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
190	8	231	0.85D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
191	8	231	0.85D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
192	8	231	0.85D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
193	8	231	0.85D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
194	8	231	0.85D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
195	8	231	0.85D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
196	8	231	0.85D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
197	8	231	0.85D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
198	8	231	0.85D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
199	8	231	0.90D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
200	8	231	0.90D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
201	8	231	0.90D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
202	8	231	0.90D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
203	8	231	0.90D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
204	8	231	0.90D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
205	8	231	0.90D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
206	8	231	0.90D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00
207	8	231	0.90D+03	0.00D+00	0.60D+02	0.00D+00	0.50D+02	0.00D+00
208	8	231	0.90D+03	0.00D+00	0.66D+02	0.00D+00	0.50D+02	0.00D+00
209	8	231	0.90D+03	0.00D+00	0.72D+02	0.00D+00	0.50D+02	0.00D+00
210	8	231	0.95D+03	0.00D+00	0.00D+00	0.00D+00	0.50D+02	0.00D+00
211	8	231	0.95D+03	0.00D+00	0.15D+02	0.00D+00	0.50D+02	0.00D+00
212	8	231	0.95D+03	0.00D+00	0.30D+02	0.00D+00	0.50D+02	0.00D+00
213	8	231	0.95D+03	0.00D+00	0.35D+02	0.00D+00	0.50D+02	0.00D+00
214	8	231	0.95D+03	0.00D+00	0.40D+02	0.00D+00	0.50D+02	0.00D+00
215	8	231	0.95D+03	0.00D+00	0.45D+02	0.00D+00	0.50D+02	0.00D+00
216	8	231	0.95D+03	0.00D+00	0.50D+02	0.00D+00	0.50D+02	0.00D+00
217	8	231	0.95D+03	0.00D+00	0.55D+02	0.00D+00	0.50D+02	0.00D+00

Table 3 (Continued)

```
 218    8   231      0.95D+03  0.00D+00  0.60D+02  0.00D+00  0.50D+02  0.00D+00
 219    8   231      0.95D+03  0.00D+00  0.66D+02  0.00D+00  0.50D+02  0.00D+00
 220    8   231      0.95D+03  0.00D+00  0.72D+02  0.00D+00  0.50D+02  0.00D+00
 221    8   231      0.10D+04  0.00D+00  0.00D+00  0.00D+00  0.50D+02  0.00D+00
 222    8   231      0.10D+04  0.00D+00  0.15D+02  0.00D+00  0.50D+02  0.00D+00
 223    8   231      0.10D+04  0.00D+00  0.30D+02  0.00D+00  0.50D+02  0.00D+00
 224    8   231      0.10D+04  0.00D+00  0.35D+02  0.00D+00  0.50D+02  0.00D+00
 225    8   231      0.10D+04  0.00D+00  0.40D+02  0.00D+00  0.50D+02  0.00D+00
 226    8   231      0.10D+04  0.00D+00  0.45D+02  0.00D+00  0.50D+02  0.00D+00
 227    8   231      0.10D+04  0.00D+00  0.50D+02  0.00D+00  0.50D+02  0.00D+00
 228    8   231      0.10D+04  0.00D+00  0.55D+02  0.00D+00  0.50D+02  0.00D+00
 229    8   231      0.10D+04  0.00D+00  0.60D+02  0.00D+00  0.50D+02  0.00D+00
 230    8   231      0.10D+04  0.00D+00  0.66D+02  0.00D+00  0.50D+02  0.00D+00
 231    8   231      0.10D+04  0.00D+00  0.72D+02  0.00D+00  0.50D+02  0.00D+00
                                                    END OF COORDINATE
   1    8    1  231    0
                                              END OF NNPLR(9)
   1  230    1    1    1
                                              END OF GNLR(I,1)
   1  230    1  232    1
                                              END OF GNLR(I,2)
   1  230    1  463    1
                                              END OF GNLR(I,3)
   1  230    1  694    1
                                              END OF GNLR(I,4)
   1  230    1  925    1
                                              END OF GNLR(I,5)
   1  230    1 1156    1
                                              END OF GNLR(I,6)
   1  230    1 1387    1
                                              END OF GNLR(I,7)
   1  230    1 1618    1
                                              END OF GNLR(I,8)
   1  230    1 1849    1
                                              END OF GNLR(I,9)
   1    9    1    1   12   13    2  232  243  244  233    1
  11    9    1   12   23   24   13  243  254  255  244    1
  21    9    1   23   34   35   24  254  265  266  255    1
  31    9    1   34   45   46   35  265  276  277  266    1
  41    9    1   45   56   57   46  276  287  288  277    1
  51    9    1   56   67   68   57  287  298  299  288    1
  61    9    1   67   78   79   68  298  309  310  299    1
  71    9    1   78   89   90   79  309  320  321  310    1
  81    9    1   89  100  101   90  320  331  332  321    1
  91    9    1  100  111  112  101  331  342  343  332    1
```

Table 3 (Continued)

101	9	1	111	122	123	112	342	353	354	343	1
111	9	1	122	133	134	123	353	364	365	354	1
121	9	1	133	144	145	134	364	375	376	365	1
131	9	1	144	155	156	145	375	386	387	376	1
141	9	1	155	166	167	156	386	397	398	387	1
151	9	1	166	177	178	167	397	408	409	398	1
161	9	1	177	188	189	178	408	419	420	409	1
171	9	1	188	199	200	189	419	430	431	420	1
181	9	1	199	210	211	200	430	441	442	431	1
191	9	1	210	221	222	211	441	452	453	442	1
201	9	1	232	243	244	233	463	474	475	464	1
211	9	1	243	254	255	244	474	485	486	475	1
221	9	1	254	265	266	255	485	496	497	486	1
231	9	1	265	276	277	266	496	507	508	497	1
241	9	1	276	287	288	277	507	518	519	508	1
251	9	1	287	298	299	288	518	529	530	519	1
261	9	1	298	309	310	299	529	540	541	530	1
271	9	1	309	320	321	310	540	551	552	541	1
281	9	1	320	331	332	321	551	562	563	552	1
291	9	1	331	342	343	332	562	573	574	563	1
301	9	1	342	353	354	343	573	584	585	574	1
311	9	1	353	364	365	354	584	595	596	585	1
321	9	1	364	375	376	365	595	606	607	596	1
331	9	1	375	386	387	376	606	617	618	607	1
341	9	1	386	397	398	387	617	628	629	618	1
351	9	1	397	408	409	398	628	639	640	629	1
361	9	1	408	419	420	409	639	650	651	640	1
371	9	1	419	430	431	420	650	661	662	651	1
381	9	1	430	441	442	431	661	672	673	662	1
391	9	1	441	452	453	442	672	683	684	673	1
401	9	1	463	474	475	464	694	705	706	695	1
411	9	1	474	485	486	475	705	716	717	706	1
421	9	1	485	496	497	486	716	727	728	717	1
431	9	1	496	507	508	497	727	738	739	728	1
441	9	1	507	518	519	508	738	749	750	739	1
451	9	1	518	529	530	519	749	760	761	750	1
461	9	1	529	540	541	530	760	771	772	761	1
471	9	1	540	551	552	541	771	782	783	772	1
481	9	1	551	562	563	552	782	793	794	783	1
491	9	1	562	573	574	563	793	804	805	794	1
501	9	1	573	584	585	574	804	815	816	805	1
511	9	1	584	595	596	585	815	826	827	816	1
521	9	1	595	606	607	596	826	837	838	827	1
531	9	1	606	617	618	607	837	848	849	838	1
541	9	1	617	628	629	618	848	859	860	849	1

Table 3 (Continued)

551	9	1	628	639	640	629	859	870	871	860	1
561	9	1	639	650	651	640	870	881	882	871	1
571	9	1	650	661	662	651	881	892	893	882	1
581	9	1	661	672	673	662	892	903	904	893	1
591	9	1	672	683	684	673	903	914	915	904	1
601	9	1	694	705	706	695	925	936	937	926	1
611	9	1	705	716	717	706	936	947	948	937	1
621	9	1	716	727	728	717	947	958	959	948	1
631	9	1	727	738	739	728	958	969	970	959	1
641	9	1	738	749	750	739	969	980	981	970	1
651	9	1	749	760	761	750	980	991	992	981	1
661	9	1	760	771	772	761	991	1002	1003	992	1
671	9	1	771	782	783	772	1002	1013	1014	1003	1
681	9	1	782	793	794	783	1013	1024	1025	1014	1
691	9	1	793	804	805	794	1024	1035	1036	1025	1
701	9	1	804	815	816	805	1035	1046	1047	1036	1
711	9	1	815	826	827	816	1046	1057	1058	1047	1
721	9	1	826	837	838	827	1057	1068	1069	1058	1
731	9	1	837	848	849	838	1068	1079	1080	1069	1
741	9	1	848	859	860	849	1079	1090	1091	1080	1
751	9	1	859	870	871	860	1090	1101	1102	1091	1
761	9	1	870	881	882	871	1101	1112	1113	1102	1
771	9	1	881	892	893	882	1112	1123	1124	1113	1
781	9	1	892	903	904	893	1123	1134	1135	1124	1
791	9	1	903	914	915	904	1134	1145	1146	1135	1
801	9	1	925	936	937	926	1156	1167	1168	1157	1
811	9	1	936	947	948	937	1167	1178	1179	1168	1
821	9	1	947	958	959	948	1178	1189	1190	1179	1
831	9	1	958	969	970	959	1189	1200	1201	1190	1
841	9	1	969	980	981	970	1200	1211	1212	1201	1
851	9	1	980	991	992	981	1211	1222	1223	1212	1
861	9	1	991	1002	1003	992	1222	1233	1234	1223	1
871	9	1	1002	1013	1014	1003	1233	1244	1245	1234	1
881	9	1	1013	1024	1025	1014	1244	1255	1256	1245	1
891	9	1	1024	1035	1036	1025	1255	1266	1267	1256	1
901	9	1	1035	1046	1047	1036	1266	1277	1278	1267	1
911	9	1	1046	1057	1058	1047	1277	1288	1289	1278	1
921	9	1	1057	1068	1069	1058	1288	1299	1300	1289	1
931	9	1	1068	1079	1080	1069	1299	1310	1311	1300	1
941	9	1	1079	1090	1091	1080	1310	1321	1322	1311	1
951	9	1	1090	1101	1102	1091	1321	1332	1333	1322	1
961	9	1	1101	1112	1113	1102	1332	1343	1344	1333	1
971	9	1	1112	1123	1124	1113	1343	1354	1355	1344	1
981	9	1	1123	1134	1135	1124	1354	1365	1366	1355	1
991	9	1	1134	1145	1146	1135	1365	1376	1377	1366	1

Table 3 (Continued)

1001	9	1	1156	1167	1168	1157	1387	1398	1399	1388	1
1011	9	1	1167	1178	1179	1168	1398	1409	1410	1399	1
1021	9	1	1178	1189	1190	1179	1409	1420	1421	1410	1
1031	9	1	1189	1200	1201	1190	1420	1431	1432	1421	1
1041	9	1	1200	1211	1212	1201	1431	1442	1443	1432	1
1051	9	1	1211	1222	1223	1212	1442	1453	1454	1443	1
1061	9	1	1222	1233	1234	1223	1453	1464	1465	1454	1
1071	9	1	1233	1244	1245	1234	1464	1475	1476	1465	1
1081	9	1	1244	1255	1256	1245	1475	1486	1487	1476	1
1091	9	1	1255	1266	1267	1256	1486	1497	1498	1487	1
1101	9	1	1266	1277	1278	1267	1497	1508	1509	1498	1
1111	9	1	1277	1288	1289	1278	1508	1519	1520	1509	1
1121	9	1	1288	1299	1300	1289	1519	1530	1531	1520	1
1131	9	1	1299	1310	1311	1300	1530	1541	1542	1531	1
1141	9	1	1310	1321	1322	1311	1541	1552	1553	1542	1
1151	9	1	1321	1332	1333	1322	1552	1563	1564	1553	1
1161	9	1	1332	1343	1344	1333	1563	1574	1575	1564	1
1171	9	1	1343	1354	1355	1344	1574	1585	1586	1575	1
1181	9	1	1354	1365	1366	1355	1585	1596	1597	1586	1
1191	9	1	1365	1376	1377	1366	1596	1607	1608	1597	1
1201	9	1	1387	1398	1399	1388	1618	1629	1630	1619	1
1211	9	1	1398	1409	1410	1399	1629	1640	1641	1630	1
1221	9	1	1409	1420	1421	1410	1640	1651	1652	1641	1
1231	9	1	1420	1431	1432	1421	1651	1662	1663	1652	1
1241	9	1	1431	1442	1443	1432	1662	1673	1674	1663	1
1251	9	1	1442	1453	1454	1443	1673	1684	1685	1674	1
1261	9	1	1453	1464	1465	1454	1684	1695	1696	1685	1
1271	9	1	1464	1475	1476	1465	1695	1706	1707	1696	1
1281	9	1	1475	1486	1487	1476	1706	1717	1718	1707	1
1291	9	1	1486	1497	1498	1487	1717	1728	1729	1718	1
1301	9	1	1497	1508	1509	1498	1728	1739	1740	1729	1
1311	9	1	1508	1519	1520	1509	1739	1750	1751	1740	1
1321	9	1	1519	1530	1531	1520	1750	1761	1762	1751	1
1331	9	1	1530	1541	1542	1531	1761	1772	1773	1762	1
1341	9	1	1541	1552	1553	1542	1772	1783	1784	1773	1
1351	9	1	1552	1563	1564	1553	1783	1794	1795	1784	1
1361	9	1	1563	1574	1575	1564	1794	1805	1806	1795	1
1371	9	1	1574	1585	1586	1575	1605	1816	1817	1806	1
1381	9	1	1585	1596	1597	1586	1816	1827	1828	1817	1
1391	9	1	1596	1607	1608	1597	1827	1838	1839	1828	1
1401	9	1	1618	1629	1630	1619	1849	1860	1861	1850	1
1411	9	1	1629	1640	1641	1630	1860	1871	1872	1861	1
1421	9	1	1640	1651	1652	1641	1871	1882	1883	1872	1
1431	9	1	1651	1662	1663	1652	1882	1893	1894	1883	1
1441	9	1	1662	1673	1674	1663	1893	1904	1905	1894	1

Table 3 (Continued)

```
1451   9    1 1673 1684 1685 1674 1904 1915 1916 1905     1
1461   9    1 1684 1695 1696 1685 1915 1926 1927 1916     1
1471   9    1 1695 1706 1707 1696 1926 1937 1938 1927     1
1481   9    1 1706 1717 1718 1707 1937 1948 1949 1938     1
1491   9    1 1717 1728 1729 1718 1948 1959 1960 1949     1
1501   9    1 1728 1739 1740 1729 1959 1970 1971 1960     1
1511   9    1 1739 1750 1751 1740 1970 1981 1982 1971     1
1521   9    1 1750 1761 1762 1751 1981 1992 1993 1982     1
1531   9    1 1761 1772 1773 1762 1992 2003 2004 1993     1
1541   9    1 1772 1783 1784 1773 2003 2014 2015 2004     1
1551   9    1 1783 1794 1795 1784 2014 2025 2026 2015     1
1561   9    1 1794 1805 1806 1795 2025 2036 2037 2025     1
1571   9    1 1805 1816 1817 1806 2036 2047 2048 2037     1
1581   9    1 1816 1827 1828 1817 2047 2058 2059 2048     1
1591   9    1 1827 1838 1839 1828 2058 2069 2070 2059     1
                                                                END OF IE
   1   8  231        0.60D+02  0.00D+00
   2   8  231        0.45D+02  0.00D+00
   3   8  231        0.30D+02  0.00D+00
   4   8  231        0.25D+02  0.00D+00
   5   8  231        0.20D+02  0.00D+00
   6   8  231        0.15D+02  0.00D+00
   7   8  231        0.10D+02  0.00D+00
   8   8  231        0.50D+01  0.00D+00
   9   8  231        0.00D+00  0.00D+00
  10   8  231       -0.60D+01  0.00D+00
  11   8  231       -0.12D+02  0.00D+00
  12   8  231        0.60D+02  0.00D+00
  13   8  231        0.45D+02  0.00D+00
  14   8  231        0.30D+02  0.00D+00
  15   8  231        0.25D+02  0.00D+00
  16   8  231        0.20D+02  0.00D+00
  17   8  231        0.15D+02  0.00D+00
  18   8  231        0.10D+02  0.00D+00
  19   8  231        0.50D+01  0.00D+00
  20   8  231        0.00D+00  0.00D+00
  21   8  231       -0.60D+01  0.00D+00
  22   8  231       -0.12D+02  0.00D+00
  23   8  231        0.60D+02  0.00D+00
  24   8  231        0.45D+02  0.00D+00
  25   8  231        0.30D+02  0.00D+00
  26   8  231        0.25D+02  0.00D+00
  27   8  231        0.20D+02  0.00D+00
  28   8  231        0.15D+02  0.00D+00
  29   8  231        0.10D+02  0.00D+00
```

Table 3 (Continued)

30	8	231	0.50D+01	0.00D+00
31	8	231	0.00D+00	0.00D+00
32	8	231	-0.60D+01	0.00D+00
33	8	231	-0.12D+02	0.00D+00
34	8	231	0.60D+02	0.00D+00
35	8	231	0.45D+02	0.00D+00
36	8	231	0.30D+02	0.00D+00
37	8	231	0.25D+02	0.00D+00
38	8	231	0.20D+02	0.00D+00
39	8	231	0.15D+02	0.00D+00
40	8	231	0.10D+02	0.00D+00
41	8	231	0.50D+01	0.00D+00
42	8	231	0.00D+00	0.00D+00
43	8	231	-0.60D+01	0.00D+00
44	8	231	-0.12D+02	0.00D+00
45	8	231	0.60D+02	0.00D+00
46	8	231	0.45D+02	0.00D+00
47	8	231	0.30D+02	0.00D+00
48	8	231	0.25D+02	0.00D+00
49	8	231	0.20D+02	0.00D+00
50	8	231	0.15D+02	0.00D+00
51	8	231	0.10D+02	0.00D+00
52	8	231	0.50D+01	0.00D+00
53	8	231	0.00D+00	0.00D+00
54	8	231	-0.60D+01	0.00D+00
55	8	231	-0.12D+02	0.00D+00
56	8	231	0.60D+02	0.00D+00
57	8	231	0.45D+02	0.00D+00
58	8	231	0.30D+02	0.00D+00
59	8	231	0.25D+02	0.00D+00
60	8	231	0.20D+02	0.00D+00
61	8	231	0.15D+02	0.00D+00
62	8	231	0.10D+02	0.00D+00
63	8	231	0.50D+01	0.00D+00
64	8	231	0.00D+00	0.00D+00
65	8	231	-0.60D+01	0.00D+00
66	8	231	-0.12D+02	0.00D+00
67	8	231	0.60D+02	0.00D+00
68	8	231	0.45D+02	0.00D+00
69	8	231	0.30D+02	0.00D+00
70	8	231	0.25D+02	0.00D+00
71	8	231	0.20D+02	0.00D+00
72	8	231	0.15D+02	0.00D+00
73	8	231	0.10D+02	0.00D+00
74	8	231	0.50D+01	0.00D+00

Table 3 (Continued)

75	8	231	0.00D+00	0.00D+00
76	8	231	-0.60D+01	0.00D+00
77	8	231	-0.12D+02	0.00D+00
78	8	231	0.60D+02	0.00D+00
79	8	231	0.45D+02	0.00D+00
80	8	231	0.30D+02	0.00D+00
81	8	231	0.25D+02	0.00D+00
82	8	231	0.20D+02	0.00D+00
83	8	231	0.15D+02	0.00D+00
84	8	231	0.10D+02	0.00D+00
85	8	231	0.50D+01	0.00D+00
86	8	231	0.00D+00	0.00D+00
87	8	231	-0.60D+01	0.00D+00
88	8	231	-0.12D+02	0.00D+00
89	8	231	0.60D+02	0.00D+00
90	8	231	0.45D+02	0.00D+00
91	8	231	0.30D+02	0.00D+00
92	8	231	0.25D+02	0.00D+00
93	8	231	0.20D+02	0.00D+00
94	8	231	0.15D+02	0.00D+00
95	8	231	0.10D+02	0.00D+00
96	8	231	0.50D+01	0.00D+00
97	8	231	0.00D+00	0.00D+00
98	8	231	-0.60D+01	0.00D+00
99	8	231	-0.12D+02	0.00D+00
100	8	231	0.60D+02	0.00D+00
101	8	231	0.45D+02	0.00D+00
102	8	231	0.30D+02	0.00D+00
103	8	231	0.25D+02	0.00D+00
104	8	231	0.20D+02	0.00D+00
105	8	231	0.15D+02	0.00D+00
106	8	231	0.10D+02	0.00D+00
107	8	231	0.50D+01	0.00D+00
108	8	231	0.00D+00	0.00D+00
109	8	231	-0.60D+01	0.00D+00
110	8	231	-0.12D+02	0.00D+00
111	8	231	0.60D+02	0.00D+00
112	8	231	0.45D+02	0.00D+00
113	8	231	0.30D+02	0.00D+00
114	8	231	0.25D+02	0.00D+00
115	8	231	0.20D+02	0.00D+00
116	8	231	0.15D+02	0.00D+00
117	8	231	0.10D+02	0.00D+00
118	8	231	0.50D+01	0.00D+00
119	8	231	0.00D+00	0.00D+00

Table 3 (Continued)

120	8	231	-0.60D+01	0.00D+00
121	8	231	-0.12D+02	0.00D+00
122	8	231	0.60D+02	0.00D+00
123	8	231	0.45D+02	0.00D+00
124	8	231	0.30D+02	0.00D+00
125	8	231	0.25D+02	0.00D+00
126	8	231	0.20D+02	0.00D+00
127	8	231	0.15D+02	0.00D+00
128	8	231	0.10D+02	0.00D+00
129	8	231	0.50D+01	0.00D+00
130	8	231	0.00D+00	0.00D+00
131	8	231	-0.60D+01	0.00D+00
132	8	231	-0.12D+02	0.00D+00
133	8	231	0.60D+02	0.00D+00
134	8	231	0.45D+02	0.00D+00
135	8	231	0.30D+02	0.00D+00
136	8	231	0.25D+02	0.00D+00
137	8	231	0.20D+02	0.00D+00
138	8	231	0.15D+02	0.00D+00
139	8	231	0.10D+02	0.00D+00
140	8	231	0.50D+01	0.00D+00
141	8	231	0.00D+00	0.00D+00
142	8	231	-0.60D+01	0.00D+00
143	8	231	-0.12D+02	0.00D+00
144	8	231	0.60D+02	0.00D+00
145	8	231	0.45D+02	0.00D+00
146	8	231	0.30D+02	0.00D+00
147	8	231	0.25D+02	0.00D+00
148	8	231	0.20D+02	0.00D+00
149	8	231	0.15D+02	0.00D+00
150	8	231	0.10D+02	0.00D+00
151	8	231	0.50D+01	0.00D+00
152	8	231	0.00D+00	0.00D+00
153	8	231	-0.60D+01	0.00D+00
154	8	231	-0.12D+02	0.00D+00
155	8	231	0.60D+02	0.00D+00
156	8	231	0.45D+02	0.00D+00
157	8	231	0.30D+02	0.00D+00
158	8	231	0.25D+02	0.00D+00
159	8	231	0.20D+02	0.00D+00
160	8	231	0.15D+02	0.00D+00
161	8	231	0.10D+02	0.00D+00
162	8	231	0.50D+01	0.00D+00
163	8	231	0.00D+00	0.00D+00
164	8	231	-0.60D+01	0.00D+00

Table 3 (Continued)

165	8	231	-0.12D+02	0.00D+00
166	8	231	0.60D+02	0.00D+00
167	8	231	0.45D+02	0.00D+00
168	8	231	0.30D+02	0.00D+00
169	8	231	0.25D+02	0.00D+00
170	8	231	0.20D+02	0.00D+00
171	8	231	0.15D+02	0.00D+00
172	8	231	0.10D+02	0.00D+00
173	8	231	0.50D+01	0.00D+00
174	8	231	0.00D+00	0.00D+00
175	8	231	-0.60D+01	0.00D+00
176	8	231	-0.12D+02	0.00D+00
177	8	231	0.60D+02	0.00D+00
178	8	231	0.45D+02	0.00D+00
179	8	231	0.30D+02	0.00D+00
180	8	231	0.25D+02	0.00D+00
181	8	231	0.20D+02	0.00D+00
182	8	231	0.15D+02	0.00D+00
183	8	231	0.10D+02	0.00D+00
184	8	231	0.50D+01	0.00D+00
185	8	231	0.00D+00	0.00D+00
186	8	231	-0.60D+01	0.00D+00
187	8	231	-0.12D+02	0.00D+00
188	8	231	0.60D+02	0.00D+00
189	8	231	0.45D+02	0.00D+00
190	8	231	0.30D+02	0.00D+00
191	8	231	0.25D+02	0.00D+00
192	8	231	0.20D+02	0.00D+00
193	8	231	0.15D+02	0.00D+00
194	8	231	0.10D+02	0.00D+00
195	8	231	0.50D+01	0.00D+00
196	8	231	0.00D+00	0.00D+00
197	8	231	-0.60D+01	0.00D+00
198	8	231	-0.12D+02	0.00D+00
199	8	231	0.60D+02	0.00D+00
200	8	231	0.45D+02	0.00D+00
201	8	231	0.30D+02	0.00D+00
202	8	231	0.25D+02	0.00D+00
203	8	231	0.20D+02	0.00D+00
204	8	231	0.15D+02	0.00D+00
205	8	231	0.10D+02	0.00D+00
206	8	231	0.50D+01	0.00D+00
207	8	231	0.00D+00	0.00D+00
208	8	231	-0.60D+01	0.00D+00
209	8	231	-0.12D+02	0.00D+00

Table 3 (Continued)

```
 210     8  231        0.60D+02  0.00D+00
 211     8  231        0.45D+02  0.00D+00
 212     8  231        0.30D+02  0.00D+00
 213     8  231        0.25D+02  0.00D+00
 214     8  231        0.20D+02  0.00D+00
 215     8  231        0.15D+02  0.00D+00
 216     8  231        0.10D+02  0.00D+00
 217     8  231        0.50D+01  0.00D+00
 218     8  231        0.00D+00  0.00D+00
 219     8  231       -0.60D+01  0.00D+00
 220     8  231       -0.12D+02  0.00D+00
 221     8  231        0.60D+02  0.00D+00
 222     8  231        0.45D+02  0.00D+00
 223     8  231        0.30D+02  0.00D+00
 224     8  231        0.25D+02  0.00D+00
 225     8  231        0.20D+02  0.00D+00
 226     8  231        0.15D+02  0.00D+00
 227     8  231        0.10D+02  0.00D+00
 228     8  231        0.50D+01  0.00D+00
 229     8  231        0.00D+00  0.00D+00
 230     8  231       -0.60D+01  0.00D+00
 231     8  231       -0.12D+02  0.00D+00
                                                                   END OF IC
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0  164    2    2
   0.0D0     60.0D0     1.0D38     60.0D0
   0.0D0     30.0D0     1.0D38     30.0D0
    1    2    3    4    5    6    7    8    9  232  233  234  235  236  237  238
  239  240  463  464  465  466  467  468  469  470  471  694  695  696  697  698
  699  700  701  702  925  926  927  928  929  930  931  932  933 1156 1157 1158
 1159 1160 1161 1162 1163 1164 1387 1388 1389 1390 1391 1392 1393 1394 1395 1618
 1619 1620 1621 1622 1623 1624 1625 1626 1849 1850 1851 1852 1853 1854 1855 1856
 1857  221  222  223  224  225  226  227  228  229  452  453  454  455  456  457
  458  459  460  683  684  685  686  687  688  689  690  691  914  915  916  917
  918  919  920  921  922 1145 1146 1147 1148 1149 1150 1151 1152 1153 1376 1377
 1378 1379 1380 1381 1382 1383 1384 1607 1608 1609 1610 1611 1612 1613 1614 1615
 1838 1839 1840 1841 1842 1843 1844 1845 1846 2069 2070 2071 2072 2073 2074 2075
 2076 2077  111  112
    1  161    1    1    0
  163    1    1    2    0
                                                               END OF IDTYP
                                    END OF JOB
```

the following statements in BLOCK DATA to specify the maximum-control integers:

```
     DATA MAXEL,MAXNP,MAXBES,MAXBNP,JBAND/1600,2079,880,882,27/
     DATA MAXNTI,MXNDTC/1,1/
     DATA LTMXNP,LMXNP,LMXBW,MXREGN/693,231,25,9/
     DATA MXSEL,MXSPR,MXSDP,MXWNP,MXWPR,MXWDP/1,1,1,1,1,1/
     DATA MXCNP,MXCES,MXCPR,MXCDP/1,1,1,1/
     DATA MXNNP,MXNES,MXNPR,MXNDP/1,1,1,1/
     DATA MXVES,MXVNP,MXVPR,MXVDP/1,1,1,1/
     DATA MXDNP,MXDPR,MXDDP/165,2,2/
     DATA MAXMAT,MXSPPM,MXMPPM/1,5,6/
```

To reflect the soil property function given by Eqs. (117) and (118), we have to modify the subroutine SPROP given in Appendix A. Lines between SPRO 405 and SPRO 535 in the subroutine SPROP must be modified according to

```
      WCR-THPROP(1,MTYP)
      WCS-THPROP(2,MTYP)
      HAA-THPROP(3,MTYP)
      ALPHA-THPROP(4,MTYP)
      BETA-THPROP(5,MTYP)
      DO 800 KG-1,8
      HNP-HKG(KG)
      HNP--HNP
C
C ------- SATURATED CONDITION
C
      IF(HNP.GT.0) GO TO 700
      TH(KG,M)-WCS
      AKR(KG,M)-1.0D0
      DTH(KG,M)-0.0D0
      GO TO 800
C
C ------- UNSATURATED CASE
C
```

```
700 THMKG=WCS+(WCS-WCR)/(1.0D0+(ALPHA*DABS(-HNP-HAA))**BETA)
    TH(KG,M)=THMKG
    AKR(KG,M)=((THMKG-WCR)/(WCS-WCR))**2
    DNOM=1.0D0+(ALPHA*DABS(-HNP-HAA))**BETA
    DTH(KG,M)=(WCS-WCR)*(ALPHA*DABS(-HNP-THAA))**(BETA-1.0D0)/DNOM**2
```

The pressure head simulated with 3DFEMWATER is plotted in Figs. 13(a) and 13(b), respectively, for longitudinal and transverse sections through the pumping well. Numerical predictions of these pressure heads are in good agreement with those given by Huyakorn (1986). This verifies the three-dimensional algorithm of 3DFEMWATER. There are some differences between the two model predictions around the pumping well, which are probably due to different boundary conditions imposed on the pumping well. Because it is not clear what boundary conditions were imposed on the pumping well by Huyakorn, we have imposed a constant total head on the three nodes in the pumping well.

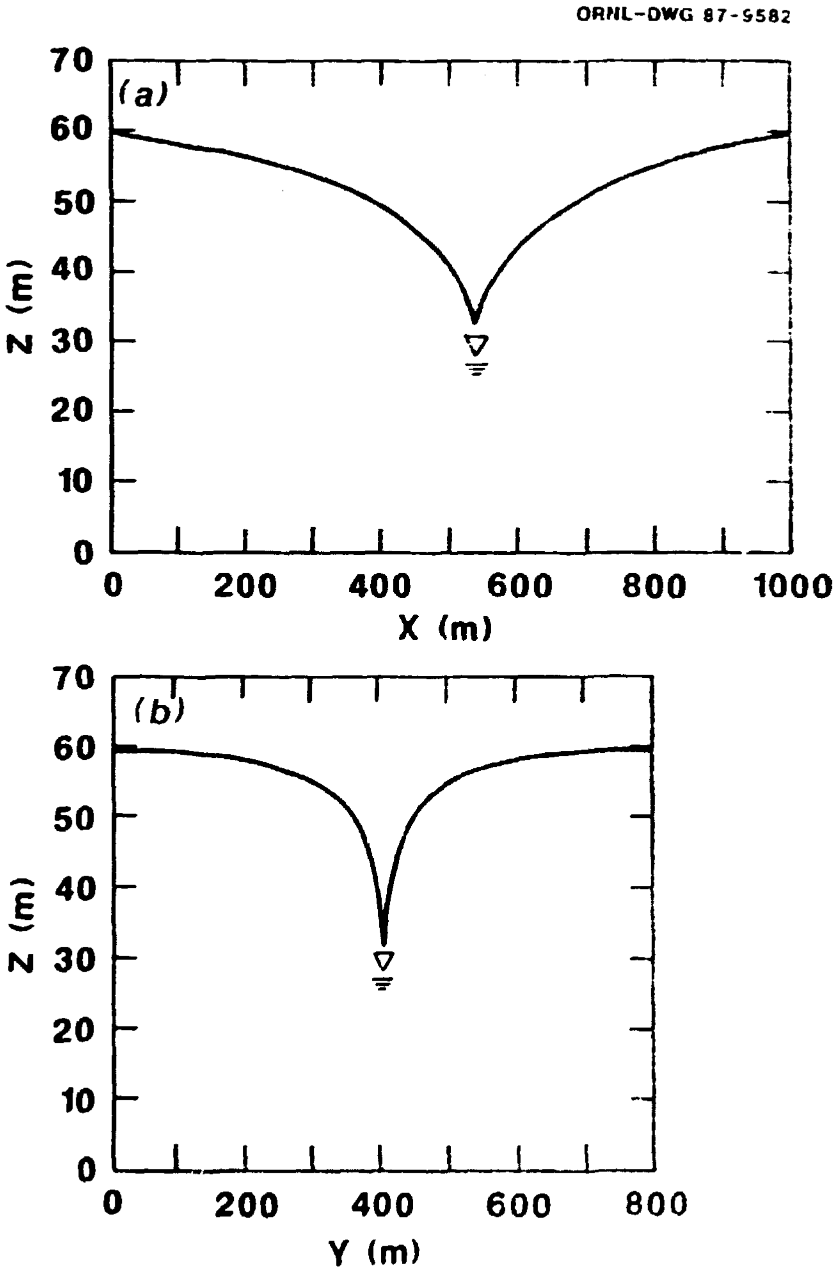

Fig. 13. Simulated steady state profiles of water table over (a) longitudinal and (b) transverse cross-sections through the well for the three-dimensional pumping problem.

7. REFERENCES

Bear, J. 1972. Dynamics of Fluids in Porous Media. American Elsevier, New York. 764 pp.

Biot, M. 1940. General theory of three-dimensional consolidation. J. Appl. Phys. 12:155-164.

Conte, S. D. 1965. Elementary Numerical Analysis. McGraw-Hill, New York. 278 pp.

Forsythe, G. E., and W. R. Wasow. 1960. Finite-Difference Methods for Partial Differential Equations. John Wiley & Sons, New York, 444 pp.

Freeze, R. A. 1972. A physics-based approach to hydrologic response modeling: Phase I: model development. Completion report for OWRR Contract No. 14-31-001-3694. Department of the Interior, Washington, D.C. 119 pp.

Huebner, K. H. 1975. The Finite Element Methods for Engineers, John Wiley & Sons, New York. 500 pp.

Huyakorn, P. S., and S. D. Thomas. 1984. Techniques for making finite elements competitive in modeling flow in variably saturated porous media. Water Resour. Res. 20(8):1099-1115.

Huyakorn, P. S. 1986. A three-dimensional finite element model for simulating water flow in variably saturated porous media, Water Resour. Res. 22(13):1790-1809.

Lapidus, L., and G. F. Pinder. 1982. Numerical Solution of Partial Differential Equations in Science and Engineering. John Wiley & Sons, New York. 677 pp.

Narasimhan, T. N., and P. A. Witherspoon. 1977. Numerical model for saturated-unsaturated flow in deformable porous media. Water Resour. Res. 13(3):657-664.

Neuman, S. P., R. A. Feddes, and E. Bresler. 1974. Finite element simulation of flow in saturated-unsaturated soils considering water uptake by plants. Report Project A10-SWC-77, Hydrodynamic and Hydraulic Engineering Laboratory, Technion, Haifa, Israel. 104 pp.

Owczarek, J. A. 1964. Fundamentals of Gas Dynamics. International Textbook, Scranton, Pa. 648 pp.

Polubarinova-Kochina, P. Ya. 1962. Theory of Groundwater Movement. Princeton University Press, Princeton, N. J. 613 pp.

Reeves, M., and J. O. Duguid. 1975. Water movement through saturated-unsaturated porous media: A finite-element Galerkin model. ORNL-4927. Oak Ridge National Laboratory, Oak Ridge, TN. 232pp.

Reisenauer, A. E., K. T. Key, T. N. Narasimhan, and R. W. Nelson. 1982. TRUST: A computer program for variably saturated flow in multidimensional, deformable media. NUREG/CR-2360, PNL-3975. Pacific Northwest Laboratory, Richland, Wash.

Richards, L. A. 1931. Capillary conduction of liquids through porous mediums. Physics. 1:318-333.

Segol, G. 1976. A three-dimensional Galerkin finite-element model for the analysis of contaminant transport in variably saturated porous media. Department of Earth Sciences, University of Waterloo, Canada.

van Genuchten, M. Th., G. F. Pinder, and W. F. Saukin. 1977. Modeling of leachate and soil interaction in an aquifer. pp. 95-103. IN S. K. Banerji (ed.), Management of Gas and Leachate in Landfills. EPA-600/9-77-026. Office of Research and Development, U.S. Environmental Protection Agency, Cincinnati, Ohio.

van Genuchten, M. Th., and G. F. Pinder. 1978. Mass transport in saturated-unsaturated Media. II. Two-dimensional solutions. Research Report 78-WR-12, Water Resources Program, Department of Civil Engineering, Princeton University, Princeton, N. J.

van Genuchten, M. Th. 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Sci. Soc. Am. J. 44:892-898

Verruijt, A. 1969. Elastic storage of aquifers. Chap. 8. IN R. Dewiest (ed.), Flow through Porous Media. McGraw-Hill Book, New York.

Wang, J. D., and J. J. Connor, 1975. Mathematical modeling of near-coastal circulation. Report No. MITSG 75-13. MIT, Cambridge, MA.

Yeh, G. T., and D. S. Ward. 1980. FEMWATER: A finite-element model of water flow through saturated-unsaturated porous media. ORNL-5567. Oak Ridge National Laboratory, Oak Ridge, TN. 153 pp.

Yeh, G. T., and R. J. Luxmoore. 1983. Modeling moisture and thermal transport in unsaturated porous media. J. Hydrol. 64:299-309.

Yeh, G. T., J. C. Hwang, and W. C. Cho. 1985. Subregion iteration of finite element method for large-field problems of groundwater solute transport. J. Hydraulics Div. ASCE(HY3):547-551.

Yeh, G. T. 1985. Comparison of successive iteration and direct methods to solve finite element equations of aquifer contaminant transport. Water Resour. Res. 21(3):272-280.

Yeh, G. T. 1986. An orthogonal upstream finite element approach to modelling aquifer contaminant transport. Water Resour. Res. 22(6): 952-964.

Yeh, G. T. 1987. Simulation of water balance in waste burial trenches using a finite element moisture movement model. ORNL/RAP/LTR-86/72. Oak Ridge National Laboratory, Oak Ridge, TN.

APPENDIX A

Listing of FORTRAN Source Program of 3DFEMWATER Code BLI (Block Iteration Version)

```
C *                                                                  * COM  005
C *     ***********************************************************  * COM  010
C *     *                                                         *  * COM  015
C *     *                     3DFEMWATER                          *  * COM  020
C *     *            WITH BLOCK ITERATION METHOD                  *  * COM  025
C *     *                                                         *  * COM  030
C *     ***********************************************************  * COM  035
C *                                                                  * COM  040
C *                                                                  * COM  045
C * THIS COMPUTER CODE IS CONTAINED IN THE FOLLOWING REPORT:          * COM  050
C * YEH, G. T., 1987. "3DFEMWATER: A THREE-DIMENSIONAL FINITE ELEMENT * COM  055
C * MODEL OF WATER FLOW THROUGH SATURATED-UNSATURATED MEDIA"          * COM  060
C * ORNL-6386, OAK RIDGE NATIONAL LABORATORY, OAK RIDE, TN 37831      * COM  065
C *                                                                  * COM  070
C *                                                                  * COM  075
C * FOR ANY QUESTION, PLEASE CONTACT G. T. YEH AT (615) 574-7285      * COM  080
C *                                                                  * COM  085
C *                                                                  * COM  090
C ******************************                                      * COM  095
C *                           *                                      * COM  100
C *  DEFINITION OF VARIABLES  *                                      * COM  105
C *                           *                                      * COM  110
C ******************************                                      * COM  115
C *                                                                  * COM  120
C * MAXEL  - MAXIMUM NO. OF ELEMENTS                                  * COM  125
C * MAXNP  - MAXIMUM NO. OF NODES                                     * COM  130
C * MAXBES - MAXIMUM NO. OF BOUNDARY ELEMENT-SIDES                    * COM  135
C * MAXBNP - MAXIMUM NO. OF BOUNDARY NODAL POINTS                     * COM  140
C * JBAND  - MAXIMUM NO. OF NON-ZERO ELEMENTS IN ANY ROW OF THE MATRIX* COM  145
C * MAXNTI - MAXIMUM NO. OF TIME STEPS                                * COM  150
C * MXNDTC - MAXIMUM NO. OF DELT CHANGES                              * COM  155
C *                                                                  * COM  160
C *                                                                  * COM  165
C * NNP    - NO. OF NODAL POINTS                                      * COM  170
C * NEL    - NO. OF ELEMENTS                                          * COM  175
C * NBNP   - NO. OF BOUNDARY NODE POINTS                              * COM  180
C * NBES   - NO. OF BOUNDARY ELEMENT-SIDES                            * COM  185
C * KGRAV  - GRAVITY TERM CONTROL, 0 - NOT INCLUDED, 1 - INCLUDED     * COM  190
C * NTI    - NO. OF TIME INCREMENTS                                   * COM  195
C * NDTCHG - NO. OF DELT CHANGES                                      * COM  200
C *                                                                  * COM  205
C * NCYL   - NO. OF CYCLES FOR ITERATING RAINFALL-SEEPAGE             * COM  210
C * NITER  - NO. OF ITERATIONS ALLOWED FOR SOLVING NONLINEAR EQATIONS * COM  215
C * KSTR   - STORE CONTROL INTEGER                                    * COM  220
C * KPRO   - LINE-PRINTING CONTROL FOR INITIAL TIME                   * COM  225
C * KDSKO  - DISK STORE CONTROL INTEGER                               * COM  230
```

```
C * KSS    - STEADY STATE SOLUTION CONTROL                          * COM  235
C *          0 - STEADY STATE SOLUTION IS WANTED EITHER AS FINAL    * COM  240
C *              SOLUTION OR AS INITIAL CONDIITION OF TRANSIENT STATE * COM  245
C *              SOLUTION                                           * COM  250
C *          1 - ONLY TRANSIENT SOLUTION IS NEEDED                  * COM  255
C * KCP    - SOIL PERMEABILITY INPUT CONTROL                        * COM  260
C *          0 - INPUT SATURATED HYDRAULIC CONDUCTIVITY TO ARRAY PROP * COM  265
C *          1 - INPUT SATURATED PERMEABILITY TO ARRAY PROP         * COM  270
C *                                                                 * COM  275
C * DELT   - TIME STEP SIZE                                         * COM  280
C * CHNG   - PERCENTAGE CHANGE OF TIME STEP SIZE                    * COM  285
C * DELMAX - MAXIMUM TIME STEP SIZE ALLOWED                         * COM  290
C * TMAX   - MAXIMUM SIMULATION TIME ALLOWED                        * COM  295
C * DELTO  - INITIAL TIME STEP SIZE                                 * COM  300
C * TOLA   - TOTLERANCE FOR ITERATING STEADY STATE SOLUTION         * COM  305
C * TOLB   - TOLERANCE FOR ITERATING TRANSIENT SOLUTION OF PRESSURE * COM  310
C * W      - TIME INTEGRATION PARAMETERS                            * COM  315
C *          0.5 - CRANK-NICOLSON CENTRAL DIFFERENCE                * COM  320
C *          1.0 - BACKWARD DIFFERENCE (COMPLETE IMPLICIT SCHEME)   * COM  325
C * OME    - ITERATION PARAMETER FOR SOLVING NONLINEAR EQUATION     * COM  330
C * OMI    - ITERATION PARAMETER FOR SOLVING THE LINEARIZED MATRIX  * COM  335
C *          0.0 TO 1.0 - UNDER RELAXATION                          * COM  340
C *          1.0 TO 2.0 - OVER-RELAXATION                           * COM  345
C * TIME   - TIME SINCE THE BEGINNING                               * COM  350
C *                                                                 * COM  355
C *                                                                 * COM  360
C * MXSEL  - MAXIMUM NO. OF SOURCE ELEMENTS                         * COM  365
C * NSEL   - NO. OF SOURCE ELEMENTS                                 * COM  370
C * MXSPR  - MAXIMUM NO. OF SOURCE PROFILES                         * COM  375
C * MXSDP  - MAXIMUM NO. OF DATA POINTS ON EACH SOURCE PROFILE      * COM  380
C * NSPR   - NO. OF SOURCE PROFILE                                  * COM  385
C * NSDP   - NO. OF DATA POINTS ON EACH OF THE SOURCE PROFILES      * COM  390
C * MXWNP  - MAXIMUM NO. OF WELL NODAL POINTS                       * COM  395
C * NWNP   - NO. OF WELL NODAL POINTS                               * COM  400
C * MXWPR  - MAXIMUM NO. OF WELL SOURCE PROFILES                    * COM  405
C * MXWDP  - MAXIMUM NO. OF DATA POINT IN EACH WELL SOURCE PROFILE  * COM  410
C * NWPR   - NO. OF WELL SOURCE/SINK PROFILES                       * COM  415
C * NWDP   - NO. OF DATA POINTS IN EACH OF ALL SOURCE/SINK PROFILES * COM  420
C *                                                                 * COM  425
C * MXCNP  - MAXIMUM NO. OF CAUCHY NODAL POINTS                     * COM  430
C * MXCES  - MAXIMUM NO. OF CAUCHY ELEMENT SIDES                    * COM  435
C * MXCPR  - MAXIMUM NO. OF CAUCHY FLUX PROFILES                    * COM  440
C * MXCDP  - MAXIMUM NO. OF CAUCHY DATA POINTS ON EACH PROFILE      * COM  445
C * NCNP   - NO. OF CAUCHY NODAL POINTS                             * COM  450
C * NCES   - NO. OF CAUCHY ELEMENT SIDES                            * COM  455
C * NCPR   - NO. OF CAUCY FLUX PROFILES                             * COM  460
C * NCDP   - NO. OF CAUCY FLUX DATA POINTS ON EACH PROFILE          * COM  465
C *                                                                 * COM  470
```

```
C * MXNNP  - MAXIMUM NO. OF NEUMANN NODAL POINTS                     * COM  475
C * MXNES  - MAXIMUM NO. OF NEUMANN ELEMENT SIDES                    * COM  480
C * MXNPR  - MAXIMUM NO. OF NEUMANN-FLUX PROFILES                    * COM  485
C * MXNDP  - MAXIMUM NO. OF NEUMANN-FLUX DATA POINTS ON EACH PROFILE * COM  490
C * NNNP   - NO. OF NEUMANN NODAL POINTS                             * COM  495
C * NNES   - NO. OF NEUMANN ELEMENT SIDES                            * COM  500
C * NNPR   - NO. OF NEUMANN-FLUX PROFILES                            * COM  505
C * NNDP   - NO. OF NEUMANN-FLUX DATA POINTS ON EACH PROFILE         * COM  510
C *                                                                  * COM  515
C *                                                                  * COM  520
C * MXVES  - MAXIMUM NO. OF VARIABLE ELEMENT SIDES                   * COM  525
C * MXVNP  - MAXIMUM NO. OF VARIABLE NODAL POINTS                    * COM  530
C * MXRPR  - MAXIMUM NO. OF RAINFALL PROFILES                        * COM  535
C * MXRDP  - MAXIMUM NO. OF DATA POINT ON EACH RAINFALL PROFILE      * COM  540
C * NVES   - NO. OF VARIABLE ELEMENT SIDES                           * COM  545
C * NVNP   - NO. OF VARIABLE NODAL POINTS                            * COM  550
C * NRPR   - NO. OF RAINFALL PROFILES                                * COM  555
C * NRDP   - NO. OF DATA POINTS ON EACH RAINFALL PROFILE             * COM  560
C *                                                                  * COM  565
C * MXDNP  - MAXIMUM NO. OF DIRICHLET NODAL POINTS                   * COM  570
C * MXDPR  - MAXIMUM NO. OF DIRICHLET TOTAL HEAD PROFILES            * COM  575
C * MXDDP  - MAXIMUM NO. OF DATA POINTS ON EACH DIRICHLET PROFILE    * COM  580
C * NDNP   - NO. OF DIRICHLET NODAL POINTS                           * COM  585
C * NDPR   - NO. OF DIRICHLET TOTAL HEAD PROFILES                    * COM  590
C * NDDP   - NO. OF DIRICHLET DATA POINTS ON EACH DIRICHLET PROFILE  * COM  595
C *                                                                  * COM  600
C * MAXMAT - MAXIMUM NO. OF MATERIAL TYPES (MEDIUM FORMATIONS)       * COM  605
C * MXSPPM - MAXIMUM NO. OF SOIL PARAMETERS PER MATERIAL TO DESCRIBE * COM  610
C *          SOIL CHARACTERISTIC CURVES.                             * COM  615
C * MXMPPM - MAXIMUM NO. OF MATERIAL PROPERTIES PER MATERIAL         * COM  620
C *                                                                  * COM  625
C * NMAT   - NO. OF MATERIAL TYPES THAT FORM THE REGION OF INTEREST  * COM  630
C * NSPPM  - NO. OF SOIL PARAMETERS PER MATERIAL                     * COM  635
C * FRATE(I) - FLOW RATE THROUGH I-TH TYPE BOUNDARY SEGMENT          * COM  640
C *            I - 1 - DIRICHLET BOUNDARY                            * COM  645
C *            I - 2 - NEUMANN BOUNDARY                              * COM  650
C *            I - 3 - CAUCHY BOUNDARY                               * COM  655
C *            I - 4 - SEEPAGE BOUNDARY                              * COM  660
C *            I - 5 - INFILTRATION BOUNDARY                         * COM  665
C *            I - 6 - NUMERICAL LOSSES                              * COM  670
C *            I - 7 - ENTIRE BOUNDARY                               * COM  675
C *            I - 8 - ARTIFICIAL SOURCE INTO THE REGION             * COM  680
C *            I - 9 - INCREASE IN THE WHOLE REGION OF INTEREST      * COM  685
C * FLOW(I) - AMOUNT OF FLOW THROUGH I-TH TYPE BOUNDARY IN THE STEP  * COM  690
C * TFLOW(I) - ACCUMULATED AMOUNT OF FLOW THROUGH I-TH TYPE BOUNDARY * COM  695
C * NOTE: POSITIVE VALUES MEAN OUT FROM THE REGION, NEGATIVE VALUES  * COM  700
C * MEAN INTO THE REGION.                                            * COM  705
C *                                                                  * COM  710
```

```
C * X(N) - X-COORDINATE OF N-TH NODAL POINT, (L): X(MAXNP)           * COM  715
C * Y(N) - Y-COORDINATE OF N-TH NODAL POINT, (L): Y(MAXNP)           * COM  720
C * Z(N) - Z-COORDINATE OF N-TH NODAL POINT, (L): X(MAXNP)           * COM  725
C * IE(M,I) - GLOBAL NODE NUMBER OF THE I-TH NODE OF M-TH ELEMENT IF * COM  730
C *            I IS BETWEEN 1 AND 8                                  * COM  735
C * IE(M,9) - INTEGER TO INDICATE THE MATERIAL TYPE OF M-TH ELEMENT  * COM  740
C * NOTE: IE IS DIMENSIONED IE(MAXEL,9)                              * COM  745
C *                                                                  * COM  750
C * CMATRIX(N,I) - AN ARRAY TO STORE THE ASSEMBLED GLOBLE MATRIX     * COM  755
C *                SHOULD BE DIMENSIONED AS C(MAXNP,JBAND)           * COM  760
C * GNOJCN(I,N) - GLOBAL NODE NUMBER OF THE I-TH CONNECTING NODE     * COM  765
C *               SURROUNDING THE N-TH GLOBAL NODE.  THIS ARRAY IS   * COM  770
C *               GENERATED BASED ON THE INPUT ARRAY IE.             * COM  775
C * RLD(N) - AN ARRAY TO STORE THE ASSEMBLED LOAD VECTOR             * COM  780
C *          SHOULD BE DIMENSIONED AS RLD(MAXNP)                     * COM  785
C * RI(N) - PRESSURE HEAD ITERATE IN SUBROUTINE BLKITR               * COM  790
C * RL(N) - A WORKING ARRAY TO CONTAIN FINAL SOLUTION FOR THE PRES-  * COM  795
C *         SURE HEAD IN SUBROUTINE BLKITR                           * COM  800
C * NTNPLR(K) - TOTAL NUMBER OF NODES FOR K-TH SUBREGION INCLUDING   * COM  805
C *             INTERIOR & GLOBAL BOUNDARY AND INTRA-BOUNDARY NODES  * COM  810
C *             THIS ARRAY IS GENERATED BASED ON THE INPUT ARRAYS IE * COM  815
C *             AND GNLR(I,K) FOR I=1,2,...,NNPLR(K).                * COM  820
C * NNPLR(K) - NUMBER OF INTERIOR NODES FOR K-TH SUBREGION, INCLUDING * COM  825
C *            GLOBAL BOUNDARY NODES.  INPUT.                        * COM  830
C * GNLR(I,K) - GLOBAL NODAL NUMBER OF I-TH LOCAL NODAL NUMBER FOR   * COM  835
C *             K-TH SUBREGION.  THIS ARRAY IS AN INPUT FOR I=1,2,...,* COM  840
C *             NNPLR(K).  BUT IT IS GENERATED BASED ON IE(NEL,8) AND * COM  845
C *             GNLR(I,K) WITH I=1,2,...,NNPLR(K) FOR I=NNPLR(K)+1 TO * COM  850
C *             NTNPLR(K).                                           * COM  855
C * LNOJCN(J,I,K) - LOCAL NODE NUMBER OF J-TH CONNECTING NODE SUR-   * COM  860
C *                 ROUNDING I-TH LOCAL NODE FOR K-TH SUBREGION.     * COM  865
C *                 THIS ARRAY IS GENERATED BASED ON IE AND GNLR.    * COM  870
C * LMAXDF(K) - MAXIMUM DIFFERENCE BETWEEN ANY TWO NODES OF AN       * COM  875
C *             ELEMENT IN K-TH SUBREGION.  THIS ARRAY IS GENERATED  * COM  880
C *             FROM LNOJCN                                          * COM  885
C * CMTRXL(N,I) - ASSEMBLED MATRIX FOR A SUBREGION                   * COM  890
C * RLDL(N) - ASSEMBLED LOAD VECTOR FOR A SUBREGION                  * COM  895
C *                                                                  * COM  900
C * H(N) - PRESSURE HEAD AT N-TH NODE, (L): H(MAXNP)                 * COM  905
C * HP(N) - PRESSURE HEAD OF N-TH NODE AT PREVIOUS TIME              * COM  910
C * HW(N) - NONLINEAR PRESSURE HEAD ITERATE OF N-TH NODE             * COM  915
C * HT(N) - TOTAL HEAD OF N-TH NODE                                  * COM  920
C * VX(N) - X-COMPONENT VELOCITY AT N-TH NODE                        * COM  925
C * VY(N) - Y-COMPONENT VELOCITY AT N-TH NODE                        * COM  930
C * VZ(N) - Z-COMPONENT VELOCITY AT N-TH NODE                        * COM  935
C * NPCNV(N) - NODAL POINT OF N-TH NONCONVERGENT NODE                * COM  940
C * TH(I,M) - MOISTURE CONTENT AT I-TH GAUSSIAN POINT OF M-TH        * COM  945
C *           ELEMENT                                                * COM  950
```

```
C * DTH(I,M) - DTH/DH - WATER CAPACITY OF I-TH GAUSSIAN POINT OF M-   * COM  955
C *            TH ELEMENT                                             * COM  960
C * AKR(I,M) - RELATIVE HYDRAULIC CONDUCTIVITY AT I-TH GAUSSIAN       * COM  965
C *            POINT OF M-TH ELEMENT                                  * COM  970
C * DCOSB(1,I) - X-DIRECTIONAL COSINE OF I-TH BOUNDARY ELEMENT SIDE   * COM  975
C * DCOSB(2,I) - Y-DIRECTIONAL COSINE OF I-TH BOUNDARY ELEMENT SIDE   * COM  980
C * DCOSB(3,I) - Z-DIRECTIONAL COSINE OF I-TH BOUNDARY ELEMENT SIDE   * COM  985
C *  NOTE: DCOSB SHOULD BE DIMENSIONED AS DCOSB(3,MAXBES).            * COM  990
C * ISB(1,I) - BOUNDARY NODE NUMBER OF THE FIRST NODE OF I-TH         * COM  995
C *            BOUNDARY ELEMENT SIDE                                  * COM 1000
C * ISB(2,I) - BOUNDARY NODE NUMBER OF THE SECOND NODE OF I-TH        * COM 1005
C *            BOUNDARY ELEMENT SIDE                                  * COM 1010
C * ISB(3,I) - BOUNDARY NODE NUMBER OF THE THIRD NODE OF I-TH         * COM 1015
C *            BOUNDARY ELEMENT SIDE                                  * COM 1020
C * ISB(4,I) - BOUNDARY NODE NUMBER OF THE FOURTH NODE OF I-TH        * COM 1025
C *            BOUNDARY ELEMENT SIDE                                  * COM 1030
C * ISB(5,I) - ELEMENT SIDE INDEX OF I-TH BOUNDARY ELEMENT SIDE       * COM 1035
C *          -1 - LEFT SIDE, -2 - FRONT SIDE, -3 -RIGHT SIDE          * COM 1040
C *          -4 - BACK SIDE, -5 - BOTTOM SIDE, -6 - TOP SIDE          * COM 1045
C * ISB(6,I) - ELEMENT NUMBER TO WHICH THE I-TH BOUNDARY ELEMENT-SIDE * COM 1050
C *            BELONG                                                 * COM 1055
C * NPBB(I) - GLOBAL NODE NUMBER OF I-TH BOUNDARY NODE                * COM 1060
C * BFLX(I) - BOUNDARY FLUX AT I-TH BOUNDARY NODE, (L**3/T)           * COM 1065
C *           SHOULD BE DIMENSIONED AS BFLX(MAXBNP)                   * COM 1070
C * BFLXP(I) - BOUNDARY FLUX AT PREVIOUS TIME OF I-TH BOUNDARY NODE   * COM 1075
C *            SOULD BE DIMENSIONED AS BFLXP(MAXBNP)                  * COM 1080
C *                                                                   * COM 1085
C *                                                                   * COM 1090
C *                                                                   * COM 1095
C * SOS(I) - SOURCE STRENGTH OF I-TH SOURCE PROFILE,                  * COM 1100
C *          ((L**3/T)/L**3): SOS(MXSPR).                             * COM 1105
C * SOSF(I,J) - SOURCE STRENGTH OF I-TH DATA POINT IN J-TH ELEMENT    * COM 1110
C *             SOURCE VS TIME PROFILE, ((L**3/T)/L**)                * COM 1115
C * TSOSF(I,J) - TIME OF I-TH DATA POINT IN J-TH SOURCE VS TIME       * COM 1120
C *              PROFILE, (T)                                         * COM 1125
C * MSEL(I) - ELEMENT NUMBER OF I-TH SOURCE/SINK ELEMENT              * COM 1130
C * ISTYP(I) - SOURCE TYPE ASSIGNED TO I-TH SOURCE/SINK ELEMENT       * COM 1135
C * WSS(I) - VALUE OF I-TH WELL SOURCE/SINK AT PRESENT TIME, (L**3/T) * COM 1140
C * WSSF(I,J) - WELL SOURCE/SINK FUNCTION OF I-TH DATA POINT OF J-TH  * COM 1145
C *             PROFILE, (L**3/T)                                     * COM 1150
C * TWSSF(I,J) - TIME OF I-TH DATA POINT OF J-TH PROFILE              * COM 1155
C * NPW(I) - GLOBAL NODE NUMBER OF I-TH WELL SOURCE/SINK POINT        * COM 1160
C * IWTYP(I) - TYPE OF WELL SOURCE/SINK ASSGINED TO I-TH SOURCE/SINK  * COM 1165
C *            POINT                                                  * COM 1170
C *                                                                   * COM 1175
C *                                                                   * COM 1180
```

```
C * QCB(I) - VALUE OF CAUCHY FLUX AT THE PRESENT TIME OF I-TH CAUCHY  * COM 1185
C *          FLUX VS TIME PROFILE, (L**3/T/L**2)                      * COM 1190
C * QCBF(I,J) - FLUX OF I-TH DATA POINT IN J-TH CAUCHY FLUX VS TIME   * COM 1195
C *             TIEM PROFILE, (L**3/T/L**2)                           * COM 1200
C * TQCBF(I,J) - TIME OF I-TH DATA POINT IN J-TH CAUCHY FLUX VS TIME  * COM 1205
C *              PROFILE                                              * COM 1210
C * ICTYP(MP) - TYPE OF CAUCHY FLUX VS TIME PROFILE ASSIGNED TO MP-TH * COM 1215
C *             CAUCHY SIDE                                           * COM 1220
C * NPCB(NP) - GLOBAL NODE NUMBER OF NP-TH CAUCHY NODE FOR INPUTTING, * COM 1225
C *            THEN IS CHANGED TO CONTAIN BOUNDARY NODE NUMBER        * COM 1230
C * ISC(1,I) - GLOBAL NODE NUMBER OF THE FIRST NODE OF I-TH CAUCHY    * COM 1235
C *            SIDE                                                   * COM 1240
C * ISC(2,I) - GLOBAL NODE NUMBER OF THE SECOND NODE OF I-TH CAUCHY   * COM 1245
C *            SIDE.                                                  * COM 1250
C * ISC(3,I) - GLOBAL NODE NUMBER OF THE THIRD NODE OF I-TH CAUCHY    * COM 1255
C *            SIDE                                                   * COM 1260
C * ISC(4,I) - GLOBAL NODE NUMBER OF THE FOURTH NODE OF I-TH CAUCHY   * COM 1265
C *            SIDE                                                   * COM 1270
C * ISC(5,I) - BOUNDARY SIDE NUMBER OF I-TH CAUCHY SIDE               * COM 1275
C *                                                                   * COM 1280
C *                                                                   * COM 1285
C * QNB(I) - VALUE OF FLUX AT THE PRESENT TIME OF I-TH NEUMANN FLUX VS* COM 1290
C *          TIME PROFILE FOR NEUMANN BOUNDARY, ((L**3/T)/L**2)       * COM 1295
C *          QNB(MXNPR)                                               * COM 1300
C * QNBF(I,J) - FLUX OF I-TH DATA POINT IN J-TH NEUMANN FLUX VS TIME  * COM 1305
C *             PROFILE, ((L**3/T)/L**2)                              * COM 1310
C * TQNBF(I,J) - TIME OF I-TH DATA POINT IN J-TH NEUMANN FLUX VS TIME * COM 1315
C *              PROFILE FOR NEUMANN BOUNDARY, (T)                    * COM 1320
C * INTYP(MP) - TYPE OF NEUMANN FLUX VS TIME PROFILE TO BE ASSIGNED TO* COM 1325
C *             MP-TH NEUMANN SIDE: INTYP(MXNNP)                      * COM 1330
C * NPNB(NP) - GLOBAL NODE NUMBER OF NP-TH NEUMANN NODE FOR INPUTTING * COM 1335
C *            THEN IS CHANGED TO CONTAIN BOUNDARY NODE NUMBER        * COM 1340
C * ISN(1,I) - GLOBAL NODE NUMBER OF THE FIRST NODE OF I-TH NEUMANN   * COM 1345
C *            SIDE                                                   * COM 1350
C * ISN(2,I) - GLOBAL NODE NUMBER OF THE SECOND NODE OF I-TH NEUMANN  * COM 1355
C *            SIDE                                                   * COM 1360
C * ISN(3,I) - GLOBAL NODE NUMBER OF THE THIRD NODE OF I-TH NEUMANN   * COM 1365
C *            SIDE                                                   * COM 1370
C * ISN(4,I) - GLOBAL NODE NUMBER OF THE FOURTH NODE OF I-TH NEUMANN  * COM 1375
C *            SIDE                                                   * COM 1380
C * ISN(5,I) - BOUNDARY SIDE NUMBER OF I-TH NEUMANN SIDE              * COM 1385
C *                                                                   * COM 1390
C * RFALL(I) - VALUE OF RAINFALL AT THE PRESENT TIME OF I-TH RAIN-    * COM 1395
C *            FALL VIS TIME PROFILE, (L/T)                           * COM 1400
C * TRF(I,J) - TIME OF I-TH DATA POINT ON J-TH RAINFALL VS TIME       * COM 1405
C *            PROFILE                                                * COM 1410
C * RF(I,J) - RAINFALL VALUE OF I-TH DATA POINT ON J-TH RAINFALL VS   * COM 1415
C *           TIME PROFILE, (L/T)                                     * COM 1420
C * IRTYP(MP) - TYPE OF RAINFALL PROFILE ASSEGNED TO MP-TH VARIABLE   * COM 1425
C *             BOUNDARY SIDE                                         * COM 1430
```

```
C * ISV(1,I) - GLOBAL NODE NUMBER OF THE FIRST NODE OF I-TH VB SIDE   * COM 1435
C *            FOR INPUTTING, THEN IS CHANGED TO CONTAIN COMPRESSED   * COM 1440
C *            VB NODE NUMBER.                                        * COM 1445
C * ISV(2,I) - GLOBAL NODE NUMBER OF THE SECOND NODE OF I-TH VB SIDE  * COM 1450
C *            FOR INPUTITING, THEN IS CHANGED TO CONTAIN COMPRESSED  * COM 1455
C *            VB NODE NUMBER.                                        * COM 1460
C * ISV(3,I) - GLOBAL NODE NUMBER OF THE THIRD NODE OF I-TH VB SIDE   * COM 1465
C *            FOR INPUTTING, THEN IS CHANGED TO CONTAIN COMPRESSED   * COM 1470
C *            VB NODE NUMBER.                                        * COM 1475
C * ISV(4,I) - GLOBAL NODE NUMBER OF THE FOURTH NODE OF I-TH VB SIDE  * COM 1480
C *            FOR INPUTTING, THEN IS CHANGED TO CONTAIN COMPRESSED   * COM 1485
C *            VB NODE NUMBER.                                        * COM 1490
C * ISV(5,I) - BOUNDARY SIDE NUMBER OF I-TH RS SIDE, BETWEEN 1 & NBES * COM 1495
C * NOTE: "ISV" SHOULD BE DIMENSIONED AS ISV(5,MXVES)                 * COM 1500
C *                                                                   * COM 1505
C * NPVB(NP) - GLOBAL NODE NUMBER OF NP-TH VB NODE FOR INPUTTING,     * COM 1510
C *            THEN IS CHANGED TO CONTAIN BOUNDARY NODE NUMBER.       * COM 1515
C *                                                                   * COM 1520
C * DCYFLX(NP) - DARCY FLUX THROUGH NP-TH RAINFALL-SEEPAGE NODE       * COM 1525
C *              (L**3/T)                                             * COM 1530
C * FLX(NP) - RAINFALL RATE THROUGH NP-TH RAINFALL-SEEPAGE NODE       * COM 1535
C *           (L**3/T)                                                * COM 1540
C * HCON(NP) - PONDING DEPTH ALLOWED FOR THE NP-TH NODE OF RAINFALL/  * COM 1545
C *            EVAPORATION SEEPAGE NODES, (L): HCON(MXRSNP)           * COM 1550
C * HMIN(NP) - MINIMUM PRESSURE HEAD ALLOWED FOR THE NP-TH RAINFALL/  * COM 1555
C *            EVAPORATION SEEPAGE NODES, (L): HMIN(MXVNP)            * COM 1560
C * NPFLX(NP) - CAUCHY BOUNDARY CONDITION INDICATOR FOR THE NP-TH     * COM 1565
C *             NODE OF RAINFALL(+)/EVAPORATION(-)-SEEPAGE NODE, 0 -  * COM 1570
C *             THIS NP-TH RS NODE IS NOT A CAUCHY NODE FOR THE       * COM 1575
C *             TIME STEP, GLOBAL NODE NUMBER OF NP-TH RS NODE - THIS * COM 1580
C *             NP-TH RS NODE IS A CAUCHY NODE                        * COM 1585
C *                                                                   * COM 1590
C * NPCON(NP) - PONDING DEPTH DIRICHLET BOUNDARY CONDITION INDICATOR  * COM 1595
C *             FOR THE NP-TH RS NODE, 0 - THIS NP-TH NODE IS NOT A   * COM 1600
C *             PONDING DEPTH DIRICHLET NODE, GLOBAL NODE NUMBER OF   * COM 1605
C *             NP-TH RS NODE - THIS NP-TH RS NODE IS A PONDING DEPTH * COM 1610
C *             DIRICHLET NODE                                        * COM 1615
C *                                                                   * COM 1620
C * NPMIN(NP) - MINIMUM PRESSURE HEAD DIRICHLET BOUNDARY CONDITION    * COM 1625
C *             INDICATOR FOR THE NP-TH RS NODE, 0 - NP-TH RS NODE IS * COM 1630
C *             NOT A MINIMUM PRESSURE HEAD DIRICHLET NODE, GLOBAL    * COM 1635
C *             NODE NUMBER OF NP-TH RS NODE - THIS NP-TH RS NODE IS  * COM 1640
C *             PRESCRIBED MINIMUM PRESSURE HEAD DIRICHLET NODE       * COM 1645
C *                                                                   * COM 1650
C *                                                                   * COM 1655
```

```
C * HDB(I) - VALUE OF TOTAL HEAD AT THE PRESENT TIME OF I-TH TOTAL    * COM 1660
C *          HEAD VS TIME PROFILE FOR DIRICHELT BOUNDARY,             * COM 1665
C *          SHOULD BE DIMENSIONED AS HDB(MXDPR)                      * COM 1670
C * HDBF(I,J) - TOTAL HEAD OF I-TH DATA POINT IN J-TH DIRICHLET TOTAL * COM 1675
C *             HEAD VS TIME PROFILE, (L): HDBF(MXDDP,MXDPR)          * COM 1680
C * THDBF(I,J) - TIME OF I-TH DATA POINT IN J-TH DIRICHLET TOTAL HEAD * COM 1685
C *              VS TIME PROFILE, (T): THDBF(MXDDP,MXDPR)             * COM 1690
C * IDTYP(NP) - TYPE OF DIRICHLET TOTAL HEAD VS TIME PROFILE TO BE    * COM 1695
C *             ASSIGNED TO NP-TH DIRICHLET NODE: IDTYP(MXDNP)        * COM 1700
C * NPDB(NP) - GLOBAL NODE NUMBER OF NP-TH DIRICHLET NODE FOR         * COM 1705
C *            INPUTTING, THEN IS CHANGED TO CONTAIN BOUNDARY NODE    * COM 1710
C *            NUMBER                                                 * COM 1715
C *                                                                   * COM 1720
C *                                                                   * COM 1725
C * PROP(J,I) - J-TH MATERIAL PROPERTY OF I-TH MATERIALS              * COM 1730
C *             J - 1 - SATURATED X-HYDRAULIC CONDUCTIVITY, (L/T),    * COM 1735
C *                     OR PERMEABILITY, (L**2).                      * COM 1740
C *             J - 2 - SATURATED Y-HYDRAULIC CONDUCTIVITY, (L/T),    * COM 1745
C *                     OR PERMEABILITY, (L**2).                      * COM 1750
C *             J - 3 - SATURATED Z-HYDRAULIC CONDUCTIVITY, (L/T),    * COM 1755
C *                     OR PERMEABILITY, (L**2).                      * COM 1760
C *             J - 4 - SATURATED XY-HYDRAULIC CONDUCTIVITY, (L/T),   * COM 1765
C *                     OR PERMEABILITY, (L**2).                      * COM 1770
C *             J - 5 - SATURATED XZ-HYDRAULIC CONDUCTIVITY, (L/T),   * COM 1775
C *                     OR PERMEABILITY, (L**2).                      * COM 1780
C *             J - 6 - SATURATED YZ-HYDRAULIC CONDUCTIVITY, (L/T),   * COM 1785
C *                     OR PERMEABILTY, (L**2).                       * COM 1790
C ******* NOTE: "PROP" SHOULD BE DIMENSIONED AS PROP(MXMPPM,MAXMAT)   * COM 1795
C *                                                                   * COM 1800
C *                                                                   * COM 1805
C * THPROP(J,I) - J-TH PARAMETER TO DESCRIBE THE MOISTURE CONTENT AS  * COM 1810
C *               FUNCTION OF PRESSURE HEAD FOR I-TH MATERIAL.        * COM 1815
C * AKPROP(J,I) - J-TH PARAMETER TO DESCRIBE THE RELATIVE HYDRAULIC   * COM 1820
C *               CONDUCTIVITY FOR I-TH MATERIAL.                     * COM 1825
C ******* NOTE:  "THPROP AND AKPROP" SHOULD BE DIMENSIONED            * COM 1830
C *******         AS (MXSPPM,MAXMAT)                                  * COM 1835
C *                                                                   * COM 1840
C *                                                                   * COM 1845
C * KPR(I) - LINE PRINT CONTROL FOR I-TH TIME STEP, SIMILAR TO KPRO   * COM 1850
C * KDSK(I) - DISK STORAGE CONTROL FOR I-TH TIME STEP SIMILAR TO KDSKO* COM 1855
C ******* NOTE: BOTH KPR AND KDSK" SHOULD BE DIMENSIONED AS (MAXNTI)  * COM 1860
C * TDTCH(I) - TIME OF I-TH TIME TO RESET TIME STEP SIZE - DELTO, (T) * COM 1865
C *            SHOULD BE DIMENSIONED AS TDTCH(MXNDTC)                 * COM 1870
C *                                                                   * COM 1875
C *                                                                   * COM 1880
```

```
C                                                                       MAIN 005
C ------- MAIN PROGRAM OF 3DFEMWATER - BLOCK ITERATION METHOD.          MAIN 010
C                                                                       MAIN 015
      IMPLICIT REAL*8(A-H,O-Z)                                          MAIN 020
      INTEGER*4 GNLR,GNOJCN                                             MAIN 025
C                                                                       MAIN 030
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC      MAIN 035
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI,NDTCHG                 MAIN 040
      COMMON /LGEOM/ LTMXNP,LMXNP,LMXBW,MXREGN,NREGN                    MAIN 045
      COMMON /CINTE/ NCYL,NITER,KSTR,KPRO,KDSKO,KSS,NPITER,IGEOM        MAIN 050
      COMMON /CREAL/ DELT,CHNG,DELMAX,TMAX,DELT0,TOLA,TOLB,W,OME,OMI    MAIN 055
C                                                                       MAIN 060
      COMMON /CS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP                      MAIN 065
      COMMON /CW/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP                      MAIN 070
C                                                                       MAIN 075
      COMMON /CCBC/ MXCNP,MXCES,MXCPR,MXCDP,NCNP,NCES,NCPR,NCDP         MAIN 080
      COMMON /CNBC/ MXNNP,MXNES,MXNPR,MXNDP,NNNP,NNES,NNPR,NNDP         MAIN 085
      COMMON /CVBC/ MXVES,MXVNP,MXRPR,MXRDP,NVES,NVNP,NRPR,NRDP         MAIN 090
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP                    MAIN 095
C                                                                       MAIN 100
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM                                MAIN 105
      COMMON /CMTL/ NMAT,NMPPM,NSPPM                                    MAIN 110
C                                                                       MAIN 115
      COMMON /CFLOW/ FRATE(10),FLOW(10),TFLOW(10)                       MAIN 120
C                                                                       MAIN 125
      DIMENSION X(6000),Y(6000),Z(6000),IE(4959,9)                      MAIN 130
      DIMENSION CMATRX(6000,27),GNOJCN(27,6000),RLD(6000)               MAIN 135
      DIMENSION RI(6000),RL(6000)                                       MAIN 140
      DIMENSION NTNPLR(20),NNPLR(20),LMAXDF(20)                         MAIN 145
      DIMENSION GNLR(900,20),LNOJCN(27,300,20)                          MAIN 150
      DIMENSION CMTRXL(300,23),RLDL(300)                                MAIN 155
      DIMENSION H(6000),HP(6000),HW(6000),HT(6000)                      MAIN 160
      DIMENSION VX(6000),VY(6000),VZ(6000)                              MAIN 165
      DIMENSION TH(8,4959),DTH(8,4959),AKR(8,4959), NPCNV(6000)         MAIN 170
C                                                                       MAIN 175
      DIMENSION DCOSB(3,1966),ISB(6,1966),NPBB(1968)                    MAIN 180
      DIMENSION BFLX(1968),BFLXP(1968)                                  MAIN 185
C                                                                       MAIN 190
      DIMENSION SOS(3),SOSF(4,3),TSOSF(4,3)                             MAIN 195
      DIMENSION ISTYP(11),MSEL(11)                                      MAIN 200
      DIMENSION WSS(5),WSSF(8,5),TWSSF(8,5)                             MAIN 205
      DIMENSION IWTYP(10),NPW(10)                                       MAIN 210
C                                                                       MAIN 215
      DIMENSION QCB(2),QCBF(4,2),TQCBF(4,2)                             MAIN 220
      DIMENSION ICTYP(171,ISC(5,171), NPCB(200)                         MAIN 225
C                                                                       MAIN 230
      DIMENSION QNB(1),QNBF(2,1),TQNBF(2,1)                             MAIN 235
      DIMENSION INTYP(551),ISN(5,551), NPNB(600)                        MAIN 240
C                                                                       MAIN 245
```

```
      DIMENSION RFALL(3),RF(24,3),TRF(24,3)
      DIMENSION IRTYP(551),ISV(5,551), NPVB(600)
      DIMENSION DCYFLX(600),FLX(600),HCON(600),HMIN(600)
      DIMENSION NPFLX(600),NPCON(600),NPMIN(600)
C
      DIMENSION HDB(20),HDBF(8,20),THDBF(8,20)
      DIMENSION IDTYP(200),NPDB(200)
C
      DIMENSION PROP(6,3)
      DIMENSION THPROP(4,3),AKPROP(4,3)
C
      DIMENSION KPR(500),KDSK(500),TDTCH(20)
C
C ------- PASS THE PROGRAM TO GW3D
C
      CALL GW3D(X,Y,Z,IE, CMATRX,GNOJCN,RLD,RI,RL, H,HP,HW,HT,
     1 VX,VY,VZ,  TH,DTH,AKR, NPCNV, DCOSB,ISB,NPBB, BFLX,BFLXP,
     2 SOS,SOSF,TSOSF,ISTYP,MSEL, WSS,WSSF,TWSSF,IWTYP,NPW,
     3 QCB,QCBF,TQCBF,ICTYP,ISC,NPCB, QNB,QNBF,TQNBF,INTYP,ISN,NPNB,
     4 RFALL,RF,TRF,IRTYP,ISV,NPVB, DCYFLX,FLX,HCON,HMIN,
     5 NPFLX,NPCON,NPMIN, HDB,HDBF,THDBF,IDTYP,NPDB,
     6 PROP,THPROP,AKPROP, KPR,KDSK,TDTCH,
     7 NTNPLR,NNPLR,LMAXDF,GNLR,LNOJCN,CMTRXL,RLDL)
C
      STOP
      END
```

```
      BLOCK DATA
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC
      COMMON /LGEOM/ LTMXNP,LMXNP,LMXBW,MXREGN,NREGN
      COMMON /CS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP
      COMMON /CW/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP
      COMMON /CCBC/ MXCNP,MXCES,MXCPR,MXCDP,NCNP,NCES,NCPR,NCDP
      COMMON /CNBC/ MXNNP,MXNES,MXNPR,MXNDP,NNNP,NNES,NNPR,NNDP
      COMMON /CVBC/ MXVES,MXVNP,MXRPR,MXRDP,NVES,NVNP,NRPR,NRDP
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM
C
      DATA MAXEL,MAXNP,MAXBES,MAXBNP,JBAND/4959,6000,1966,1968,27/
      DATA MAXNTI,MXNDTC/500,20/
      DATA LTMXNP,LMXNP,LMXBW,MXREGN/900,300,23,20/
```

```
      DATA MXSEL,MXSPR,MXSDP,MXWNP,MXWPR,MXWDP/11,3,4,10,5,8/           BLOC 090
      DATA MXCNP,MXCES,MXCPR,MXCDP/200,171,2,4/                         BLOC 095
      DATA MXNNP,MXNES,MXNPR,MXNDP/600,551,1,2/                         BLOC 100
      DATA MXVES,MXVNP,MXRPR,MXRDP/551,600,3,24/                        BLOC 105
      DATA MXDNP,MXDPR,MXDDP/200,20,8/                                  BLOC 110
      DATA MAXMAT,MXSPPM,MXMPPM/3,4,6/                                  BLOC 115
C                                                                       BLOC 120
      END                                                               BLOC 125


      SUBROUTINE GW3D(X,Y,Z,IE, CMATRX,GNOJCN,RLD,RI,RL, H,HP,HW,HT,    GW3D 005
     1 VX,VY,VZ,  TH,DTH,AKR, NPCNV, DCOSB,ISB,NPBB, BFLX,BFLXP,        GW3D 010
     2 SOS,SOSF,TSOSF,ISTYP,MSEL, WSS,WSSF,TWSSF,IWTYP,NPW,             GW3D 015
     3 QCB,QCBF,TQCBF,ICTYP,ISC,NPCB, QNB,QNBF,TQNBF,INTYP,ISN,NPNB,    GW3D 020
     4 RFALL,RF,TRF,IRTYP,ISV,NPVB, DCYFLX,FLX,HCON,HMIN,               GW3D 025
     5 NPFLX,NPCON,NPMIN, HDB,HDBF,THDBF,IDTYP,NPDB,                    GW3D 030
     6 PROP,THPROP,AKPROP, KPR,KDSK,TDTCH,                              GW3D 035
     7 NTNPLR,NNPLR,LMAXDF,GNLR,LNOJCN,CMTRXL,RLDL)                     GW3D 040
C                                                                       GW3D 045
      IMPLICIT REAL*8(A-H,O-Z)                                          GW3D 050
      REAL*4 SUBHD                                                      GW3D 055
      INTEGER*4 GNLR,GNOJCN                                             GW3D 060
C                                                                       GW3D 065
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC      GW3D 070
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI,NDTCHG                 GW3D 075
      COMMON /LGEOM/ LTMXNP,LMXNP,LMXBW,MXREGN,NREGN                    GW3D 080
      COMMON /CINTE/ NCYL,NITER,KSTR,KPRO,KDSKO,KSS,NPITER,IGEOM        GW3D 085
      COMMON /CREAL/ DELT,CHNG,DELMAX,TMAX,DELTO,TOLA,TOLB,W,OME,OMI    GW3D 090
C                                                                       GW3D 095
      COMMON /CS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP                      GW3D 100
      COMMON /CW/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP                      GW3D 105
C                                                                       GW3D 110
      COMMON /CCBC/ MXCNP,MXCES,MXCPR,MXCDP,NCNP,NCES,NCPR,NCDP         GW3D 115
      COMMON /CNBC/ MXNNP,MXNES,MXNPR,MXNDP,NNNP,NNES,NNPR,NNDP         GW3D 120
      COMMON /CVBC/ MXVES,MXVNP,MXRPR,MXRDP,NVES,NVNP,NRPR,NRDP         GW3D 125
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP                    GW3D 130
C                                                                       GW3D 135
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM                                GW3D 140
      COMMON /CMTL/ NMAT,NMPPM,NSPPM                                    GW3D 145
C                                                                       GW3D 150
      COMMON /CFLOW/ FRATE(10),FLOW(10),TFLOW(10)                       GW3D 155
C                                                                       GW3D 160
      DIMENSION X(MAXNP),Y(MAXNP),Z(MAXNP),IE(MAXEL,9)                  GW3D 165
      DIMENSION CMATRX(MAXNP,JBAND),GNOJCN(JBAND,MAXNP),RLD(MAXNP)      GW3D 170
      DIMENSION RI(MAXNP),RL(MAXNP)                                     GW3D 175
      DIMENSION NTNPLR(MXREGN),NNPLR(MXREGN),LMAXDF(MXREGN)             GW3D 180
      DIMENSION GNLR(LTMXNP,MXREGN),LNOJCN(JBAND,LMXNP,MXREGN)          GW3D 185
      DIMENSION CMTRXL(LMXNP,LMXBW),RLDL(LMXNP)                         GW3D 190
      DIMENSION H(MAXNP),HP(MAXNP),HW(MAXNP),HT(MAXNP)                  GW3D 195
```

```
      DIMENSION VX(MAXNP),VY(MAXNP),VZ(MAXNP)
      DIMENSION TH(8,MAXEL),DTH(8,MAXEL),AKR(8,MAXEL), NPCNV(MAXNP)
C
      DIMENSION DCOSB(3,MAXBES),ISB(6,MAXBES), NPBB(MAXBNP)
      DIMENSION BFLX(MAXBNP),BFLXP(MAXBNP)
C
      DIMENSION SOS(MXSPR),SOSF(MXSDP,MXSPR),TSOSF(MXSDP,MXSPR)
      DIMENSION ISTYP(MXSEL),MSEL(MXSEL)
      DIMENSION WSS(MXWPR),WSSF(MXWDP,MXWPR),TWSSF(MXWDP,MXWPR)
      DIMENSION IWTYP(MXWNP),NPW(MXWNP)
C
      DIMENSION QCB(MXCPR),QCBF(MXCDP,MXCPR),TQCBF(MXCDP,MXCPR)
      DIMENSION ICTYP(MXCES),ISC(5,MXCES), NPCB(MXCNP)
C
      DIMENSION QNB(MXNPR),QNBF(MXNDP,MXNPR),TQNBF(MXNDP,MXNPR)
      DIMENSION INTYP(MXNES),ISN(5,MXNES), NPNB(MXNNP)
C
      DIMENSION RFALL(MXRPR),RF(MXRDP,MXRPR),TRF(MXRDP,MXRPR)
      DIMENSION IRTYP(MXVES),ISV(5,MXVES), NPVB(MXVNP)
      DIMENSION DCYFLX(MXVNP),FLX(MXVNP),HCON(MXVNP),HMIN(MXVNP)
      DIMENSION NPFLX(MXVNP),NPCON(MXVNP),NPMIN(MXVNP)
C
      DIMENSION HDB(MXDPR),HDBF(MXDDP,MXDPR),THDBF(MXDDP,MXDPR)
      DIMENSION IDTYP(MXDNP),NPDB(MXDNP)
C
      DIMENSION PROP(MXMPPM,MAXMAT)
      DIMENSION THPROP(MXSPPM,MAXMAT),AKPROP(MXSPPM,MAXMAT)
C
      DIMENSION KPR(MAXNTI),KDSK(MAXNTI),TDTCH(MXNDTC)
C
      DIMENSION TITLE(9)
      DIMENSION SUBHD(8,3)
C
      DATA SUBHD/4HINPU,4HT IN,4HITIA,4HL CO,4HNDIT,4HIONS,2*4H    ,
     > 4HSTEA,4HDY-S,4HTATE,4H INI,4HTIAL,4H CON,4HDITI,4HONS , 8*
     > 4H    /
C
C ******* DATA SET 1: PROBLEM IDENTIFICATION AND DESCRIPTION
C
  100 READ 10, NPROB,(TITLE(I),I=1,9),IGEOM,IBUG,ICHNG
C
      IF (NPROB.LE.0) GO TO 990
C
      PRINT 1000, NPROB,(TITLE(I),I=1,9),IGEOM,IBUG,ICHNG
C
C ------- READ AND PRINT INPUT DATA BY CALLING DATAIN
C
      KOUT=0
      TIME=0.0
```

```
C
C ******* DATA SETS 2 THROUGH 17 WILL BE READ IN DATAIN
C
      CALL DATAIN(TITLE,NPROB, KPR,KDSK,TDTCH,
     1 PROP,THPROP,AKPROP, X,Y,Z,IE,H, DCOSB,ISB,NPBB,
     2 SOSF,TSOSF,ISTYP,MSEL, WSSF,TWSSF,IWTYP,NPW,
     3 QCBF,TQCBF,ICTYP,ISC,NPCB,  QNBF,TQNBF,INTYP,ISN,NPNB,
     4 RF,TRF,IRTYP,ISV,NPVB, HCON,HMIN,  HDBF,THDBF,IDTYP,NPDB,
     5 ISTOP, NNPLR,GNLR)
C
      IF(IGEOM.LE.3) CALL PAGEN(GNOJCN,LNOJCN,LMAXDF,NTNPLR,GNLR,
     1 IE,NNPLR)
      REWIND 4
      IF(IGEOM.LE.3) WRITE(4) ((GNOJCN(J,I),J=1,JBAND),I=1,NNP),
     1 (NTNPLR(K),K=1,MXREGN),(LMAXDF(K),K=1,MXREGN),
     2 (((LNOJCN(J,I,K),J=1,JBAND),I=1,LMXNP),K=1,MXREGN),
     3 ((GNLR(I,K),I=1,LTMXNP),K=1,MXREGN),(NNPLR(K),K=1,MXREGN)
      IF(IGEOM.GT.3) READ(4) ((GNOJCN(J,I),J=1,JBAND),I=1,NNP),
     1 (NTNPLR(K),K=1,MXREGN),(LMAXDF(K),K=1,MXREGN),
     2 (((LNOJCN(J,I,K),J=1,JBAND),I=1,LMXNP),K=1,MXREGN),
     3 ((GNLR(I,K),I=1,LTMXNP),K=1,MXREGN),(NNPLR(K),K=1,MXREGN)
C
      KDIG=0
      IF (ISTOP.GT.0) GO TO 990
C
C ------- PREPARE INITIAL OR PRE-INITIAL VARIABLES
C
      IF(NSEL.NE.0) CALL INTERP(SOS,TSOSF,SOSF,TIME,
     1                 MXSPR,MXSDP,NSPR,NSDP)
      IF(NWNP.NE.0) CALL INTERP(WSS,TWSSF,WSSF,TIME,
     1                 MXWPR,MXWDP,NWPR,NWDP)
C
      IF(NCES.NE.0) CALL INTERP(QCB,TQCBF,QCBF,TIME,
     1                 MXCPR,MXCDP,NCPR,NCDP)
      IF(NNES.NE.0) CALL INTERP(QNB,TQNBF,QNBF,TIME,
     1                 MXNPR,MXNDP,NNPR,NNDP)
      IF(NVES.NE.0) CALL INTERP(RFALL,TRF,RF,TIME,
     1                 MXRPR,MXRDP,NRPR,NRDP)
      IF(NDNP.NE.0) CALL INTERP(HDB,THDBF,HDBF,TIME,
     1                 MXDPR,MXDDP,NDPR,NDDP)
C
C ------- PUT DIRICHLET BOUNDARY VALUES TO INITIAL CONDITIONS
C
      DO 130 I=1,NDNP
      NI=NPDB(I)
      NP=NPBB(NI)
      ITYP=IDTYP(I)
      H(NP)=HDB(ITYP)-Z(NP)*DFLOAT(KGRAV)
  130 CONTINUE
```

```
C
      CALL SPROP(TH,DTH,AKR, IE,H,THPROP,AKPROP)
C
      CALL VELT(VX,VY,VZ,CMATRX,X,Y,Z,IE,H,HT,AKR,PROP)
      KFLOW--1
C
      CALL SFLOW(X,Y,Z,IE, H,HP,VX,VY,VZ,TH,DTH,
     1 BFLX,BFLXP,DCOSB,ISB,NPBB, MSEL,SOS,ISTYP, NPW,WSS,IWTYP,
     2 NPVB,NPDB,NPCB,NPNB,   DELT, KFLOW)
C
      DO 140 I-1,9
      IF(I.EQ.9) GO TO 140
      FLOW(I)-0.0
      TFLOW(I)-0.0
  140 CONTINUE
      FLOW(9)-0.0
C
C ------- PRINT INITIAL OR PRE-INITIAL VARIABLES
C
      KDIAG-0
C
      CALL PRINTT(VX,VY,VZ,H,HT,TH, NPBB,BFLX, NPVB,DCYFLX,NPCON,
     1NPFLX,NPMIN, SUBHD(1,1), TIME,DELT,KPRO,KOUT,KDIAG,-1)
C
      IF(KSTR.EQ.1 .AND. KSS.EQ.1  .AND. KDSKO.EQ.1)
     > CALL STORE(X,Y,Z,IE, H,HT,TH,VX,VY,VZ,DCOSB,ISB,NPBB,
     1 NNPLR,GNLR, TITLE, TIME, NPROB)
C
      IF (KSS.NE.0) GO TO 500
C
C $$$$$$$
C $$$$$$$ PERFORM STEADY-STATE CALCULATION
C $$$$$$$
C
      IF (NVES.EQ.0) GO TO 170
C
      DO 150 NPP-1,NVNP
      NI-NPVB(NPP)
      NPCON(NPP)-NPBB(NI)
      NPMIN(NPP)-0
  150 NPFLX(NPP)-0
C
      NCHG--1
      CALL BCPREP(IE,X,Y,Z,H,VX,VY,VZ,DCOSB,ISB,NPVB,ISV,DCYFLX,FLX,
     > HCON,HMIN,NPFLX,NPCON,NPMIN, IRTYP,RFALL, NCHG)
C
  170 DO 180 NP-1,NNP
  180 HP(NP)-H(NP)
```

```
C                                                                        GW3D 930
      KDIG-KDIG+1                                                        GW3D 935
      IF(IBUG.NE.0) PRINT 10400,KDIG,TIME,DELT                           GW3D 940
C                                                                        GW3D 945
C ------- ITERATION LOOP ON SEEPAGE-RAINFALL BOUNDARY CONDITIONS BEGINS GW3D 950
C                                                                        GW3D 955
      EPS-0.5D0*TOLA                                                     GW3D 960
C                                                                        GW3D 965
      DO 390 ICY-1,NCYL                                                  GW3D 970
C                                                                        GW3D 975
      DO 210 NP-1,NNP                                                    GW3D 980
      HW(NP)-OME*H(NP)+(1.0D0-OME)*HP(NP)                                GW3D 985
      RI(NP)-HW(NP)                                                      GW3D 990
  210 CONTINUE                                                           GW3D 995
C                                                                        GW3D1000
C ------- ITERATION LOOP ON THE NON-LINEAR EQUATION BEGINS               GW3D1005
C                                                                        GW3D1010
      IF(IBUG.NE.0) PRINT 10401, ICY                                     GW3D1015
C                                                                        GW3D1020
C ------- PUT DIRICHLET BOUNDARY VALUES OF THE VARIABLE BOUNDARY         GW3D1025
C ------- INTO H, RI, HW, AND RL                                         GW3D1030
C                                                                        GW3D1035
      IF(NVES.EQ.0) GO TO 250                                            GW3D1040
      DO 230 NPP-1,NVNP                                                  GW3D1045
      NI-NPMIN(NPP)                                                      GW3D1050
      IF(NI.EQ.0) GO TO 220                                              GW3D1055
      H(NI)-HMIN(NPP)                                                    GW3D1060
      RI(NI)-HMIN(NPP)                                                   GW3D1065
      HW(NI)-HMIN(NPP)                                                   GW3D1070
      RL(NI)-HMIN(NPP)                                                   GW3D1075
      GO TO 230                                                          GW3D1080
  220 NI-NPCON(NPP)                                                      GW3D1085
      IF(NI.EQ.0) GO TO 230                                              GW3D1090
      H(NI)-HCON(NPP)                                                    GW3D1095
      RI(NI)-HCON(NPP)                                                   GW3D1100
      HW(NI)-HCON(NPP)                                                   GW3D1105
      RL(NI)-HCON(NPP)                                                   GW3D1110
  230 CONTINUE                                                           GW3D1115
  250 CONTINUE                                                           GW3D1120
C                                                                        GW3D1125
      DO 350 IT-1,NITER                                                  GW3D1130
C                                                                        GW3D1135
C ------- EVALUATE SOIL PROPERTIES FOR PREVIOUS ITERATE                  GW3D1140
C                                                                        GW3D1145
      CALL SPROP(TH,DTH,AKR, IE,HW,THPROP,AKPROP)                        GW3D1150
```

```
C
C ------- ASSEMBLE STEAD-STATE ELEMENT MATRICES QA AND QB INTO THE
C ------- GLOBAL MATRIX C AND CONSTRUCT GLOBAL LOAD VECTOR R FROM
C ------- ELEMENT LOAD VECTOR RQ.
C
      CALL ASEMBL(X,Y,Z,IE, CMATRX,RLD,GNOJCN,HW,HP,DTH,AKR,PROP,
     > SOS,MSEL,ISTYP,WSS,NPW,IWTYP, KSS,W,DELT)
C
C ------- APPLY STEADY-STATE BOUNDARY CONDITIONS
C
      CALL BC(CMATRX,RLD,GNOJCN, IE,X,Y,Z, AKR,PROP, DCOSB,ISB,NPBB,
     1 QCB,ISC,ICTYP, QNB,ISN,INTYP, FLX,HCON,HMIN,NPFLX,NPCON,NPMIN,
     2 HDB,IDTYP,NPDB, KSS)
C
C ------- SOLVE THE MATRIX EQUATION BY BLOCK ITERATION
C
      CALL BLKITR(RL,RI,CMTRXL,RLDL, CMATRX,RLD, GNLR,LNOJCN,NNPLR,
     1 LMAXDF, EPS,NPITER,IBUG,KPRO,OMI)
C
C ------- OBTAIN MAXIMUM RELATIVE DEVIATION FROM PREVIOUS ITERATE
C
      NPP=0
      RD=-1.0D0
      RES=-1.0D0
      DO 320 NP=1,NNP
      RESNP=DABS(RL(NP)-H(NP))
      RES=DMAX1(RES,RESNP)
      IF(H(NP).NE.0.0D0) RD=DMAX1(RD,DABS(RESNP/H(NP)))
      IF(RESNP .LE. TOLA) GO TO 320
      NPP=NPP+1
      NPCNV(NPP)=NP
  320 CONTINUE
C
      NNCVN=NPP
C
C ------- UPDATE PRESSURE WITH CURRENT ITERATE
C
      DO 330 NP=1,NNP
      H(NP)=OME*RL(NP)+(1.0D0-OME)*H(NP)
      RI(NP)=H(NP)
      HW(NP)=H(NP)
  330 CONTINUE
C
C ------- ESCAPE FROM ITERATION LOOP IF THE MAXIMUM RESIDUAL IS
C ------- SUFFICIENTLY SMALL
C
      IF(IBUG.NE.0) PRINT 10200, IT,RES,RD,NNCVN
      IF(IT.EQ.1) GO TO 350
      IF(RES.LT.TOLA) GO TO 360
```

```
C
  350 CONTINUE
C
C ------- END OF ITERATION LOOP ON THE NON-LINEAR EQUATION
C
      PRINT 10210, ICY,IT,NITER,RES,RD,NNCVN
C
C ------- PRINT NONCONVERGENING NODES
C
      IF(IBUG.EQ.0) GO TO 360
      PRINT 10500
      PRINT 10600, (NPCNV(NPP),NPP=1,NNCVN)
C
  360 IF(ICHNG.EQ.0) GO TO 380
      IF(NVES.EQ.0) GO TO 380
C
C ------- PRINT RAINFALL-SEEPAGE B. C. CHANGE INFORMATION
C
      PRINT 10402, ICY
      DO 370 I=1,NVNP
      NI=NPVB(I)
      NP=NPBB(NI)
      PRINT 10403, I,NP,NPCON(I),HCON(I),NPMIN(I),HMIN(I),NPFLX(I),
     1   FLX(I),DCYFLX(I)
  370 CONTINUE
C
C ------- CALCULATE DARCY'S VELOCITY
C
  380 CALL SPROP(TH,DTH,AKR, IE,H,THPROP,AKPROP)
C
      CALL VELT(VX,VY,VZ,CMATRX,X,Y,Z,IE,H,HT,AKR,PROP)
C
      IF(NVES .EQ. 0) GO TO 440
C
C ------- PREPARE BOUNDARY CONDITIONS ON THE VARIABLE-TYPE BOUNDARY FOR
C ------- NEXT CYCLE COMPUTATIONS.
C
      CALL BCPREP(IE,X,Y,Z, H,VX,VY,VZ, DCOSB,ISB, NPVB,ISV,DCYFLX,FLX,
     > HCON,HMIN,NPFLX,NPCON,NPMIN,IRTYP,RFALL, NCHG)
C
      IF(NCHG.EQ.0) GO TO 440
  390 CONTINUE
C
C ------- END OF ITERATION LOOP ON SEEPAGE-RAINFALL BOUNDARY CONDITIONS
C
      PRINT 10610, ICY,IT,NCYL,NITER,RES,RD,NNCVN
C
  440 IF(NNCVN.EQ.0) GO TO 445
      PRINT 10610, ICY,IT,NCYL,NITER,RES,RD,NNCVN
```

```
C                                                                       GW3D1645
C ------- COMPUTE FLUXES THROUGH ALL TYPES OF BOUNDARIES.               GW3D1650
C                                                                       GW3D1655
  445 KFLOW=0                                                           GW3D1660
      CALL SFLOW(X,Y,Z,IE, H,HP,VX,VY,VZ,TH,DTH,                        GW3D1665
     1 BFLX,BFLXP,DCOSB,ISB,NPBB, MSEL,SOS,ISTYP,NPW,WSS,IWTYP,         GW3D1670
     2 NPVB,NPDB,NPCB,NPNB,    DELT, KFLOW)                             GW3D1675
C                                                                       GW3D1680
      DO 450 I=1,9                                                      GW3D1685
      IF(I.EQ.9) GO TO 450                                              GW3D1690
      FLOW(I)=0.0                                                       GW3D1695
      TFLOW(I)=0.0                                                      GW3D1700
  450 CONTINUE                                                          GW3D1705
      FLOW(9)=0.0                                                       GW3D1710
C                                                                       GW3D1715
C ------- PRINT STEADY-STATE VARIABLES                                  GW3D1720
C                                                                       GW3D1725
      CALL PRINTT(VX,VY,VZ,H,HT,TH, NPBB,BFLX, NPVB,DCYFLX,NPCON,       GW3D1730
     1 NPFLX,NPMIN, SUBHD(1,2), TIME,DELT, KPRO,KOUT,KDIAG,0)           GW3D1735
C                                                                       GW3D1740
      IF(KSTR.EQ.1 .AND. KDSKO.EQ.1) CALL STORE(X,Y,Z,IE,               GW3D1745
     > H,HT,TH,VX,VY,VZ,DCOSB,ISB,NPBB, NNPLR,GNLR, TITLE,TIME,NPROB)   GW3D1750
C                                                                       GW3D1755
      IF (NTI.EQ.0) GO TO 100                                           GW3D1760
C                                                                       GW3D1765
      KSS=1                                                             GW3D1770
C                                                                       GW3D1775
C $$$$$$$                                                               GW3D1780
C $$$$$$$ PERFORM TRANSIENT-STATE CALCULATION                           GW3D1785
C $$$$$$$                                                               GW3D1790
C                                                                       GW3D1795
  500 IF (NVES.EQ.0) GO TO 550                                          GW3D1800
C                                                                       GW3D1805
      DO 510 NPP=1,NVNP                                                 GW3D1810
      NI=NPVB(NPP)                                                      GW3D1815
      NPCON(NPP)=NPBB(NI)                                               GW3D1820
      NPMIN(NPP)=0                                                      GW3D1825
  510 NPFLX(NPP)=0                                                      GW3D1830
C                                                                       GW3D1835
      NCHG=-1                                                           GW3D1840
C                                                                       GW3D1845
  550 TIME=TIME+DELT                                                    GW3D1850
      W1=W                                                              GW3D1855
      W2=1.0D0-W                                                        GW3D1860
      KFLOW=1                                                           GW3D1865
      TFLOW(9)=0.0                                                      GW3D1870
```

```
C                                                                       GW3D1875
C  ------- BEGIN THE TIME-MARCHING LOOP                                 GW3D1880
C                                                                       GW3D1885
      EPS=0.5D0*TOLB                                                    GW3D1890
      IDELT=0                                                           GW3D1895
      DO 890 ITM=1,NTI                                                  GW3D1900
      ITMITM=ITM                                                        GW3D1905
C                                                                       GW3D1910
C ------- PREPARE TRANSIENT BOUNDARY CONDITIONS AND SOURCE FOR THE STEP GW3D1915
C                                                                       GW3D1920
      IF(NSEL.NE.0) CALL INTERP(SOS,TSOSF,SOSF,TIME,MXSPR,MXSDP,        GW3D1925
     1 NSPR,NSDP)                                                       GW3D1930
      IF(NWNP.NE.0) CALL INTERP(WSS,TWSSF,WSSF,TIME,MXWPR,MXWDP,        GW3D1935
     1 NWPR,NWDP)                                                       GW3D1940
C                                                                       GW3D1945
      IF(NCES.NE.0) CALL INTERP(QCB,TQCBF,QCBF,TIME,                    GW3D1950
     1                    MXCPR,MXCDP,NCPR,NCDP)                        GW3D1955
      IF(NNES.NE.0) CALL INTERP(QNB,TQNBF,QNBF,TIME,                    GW3D1960
     1                    MXNPR,MXNDP,NNPR,NNDP)                        GW3D1965
      IF(NVES.NE.0) CALL INTERP(RFALL,TRF,RF,TIME,                      GW3D1970
     1                    MXRPR,MXRDP,NRPR,NRDP)                        GW3D1975
      IF(NDNP.NE.0) CALL INTERP(HDB,THDBF,HDBF,TIME,                    GW3D1980
     1                    MXDPR,MXDDP,NDPR,NDDP)                        GW3D1985
C                                                                       GW3D1990
      IF(NVES.EQ.0) GO TO 560                                           GW3D1995
      NCHG=-1                                                           GW3D2000
      CALL BCPREP(IE,X,Y,Z, H,VX,VY,VZ, DCOSB,ISB, NPVB,ISV,DCYFLX,FLX, GW3D2005
     1 HCON,HMIN,NPFLX,NPCON,NPMIN, IRTYP,RFALL, NCHG)                  GW3D2010
C                                                                       GW3D2015
  560 DO 570 NP=1,NNP                                                   GW3D2020
      RL(NP)=H(NP)                                                      GW3D2025
      HP(NP)=H(NP)                                                      GW3D2030
  570 CONTINUE                                                          GW3D2035
C                                                                       GW3D2040
      KDIG=KDIG+1                                                       GW3D2045
      IF(IBUG.NE.0 .AND. KPR(ITM).NE.0) PRINT 10400, KDIG,TIME,DELT     GW3D2050
C                                                                       GW3D2055
C ------- BEGIN ITERATION LOOP ON SEEPAGE-RAINFALL BOUNDARY CONDITIONS  GW3D2060
C                                                                       GW3D2065
      DO 690 ICY=1,NCYL                                                 GW3D2070
      IF(IBUG.NE.0 .AND. KPR(ITM).NE.0) PRINT 10401, ICY                GW3D2075
C                                                                       GW3D2080
      DO 580 NP=1,NNP                                                   GW3D2085
      H(NP)=OME*RL(NP)+(1.0D0-OME)*H(NP)                                GW3D2090
      RL(NP)=H(NP)                                                      GW3D2095
      HW(NP)=W1*(OME*H(NP)+(1.0D0-OME)*HP(NP))+W2*HP(NP)                GW3D2100
  580 CONTINUE                                                          GW3D2105
```

```
C                                                                       GW3D2110
C ------- BEGIN ITERATION LOOP ON THE NON-LINEAR EQUATION               GW3D2115
C                                                                       GW3D2120
C ------- PUT DIRICHLET BOUNDARY VALUES OF THE VARIABLE BOUNDARY        GW3D2125
C ------- INTO H, RI, HW, AND RL                                        GW3D2130
C                                                                       GW3D2135
      IF(NVES.EQ.0) GO TO 595                                           GW3D2140
      DO 590 NPP=1,NVNP                                                 GW3D2145
      NI=NPMIN(NPP)                                                     GW3D2150
      IF(NI.EQ.0) GO TO 585                                             GW3D2155
      H(NI)=HMIN(NPP)                                                   GW3D2160
      RI(NI)=HMIN(NPP)                                                  GW3D2165
      HW(NI)=HMIN(NPP)                                                  GW3D2170
      RL(NI)=HMIN(NPP)                                                  GW3D2175
      GO TO 590                                                         GW3D2180
  585 NI=NPCON(NPP)                                                     GW3D2185
      IF(NI.EQ.0) GO TO 590                                             GW3D2190
      H(NI)=HCON(NPP)                                                   GW3D2195
      RI(NI)=HCON(NPP)                                                  GW3D2200
      HW(NI)=HCON(NPP)                                                  GW3D2205
      RL(NI)=HCON(NPP)                                                  GW3D2210
  590 CONTINUE                                                          GW3D2215
  595 CONTINUE                                                          GW3D2220
C                                                                       GW3D2225
      DO 650 IT=1,NITER                                                 GW3D2230
C                                                                       GW3D2235
C ------- EVALUATE SOIL PROPERTIES FOR PREVIOUS ITERATE                 GW3D2240
C                                                                       GW3D2245
      CALL SPROP(TH,DTH,AKR, IE,HW,THPROP,AKPROP)                       GW3D2250
C                                                                       GW3D2255
C ------- ASSEMBLE ELEMENT MATRICES QA AND QB INTO THE GLOBAL MATRIX C  GW3D2260
C ------- AND CONSTRUCT THE GLOBAL LOAD VECTOR R FROM ELEMENT LOAD      GW3D2265
C ------- VECTOR RQ.                                                    GW3D2270
C                                                                       GW3D2275
      CALL ASEMBL(X,Y,Z,IE, CMATRX,RLD,GNOJCN,HW,HP,DTH, AKR,PROP,      GW3D2280
     > SOS,MSEL,ISTYP,WSS,NPW,IWTYP, KSS,W,DELT)                        GW3D2285
C                                                                       GW3D2290
C ------- APPLY BOUNDARY CONDITIONS TO MODIFY THE GLOBAL MATRIX C AND   GW3D2295
C ------- THE LOAD VECTOR R.                                            GW3D2300
C                                                                       GW3D2305
      CALL BC(CMATRX,RLD,GNOJCN, IE,X,Y,Z, AKR,PROP, DCOSB,ISB,NPBB,    GW3D2310
     1 QCB,ISC,ICTYP, QNB,ISN,INTYP, FLX,HCON,HMIN,NPFLX,NPCON,NPMIN,   GW3D2315
     2 HDB,IDTYP,NPDB, KSS)                                             GW3D2320
C                                                                       GW3D2325
C ------- SOLVE THE MATRIX EQUAITON BY BLOCK ITERATION                  GW3D2330
C                                                                       GW3D2335
      CALL BLKITR(RL,RI,CMTRXL,RLDL, CMATRX,RLD, GNLR,LNOJCN,NNPLR,     GW3D2340
     1 LMAXDF, EPS,NPITER,IBUG,KPR(ITM),OMI)                            GW3D2345
```

```
C                                                                       GW3D2350
C ------- OBTAIN MAXIMUM RELATIVE DEVIATION FROM PREVIOUS ITERATE       GW3D2355
C                                                                       GW3D2360
      NPP=0                                                             GW3D2365
      RD=-1.0D0                                                         GW3D2370
      RES=-1.0D0                                                        GW3D2375
      DO 620 NP=1,NNP                                                   GW3D2380
      RESNP=DABS(RL(NP)-H(NP))                                          GW3D2385
      RES=DMAX1(RES,RESNP)                                              GW3D2390
      IF(H(NP) .NE. 0.0D0) RD=DMAX1(RD,DABS(RESNP/H(NP)))               GW3D2395
      IF(RESNP .LE. TOLB) GO TO 620                                     GW3D2400
      NPP=NPP+1                                                         GW3D2405
      NPCNV(NPP)=NP                                                     GW3D2410
  620 CONTINUE                                                          GW3D2415
C                                                                       GW3D2420
      NNCVN=NPP                                                         GW3D2425
C                                                                       GW3D2430
C ------- UPDATE PRESSURE WITH CURRENT ITERATE                          GW3D2435
C                                                                       GW3D2440
      DO 630 NP=1,NNP                                                   GW3D2445
      H(NP)=OME*RL(NP)+(1.0D0-OME)*H(NP)                                GW3D2450
      RI(NP)=H(NP)                                                      GW3D2455
      HW(NP)=W1*H(NP)+W2*HP(NP)                                         GW3D2460
  630 CONTINUE                                                          GW3D2465
C                                                                       GW3D2470
C ------- ESCAPE FROM ITERATION LOOP IF THE MAXIMUM RESIDUAL IS         GW3D2475
C ------- SUFFICIENTLY SMALL.                                           GW3D2480
C                                                                       GW3D2485
      IF(IBUG.NE.0 .AND. KPR(ITM).NE.0) PRINT 10200, IT,RES,RD,NNCVN    GW3D2490
      IF(IT.EQ.1 .AND. ITM.EQ.1) GO TO 650                              GW3D2495
      IF(RES.LT.TOLB) GO TO 660                                         GW3D2500
C                                                                       GW3D2505
  650 CONTINUE                                                          GW3D2510
C                                                                       GW3D2515
C ------- END THE ITERATION LOOP ON THE NON-LINEAR EQUATION             GW3D2520
C                                                                       GW3D2525
      PRINT 10710, ITM,ICY,IT,NITER,RES,RD,NNCVN                        GW3D2530
C                                                                       GW3D2535
      IF(IBUG.EQ.0 .OR. KPR(ITM).EQ.0) GO TO 660                        GW3D2540
C                                                                       GW3D2545
C ------- PRINT NONCONVERGING NODES                                     GW3D2550
C                                                                       GW3D2555
      PRINT 10500                                                       GW3D2560
      PRINT 10600, (NPCNV(NPP),NPP=1,NNCVN)                             GW3D2565
C                                                                       GW3D2570
  660 IF(ICHNG.EQ.0 .OR. KPR(ITM).EQ.0) GO TO 680                       GW3D2575
      IF(NVES.EQ.0) GO TO 680                                           GW3D2580
```

```
C
C ------- PRINT RAINFALL-SEEPAGE BOUNDARY CONDITION CHANGE INFORMATION
C
      PRINT 10402, ICY
      DO 670 I-1,NVNP
      NI-NPVB(I)
      NP-NPBB(NI)
      PRINT 10403, I,NP,NPCON(I),HCON(I),NPMIN(I),HMIN(I),NPFLX(I),
     1    FLX(I),DCYFLX(I)
  670 CONTINUE
C
C ------- CALCULATE DARCY'S VELOCITY
C
  680 CALL SPROP(TH,DTH,AKR, IE,H,THPROP,AKPROP)
C
      CALL VELT(VX,VY,VZ,CMATRX,X,Y,Z,IE,H,HT,AKR,PROP)
C
      IF(NVES.EQ.0) GO TO 710
C
      CALL BCPREP(IE,X,Y,Z,H,VX,VY,VZ, DCOSB,ISB, NPVB,ISV,DCYFLX,FLX,
     > HCON,HMIN,NPFLX,NPCON,NPMIN,IRTYP,RFALL, NCHG)
C
      IF(NCHG.EQ.0) GO TO 710
C
  690 CONTINUE
C
C ------- END ITERATION LOOP ON SEEPAGE-RAINFALL BOUNDARY CONDITIONS
C
      PRINT 10810, ITM,ICY,IT,NCYL,NITER,RES,RD,NNCVN
  710 IF(NNCVN.EQ.0) GO TO 740
      PRINT 10810, ITM,ICY,IT,NCYL,NITER,RES,RD,NNCVN
C
C ------- COMPUTE FLUXES THROUGH ALL TYPES OF BOUNDARIES
C
  740 CALL SFLOW(X,Y,Z,IE, H,HP,VX,VY,VZ,TH,DTH,
     1 BFLX,BFLXP,DCOSB,ISB,NPBB, MSEL,SOS,ISTYP,NPW,WSS,IWTYP,
     2 NPVB,NPDB,NPCB,NPNB,  DELT, KFLOW)
C
C ------- PRINT VARIABLES AT EACH TIME STEP
C
      CALL PRINTT(VX,VY,VZ,H,HT,TH, NPBB,BFLX, NPVB,DCYFLX,NPCON,
     1 NPFLX,NPMIN,SUBHD(1,3), TIME,DELT, KPR(ITM),KOUT,KDIAG,ITMITM)
C
      IF(KSTR.EQ.1 .AND.KDSK(ITM).EQ.1) CALL STORE(X,Y,Z,IE,
     > H,HT,TH,VX,VY,VZ, DCOSB,ISB,NPBB, NNPLR,GNLR, TITLE, TIME,NPROB)
C
C ------- PREPARE FOR NEXT TIME STEP
C
      IF(TIME.GT.TMAX) GO TO 100
      DELT-DELT*(1.0D0+CHNG)
```

```
      DELT=DMIN1(DELT,DELMAX)                                           GW3D2835
      IF(IDELT.EQ.0) GO TO 880                                          GW3D2840
      IF(TIME.EQ.TDTCH(IDELT)) DELT=DELTO                               GW3D2845
  880 TIME=TIME+DELT                                                    GW3D2850
      IF(TIME.LT.TDTCH(IDELT+1)) GO TO 890                              GW3D2855
      IDELT=IDELT+1                                                     GW3D2860
      TIME=TIME-DELT                                                    GW3D2865
      DELT=TDTCH(IDELT)-TIME                                            GW3D2870
      IF(DELT.LE.0.0) DELT=DELTO                                        GW3D2875
      TIME=TIME+DELT                                                    GW3D2880
  890 CONTINUE                                                          GW3D2885
C                                                                       GW3D2890
C ----- END OF TIME-MARCHING LOOP                                       GW3D2895
C                                                                       GW3D2900
      GO TO 100                                                         GW3D2905
C                                                                       GW3D2910
  990 RETURN                                                            GW3D2915
C                                                                       GW3D2920
   10 FORMAT(I5,9A8,3I1)                                                GW3D2925
 1000 FORMAT('1 PROBLEM',I5,'.. ',9A8,1X,3I1/)                          GW3D2930
10200 FORMAT(5X,I10,3X,E12.4,3X,E12.4,15X,I10)                          GW3D2935
10400 FORMAT('1','****************************************************',GW3D2940
     1 '**************************************************************',GW3D2945
     2 '*****'///' DIAGNOSTIC TABLE',I4,'.. AT TIME =',1PD12.4,         GW3D2950
     3 ' ,(DELT =',1PD12.4,')')                                         GW3D2955
10401 FORMAT(//' TABLE OF ITERATIVE PARAMETERS FOR',I3,'-TH CYCLE'//6X, GW3D2960
     1 'ITERATION',7X,'RESIDUAL',6X,'DEVIATION',6X,                     GW3D2965
     2 'NO. NON-CONV. NODES')                                           GW3D2970
10402 FORMAT(//' TABLE OF RAINFALL/EVAPORATION-SEEPAGE B. C. USED FOR', GW3D2975
     1 I3,'-TH CYCLE'//7X,' I  NPVB  NPCON     HCON     NPMIN     HMIN',GW3D2980
     2 '     NPFLX        FLX          DCYFLX FROM PREVIOUS CYCLE'/7X,  GW3D2985
     3         ' -  ----  -----        ----      -----      ----',      GW3D2990
     4 '     -----        ---          -------------------------')      GW3D2995
10403 FORMAT(1H ,I8,I6,I7,D12.4,I7,D12.4,I7,D12.4,12X,D12.4)            GW3D3000
10500 FORMAT(//' TABLE OF NON-CONVERGING NODES')                        GW3D3005
10600 FORMAT(/(5X,20I5))                                                GW3D3010
10210 FORMAT(1H0,'WARING: NON-CONVERGENCE OCCUR DURING STEADY STATE SOLUGW3D3015
     1TION AT',I3,' -TH CYCLE'/1H ,'IT = ',I3,'  .GT.  MAXIT = ',I3,    GW3D3020
     2 ',  RES =',D12.5,',  RD =',D12.4,',  NNCVN =',I4)                GW3D3025
10610 FORMAT(1H0,'ABSOLUTELY WARNING: STEADY STATE SOLUTION IS NG'/1H , GW3D3030
     > 'ICY = ',I3,'  IT = ',I3,'  MAXCY = ',I3,'  MAXIT = ',I3,        GW3D3035
     > ',  RES =',D12.4,',  RD =',D12.4,',  NNCVN =',I4)                GW3D3040
10710 FORMAT(1H0,'WARNING: NON-CONVERGENCE OCCUR AT',I5,' -TH TIME STEP'GW3D3045
     >,I3,' -TH CYCLE'/1H ,'IT = ',I3,'  .GT.  MAXIT = ',I3,2D12.4,I5)  GW3D3050
10810 FORMAT(1H0,'ABSOLUTELY WARNING: TRANSIENT SOLUTION IS NG AT ',I5, GW3D3055
     > ' -TH TIME STEP'/1H ,'ICY = ',I3,'  IT =  ,I3,'  MAXCY = ',I3,   GW3D3060
     > '  MAXIT = ',I3,',  RES =',D12.4,',  RD =',D12.4,',  NNCVN =',I4)GW3D3065
C                                                                       GW3D3070
      END                                                               GW3D3075
```

```
      SUBROUTINE DATAIN(TITLE,NPROB, KPR,KDSK,TDTCH,                    DATA 005
     1 PROP,THPROP,AKPROP, X,Y,Z,IE,H, DCOSB,ISB,NPBB,                  DATA 010
     2 SOSF,TSOSF,ISTYP,MSEL, WSSF,TWSSF,IWTYP,NPW,                     DATA 015
     3 QCBF,TQCBF,ICTYP,ISC,NPCB,  QNBF,TQNBF,INTYP,ISN,NPNB,           DATA 020
     4 RF,TRF,IRTYP,ISV,NPVB, HCON,HMIN,  HDBF,THDBF,IDTYP,NPDB,        DATA 025
     5 ISTOP,NNPLR,GNLR)                                                DATA 030
C                                                                       DATA 035
C********1*********2*********3*********4*********5*********6*********7**DATA 040
C ------- TO READ AND PRINT SYSTEM PARAMETERS, GEOMETRY, BOUNDARY AND   DATA 045
C ------- INITIAL CONDITIONS, AND PROPERTIES OF THE MEDIA.              DATA 050
C********1*********2*********3*********4*********5*********6*********7**DATA 055
C                                                                       DATA 060
      IMPLICIT REAL*8(A-H,O-Z)                                          DATA 065
      REAL*4 PMAT,THPAR,AKPAR                                           DATA 070
      INTEGER*4 GNLR                                                    DATA 075
C                                                                       DATA 080
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC      DATA 085
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI,NDTCHG                 DATA 090
      COMMON /LGEOM/ LTMXNP,LMXNP,LMXBW,MXREGN,NREGN                    DATA 095
      COMMON /CINTE/ NCYL,NITER,KSTR,KPRO,KDSKO,KSS,NPITER,IGEOM        DATA 100
      COMMON /CREAL/ DELT,CHNG,DELMAX,TMAX,DELTO,TOLA,TOLB,W,OME,OMI    DATA 105
C                                                                       DATA 110
      COMMON /CS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP                      DATA 115
      COMMON /CW/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP                      DATA 120
C                                                                       DATA 125
      COMMON /CCBC/ MXCNP,MXCES,MXCPR,MXCDP,NCNP,NCES,NCPR,NCDP         DATA 130
      COMMON /CNBC/ MXNNP,MXNES,MXNPR,MXNDP,NNNP,NNES,NNPR,NNDP         DATA 135
      COMMON /CVBC/ MXVES,MXVNP,MXRPR,MXRDP,NVES,NVNP,NRPR,NRDP         DATA 140
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP                    DATA 145
C                                                                       DATA 150
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM                                DATA 155
      COMMON /CMTL/ NMAT,NMPPM,NSPPM                                    DATA 160
C                                                                       DATA 165
      DIMENSION TITLE(9)                                                DATA 170
C                                                                       DATA 175
      DIMENSION KPR(MAXNTI),KDSK(MAXNTI),TDTCH(MXNDTC)                  DATA 180
C                                                                       DATA 185
      DIMENSION PROP(MXMPPM,MAXMAT)                                     DATA 190
      DIMENSION THPROP(MXSPPM,MAXMAT),AKPROP(MXSPPM,MAXMAT)             DATA 195
C                                                                       DATA 200
      DIMENSION X(MAXNP),Y(MAXNP),Z(MAXNP),IE(MAXEL,9), H(MAXNP)        DATA 205
      DIMENSION DCOSB(3,MAXBES),ISB(6,MAXBES), NPBB(MAXBNP)             DATA 210
C                                                                       DATA 215
      DIMENSION MSEL(MXSEL),ISTYP(MXSEL)                                DATA 220
      DIMENSION SOSF(MXSDP,MXSPR),TSOSF(MXSDP,MXSPR)                    DATA 225
      DIMENSION NPW(MXWNP),IWTYP(MXWNP)                                 DATA 230
      DIMENSION WSSF(MXWDP,MXWPR),TWSSF(MXWDP,MXWPR)                    DATA 235
C                                                                       DATA 240
      DIMENSION QCBF(MXCDP,MXCPR),TQCBF(MXCDP,MXCPR),ICTYP(MXCES)       DATA 245
      DIMENSION ISC(5,MXCES),NPCB(MXCNP)                                DATA 250
```

```
      DIMENSION QNBF(MXNDP,MXNPR),TQNBF(MXNDP,MXNPR),INTYP(MXNES)
      DIMENSION ISN(5,MXNES),NPNB(MXNNP)
      DIMENSION RF(MXRDP,MXRPR),TRF(MXRDP,MXRPR), IRTYP(MXVES)
      DIMENSION ISV(5,MXVES),NPVB(MXVNP)
      DIMENSION HCON(MXVNP),HMIN(MXVNP)
      DIMENSION HDBF(MXDDP,MXDPR),THDBF(MXDDP,MXDPR),IDTYP(MXDNP)
      DIMENSION NPDB(MXDNP)
      DIMENSION NNPLR(MXREGN),GNLR(LTMXNP,MXREGN)
C
      DIMENSION NIMI(4),NJMJ(4),IEM(8)
      DIMENSION PMAT(3,6),AKPAR(3,8),THPAR(3,8)
C
      DATA PMAT/4H   S,4HAT K,4HXX  ,4H   S,4HAT K,4HYY  ,4H   S,
     > 4HAT K,4HZZ  ,4H   S,4HAT K,4HXY  ,4H   S,4HAT K,4HXZ  ,4H   S,
     >4HAT K,4HYZ  /
C
      DATA THPAR/4H    ,4H TH1,4H    ,4H    ,4H TH2,4H    ,4H    ,
     > 4H TH3,4H    ,4H    ,4H TH4,4H    ,4H    ,4H TH5,4H    ,4H    ,
     > 4H TH6,4H    ,4H    ,4H TH7,4H    ,4H    ,4H TH8,4H    /
C
      DATA AKPAR/4H    ,4H  K1,4H    ,4H    ,4H  K2,4H    ,4H    ,
     > 4H  K3,4H    ,4H    ,4H  K4,4H    ,4H    ,4H  K5,4H    ,
     > 4H    ,4H  K6,4H    ,4H    ,4H  K7,4H    ,4H    ,4H  K8,4H    /
C
      ISTOP-0
C
C ******* DATA SET 2: BASIC INTEGER PARAMETERS
C
      READ 10, NNP,NEL,NMAT,NCM,NTI,KSS,NSPPM,NMPPM,KSTR,
     1 KCP,KGRAV,NITER,NCYL,NDTCHG,NPITER,NREGN
      IF(NDTCHG.LE.0) NDTCHG-1
      PRINT 1000, NNP,NEL,NMAT,NCM,NTI,KSS,NSPPM,NMPPM,KSTR,
     1 KCP,KGRAV,NITER,NCYL,NDTCHG,NPITER,NREGN
C
C ******* DATA SET 3: BASIC REAL PARAMETERS
C
      READ 20, DELT,CHNG,DELMAX,TMAX,TOLA,TOLB,RHO,GRAV,VISC,W,
     1 OME,OMI
C
      DELTO-DELT
      IF(TMAX.LE.0.0) TMAX-1.0D38
C
      PRINT 1100, DELT,CHNG,DELMAX,TMAX,TOLA,TOLB,RHO,GRAV,VISC,W,
     1 OME,OMI
C
C ******* DATA SET 4: LINE PRINT CONTROL AND DISK STORE CONTROL
C
      READ 30, KPRO,(KPR(ITM),ITM-1,NTI)
      READ 30, KDSKO,(KDSK(ITM),ITM-1,NTI)
      READ 20, (TDTCH(I),I-1,NDTCHG)
```

```
C
      PRINT 1200
      PRINT 1210, KPRO,(KPR(ITM),ITM-1,NTI)
      PRINT 1300
      PRINT 1210,  KDSKO,(KDSK(ITM),ITM-1,NTI)
      PRINT 1350,  (TDTCH(I),I-1,NDTCHG)
C
C ******* DATA SET 5: MATERIAL PROPERTIES
C
      PRINT 1400, ((PMAT(I,J),I-1,3),J-1,NMPPM)
      DO 100 I-1,NMAT
      READ 20, (PROP(J,I),J-1,NMPPM)
  100 PRINT 1410, I,(PROP(J,I),J-1,NMPPM)
C
C ******* DATA SET 6: SOIL PROPERTIES
C
C
C -------- READ AND PRINT PARAMETERS REQUIRE TO COMPUTE
C -------- MOISTURE CONTENT GIVEN THE PRESSURE HEAD
C
      IF (NSPPM.EQ.0) GO TO 200
      PRINT 1500, ((THPAR(I,J),I-1,3),J-1,8)
      DO 105 I-1,NMAT
      READ 20, (THPROP(J,I),J-1,NSPPM)
      PRINT 1510, I,(THPROP(J,I),J-1,NSPPM)
  105 CONTINUE
C
C ------- READ AND PRINT PARAMETERS REQUIRED TO COMPUTE RELATIVE
C ------- HYDRAULIC CONDUCTIVITY
C
      PRINT 1600, ((AKPAR(I,J),I-1,3),J-1,8)
      DO 110 I-1,NMAT
      READ 20, (AKPROP(J,I),J-1,NSPPM)
      PRINT 1510, I,(AKPROP(J,I),J-1,NSPPM)
  110 CONTINUE
      IF (KCP.EQ.0) GO TO 200
C
C ------- CONVERT FROM SATURATED PERMEABILITY TO SATURATED CONDUCTIVITY
C
      DO 190 I-1,NMAT
      PKCF-RHO*GRAV/VISC
      DO 190 J-1,6
      PROP(J,I)-PROP(J,I)*PKCF
  190 CONTINUE
C
C ******* DATA SET 7: NODE COORDINATES
C
C ------- READ NODAL POINT COORDINATES
C
  200 NPI-0
```

```
  210 READ 40, NI,NSEQ,NIAD,XI,YI,ZI,XIAD,YIAD,ZIAD
      IF(NI.EQ.0) GO TO 240
      NJ=NI+NSEQ
      DO 220 NP=NI,NJ
      I=NI+NIAD*(NP-NI)
      X(I)=XI+XIAD*DFLOAT(NP-NI)
      Y(I)=YI+YIAD*DFLOAT(NP-NI)
      Z(I)=ZI+ZIAD*DFLOAT(NP-NI)
      NPI=NPI+1
  220 CONTINUE
      GO TO 210
  240 IF(NPI.EQ.NNP) GO TO 250
      PRINT 2000
      STOP
  250 CONTINUE
C
C ------- PRINT NODAL POINT COORDINATES
C
      IF(MOD(IGEOM,2).EQ.0) GO TO 270
      LINE=0
      DO 265 NP=1,NNP,3
      NJMN=NP
      NJMX=MIN0(NP+2,NNP)
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 2100
      PRINT 2110, (NJ,X(NJ),Y(NJ),Z(NJ),NJ=NJMN,NJMX)
  265 CONTINUE
  270 CONTINUE
C
C ******* DATA SET 8: SUBREGION DATA
C
      CALL READN(NNPLR,MXREGN,NREGN)
      IF(MOD(IGEOM,2).NE.0) PRINT 2200
      DO 280 K=1,NREGN
      LNNP=NNPLR(K)
      CALL READN(GNLR(1,K),LTMXNP,LNNP)
      IF(MOD(IGEOM,2).NE.0) PRINT 2210, K,(GNLR(I,K),I=1,LNNP)
  280 CONTINUE
C
C ******* DATA SET 9: ELEMENT DATA
C
C ------- READ ELEMENT INDICES AND COMPUTE MAXIMUM NODAL DIFFERENCE FOR
C ------- EACH ELEMENT
C
      MMP=0
  300 READ 10, MI,NSEQ,MIAD,(IEM(I),I=1,8),IEMAD
      IF(MI.EQ.0) GO TO 330
C
      MJ=MI+NSEQ
      DO 320 MP=MI,MJ
```

```
      K=MI+(MP-MI)*MIAD
      DO 310 IQ=1,8
      NI=IEM(IQ)+(MP-MI)*IEMAD
  310 IE(M,IQ)=NI
      MMP=MMP+1
  320 CONTINUE
      GO TO 300
C
  330 IF(MMP.EQ.NEL) GO TO 350
      PRINT 3000, MMP,NEL
      STOP
C
  350 DO 360 M=1,NEL
  360 IE(M,9)=1
C
C ******* DATA SET 10: MATERIAL CORRECTIONS
C
      IF (NCM.LE.0) GO TO 405
      CALL READN(IE(1,9),MAXEL,NCM)
C
C ------- PRINT ELEMENT INCIDENCE AND MATERIAL TYPES FOR EACH ELEMENT
C
  405 CONTINUE
      IF(MOD(IGEOM,2).EQ.0) GO TO 415
      LINE=0
      DO 410 NI=1,NEL,2
      NJMN=NI
      NJMX=MIN0(NI+1,NEL)
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 2700
      PRINT 3100, (NJ,(IE(NJ,K),K=1,9),NJ=NJMN,NJMX)
  410 CONTINUE
  415 CONTINUE
C
C ------- CHECK IF MATERIAL TYPE FOR EACH ELEMENT IS CORRECT
C
      DO 420 M=1,NEL
      MTYP=IE(M,9)
      IF(MTYP.GT.0 .AND. MTYP.LE.NMAT) GO TO 420
      PRINT 4200, M
      ISTOP=ISTOP+1
  420 CONTINUE
C
      IF(ISTOP.EQ.0) GO TO 430
      PRINT 4300, ISTOP
      STOP
C
C ------- IDENTIFY BOUNDARY ELEMENTS AND COMPUTE DIRECTIONAL COSINES
C
  430 CONTINUE
```

```
      IF(IGEOM.LE.1) CALL SURF(X,Y,Z,IE, DCOSB,ISB,NPBB)
      REWIND 3
      IF(IGEOM.LE.1) WRITE(3) NBES,NBNP,((DCOSB(J,I),J-1,3),I-1,NBES),
     1 ((ISB(J,I),J-1,6),I-1,NBES),(NPBB(I),I-1,NBNP)
      IF(IGEOM.GT.1) READ(3) NBES,NBNP,((DCOSB(J,I),J-1,3),I-1,NBES),
     1 ((ISB(J,I),J-1,6),I-1,NBES),(NPBB(I),I-1,NBNP)
C
C ******* DATA SET 11: INITIAL CONDITIONS
C
C
C ------- READ INITIAL OR PRE-INITIAL CONDITIONS VIA CARDS
C
      CALL READR(H,MAXNP,NNP)
C
C ******* DATA SET 12: INTEGERS CONTROLLING SOURCES AND B.C.
C
      READ 10, NSEL,NSPR,NSDP, NWNP,NWPR,NWDP, NCES,NCNP,NCPR,NCDP,
     1 NNES,NNNP,NNPR,NNDP, NVES,NVNP,NRPR,NRDP, NDNP,NDPR,NDDP
C
      PRINT 5100, NSEL,NSPR,NSDP, NWNP,NWPR,NWDP
      PRINT 5150, NCES,NCNP,NCPR,NCDP,NNES,NNNP,NNPR,NNDP,
     1 NVES,NVNP,NRPR,NRDP, NDNP,NDPR,NDDP
C
C ******* DATA SET 13: SOURCE DATA
C
      IF(NSEL.EQ.0) GO TO 560
      PRINT 5300
      DO 510 I-1,NSPR
      READ 20, (TSOSF(J,I),SOSF(J,I),J-1,NSDP)
      PRINT 5500, I
      PRINT 5510, (TSOSF(J,I),SOSF(J,I),J-1,NSDP)
  510 CONTINUE
C
C ------- READ SOURCE TYPE ASSIGNED TO EACH ELEMENT
C
      READ 10, (MSEL(M),M-1,NSEL)
      CALL READN(ISTYP,MXSEL,NSEL)
C
C ------- PRINT ELEMENT SOURCE/SINK PROFILES AND TYPE
C
      LINE-0
      DO 520 I-1,NSEL,5
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ 0) PRINT 5600
      NJMN-I
      NJMX-MIN0(I+4,NSEL)
      PRINT 5650, (J,MSEL(J),ISTYP(J),J-NJMN,NJMX)
  520 CONTINUE
```

```
C                                                                       DATA1495
C -------  READ AND WRITE WELL SOURCE/SINK PROFILES                      DATA1500
C                                                                       DATA1505
  560 IF(NWNP.EQ.0) GO TO 600                                           DATA1510
      PRINT 5700                                                        DATA1515
      DO 570 I-1,NWPR                                                   DATA1520
      READ 20, (TWSSF(J,I),WSSF(J,I),J-1,NWDP)                          DATA1525
      PRINT 5710, I                                                     DATA1530
      PRINT 5510, (TWSSF(J,I),WSSF(J,I),J-1,NWDP)                       DATA1535
  570 CONTINUE                                                          DATA1540
C                                                                       DATA1545
C ------- READ WELL SOURCE/SINK NODES AND TYPE OF PROFILES ASSIGNED TO  DATA1550
C ------- EACH OF NWNP NODES.                                           DATA1555
C                                                                       DATA1560
      READ 10, (NPW(I),I-1,NWNP)                                        DATA1565
      CALL READN(IWTYP,MXWNP,NWNP)                                      DATA1570
C                                                                       DATA1575
C ------- PRINT GLOBAL WELL NODE NUMBERS AND PROFILE TYPE OF WELL NODE  DATA1580
C                                                                       DATA1585
      LINE-0                                                            DATA1590
      DO 590 I-1,NWNP,5                                                 DATA1595
      LINE-LINE+1                                                       DATA1600
      IF(MOD(LINE-1,50).EQ.0) PRINT 5800                                DATA1605
      NJMN-I                                                            DATA1610
      NJMX-MIN0(I+4,NWNP)                                               DATA1615
      PRINT 5850, (J,NPW(J),IWTYP(J),J-NJMN,NJMX)                       DATA1620
  590 CONTINUE                                                          DATA1625
C                                                                       DATA1630
C ******* DATA SET 14: RAINFALL/EVAPORATION-SEEPAGE BOUNDARY CONDITIONS DATA1635
C                                                                       DATA1640
  600 IF(NVES.EQ.0) GO TO 700                                           DATA1645
C                                                                       DATA1650
      PRINT 6000                                                        DATA1655
C                                                                       DATA1660
C ------- READ AND WRITE RAINFALL (+)/EVAPORATION (-) PROFILES          DATA1665
C                                                                       DATA1670
      PRINT 6100                                                        DATA1675
      DO 610 I-1,NRPR                                                   DATA1680
         READ 20, (TRF(J,I),RF(J,I),J-1,NRDP)                           DATA1685
      PRINT 6150, I                                                     DATA1690
      PRINT 5510, (TRF(J,I),RF(J,I),J-1,NRDP)                           DATA1695
  610 CONTINUE                                                          DATA1700
C                                                                       DATA1705
C ------- READ RAINFALL/EVAPORATION TYPE ASSIGNED TO EACH RS SIDE       DATA1710
C                                                                       DATA1715
      CALL READN(IRTYP,MXVES,NVES)                                      DATA1720
C                                                                       DATA1725
C ------- READ FOUR GLOBAL NODE NUMBER FOR EACH OF ALL VARIABLE         DATA1730
C ------- BOUNDARY ELEMENT SIDES.                                       DATA1735
C                                                                       DATA1740
```

```
      MPI=0
  620 READ 10, MI,NSEQ,MIAD,I1,I2,I3,I4,I1AD,I2AD,I3AD,I4AD
      IF(MI.EQ.0) GO TO 630
      MJ=MI+NSEQ
      DO 625 MP=MI,MJ
      I=MI+(MP-MI)*MIAD
      ISV(1,I)=I1+(MP-MI)*I1AD
      ISV(2,I)=I2+(MP-MI)*I2AD
      ISV(3,I)=I3+(MP-MI)*I3AD
      ISV(4,I)=I4+(MP-MI)*I4AD
      MPI=MPI+1
  625 CONTINUE
      GO TO 620
  630 IF(MPI.EQ.NVES) GO TO 635
      PRINT 6300
      STOP
C
C ------- PRINT INPUTTED GLOBAL NODAL NUMBER AND RAINFALL TYPES OF ALL
C ------- VARIABLE BOUNDARY ELEMENT SIDES.
C
  635 LINE=0
      DO 640 MP=1,NVES,3
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 6400
      NJMN=MP
      NJMX=MIN0(MP+2,NVES)
      PRINT 6450, (J,(ISV(I,J),I=1,4),IRTYP(J),J=NJMN,NJMX)
  640 CONTINUE
C
C ------- READ GLOBAL NODAL NUMBER FOR EACH OF ALL VARIABLE NODES.
C
      CALL READN(NPVB,MXVNP,NVNP)
C
C ------- READ PONDING DEPTH AND MINIMUM HEAD FOR EACH OF ALL RS NODES
C
      CALL READR(HCON,MXVNP,NVNP)
      CALL READR(HMIN,MXVNP,NVNP)
C
C ------- PRINT GLOBAL NODAL NUMBER, PONDING DEPTH AND MINIMUM PRESURE
C ------- RESSURE HEAD FOR ALL VARIABLE BOUNDARY NODES
C
      LINE=0
      DO 645 I=1,NVNP,3
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 6500
      NJMN=I
      NJMX=MIN0(I+2,NVNP)
      PRINT 6550, (J,NPVB(J),HCON(J),HMIN(J),J=NJMN,NJMX)
  645 CONTINUE
```

```
C                                                                       DATA1990
C ------- COMPUTE BOUNDARY SIDE NUMBER FOR EACH OF ALL VARIABLE         DATA1995
C ------- BOUNDARY SIDES.                                               DATA2000
C                                                                       DATA2005
      DO 659 MI-1,NVES                                                  DATA2010
      DO 651 IQ-1,4                                                     DATA2015
  651 NIMI(IQ)-ISV(IQ,MI)                                               DATA2020
C                                                                       DATA2025
      DO 657 MJ-1,NBES                                                  DATA2030
      DO 652 JQ-1,4                                                     DATA2035
      IJ-ISB(JQ,MJ)                                                     DATA2040
  652 NJMJ(JQ)-NPBB(IJ)                                                 DATA2045
      IEQ-0                                                             DATA2050
      DO 656 IQ-1,4                                                     DATA2055
      NI-NIMI(IQ)                                                       DATA2060
      DO 653 JQ-1,4                                                     DATA2065
      NJ-NJMJ(JQ)                                                       DATA2070
      IF(NJ.EQ.NI) GO TO 655                                            DATA2075
  653 CONTINUE                                                          DATA2080
      GO TO 657                                                         DATA2085
  655 IEQ-IEQ+1                                                         DATA2090
  656 CONTINUE                                                          DATA2095
      IF(IEQ.EQ.4) GO TO 658                                            DATA2100
  657 CONTINUE                                                          DATA2105
C                                                                       DATA2110
      PRINT 6570, MI                                                    DATA2115
      STOP                                                              DATA2120
  658 ISV(5,MI)-MJ                                                      DATA2125
C                                                                       DATA2130
  659 CONTINUE                                                          DATA2135
C                                                                       DATA2140
C ------- CHANGE NPVB FROM CONTAINING GLOBAL NODAL NUMBER TO            DATA2145
C ------- CONTAINING BOUNDARY NODAL NUMBER.                             DATA2150
C                                                                       DATA2155
      DO 669 NP-1,NVNP                                                  DATA2160
      NI-NPVB(NP)                                                       DATA2165
C                                                                       DATA2170
      DO 665 I-1,NBNP                                                   DATA2175
      NJ-NPBB(I)                                                        DATA2180
      IF(NJ.NE.NI) GO TO 665                                            DATA2185
      NII-I                                                             DATA2190
      GO TO 667                                                         DATA2195
  665 CONTINUE                                                          DATA2200
C                                                                       DATA2205
      PRINT 6670, NP                                                    DATA2210
      STOP                                                              DATA2215
  667 NPVB(NP)-NII                                                      DATA2220
C                                                                       DATA2225
  669 CONTINUE                                                          DATA2230
```

```
C
C --------- PRINT COMPUTED BOUNDARY NODAL NUMBER FOR ALL VB NODES
C
      LINE-0
      DO 670 I-1,NVNP,10
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 6700
      NJMN-I
      NJMX-MINO(I+9,NVNP)
      PRINT 6750, (J,NPVB(J),J-NJMN,NJMX)
  670 CONTINUE
C
C ------- CHANGE ISV(I,MP) I-1,4 FROM CONTAINING GLOBAL NODAL
C ------- NUMBER TO CONTAINING BOUNDARY NODAL NUMBER.
C
      DO 690 MP-1,NVES
      MPB-ISV(5,MP)
      DO 685 IQ-1,4
      NB-ISB(IQ,MPB)
      DO 675 I-1,NVNP
      NI-NPVB(I)
      IF(NI.NE.NB) GO TO 675
      NII-I
      GO TO 680
  675 CONTINUE
      PRINT 6751, IQ,MP
      STOP
  680 ISV(IQ,MP)-NII
  685 CONTINUE
  690 CONTINUE
C
C ------- PRINT COMPUTED BOUNDARY NODAL NUMBER & SIDE NUMBER AND
C ------- RAINFALL TYPES FOR ALL VB SIDES
C
      LINE-0
      DO 695 MP-1,NVES,3
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 6900
      NJMN-MP
      NJMX-MINO(MP+2,NVES)
      PRINT 6950, (J,(ISV(I,J),I-1,5),IRTYP(I),J-NJMN,NJMX)
  695 CONTINUE
C
C ******* DATA SET 15: DIRICHLET BOUNDARY CONDITIONS
C
C ------- READ AND PRINT TOTAL DIRICHLET HEAD PROFILES
C
  700 IF(NDNP.EQ.0) GO TO 800
      PRINT 7000
      DO 710 I-1,NDPR
```

```
      READ 20, (THDBF(J,I),HDBF(J,I),J-1,NDDP)                          DATA2485
      PRINT 7100, I                                                     DATA2490
      PRINT 5510, (THDBF(J,I),HDBF(J,I),J-1,NDDP)                       DATA2495
  710 CONTINUE                                                          DATA2500
C                                                                       DATA2505
C ------- READ GLOBAL NODAL NUMBER OF ALL DIRICHLET NODES AND           DATA2510
C ------- THE TYPE OF TOTAL HEAD ASSIGNED TO EACH OF THEM.              DATA2515
C                                                                       DATA2520
      READ 10, (NPDB(I),I-1,NDNP)                                       DATA2525
      CALL READN(IDTYP,MXDNP,NDNP)                                      DATA2530
C                                                                       DATA2535
C --------- PRINT GLOBAL NODAL NUMBER AND PROFILE OF DIRICHLET NODES    DATA2540
C                                                                       DATA2545
      LINE-0                                                            DATA2550
      DO 720 I-1,NDNP,5                                                 DATA2555
      LINE-LINE+1                                                       DATA2560
      IF(MOD(LINE-1,50).EQ.0) PRINT 7200                                DATA2565
      NJMN-I                                                            DATA2570
      NJMX-MINO(I+4,NDNP)                                               DATA2575
      PRINT 7250, (J,NPDB(J),IDTYP(J),J-NJMN.NJMX)                      DATA2580
  720 CONTINUE                                                          DATA2585
C                                                                       DATA2590
C ------- CHANGE NPDB FROM CONTAINING GLOBAL NODAL NUMBER TO            DATA2595
C ------- CONTAINING BOUNDARY NODAL NUMBER.                             DATA2600
C                                                                       DATA2605
      DO 769 NP-1,NDNP                                                  DATA2610
      NI-NPDB(NP)                                                       DATA2615
C                                                                       DATA2620
      DO 765 I-1,NBNP                                                   DATA2625
      NJ-NPBB(I)                                                        DATA2630
      IF(NJ.NE.NI) GO TO 765                                            DATA2635
      NII-I                                                             DATA2640
      GO TO 767                                                         DATA2645
  765 CONTINUE                                                          DATA2650
C                                                                       DATA2655
      PRINT 7670, NP                                                    DATA2660
      STOP                                                              DATA2665
  767 NPDB(NP)-NII                                                      DATA2670
C                                                                       DATA2675
  769 CONTINUE                                                          DATA2680
C                                                                       DATA2685
C --------- PRINT COMPUTED BOUNDARY NODAL NUMBER AND TYPE OF PROFILES   DATA2690
C --------- FOR DIRICHLET BOUNDARY NODES                                DATA2695
C                                                                       DATA2700
      LINE-0                                                            DATA2705
      DO 770 I-1,NDNP,5                                                 DATA2710
      LINE-LINE+1                                                       DATA2715
      IF(MOD(LINE-1,50).EQ.0) PRINT 7700                                DATA2720
      NJMN-I                                                            DATA2725
      NJMX-MINO(I+4,NDNP)                                               DATA2730
```

```
      PRINT 7750, (J,NPDB(J),IDTYP(J),J=NJMN,NJMX)
  770 CONTINUE
C
C ******* DATA SET 16: CAUCHY BOUNDARY CONDITIONS
C
C ------- READ AND PRINT CAUCHY FLUX PROFILES
C
  800 IF(NCES.EQ.0) GO TO 900
      PRINT 8000
      DO 810 I=1,NCPR
      READ 20, (TQCBF(J,I),QCBF(J,I),J=1,NCDP)
      PRINT 8100, I
      PRINT 5510, (TQCBF(J,I),QCBF(J,I),J=1,NCDP)
  810 CONTINUE
C
C ------- READ CAUCHY FLUX TYPE ASSIGNED TO EACH CAUCHY SIDE
C
      CALL READN(ICTYP,MXCES,NCES)
C
C ------- READ FOUR GLOBAL NODE NUMBER FOR EACH OF ALL CAUCHY SIDES
C
      MPI=0
  820 READ 10, MI,NSEQ,MIAD,I1,I2,I3,I4,I1AD,I2AD,I3AD,I4AD
      IF(MI.EQ.0) GO TO 830
      MJ=MI+NSEQ
      DO 825 MP=MI,MJ
      I=MI+(MP-MI)*MIAD
      ISC(1,I)=I1+(MP-MI)*I1AD
      ISC(2,I)=I2+(MP-MI)*I2AD
      ISC(3,I)=I3+(MP-MI)*I3AD
      ISC(4,I)=I4+(MP-MI)*I4AD
      MPI=MPI+1
  825 CONTINUE
      GO TO 820
  830 IF(MPI.EQ.NCES) GO TO 835
      PRINT 8300
      STOP
C
C ------- PRINT INPUTTED GLOBAL NODAL NUMBER AND CAUCHY FLUX TYPES
C ------- FOR ALL CAUCHY BOUNDARY ELEMENT SIDES.
C
  835 LINE=0
      DO 840 MP=1,NCES,3
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 8400
      NJMN=MP
      NJMX=MIN0(MP+2,NCES)
      PRINT 8450, (J,(ISC(I,J),I=1,4),ICTYP(J),J=NJMN,NJMX)
  840 CONTINUE
```

```
C
C ------- READ GLOBAL NODAL NUMBER FOR EACH OF ALL CAUCHY NODES.
C
      READ 10, (NPCB(I),I=1,NCNP)
C
C ------- PRINT GLOBAL NODAL NUMBER FOR ALL CAUCHY NODES
C
      LINE=0
      DO 845 I=1,NCNP,10
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 8500
      NJMN=I
      NJMX=MIN0(I+9,NCNP)
      PRINT 8550, (J,NPCB(J),J=NJMN,NJMX)
  845 CONTINUE
C
C ------- COMPUTE BOUNDARY SIDE NUMBER FOR ALL CAUSHY SIDES
C
      DO 859 MI=1,NCES
      DO 851 IQ=1,4
  851 NIMI(IQ)=ISC(IQ,MI)
C
      DO 857 MJ=1,NBES
      DO 852 JQ=1,4
      IJ=ISB(JQ,MJ)
  852 NJMJ(JQ)=NPBB(IJ)
      IEQ=0
      DO 856 IQ=1,4
      NI=NIMI(IQ)
      DO 853 JQ=1,4
      NJ=NJMJ(JQ)
      IF(NJ.EQ.NI) GO TO 855
  853 CONTINUE
      GO TO 857
  855 IEQ=IEQ+1
  856 CONTINUE
      IF(IEQ.EQ.4) GO TO 858
  857 CONTINUE
C
      PRINT 8570, MI
      STOP
  858 ISC(5,MI)=MJ
C
  859 CONTINUE
C
C ------- CHANGE NPCB FROM CONTAINING GLOBAL NODAL NUMBER TO
C ------- CONTAINING BOUNDARY NODAL NUMBER.
C
      DO 869 NP=1,NCNP
      NI=NPCB(NP)
```

```
C
      DO 865 I-1,NBNP
      NJ-NPBB(I)
      IF(NJ.NE.NI) GO TO 865
      NII-I
      GO TO 867
  865 CONTINUE
C
      PRINT 8670, NP
      STOP
  867 NPCB(NP)-NII
C
  869 CONTINUE
C
C --------- PRINT COMPUTED BOUNDARY NODAL NUMBER FOR ALL CAUCHY NODES
C
      LINE-0
      DO 870 I-1,NCNP 10
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 8700
      NJMN-I
      NJMX-MIN0(I+9,NCNP)
      PRINT 8750, (J,NPCB(J),J-NJMN,NJMX)
  870 CONTINUE
C
C ******* DATA SET 17:  NEUMANN BOUNDARY CONDITIONS
C
C ------- READ AND PRINT NEUMANN FLUX PROFILES
C
  900 IF(NNES.EQ.0) GO TO 999
      PRINT 9000
      DO 910 I-1,NNPR
      READ 20, (TQNBF(J,I),QNBF(J,I),J-1,NNDP)
      PRINT 9100, I
      PRINT 5510, (TQNBF(J,I),QNBF(J,I),J-1,NNDP)
  910 CONTINUE
C
C ------- READ NEUMANN FLUX TYPE ASSIGNED TO EACH NEUMANN SIDE
C
      CALL READN(INTYP,MXNES,NNES)
C
C ------- READ FOUR GLOBAL NODE NUMBER FOR EACH OF ALL NEUMANN SIDES
C
      MPI-0
  920 READ 10, MI,NSEQ,MIAD,I1,I2,I3,I4,I1AD,I2AD,I3AD,I4AD
      IF(MI.EQ.0) GO TO 930
      MJ-MI+NSEQ
      DO 925 MP-MI,MJ
      I-MI+(MP-MI)*MIAD
      ISN(1,I)-I1+(MP-MI)*I1AD
```

```
      ISN(2,I)-I2+(MP-MI)*I2AD                                          DATA3480
      ISN(3,I)-I3+(MP-MI)*I3AD                                          DATA3485
      ISN(4,I)-I4+(MP-MI)*I4AD                                          DATA3490
      MPI-MPI+1                                                         DATA3495
  925 CONTINUE                                                          DATA3500
      GO TO 920                                                         DATA3505
  930 IF(MPI.EQ.NNES) GO TO 935                                         DATA3510
      PRINT 9300                                                        DATA3515
      STOP                                                              DATA3520
C                                                                       DATA3525
C ------- PRINT INPUTTED GLOBAL NODAL NUMBER AND NEUMANN FLUX TYPES     DATA3530
C ------- FOR ALL NEUMANN BOUNDARY ELEMENT SIDES.                       DATA3535
C                                                                       DATA3540
  935 LINE-0                                                            DATA3545
      DO 940 MP-1,NNES,3                                                DATA3550
      LINE-LINE+1                                                       DATA3555
      IF(MOD(LINE-1,50).EQ.0) PRINT 9400                                DATA3560
      NJMN-MP                                                           DATA3565
      NJMX-MIN0(MP+2,NNES)                                              DATA3570
      PRINT 9450, (J,(ISN(I,J),I-1,4),INTYP(J),J-NJMN,NJMX)             DATA3575
  940 CONTINUE                                                          DATA3580
C                                                                       DATA3585
C ------- READ GLOBAL NODAL NUMBER FOR EACH OF ALL NEUMANN NODES.       DATA3590
C                                                                       DATA3595
      READ 10, (NPNB(I),I-1,NNNP)                                       DATA3600
C                                                                       DATA3605
C ------- PRINT GLOBAL NODAL NUMBER FOR ALL NEUMANN NODES               DATA3610
C                                                                       DATA3615
      LINE-0                                                            DATA3620
      DO 945 I-1,NNNP,10                                                DATA3625
      LINE-LINE+1                                                       DATA3630
      IF(MOD(LINE-1,50).EQ.0) PRINT 9500                                DATA3635
      NJMN-I                                                            DATA3640
      NJMX-MIN0(I+9,NNNP)                                               DATA3645
      PRINT 9550, (J,NPNB(J),J-NJMN,NJMX)                               DATA3650
  945 CONTINUE                                                          DATA3655
C                                                                       DATA3660
C ------- COMPUTE BOUNDARY SIDE NUMBER FOR EACH OF NEUMANN              DATA3665
C ------- BOUNDARY SIDES.                                               DATA3670
C                                                                       DATA3675
      DO 959 MI-1,NNES                                                  DATA3680
      DO 951 IQ-1,4                                                     DATA3685
  951 NIMI(IQ)-ISN(IQ,MI)                                               DATA3690
C                                                                       DATA3695
      DO 957 MJ-1,NBES                                                  DATA3700
      DO 952 JQ-1,4                                                     DATA3705
      IJ-ISB(JQ,MJ)                                                     DATA3710
  952 NJMJ(JQ)-NPBB(IJ)                                                 DATA3715
      IEQ-0                                                             DATA3720
      DO 956 IQ-1,4                                                     DATA3725
```

```
      NI=NIMI(IQ)                                                       DATA3730
      DO 953 JQ=1,4                                                     DATA3735
      NJ=NJMJ(JQ)                                                       DATA3740
      IF(NJ.EQ.NI) GO TO 955                                            DATA3745
  953 CONTINUE                                                          DATA3750
      GO TO 957                                                         DATA3755
  955 IEQ=IEQ+1                                                         DATA3760
  956 CONTINUE                                                          DATA3765
      IF(IEQ.EQ.4) GO TO 958                                            DATA3770
  957 CONTINUE                                                          DATA3775
C                                                                       DATA3780
      PRINT 9570, MI                                                    DATA3785
      STOP                                                              DATA3790
  958 ISN(5,MI)=MJ                                                      DATA3795
C                                                                       DATA3800
  959 CONTINUE                                                          DATA3805
C                                                                       DATA3810
C ------- CHANGE NPNB FROM CONTAINING GLOBAL NODAL NUMBER TO            DATA3815
C ------- CONTAINING BOUNDARY NODAL NUMBER.                             DATA3820
C                                                                       DATA3825
      DO 969 NP=1,NNNP                                                  DATA3830
      NI=NPNB(NP)                                                       DATA3835
C                                                                       DATA3840
      DO 965 I=1,NBNP                                                   DATA3845
      NJ=NPBB(I)                                                        DATA3850
      IF(NJ.NE.NI) GO TO 965                                            DATA3855
      NII=I                                                             DATA3860
      GO TO 967                                                         DATA3865
  965 CONTINUE                                                          DATA3870
C                                                                       DATA3875
      PRINT 9670, NP                                                    DATA3880
      STOP                                                              DATA3885
  967 NPNB(NP)=NII                                                      DATA3890
C                                                                       DATA3895
  969 CONTINUE                                                          DATA3900
C                                                                       DATA3905
C --------- PRINT COMPUTED BOUNDARY NODAL NUMBER FOR ALL NEUMANN NODES  DATA3910
C                                                                       DATA3915
      LINE=0                                                            DATA3920
      DO 970 I=1,NNNP,10                                                DATA3925
      LINE=LINE+1                                                       DATA3930
      IF(MOD(LINE-1,50).EQ.0) PRINT 9700                                DATA3935
      NJMN=I                                                            DATA3940
      NJMX=MIN0(I+9,NNNP)                                               DATA3945
      PRINT 9750, (J,NPNB(J),J=NJMN,NJMX)                               DATA3950
  970 CONTINUE                                                          DATA3955
C                                                                       DATA3960
  999 CONTINUE                                                          DATA3965
C                                                                       DATA3970
      RETURN                                                            DATA3975
```

```
C                                                                       DATA3980
   10 FORMAT(16I5)                                                      DATA3985
   20 FORMAT(8D10.3)                                                    DATA3990
   30 FORMAT(80I1)                                                      DATA3995
   40 FORMAT(3I5,5X,6D10.3)                                             DATA4000
C                                                                       DATA4005
 1000 FORMAT(40H0      **** BAISC INTEGER PARAMETERS ****//5X,          DATA4010
     1 40H NUMBER OF NODAL POINTS. . . . . . . . .,I5/ 5X,              DATA4015
     2 40H NUMBER OF ELEMENTS. . . . . . . . . . .,I5/ 5X,              DATA4020
     3 40H NUMBER OF DIFFERENT MATERIALS . . . . .,I5/ 5X,              DATA4025
     4 40H NUMBER OF CORRECTION MATERIALS. . . . .,I5/ 5X,              DATA4030
     5 40H NUMBER OF TIME INCREMENTS . . . . . . .,I5//5X,              DATA4035
     6 40H STEADY-STATE I.C. CONTROL . . . . . . .,I5/ 5X,              DATA4040
     8 40H NUMBER OF SOIL PARAMETERS . . . . . . .,I5/ 5X,              DATA4045
     9 40H NUMBER OF MATERIAL PROPERTIES . . . . .,I5//5X,              DATA4050
     A 40H AUXILIARY STORAGE CONTROL . . . . . . .,I5/ 5X,              DATA4055
     B 40H CONDUCTIVITY-PERMEABILITY CONTROL . . .,I5/ 5X,              DATA4060
     C 40H GRAVITY CONTROL . . . . . . . . . . . .,I5/ 5X,              DATA4065
     E 40H NO. OF ITERATIONS PER CYCLE . . . . . .,I5/ 5X,              DATA4070
     F 40H NO. OF CYCLES PER TIME STEP . . . . . .,I5/ 5X,              DATA4075
     G 40H NO. OF TIMES TO RESET TIME STEP SIZE  .,I5/ 5X,              DATA4080
     H 40H NO. OF BLOCKWISE ITERATIONS ALLOWED .  ,I5/ 5X,              DATA4085
     I 40H NO. OF SUBREGIONS .. . . . . . . . . . ,I5/)                 DATA4090
 1100 FORMAT(5X,40H TIME INCREMENT. . . . . . . . . . . . . .,E15.6/ 5X, DATA4095
     1 40H MULTIPLIER FOR INCREASING DELT. . . . .,E15.6/ 5X,           DATA4100
     2 40H MAXIMUM VALUE OF DELT . . . . . . . . .,E15.6/ 5X,           DATA4105
     3 40H MAXIMUM VALUE OF TIME . . . . . . . . .,E15.6//5X,           DATA4110
     5 40H STEADY-STATE TOLERANCE. . . . . . . . .,E15.6/ 5X,           DATA4115
     6 40H TRANSIENT-STATE TOLERANCE . . . . . . .,E15.6//5X,           DATA4120
     7 40H DENSITY OF WATER. . . . . . . . . . . .,E15.6/ 5X,           DATA4125
     8 40H ACCELERATION OF GRAVITY . . . . . . . .,E15.6/ 5X,           DATA4130
     9 40H VISCOSITY OF WATER. . . . . . . . . . .,E15.6//5X,           DATA4135
     A 40H TIME-INTEGRATION PARAMETER. . . . . . .,E15.6/ 5X,           DATA4140
     B 40H ITERATION PARAMETER FOR NONLINEAR EQ.  ,E15.6/ 5X,           DATA4145
     C 40H RELAXATION PARAMETER FOR POINTWISE SOL.,E15.6//)             DATA4150
 1200 FORMAT(//6X,14HOUTPUT CONTROL)                                    DATA4155
 1210 FORMAT(10X,50I2)                                                  DATA4160
 1300 FORMAT(//6X,19HDISK OUTPUT CONTROL)                               DATA4165
 1350 FORMAT(1H0,6X,'TIME OF CHANGING DELT'/(10X,8D12.4))               DATA4170
 1400 FORMAT(36H1      **** MATERIAL PROPERTIES **** // 9H MAT. NO., 9( DATA4175
     > 3A4))                                                            DATA4180
 1410 FORMAT(I8,9D12.4)                                                 DATA4185
 1500 FORMAT(44H1INPUT TABLE 3.  MOISTURE-CONTENT PARAMETERS//          DATA4190
     > 9H MAT. NO.,8(3A4))                                              DATA4195
 1510 FORMAT(I8,9D12.4/(8X,9D12.4))                                     DATA4200
 1600 FORMAT(40H1      **** CONDUCTIVITY PARAMETERS **** // 9H MAT. NO., DATA4205
     > 8(3A4))                                                          DATA4210
 2000 FORMAT(1H0/5X,'*** ERROR IN READING COORDINATE STOP ***'/)        DATA4215
```

```
2100 FORMAT(38H1       **** NODAL COORDINATE DATA **** //1X,              DATA4220
    > 3(1X.5H NODE,11H     X      ,11H     Y      ,11H     Z      ,1X)/1X,DATA4225
    > 3(1X,5H ****,33H *********  *********  ********* ,1X))              DATA4230
2110 FORMAT(1H ,3(1X,I5,3D11.3,1X))                                      DATA4235
2200 FORMAT(1H1/5X,25H *** OUTPUT GNLR(I,K) ***)                         DATA4240
2210 FORMAT(1H0,5X,26H ---- SUBREGION NUMBER K =,I4/(6X,10I5))           DATA4245
2700 FORMAT(62H1       **** ELEMENT DATA: GLOBAL INDICES OF ELEMENT NODESDATA4250
    1 **** //2(5X,5H  ELM,5H NOD1,5H NOD2,5H NOD3,5H NOD4,5H NOD5,       DATA4255
    2 5H NOD6,5H NOD7,5H NOD8,5H MTYP)/2(5X,5H  ---,5H ----,5H ----,     DATA4260
    3 5H ----,5H ----,5H ----,5H ----,5H ----,5H ----,5H ----))          DATA4265
3000 FORMAT(////'ERROR IN READING IE, MMP =',I5,' NEL =',I5,' STOP')     DATA4270
3100 FORMAT(2(5X,10I5))                                                  DATA4275
4200 FORMAT(////40H ERROR IN MATERIAL TYPE CODE FOR ELEMENT,I5///)       DATA4280
4300 FORMAT(////28H EXECUTION HALTED BECAUSE OF,I5,13H FATAL ERRORS///)DATA4285
5100 FORMAT('1       **** TRANSIENT INTEGERS **** '// 5X,                DATA4290
    1 ' NO. OF SOURCE ELEMENTS . . . . . . . . . ',I5/ 5X,               DATA4295
    2 ' NO. OF SOURCE PROFILES  . . . . . . . . .',I5/ 5X,               DATA4300
    3 ' NO. OF DATA POINTS FOR EACH SOURCE PROF',I5/ 5X,                 DATA4305
    4 ' NO. OF WELL SOURCES/SINKS NODES . . .  ',I5/ 5X,                 DATA4310
    5 ' NO. OF WELL SOURCE PROFILES . . . . . .',I5/ 5X,                 DATA4315
    6 ' NO. OF DATA POINTS IN EACH WELL PROF. .',I5/)                    DATA4320
5150 FORMAT(1H /5X,                                                      DATA4325
    1 40H NO. OF CAUCHY SIDES . . . . . . . . . . .,I5/ 5X,              DATA4330
    2 40H NO. OF CAUCHY NODES . . . . . . . . . . .,I5/ 5X,              DATA4335
    3 40H NO. OF CAUCHY FLUX PROFILES . . . . . . .,I5/ 5X,              DATA4340
    4 40H NO. OF DATA POINTS IN EACH CAUCHY PROF.,I5/ 5X,                DATA4345
    5 40H NO. OF NEUMANN SIDES . . . . . . . . . . ,I5/ 5X,              DATA4350
    6 40H NO. OF NEUMANN NODES . . . . . . . . . . ,I5/ 5X,              DATA4355
    7 40H NO. OF NEUMANN FLUXES . . . . . . . . . .,I5/ 5X,              DATA4360
    8 40H NO. OF DATA POINTS IN NEUMANN PROF. . .,I5/ 5X,                DATA4365
    9 40H NO. OF VARIABLE BOUNDARY SIDES. . . . .,I5/ 5X,                DATA4370
    A 40H NO. OF VARIABLE BOUNDARY NODES. . . . .,I5/ 5X,                DATA4375
    B 40H NO. OF RAINFALL PROFILES . . . . . . . ,I5/ 5X,                DATA4380
    C 40H NO. OF DATA POINTS IN RAINFALL PROF. . ,I5/ 5X,                DATA4385
    D 40H NO. OF DIRICHLET NODES . . . . . . . . . ,I5/ 5X,              DATA4390
    E 40H NO. OF DIRICHLET TOTAL HEAD PROF. . . .,I5/ 5X,                DATA4395
    F 40H NO. OF DATA POINTS IN DIRICHLET PROF. .,I5/)                   DATA4400
5300 FORMAT(1H1/5X,27H *** SOURCE INFORMATION ***)                       DATA4405
5500 FORMAT(1H0/5X,12H PROFILE NO.,I2,/ 5(4X,4HTIME,6X,6HSOURCE,2X)/     DATA4410
    > 5(4X,4H----,6X,6H------,2X))                                       DATA4415
5510 FORMAT(1H .5(2D11.3))                                               DATA4420
5600 FORMAT(1H0//10X,65H ELEMENT NUMBER AND PROFILE TYPES OF ELEMENT     DATA4425
    1                     //5X,5(5H    I,5H MSEL,5H STYP,5X))            DATA4430
5650 FORMAT(1H ,4X,5(3I5,5X))                                            DATA4435
5700 FORMAT(1H0///5X,37H *** WELL SOURCE/SINK INFORMATION ***)           DATA4440
5710 FORMAT(1H0/5X,12H PROFILE NO.,I2/ 4(4X,4HTIME,6X,6HSOURCE,2X)/      DATA4445
    1  4(4X,4H----,6X,6H -----,2X))                                      DATA4450
5800 FORMAT(1H0//10X,65H GLOBAL NODAL NUMBER AND PROFILE TYPE OF WELLS DATA4455
    1OUCE/SINK NODES   //5X,5(5H    I,5H  NPW,5H WTYP,5X))              DATA4460
```

```
 5850 FORMAT(1H ,4X,5(3I5,5X))                                              DATA4465
 6000 FORMAT(1H1/5X,46H **** RAINFALL-SEEPAGE BOUNDARY CONDITIONS ***)   DATA4470
 6100 FORMAT(1H0///10X,25H --- RAINFALL PROFILE ---)                       DATA4475
 6150 FORMAT(1H0/5X,12H PROFILE NO.,I2,//5(4X,4HTIME,6X,6H RAINS,2X)/   DATA4480
     > 5(4X,4H----,6X,6H------,2X))                                        DATA4485
 6300 FORMAT(1H0,10X,61H *** ERROR IN READING RAINFALL-SEEPAGE ELEMENT  DATA4490
     1SIDE STOP ***)                                                       DATA4495
 6400 FORMAT(1H0/10X,36H --- INPUTTED VARIABLE SIDE DATA ---//5X,       DATA4500
     2 3(5H   MP,5H  GN1,5H  GN2,5H  GN3,5H  GN4,5H RTYP,5X)/5X,        DATA4505
     3 3(30H   --  ---  ---  ---  --- ----,5X))                            DATA4510
 6450 FORMAT(1H ,4X,3(6I5,5X))                                             DATA4515
 6500 FORMAT(1H0/10X,36H --- INPUTTED VARIABLE NODE DATA ---//1X,       DATA4520
     1 3(1X,5H    I,5H NPVB,12H     HCON    ,12H     HMIN    ,1X)/1X,   DATA4525
     2 3(1X,5H    -,5H ----,12H     ----    ,12H     ----    ,1X))      DATA4530
 6550 FORMAT(1H ,3(1X,2I5,2D12.4,1X))                                      DATA4535
 6570 FORMAT(1H1/5X,44H CANNOT FIND A BOUNDARY SIDE COINCIDING WITH,    DATA4540
     1 I3,36H-TH VARIABLE BOUNDARY SIDE: STOP ***)                        DATA4545
 6670 FORMAT(1H1/5X,44H *** CANNOT FIND A BOUNDARY NODAL NUMBER FOR,    DATA4550
     1 I3,31H-TH VARIABLE BOUNDARY NODE STOP)                             DATA4555
 6700 FORMAT(1H0//10X,47H COMPUTED BOUNDARY NODAL NUMBER OF ALL VB NODESDATA4560
     1 //5X,10(5H    I,5H NPVB,2X)/5X,10(5H    -,5H ----,2X))            DATA4565
 6750 FORMAT(1H ,4X,10(2I5,2X))                                            DATA4570
 6751 FORMAT(1H0,5X,42H *** CAN NOT FIND A COMPRESSED RS NODE FOR,      DATA4575
     1 I2,12H-TH POINT OF,I4,20H-TH RS SIDE STOP ***)                     DATA4580
 6900 FORMAT(1H0/10X,30H --- COMPUTED VB SIDE DATA ---,//1X,             DATA4585
     1 3(5H   MP,5H CNP1,5H CNP2,5H CNP3,5H CNP4,5H  MPB,5H RTYP,1X)/1X,DATA4590
     2 3(5H   --,5H ----,5H ----,5H ----,5H ----,5H ----,5H ----,1X))   DATA4595
 6950 FORMAT(1H ,3(7I5,1X))                                                DATA4600
 7000 FORMAT(1H1/5X,40H **** DIRICHLET BOUNDARY CONDITIONS ****)         DATA4605
 7100 FORMAT(1H0/5X,12H PROFILE NO.,I2,/ 4(4X,4HTIME,6X,6H HEAD ,2X)/   DATA4610
     > 4(4X,4H----,6X,6H ---- ,2X))                                        DATA4615
 7200 FORMAT(1H0//10X,65H GLOBAL NODAL NUMBER AND PROFILE TYPE OF DIRICHDATA4620
     1LET BOUNDARY NODES//5X,5(5H    I,5H NPDB,5H TYPE,5X)/              DATA4625
     1 5X,5(5H    -,5H ----,5H ----,5X))                                   DATA4630
 7250 FORMAT(1H ,4X,5(3I5,5X))                                              DATA4635
 7670 FORMAT(1H1/5X,44H *** CANNOT FIND A BOUNDARY NODAL NUMBER FOR,    DATA4640
     1 I3,32H-TH DIRICHLET BOUNDARY NODE STOP)                            DATA4645
 7700 FORMAT(1H0//10X,66H COMPUTED BOUNDARY NODAL NUMBER & TYPE OF DIRICDATA4650
     1HLET BOUNDARY NODES//5X,5(5H    I,5H NPDB,5H TYPE,5X)/             DATA4655
     1 5X,5(5H    -,5H ----,5H ----,5X))                                   DATA4660
 7750 FORMAT(1H ,4X,5(3I5,5X))                                              DATA4665
 8000 FORMAT(1H1/5X,38H **** CAUCHY  BOUNDARY CONDITIONS ****)           DATA4670
 8100 FORMAT(1H0/5X,12H PROFILE NO.,I2/ 4(4X,4HTIME,6X,6H FLUX ,2X)/    DATA4675
     > 4(4X,4H----,6X,6H ---- ,2X))                                        DATA4680
 8300 FORMAT(1H0,10X,61H *** ERROR IN READING  CAUCHY  BOUNDARY ELEMEN  DATA4685
     1T SIDE STOP *)                                                       DATA4690
 8400 FORMAT(1H0/10X,34H --- INPUTTED CAUCHY SIDE DATA ---//5X,         DATA4695
     1 3(5H   MP,5H  GN1,5H  GN2,5H  GN3,5H  GN4,5H CTYP,5X)/5X,        DATA4700
     3 3(30H   --  ---  ---  ---  --- ----,5X))                            DATA4705
```

```
 8450 FORMAT(1H ,4X,3(6I5,5X))                                        DATA4710
 8500 FORMAT(1H0/10X,34H --- INPUTTED CAUCHY NODE DATA ---//5X,       DATA4715
     1 10(5H     I,5H NPCB,2X)/5X,10(5H     -,5H ----,2X))            DATA4720
 8550 FORMAT(1H ,4X,10(2I5,2X))                                       DATA4725
 8570 FORMAT(1H1/5X,44H CANNOT FIND A BOUNDARY SIDE COINCIDING WITH,  DATA4730
     1 I3,36H-TH CAUCY BOUNDARY SIDE:     STOP ***)                   DATA4735
 8670 FORMAT(1H1/5X,44H *** CANNOT FIND A BOUNDARY NODAL NUMBER FOR,  DATA4740
     1 I3,31H-TH CAUCHY BOUNDARY NODE:  STOP)                         DATA4745
 8700 FORMAT(1H0/10X,34H --- COMPUTED CAUCHY NODE DATA ---//5X,       DATA4750
     1 10(5H     I,5H NPCB,2X)/5X,10(5H     -,5H ----,2X))            DATA4755
 8750 FORMAT(1H ,4X,10(2I5,2X))                                       DATA4760
 9000 FORMAT(1H1/5X,38H **** NEUMANN BOUNDARY CONDITIONS ****)        DATA4765
 9100 FORMAT(1H0/5X,12H PROFILE NO.,I2/ 4(4X,4HTIME,6X,6H FLUX ,2X)/  DATA4770
     > 4(4X,4H----,6X,6H ---- ,2X))                                   DATA4775
 9300 FORMAT(1H0,10X,61H *** ERROR IN READING  NEUMANN BOUNDARY ELEMEN DATA4780
     >T SIDE STOP *)                                                  DATA4785
 9400 FORMAT(1H0/10X,35H --- INPUTTED NEUMANN SIDE DATA ---//5X,      DATA4790
     1 3(5H   MP,5H  GN1,5H  GN2,5H  GN3,5H  GN4,5H NTYP,5X)/5X,      DATA4795
     2 3(30H   --  ---  ---  ---  ---  ---,5X))                       DATA4800
 9450 FORMAT(1H ,4X,3(6I5,5X))                                        DATA4805
 9500 FORMAT(1H0/10X,35H --- INPUTTED NEUMANN NODE DATA ---//5X,      DATA4810
     1 10(5H     I,5H NPNB,2X)/5X,10(5H     -,5H ----,2X))            DATA4815
 9550 FORMAT(1H ,4X,10(2I5,2X))                                       DATA4820
 9570 FORMAT(1H1/5X,44H CANNOT FIND A BOUNDARY SIDE COINCIDING WITH,  DATA4825
     1 I3,37H-TH NEUMANN BOUNDARY SIDE:     STOP ***)                 DATA4830
 9670 FORMAT(1H1/5X,44H *** CANNOT FIND A BOUNDARY NODAL NUMBER FOR,  DATA4835
     1 I3,31H-TH NEUMANN BOUNDARY NODE: STOP)                         DATA4840
 9700 FORMAT(1H0/10X,35H --- COMPUTED NEUMANN NODE DATA ---//5X,      DATA4845
     1 10(5H     I,5H NPNB,2X)/5X,10(5H     -,5H ----,2X))            DATA4850
 9750 FORMAT(1H ,4X,10(2I5,2X))                                       DATA4855
C                                                                     DATA4860
      END                                                             DATA4865


      SUBROUTINE SURF(X,Y,Z,IE, DCOSB,ISB,NPBB)                       SURF 005
C                                                                     SURF 010
C********1*********2*********3*********4*********5*********6*********7**SURF 015
C                                                                     SURF 020
C ------- TO GENERATE BOUNDARY GEOMETRY.                              SURF 025
C                                                                     SURF 030
C********1*********2*********3*********4*********5*********6*********7**SURF 035
C                                                                     SURF 040
C ------- INPUT: X(NNP), Y(NNP), Z(NNP), IE(NEL,9).                   SURF 045
C                                                                     SURF 050
C ------- OUTPUT: DCOSB(3,NBES), ISB(6,NBES), NPBB(NBNP), NBES, NBNP.  SURF 055
C                                                                     SURF 060
C *******1*********2*********3*********4*********5*********6*********7**SURF 065
C                                                                     SURF 070
```

```
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXDTCH
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KCRAV,NTI,NDTCHG
      COMMON /CINTE/ NCYL,NITER,KSTR,KPRO,KDSKO,KSS,NPITER,IGEOM
C
      DIMENSION X(MAXNP),Y(MAXNP),Z(MAXNP),IE(MAXEL,9)
      DIMENSION DCOSB(3,MAXBES),ISB(6,MAXBES),NPBB(MAXBNP)
C
      DIMENSION IEMI(4),IEMJ(4)
C
      DIMENSION KGB(4,6)
C
      DATA KGB/1,4,8,5, 1,2,6,5, 2,3,7,6, 4,3,7,8, 1,2,3,4, 5,6,7,8/
C
      NBES=0
      NBNP=0
      DO 390 MI=1,NEL
      DO 380 LS=1,6
C
C ------- STORE FOUR GLOBAL NODAL NUMBERS OF LS-TH SIDE OF MI-TH
C ------- ELEMENT IN ARRAY IEMI(4) FOR LATER USE.
C
      DO 120 IQ=1,4
      K=KGB(IQ,LS)
      IEMI(IQ)=IE(MI,K)
  120 CONTINUE
C
C ------- CHECK IF THE LS-TH SIDE OF ELEMENT MI IS A BOUNDARY
C ------- SIDE BY LOOPING OVER EACH OF THE SIX SIDES OF ALL ELEMENTS
C ------- OTHER THAN THE ELEMENT MI.
C
      DO 290 MJ=1,NEL
      IF(MJ.EQ.MI) GO TO 290
C
      DO 280 J=1,6
C
C ------- STORE THE FOUR GLOBAL NODAL NUMBERS OF J-TH SIDE OF ELEMENT
C ------- MJ IN ARRAY IEMJ(4).
C
      DO 140 JQ=1,4
      K=KGB(JQ,J)
      IEMJ(JQ)=IE(MJ,K)
  140 CONTINUE
C
C ------- CHECK IF IEMI(4) CONTAINS THE SAME FOUR GLOBAL NODES AS
C ------- IEMJ(4).
C
      IEQ=0
```

```
      DO 180 IQ-1,4
      NI-IEMI(IQ)
      DO 160 JQ-1,4
      NJ-IEMJ(JQ)
      IF(NJ.EQ.NI) GO TO 170
  160 CONTINUE
      GO TO 180
  170 IEQ-IEQ+1
  180 CONTINUE
C
C ------- IF IEQ.EQ.4, THEN IEMJ(4) CONTAINS THE SAME FOUR GLOBAL
C ------- NODES AS IEMI(4).  THUS THE LS-TH SIDE OF ELEMENT MI IS NOT
C ------- A BOUNDARY SIDE.  HENCE GO TO 380 TO START A NEW SIDE.
C
      IF(IEQ.EQ.4) GO TO 380
C
  280 CONTINUE
  290 CONTINUE
C
C ------- AFTER LOOPING OVER J FROM 1 TO 6 AND MJ FROM 1 TO NEL, WE
C ------- STILL CANNOT FIND A SIDE CONTAINING THE SAME FOUR GLOBAL
C ------- NODES AS THE LS-TH SIDE OF ELEMENT MI.  THUS THE LS-TH SIDE
C ------- OF ELEMENT MI IS A BOUNDARY SIDE.
C
      NBES-NBES+1
      ISB(5,NBES)-LS
      ISB(6,NBES)-MI
C
C ------- COMPUTE DIRECTIONAL COSINES FOR THE NBES-TH SIDE.
C
      K-KGB(1,LS)
      NI-IE(MI,K)
      K-KGB(2,LS)
      NJ-IE(MI,K)
      A1-X(NJ)-X(NI)
      A2-Y(NJ)-Y(NI)
      A3-Z(NJ)-Z(NI)
      K-KGB(3,LS)
      NJ-IE(MI,K)
      B1-X(NJ)-X(NI)
      B2-Y(NJ)-Y(NI)
      B3-Z(NJ)-Z(NI)
      AB23-A2*B3-A3*B2
      AB31-A3*B1-A1*B3
      AB12-A1*B2-A2*B1
      AREA-DSQRT(AB23*AB23+AB31*AB31+AB12*AB12)
      DCOSB(1,NBES)-AB23/AREA
      DCOSB(2,NBES)-AB31/AREA
      DCOSB(3,NBES)-AB12/AREA
```

```
      IF(LS.EQ.2 .OR. LS.EQ.3 .OR. LS.EQ.6) GO TO 305
      DCOSB(1,NBES)--DCOSB(1,NBES)
      DCOSB(2,NBES)--DCOSB(2,NBES)
      DCOSB(3,NBES)--DCOSB(3,NBES)
C
  305 IF(NBES.GT.1) GO TO 320
      DO 310 IQ-1,4
      K-KGB(IQ,LS)
      NI-IE(MI,K)
      NBNP-NBNP+1
      NPBB(NBNP)-NI
  310 CONTINUE
  320 DO 340 IQ-1,4
      K-KGB(IQ,LS)
      NI-IE(MI,K)
      DO 330 IJ-1,NBNP
      NJ-NPBB(IJ)
      IF(NJ.EQ.NI) GO TO 340
  330 CONTINUE
      NBNP-NBNP+1
      NPBB(NBNP)-NI
  340 CONTINUE
C
  380 CONTINUE
  390 CONTINUE
C
C ------ PRINT BOUNDARY NODE INFORMATION
C
      IF(MOD(IGEOM,2).EQ.0) GO TO 396
      LINE-0
      DO 395 I-1,NBNP,10
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 3900
      NJMN-I
      NJMX-MIN0(I+9,NBNP)
      PRINT 3950, (J,NPBB(J),J-NJMN,NJMX)
  395 CONTINUE
  396 CONTINUE
C
C ------- COMPUTE THE COMPRESSED BOUNDARY NODE NUMBER FOR EACH OF THE
C ------- FOUR NODES OF A BOUNDARY SIDE
C
      DO 490 MP-1,NBES
      LS-ISB(5,MP)
      M-ISB(6,MP)
      DO 460 IQ-1,4
      I-KGB(IQ,LS)
      NI-IE(M,I)
```

```
      DO 420 NB-1,NBNP
      NP-NPBB(NB)
      IF(NP.NE.NI) GO TO 420
      NII-NB
      GO TO 440
  420 CONTINUE
      PRINT 4000, IQ,MP
 4000 FORMAT(1H0,10X,' CAN NOT FIND A COMPRESSED BOUNDARY NODE FOR',
     1 I2,'-TH POINT OF',I4,'-TH BOUNDARY SIDE  STOP')
      STOP
  440 ISB(IQ,MP)-NII
  460 CONTINUE
  490 CONTINUE
C
C ------- PRINT BOUNDARY SIDE INFORMATION
C
      IF(MOD(IGEOM,2).EQ.0) GO TO 696
      LINE-0
      DO 695 MP-1,NBES
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 6900
      PRINT 6950, MP,(DCOSB(I,MP),I-i,3),(ISB(I,MP),I-1,6)
  695 CONTINUE
  696 CONTINUE
C
 3900 FORMAT(1H1//10X,' **** COMPUTED BOUNDARY NODE DATA ****'//5X,
     1 10('    I NPBB',2X)/5X,10('    - ----',2X))
 3950 FORMAT(1H ,4X,10(2I5,2X))
 6900 FORMAT(1H1//10X,' *** COMPUTED BOUNDARY ELEMENT SIDE INFORMATION',
     1 '***'//5X,'   MP      DCOSXB            DCOSYB           DCOSZB',
     2 '      BP1  BP2  BP3  BP4   LS    M'/5X,
     3 '   --      ------          ------          ------      ',
     4 '   ---  ---  ---  ---   --    -')
 6950 FORMAT(1H ,4X,I5,3D15.6,6I5)
C
      RETURN
      END
```

```
      SUBROUTINE PAGEN(GNOJCN,LNOJCN,LMAXDF,NTNR, GNP, IE,NNR)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO GENERATE POINTER ARRAYS.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: IE(NEL,9), GNP(NNR(K),NREGN), NNR(NREGN).
C
C ------- OUPUT: GNOJCN(JBAND,NNP), GNP(NNR(K)+1 TO NTNR(K),NREGN),
C -------               NTNR(NREGN), LNOJCN(JBAND,NNR(K),NREGN), LMAXDF(NREGN).
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 GNP,GNOJCN
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI,NDTCHG
      COMMON /LGEOM/ MXTNR,MXNR,LMXBW,MXKR,NREGN
      COMMON /CINTE/ NCYL,NITER,KSTR,KPRO,KDSKO,KSS,NPITER,IGEOM
C
      DIMENSION GNOJCN(JBAND,MAXNP),LNOJCN(JBAND,MXNR,MXKR),LMAXDF(MXKR)
      DIMENSION IE(MAXEL,9),GNP(MXTNR,MXKR),NNR(MXKR),NTNR(MXKR)
      DIMENSION IEM(8)
C
C ******* GENERATE GNOJCN(JBAND,MAXNP) BASED ON IE(MAXEL,9)
C
      DO 490 NP=1,NNP
C
      DO 110 I=1,JBAND
  110 GNOJCN(I,NP)=0
C
      NLNOD=0
      KOUNT=0
C
      DO 390 M=1,NEL
C
      DO 160 IQ=1,8
  160 IEM(IQ)=IE(M,IQ)
C
      DO 170 IQ=1,8
      NI=IEM(IQ)
      IF(NI.NE.NP) GO TO 170
      GO TO 220
  170 CONTINUE
      GO TO 390
```

```
C                                                                       PAGE 245
  220 DO 290 IQ-1,8                                                     PAGE 250
      NI-IE(M,IQ)                                                       PAGE 255
C                                                                       PAGE 260
C ------- COMPRESS THE NODES TO 1 TO JBAND                              PAGE 265
C                                                                       PAGE 270
      NLNOD-NLNOD+1                                                     PAGE 275
      IF(NLNOD.GT.1) GO TO 230                                          PAGE 280
C                                                                       PAGE 285
C ------- THE FIRST NODE IS ENCOUNTERED                                 PAGE 290
C                                                                       PAGE 295
      KOUNT-KOUNT+1                                                     PAGE 300
      GNOJCN(KOUNT,NP)-NI                                               PAGE 305
      GO TO 290                                                         PAGE 310
C                                                                       PAGE 315
C ------- IF NLNOD IS GREATER THAN 1, WE HAVE TO CHECK IF NI IS THE     PAGE 320
C ------- NODE ALREADY COMPRESSED? IF YES, SKIP. IF NOT INCREASE THE    PAGE 325
C ------- KOUNT.                                                        PAGE 330
C                                                                       PAGE 335
  230 DO 240 J-1,KOUNT                                                  PAGE 340
      NJ-GNOJCN(J,NP)                                                   PAGE 345
      IF(NI.EQ.NJ) GO TO 290                                            PAGE 350
  240 CONTINUE                                                          PAGE 355
C                                                                       PAGE 360
C ------- THE NODE NI HAS NOT BEEN COMPRESSED YET.  HENCE COMPRESS IT.  PAGE 365
C                                                                       PAGE 370
      KOUNT-KOUNT+1                                                     PAGE 375
      GNOJCN(KOUNT,NP)-NI                                               PAGE 380
C                                                                       PAGE 385
  290 CONTINUE                                                          PAGE 390
C                                                                       PAGE 395
  390 CONTINUE                                                          PAGE 400
C                                                                       PAGE 405
      IF(KOUNT.LE.JBAND) GO TO 490                                      PAGE 410
      KONT-KOUNT-1                                                      PAGE 415
      JBND-JBAND-1                                                      PAGE 420
      WRITE(6,1000) NP,KONT,JBND                                        PAGE 425
      STOP                                                              PAGE 430
C                                                                       PAGE 435
  490 CONTINUE                                                          PAGE 440
C                                                                       PAGE 445
C ******* PRINT GENERATED ARRAY GNOJCN                                  PAGE 450
C                                                                       PAGE 455
      IF(MOD(IGEOM,2).EQ.0) GO TO 595                                   PAGE 460
      LINE-0                                                            PAGE 465
      DO 590 NP-1,NNP                                                   PAGE 470
      LINE-LINE+1                                                       PAGE 475
      IF(MOD(LINE-1,50).EQ.0) PRINT 5000                                PAGE 480
      PRINT 5100, NP,(GNOJCN(I,NP),I-1,JBAND)                           PAGE 485
  590 CONTINUE                                                          PAGE 490
```

```
  595 CONTINUE
C
C ******* 1. FILL-UP GNP(I,MXKR) FROM I-MXNR+1 TO I-MXTNR BASED ON
C *******    IE(MAXEL,8) AND ON GNP(I,MXKR) FROM I-1 TO I-MXNR,
C ******* 2. GENERATE NTNR(MXKR) ALSO BASED ON IE(MAXEL,8) AND ON
C *******    GNP(I,MXKR) FROM I-1 TO I-MXNR.
C
      DO 690 K-1,NREGN
      LNNP-NNR(K)
      LTNNP-LNNP
      DO 680 M-1,NEL
C
      DO 630 IQ-1,8
      NI-IE(M,IQ)
      DO 620 J-1,LNNP
      NJ-GNP(J,K)
      IF(NI.NE.NJ) GO TO 620
      GO TO 650
  620 CONTINUE
  630 CONTINUE
C
C ------- NONE OF THE EIGHT NODES IS ONE OF THE INTERIOR NODES
C ------- HENCE SKIP THIS ELEMENT AND GO TO 680.
C
      GO TO 680
C
C ------- ONE OF THE EIGHT NODES IS ONE OF THE INTERIOR NODES,
C ------- HENCE EACH OF THE EIGHT NODES IS EITHER AN INTERIOR NODE
C ------- OR AN INTRA-BOUNDARY NODE.
C
  650 DO 670 IQ-1,8
      NI-IE(M,IQ)
C
C ------- CHECK IF NI IS ONE OF THE POINTS IN GNP(I,K) FOR I-1,LTNNP
C
      DO 660 J-1,LTNNP
      NJ-GNP(J,K)
C
C ------- NJ.EQ.NI MEANS NI IS ALREADY ONE OF THE GNP(I,K) POINTS,
C ------- HENCE SKIP AND GO TO 670
C
      IF(NJ.EQ.NI) GO TO 670
  660 CONTINUE
C
C ------- THE IQ-TH NODE, I.E.  NI, IS NOT ONE OF THE GNP(I,K) POINTS,
C ------- HENCE FACTOR THIS NODE INTO THE ARRAY GNP(I,K) BY FILLING-UP.
C
      LTNNP-LTNNP+1
      GNP(LTNNP,K)-NI
  670 CONTINUE
```

```
C                                                                       PAGE 745
  680 CONTINUE                                                          PAGE 750
C                                                                       PAGE 755
      NTNR(K)=LTNNP                                                     PAGE 760
C                                                                       PAGE 765
  690 CONTINUE                                                          PAGE 770
C                                                                       PAGE 775
C ******* GENERATE LNOJCN(JBAND,MXNR,MXKR) BASED ON GNP(MXTNR,MXKR)     PAGE 780
C ******* AND GNOJCN(JBAND,MAXNP) AND PRODUCE LMAXDF(MXKR) BASED ON     PAGE 785
C ******* LNOJCN(JBAND,MXNR,MXKR) FOR ALL SUBREGIONS.                   PAGE 790
C                                                                       PAGE 795
      DO 790 K=1,NREGN                                                  PAGE 800
      LNNP=NNR(K)                                                       PAGE 805
      LTNNP=NTNR(K)                                                     PAGE 810
C                                                                       PAGE 815
      DO 710 J=1,JBAND                                                  PAGE 820
      DO 710 LI=1,LNNP                                                  PAGE 825
      LNOJCN(J,LI,K)=0                                                  PAGE 830
  710 CONTINUE                                                          PAGE 835
C                                                                       PAGE 840
C -----  GENERATE LNOJCN(JBAND,MXNR,MXKR)                               PAGE 845
C                                                                       PAGE 850
      DO 770 LI=1,LNNP                                                  PAGE 855
      NI=GNP(LI,K)                                                      PAGE 860
C                                                                       PAGE 865
      DO 750 J=1,JBAND                                                  PAGE 870
      NJ=GNOJCN(J,NI)                                                   PAGE 875
C                                                                       PAGE 880
      IF(NJ.EQ.0) GO TO 740                                             PAGE 885
C                                                                       PAGE 890
      DO 720 LJ=1,LTNNP                                                 PAGE 895
      NLJ=GNP(LJ,K)                                                     PAGE 900
      IF(NLJ.EQ.NJ) GO TO 730                                           PAGE 905
  720 CONTINUE                                                          PAGE 910
C                                                                       PAGE 915
      WRITE (6,7000) K,J,NJ,NI,LI                                       PAGE 920
      STOP                                                              PAGE 925
C                                                                       PAGE 930
  730 LNOJCN(J,LI,K)=LJ                                                 PAGE 935
      GO TO 750                                                         PAGE 940
C                                                                       PAGE 945
  740 LNOJCN(J,LI,K)=0                                                  PAGE 950
C                                                                       PAGE 955
  750 CONTINUE                                                          PAGE 960
  770 CONTINUE                                                          PAGE 965
C                                                                       PAGE 970
C ------ GENERATE LMAXDF(MXKR)                                          PAGE 975
C                                                                       PAGE 980
```

```
      MAXDF=0                                                           PAGE 985
      DO 780 LI=1,LNNP                                                  PAGE 990
      DO 780 J=1,JBAND                                                  PAGE 995
      LJ=LNOJCN(J,LI,K)                                                 PAGE1000
      IF(LJ.GT.LNNP .OR. LJ.EQ.0) GO TO 780                             PAGE1005
      IDIF=LI-LJ                                                        PAGE1010
      IF(IDIF.LT.0) IDIF=LJ-LI                                          PAGE1015
      IF(MAXDF.LT.IDIF) MAXDF=IDIF                                      PAGE1020
  780 CONTINUE                                                          PAGE1025
C                                                                       PAGE1030
      LMAXDF(K)=MAXDF                                                   PAGE1035
C                                                                       PAGE1040
  790 CONTINUE                                                          PAGE1045
C                                                                       PAGE1050
C *******  PRINT GENERATED ARRAYS LNOJCN(J,AND,MXNR,MXKR) AND           PAGE1055
C *******  GNP(MXTNR,MXKR).                                             PAGE1060
C                                                                       PAGE1065
      IF(MOD(IGEOM,2).EQ.0) GO TO 895                                   PAGE1070
      DO 890 K=1,NREGN                                                  PAGE1075
      PRINT 8000, K                                                     PAGE1080
C                                                                       PAGE1085
      LNNP=NNR(K)                                                       PAGE1090
      LNNP1=LNNP+1                                                      PAGE1095
      LTNNP=NTNR(K)                                                     PAGE1100
C                                                                       PAGE1105
      DO 820 I=1,LNNP                                                   PAGE1110
      PRINT 8200, I,GNP(I,K),(LNOJCN(J,I,K),J=1,JBAND)                  PAGE1115
  820 CONTINUE                                                          PAGE1120
C                                                                       PAGE1125
      DO 830 I=LNNP1,LTNNP                                              PAGE1130
      PRINT 8200, I,GNP(I,K)                                            PAGE1135
  830 CONTINUE                                                          PAGE1140
C                                                                       PAGE1145
  890 CONTINUE                                                          PAGE1150
  895 CONTINUE                                                          PAGE1155
C                                                                       PAGE1160
      WRITE(6,9000) (LMAXDF(K),K=1,NREGN)                               PAGE1165
C                                                                       PAGE1170
 1000 FORMAT(1H0//5X,' ***',I4,'-TH NODE HAS',I4,' NODES SURROUNDING',  PAGE1175
     1 ' IT, WHICH IS MORE THAN JBAND - 1 =',I5,'  STOP ***')           PAGE1180
 5000 FORMAT(1H1/5X,' ** GENERATED SURROUNDING NODES OF ALL NODES *',// PAGE1185
     1 5X,'  NP   1   2   3   4   5   6   7   8',                       PAGE1190
     2 '   9  10  11  12  13  14  15  16  17',                          PAGE1195
     3 '  18  19  20  21  22  23  24  25  26  27',                      PAGE1200
     4 /5X,28('  --'))                                                  PAGE1205
 5100 FORMAT(1H ,4X,28I4)                                               PAGE1210
```

```
 7000 FORMAT(1H0//5X,' WE CANNOT FIND A LOCAL NODE IN REGION',I3,       PAGE1215
     1 ' CORRESPONDING TO',I3,'-TH COMPRESSED GLOBAL NODE',I4,          PAGE1220
     2 ' CONNECTING TO GLOBAL NODE',I4,' OR CONNECTING TO THE LOCAL ',  PAGE1225
     3 ' NODE ',I3)                                                     PAGE1230
 8000 FORMAT(1H1,10X,' *** ARRAY GNLR AND LNOJCN ***'///5X,             PAGE1235
     1 ' -- SUBREGION K =',I3,' --'//5X,                                PAGE1240
     2 ' LNODE GNODE  SURROUNDING AND INCLUDING LOCAL NODES'/5X,        PAGE1245
     3 ' ----- -----  ------------------------------------')            PAGE1250
 8200 FORMAT(1H ,4X,2I6,27I4)                                           PAGE1255
 9000 FORMAT(1H0//5X,' LIST OF HALF BAND WIDTH FOR ALL SUBREGIONS'/5X,  PAGE1260
     1 (30I4))                                                          PAGE1265
C                                                                       PAGE1270
      RETURN                                                            PAGE1275
      END                                                               PAGE1280
```

```
      SUBROUTINE VELT(VX,VY,VZ, CMATRX, X,Y,Z,IE,H,HT,AKR, PROP)        VELT 005
C                                                                       VELT 010
C********1*********2*********3*********4*********5*********6*********7**VELT 015
C                                                                       VELT 020
C ------- TO COMPUTE DARCY'S VELOCITY.                                  VELT 025
C                                                                       VELT 030
C********1*********2*********3*********4*********5*********6*********7**VELT 035
C                                                                       VELT 040
C ------- INPUT: X(NNP), Y(NNP), Z(NNP), IE(NEL,9), H(NNP), HT(NNP),    VELT 045
C -------        AKR(8,NEL), PROP(NMPPM,NMAT)                           VELT 050
C                                                                       VELT 055
C ------- OUTPUT: VX(NNP), VY(NNP), VZ(NNP).                            VELT 060
C                                                                       VELT 065
C********1*********2*********3*********4*********5*********6*********7**VELT 070
C                                                                       VELT 075
      IMPLICIT REAL*8(A-H,O-Z)                                          VELT 080
C                                                                       VELT 085
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC      VELT 090
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI,NDTCHG                 VELT 095
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM                                VELT 100
C                                                                       VELT 105
      DIMENSION VX(MAXNP),VY(MAXNP),VZ(MAXNP)                           VELT 110
      DIMENSION CMATRX(MAXNP,JBAND)                                     VELT 115
      DIMENSION X(MAXNP),Y(MAXNP),Z(MAXNP),IE(MAXEL,9)                  VELT 120
      DIMENSION H(MAXNP),HT(MAXNP),AKR(8,MAXEL),PROP(MXMPPM,MAXMAT)     VELT 125
C                                                                       VELT 130
      DIMENSION QB(8,8),QRX(8),QRY(8),QRZ(8),XQ(8),YQ(8),ZQ(8), HTQ(8)  VELT 135
      DIMENSION AKXG(8),AKYG(8),AKZG(8),AKXYG(8),AKXZG(9),AKYZG(8)      VELT 140
```

```
C
      AGRAV-DFLOAT(KGRAV)
C
C ------- INITIATE THE DARCY VELOCITY COMPONETNTS
C
      DO 100 NP-1,NNP
      VX(NP)-0.0
      VY(NP)-0.0
  100 VZ(NP)-0.0
C
C ------- CALCULATE THE TOTAL HEAD HT(NP)
C
      DO 105 NP-1,NNP
  105 HT(NP)-H(NP) +AGRAV*Z(NP)
C
C ------- COMPUTE DARCY VELOCITIES BY APPLYING FINITE ELEMENT METHOD TO
C ------- DARCY LAW.
C
C
C ------- INITIATE MATRIX CMATRX(NP,IB)
C
      DO 160 NP-1,NNP
  160 CMATRX(NP,1)-0.0
C
C ------- COMPUTE ELEMENT MATRICES QB(IQ,JQ), QRX(IQ),QRY(IQ),& QRZ(IQ)
C
      DO 290 M-1,NEL
      MTYP-IE(M,9)
C
      DO 210 IQ-1,8
      NP-IE(M,IQ)
      XQ(IQ)-X(NP)
      YQ(IQ)-Y(NP)
      ZQ(IQ)-Z(NP)
  210 HTQ(IQ)-HT(NP)
C
      DO 215 KG-1,8
      AKXG(KG)-AKR(KG,M)*PROP(1,MTYP)
      AKYG(KG)-AKR(KG,M)*PROP(2,MTYP)
      AKZG(KG)-AKR(KG,M)*PROP(3,MTYP)
      AKXYG(KG)-AKR(KG,M)*PROP(4,MTYP)
      AKXZG(KG)-AKR(KG,M)*PROP(5,MTYP)
  215 AKYZG(KG)-AKR(KG,M)*PROP(6,MTYP)
C
      CALL Q8DV(QB,QRX,QRY,QRZ, XQ,YQ,ZQ,HTQ,AKXG,AKYG,AKZG,
     1 AKXYG,AKXZG,AKYZG)
```

```
C
C ------- ASSEMBLE QB(IQ,JQ) INTO THE GLOBAL MATRIX CMATRX(NP,IB) AND
C ------- FORM THE LOAD VECTOR VX(NP), VY(NP), AND VZ(NP).
C
      DO 280 IQ-1,8
      NI-IE(M,IQ)
      DO 240 JQ-1,8
      CMATRX(NI,1)-CMATRX(NI,1)+QB(IQ,JQ)
  240 CONTINUE
C
      VX(NI)-VX(NI)+QRX(IQ)
      VY(NI)-VY(NI)+QRY(IQ)
      VZ(NI)-VZ(NI)+QRZ(IQ)
  280 CONTINUE
C
  290 CONTINUE
C
C ------- SOLVE THE MATRIX EQUATION CX-B
C
      DO 370 NP-1,NNP
      VX(NP)-VX(NP)/CMATRX(NP,1)
      VY(NP)-VY(NP)/CMATRX(NP,1)
  370 VZ(NP)-VZ(NP)/CMATRX(NP,1)
C
      RETURN
      END
```

```
      SUBROUTINE Q8DV(QB,QRX,QRY,QRZ, XQ,YQ,ZQ,HTQ,AKXG,AKYG,AKZG,
     1 AKXYG,AKXZG,AKYZG)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO COMPUTE THE INTEGRATION OF N(I)*N(J) AND -N(I)*K.GRAD(HT)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: XQ(8), YQ(8), ZQ(8), HTQ(8), AKXG(8), AKYG(8), AKZG(8),
C -------        AKXYG(8), AKXZG(8), AKYZG(8).
C
C ------- OUTPUT: QB(8,8), QRX(8), QRY(8), QRZ(8).
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N(8)
```

```
C                                                                       Q8DV 095
      DIMENSION QB(8,8),QRX(8),QRY(8),QRZ(8),XQ(8),YQ(8),ZQ(8),HTQ(8)   Q8DV 100
      DIMENSION AKXG(8),AKYG(8),AKZG(8),AKXYG(8),AKXZG(8),AKYZG(8)      Q8DV 105
      DIMENSION S(8),T(8),U(8), DNX(8),DNY(8),DNZ(8)                    Q8DV 110
C                                                                       Q8DV 115
      DATA P /0.577350269189626DO/                                      Q8DV 120
      DATA S/-1.0D0,1.0D0,1.0D0,-1.0D0, -1.0D0,1.0D0,1.0D0,-1.0D0/      Q8DV 125
      DATA T/-1.0D0,-1.0D0,1.0D0,1.0D0, -1.0D0,-1.0D0,1.0D0,1.0D0/      Q8DV 130
      DATA U/-1.0D0,-1.0D0,-1.0D0,-1.0D0, 1.0D0,1.0D0,1.0D0,1.0D0/      Q8DV 135
C                                                                       Q8DV 140
C ------- INITIATE MATRICES QB(IQ,JQ), QRX(IQ),QRY(IQ) & QRZ(IQ)        Q8DV 145
C                                                                       Q8DV 150
      DO 100 IQ=1,8                                                     Q8DV 155
      QRX(IQ)=0.0                                                       Q8DV 160
      QRY(IQ)=0.0                                                       Q8DV 165
      QRZ(IQ)=0.0                                                       Q8DV 170
      DO 100 JQ=1,8                                                     Q8DV 175
  100 QB(IQ,JQ)=0.0                                                     Q8DV 180
C                                                                       Q8DV 185
C ------- SUMMATION OF THE INTEGRAND OVER THE GAUSSIAN POINTS           Q8DV 190
C                                                                       Q8DV 195
      DO 490 KG=1,8                                                     Q8DV 200
C                                                                       Q8DV 205
C ------- DETERMINE LOACAL COORDINATE OF GAUSSIAN POINT KG              Q8DV 210
C                                                                       Q8DV 215
      SS=P*S(KG)                                                        Q8DV 220
      TT=P*T(KG)                                                        Q8DV 225
      UU=P*U(KG)                                                        Q8DV 230
C                                                                       Q8DV 235
C ------- CALCULATE VALUES OF BASIS FUNCTIONS N(IQ) AND THEIR           Q8DV 240
C ------- DERIVATIVES DNX(IQ), DNY(IQ), AND DNZ(IQ), W.R.T. TO          Q8DV 245
C ------- X, Y, AND Z, RESPECTIVELY, AT THE GAUSSIAN POINT KG.          Q8DV 250
C                                                                       Q8DV 255
      CALL BASE(N,DNX,DNY,DNZ,DJAC,XQ,YQ,ZQ,SS,TT,UU)                   Q8DV 260
C                                                                       Q8DV 265
      AKXK=AKXG(KG)                                                     Q8DV 270
      AKYK=AKYG(KG)                                                     Q8DV 275
      AKZK=AKZG(KG)                                                     Q8DV 280
      AKXYK=AKXYG(KG)                                                   Q8DV 285
      AKXZK=AKXZG(KG)                                                   Q8DV 290
      AKYZK=AKYZG(KG)                                                   Q8DV 295
C                                                                       Q8DV 300
C ------- ACCUMULATE THE SUMS TO OBTAIN THE MATRIX INTEGRALS QB(IQ,JQ)  Q8DV 305
C ------- AND QRX(IQ), QRY(IQ), AND QRZ(IQ)                             Q8DV 310
C                                                                       Q8DV 315
      DO 390 IQ=1,8                                                     Q8DV 320
      DO 390 JQ=1,8                                                     Q8DV 325
      QB(IQ,JQ)=QB(IQ,JQ)+ N(IQ)*N(JQ)*DJAC                             Q8DV 330
```

```
      QRX(IQ)=QRX(IQ)-N(IQ)*HTQ(JQ)*(AKXK*DNX(JQ)+AKXYK*DNY(JQ)+
     1 AKXZK*DNZ(JQ))*DJAC
      QRY(IQ)=QRY(IQ)-N(IQ)*HTQ(JQ)*(AKXYK*DNX(JQ)+AKYK*DNY(JQ)+
     1 AKYZK*DNZ(JQ))*DJAC
      QRZ(IQ)=QRZ(IQ)-N(IQ)*HTQ(JQ)*(AKXZK*DNX(JQ)+AKYZK*DNY(JQ)+
     1 AKZK*DNZ(JQ))*DJAC
  390 CONTINUE
C
  490 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SPROP(TH,DTH,AKR, IE,H,THPROP,AKPROP)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO COMPUTE RELATIVE HYDRAULIC CONDUCTIVITY OR PERMEABILITY,
C ------- MOISTURE CONTENT, AND WATER CAPACITY GIVEN THE PRESSSURE
C ------- HEAD.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: IE(NEL,9), H(NNP), THPROP(NSPPM,NMAT),
C -------               AKPROP(NSPPM,NMAT).
C
C ------- OUTPUT: AKR(8,NEL), TH(8,NEL), DTH(8,NEL).
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 N(8)
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI,NDTCHG
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM
C
      DIMENSION TH(8,MAXEL),DTH(8,MAXEL),AKR(8,MAXEL)
      DIMENSION IE(MAXEL,9),H(MAXNP)
      DIMENSION THPROP(MXSPPM,MAXMAT),AKPROP(MXSPPM,MAXMAT)
C
      DIMENSION HQ(8), S(8),T(8),U(8),HKG(8)
```

```
C
      DATA P/0.577350269189626/
      DATA S/-1.0D0,1.0D0,1.0D0,-1.0D0, -1.0D0,1.0D0,1.0D0,-1.0D0/
      DATA T/-1.0D0,-1.0D0,1.0D0,1.0D0, -1.0D0,-1.0D0,1.0D0,1.0D0/
      DATA U/-1.0D0,-1.0D0,-1.0D0,-1.0D0, 1.0D0,1.0D0,1.0D0,1.0D0/
C
      DO 900 M=1,NEL
C
      DO 110 IQ=1,8
      NP=IE(M,IQ)
  110 HQ(IQ)=H(NP)
C
C ------- EVALUATE PRESSURE AT EIGHT GAUSSIAN POINTS.
C
      DO 150 KG=1,8
      SS=P*S(KG)
      TT=P*T(KG)
      UU=P*U(KG)
      SM=1.0D0-SS
      SP=1.0D0+SS
      TM=1.0D0-TT
      TP=1.0D0+TT
      UM=1.0D0-UU
      UP=1.0D0+UU
C
      N(1)=0.125D0*SM*TM*UM
      N(2)=0.125D0*SP*TM*UM
      N(3)=0.125D0*SP*TP*UM
      N(4)=0.125D0*SM*TP*UM
      N(5)=0.125D0*SM*TM*UP
      N(6)=0.125D0*SP*TM*UP
      N(7)=0.125D0*SP*TP*UP
      N(8)=0.125D0*SM*TP*UP
C
      HKG(KG)=0.0
      DO 120 IQ=1,8
  120 HKG(KG)=HKG(KG)+HQ(IQ)*N(IQ)
  150 CONTINUE
C
      MTYP=IE(M,9)
C
C
C ------- TH, DTH/DH, AND AKR ARE OBTAINED WITH ANALYTICAL FUNCTIONS.
C ------- THE READER MUST SUPPLY THE FUNCTIONAL FORM OF FKR AND FTH
```

```
C
C ------- THE FOLLOWING IS JUST AN EXAMPLE.
C ------- WCR- THPROP(1,MTYP)-0.065, 0.050 FOR TWO SAMPLE MATERUALS
C ------- WCS-THPROP(2,MTYP)-0.364, 0.341 FOR TWO SAMPLE MATERIALS
C ------- RN-THPROP(3,MTYP)-1.092217, 1.546937 FOR TWO SAMPLE MATERIALS
C ------- ALPH-THPROP(4,MTYP)-0.109, 0.002166 FOR TWO SAMPLE MATERIALS
C
          WCR-THPROP(1,MTYP)
          WCS-THPROP(2,MTYP)
          RN-PROP(3,MTYP)
          ALPH-PROP(4,MTYP)
          RM-1.0D0-1.0D0/RN
          DO 800 KG-1,8
          HNP-HKG(KG)
          HNP--HNP
C
C ------- SATURATED CONDITION
C
          IF(HNP.GT.0.0) GO TO 700
          TH(KG,M)-WCS
          DTH(KG,M)-0.0D0
          AKR(KG,M)-1.0D0
          GO TO 800
C
C ------- UNSATURATED CASE
C
  700 THMKG-WCR+(WCS-WCR)/(1.0D0+(ALPH*HNP)**RN)**RM
      TH(KG,M)-THMKG
      RWC-(THMKG-WCR)/(WCS-WCR)
      TERM-(1.0D0-RWC**(1.0D0/RM))**RM
      RK-DSQRT(RWC)*(1.0D0-TERM)*(1.0D0-TERM)
      AKR(KG,M)-RK
      DTH(KG,M)-ALPH*(RN-1.0D0)*TERM*RWC**(1.0D0/RM)*(WCS-WCR)
C
  800 CONTINUE
  900 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BCPREP(IE,X,Y,Z,  H,VX,VY,VZ, DCOSB,ISB, NPVB,ISV,
     > DCYFLX,FLX,HCON,HMIN,NPFLX,NPCON,NPMIN, IRTYP,RFALL, NCHG)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO DETERMINE WHETHER THE DIRICHLET B. C. WITH PONDING DEPTH,
C ------- OR THE DIRICHELT B. C. WITH MINIMUM PRESSURE HEAD, OR THE FLUX
C ------- B. C. WITH PRESCRIBED FLUX TO BE APPLIED ON THE VARIABLE
C ------- BOUNDARIES, NORMALLY THE AIR-MEDIA INTERFACE.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: IE(NEL,9), X(NNP), Y(NNP), Z(NNP), H(NNP), VX(NNP),
C -------        VY(NNP), VZ(NNP), DCOSB(3,NBES), ISB(6,NBES),
C -------        NPNV(NVNP), ISV(5,NVES), IRTYP(NVES), RFALL(NRPR),
C -------        HCON(NVNP), HMIN(NVNP), NCHG.
C
C ------- OUTPUT: NPCON(NVNP), NPFLX(NVNP), NPMIN(NVNP), FLX(NVNP),
C -------         DCYFLX(NVNP), NCHG.
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI
      COMMON /CVBC/ MXVES,MXVNP,MXRPR,MXRDP,NVES,NVNP,NRPR,NRDP
C
      DIMENSION IE(MAXEL,9),X(MAXNP),Y(MAXNP),Z(MAXNP)
      DIMENSION H(MAXNP),VX(MAXNP),VY(MAXNP),VZ(MAXNP)
C
      DIMENSION DCOSB(3,MAXBES),ISB(6,MAXBES),ISV(5,MXVES),NPVB(MXVNP)
      DIMENSION DCYFLX(MXVNP),FLX(MXVNP),HCON(MXVNP),HMIN(MXVNP)
      DIMENSION NPFLX(MXVNP),NPCON(MXVNP),NPMIN(MXVNP)
      DIMENSION RFALL(MXRPR),IRTYP(MXVES)
C
      DIMENSION R1Q(4),R2Q(4),XQ(4),YQ(4),ZQ(4),F1Q(4),F2Q(4)
C
      DIMENSION KGB(4,6)
C
      DATA KGB/1,4,8,5, 1,2,6,5, 2,3,7,6, 4,3,7,8, 1,2,3,4, 5,6,7,8/
C
C ------- DETERMINE NORMAL RAINFALLS FLX(I) AND DARCY FLUXES DCYFLX(I)
C ------- FOR EACH NODAL POINT ON THE VARIABLE ELEMENT-SIDE.
C
      DO 210 NP=1,NVNP
         FLX(NP)=0.
  210    DCYFLX(NP)=0.
```

```
C                                                                       BCPR 245
      DO 290 MP=1,NVES                                                  BCPR 250
      ITYP=IRTYP(MP)                                                    BCPR 255
      RFMP=RFALL(ITYP)                                                  BCPR 260
C                                                                       BCPR 265
      MPB=ISV(5,MP)                                                     BCPR 270
      LS=ISB(5,MPB)                                                     BCPR 275
      M=ISB(6,MPB)                                                      BCPR 280
C                                                                       BCPR 285
      PROJ=DCOSB(3,MPB)                                                 BCPR 290
      RFMPN=-RFMP*PROJ                                                  BCPR 295
C                                                                       BCPR 300
      DO 230 IQ=1,4                                                     BCPR 305
      I=KGB(IQ,LS)                                                      BCPR 310
      NI=IE(M,I)                                                        BCPR 315
      XQ(IQ)=X(NI)                                                      BCPR 320
      YQ(IQ)=Y(NI)                                                      BCPR 325
      ZQ(IQ)=Z(NI)                                                      BCPR 330
      F1Q(IQ)=RFMPN                                                     BCPR 335
      F2Q(IQ)=VX(NI)*DCOSB(1,MPB)+DCOSB(2,MPB)*VY(NI)+DCOSB(3,MPB)*     BCPR 340
     1        VZ(NI)                                                    BCPR 345
  230 CONTINUE                                                          BCPR 350
C                                                                       BCPR 355
C -------- COMPUTE SURFACE INTEGRAL OF N(IQ).F                          BCPR 360
C                                                                       BCPR 365
      CALL Q4S(R1Q,R2Q,XQ,YQ,ZQ,F1Q,F2Q)                                BCPR 370
C                                                                       BCPR 375
      DO 270 IQ=1,4                                                     BCPR 380
      I=ISV(IQ,MP)                                                      BCPR 385
      FLX(I)=FLX(I)+R1Q(IQ)                                             BCPR 390
      DCYFLX(I)=DCYFLX(I)+R2Q(IQ)                                       BCPR 395
  270 CONTINUE                                                          BCPR 400
C                                                                       BCPR 405
  290   CONTINUE                                                        BCPR 410
C                                                                       BCPR 415
C ------- CHANGE TO FLUX OR HEAD CONDITIONS, AS NECESSARY, AND SO       BCPR 420
C ------- INDICATE IN THE ARRAYS NPFLX(NPP) AND NPCON(NPP).             BCPR 425
C                                                                       BCPR 430
      IF (NCHG.NE.(-1)) GO TO 300                                       BCPR 435
      NCHG=0                                                            BCPR 440
      RETURN                                                            BCPR 445
C                                                                       BCPR 450
  300 NCHG=0                                                            BCPR 455
C                                                                       BCPR 460
      DO 390 NPP=1,NVNP                                                 BCPR 465
C                                                                       BCPR 470
      DCYNNP=DCYFLX(NPP)                                                BCPR 475
      FLXNNP=FLX(NPP)                                                   BCPR 480
      HCONNP=HCON(NPP)                                                  BCPR 485
      HMINNP=HMIN(NPP)                                                  BCPR 490
```

```
C
      IF(FLXNNP.GT.0.0) GO TO 350
C
C **** RAINFALL(INFILTRATION)-SEEPAGE CONDITIONS PREVAIL DURING RAINFALL
C
C
C ------- CHECK IF THE CHANGE FROM RAINFALL-FLUX (NEUMANN) CONDITION TO
C ------- PONDING (DIRICHLET) CONDITION IS NECESSARY?
C
         NP=NPFLX(NPP)
         IF (NP.EQ.0) GO TO 310
      IF(HCONNP.GE.H(NP)) GO TO 390
         NPCON(NPP)=NPFLX(NPP)
         NPFLX(NPP)=0
         NCHG=NCHG+1
         GO TO 390
C
C ------- CHECK IF THE CHANGE FROM PONDING (DIRICHLET) CONDITION TO
C ------- RAINFALL-FLUX (NEUMANN) CONDITION IS NECESSARY?
C
  310 NP=NPCON(NPP)
      IF(NP.EQ.0) GO TO 320
      IF(FLXNNP.LE.DCYNNP) GO TO 390
         NPFLX(NPP)=NPCON(NPP)
         NPCON(NPP)=0
         NCHG=NCHG+1
      GO TO 390
C
C ------- CHANGE MINIMUM PRESSURE CONDITION TO RAINFALL-FLUX CONDITION
C ------- SINCE A MINIMUM PRESSURE CONDITION IS NOT LIKELY TO BE
C ------- DURING RAINFALL PERIOD
C
  320 NP=NPMIN(NPP)
      IF(NP.EQ.0) GO TO 390
      NPFLX(NPP)=NPMIN(NPP)
      NPMIN(NPP)=0
      NCHG=NCHG+1
      GO TO 390
C
C **** EVAPORATION-SEEPAGE CONDITIONS PREVAIL DURING NON-RAINFALL PERIOD
C
C
C ------- CHECK IF THE CHANGE FROM EVAPORATION-FLUX CONDITION TO
C ------- MINIMUM PRESSURE HEAD CONDITION IS NECESSARY?
C
  350 NP=NPFLX(NPP)
      IF(NP.EQ.0) GO TO 360
      IF(HMINNP.LE.H(NP)) GO TO 390
      NPMIN(NPP)=NPFLX(NPP)
```

```
      NPFLX(NPP)=0                                                      BCPR 740
      NCHG=NCHG+1                                                       BCPR 745
      GO TO 390                                                         BCPR 750
C                                                                       BCPR 755
C ------- CHECK IF THE CHANGE FROM PONDING CONDITION TO EVAPORATION-FLUXBCPR 760
C ------- CONDITION IS NECESSARY?                                       BCPR 765
C                                                                       BCPR 770
  360 NP=NPCON(NPP)                                                     BCPR 775
      IF(NP.EQ.0) GO TO 370                                             BCPR 780
      IF(DCYNNP.GE.0.0) GO TO 390                                       BCPR 785
      NPFLX(NPP)=NPCON(NPP)                                             BCPR 790
      NPCON(NPP)=0                                                      BCPR 795
      NCHG=NCHG+1                                                       BCPR 800
      GO TO 390                                                         BCPR 805
C                                                                       BCPR 810
C ------- CHECK IF THE CHANGE FROM MINIMUM PRESSURE HEAD CONDITION TO   BCPR 815
C ------- EVAPORATION-FLUX CONDITION IS NECESSARY?                      BCPR 820
C                                                                       BCPR 825
  370 NP=NPMIN(NPP)                                                     BCPR 830
      IF(NP.EQ.0) GO TO 390                                             BCPR 835
C     IF(DCYNNP.LT.0.0) GO TO 380                                       BCPR 840
      IF(DCYNNP.LT.FLXNNP) GO TO 390                                    BCPR 845
      NPFLX(NPP)=NPMIN(NPP)                                             BCPR 850
      NPMIN(NPP)=0                                                      BCPR 855
      NCHG=NCHG+1                                                       BCPR 860
C     GO TO 390                                                         BCPR 865
C                                                                       BCPR 870
C 380 PRINT 1000, NPP                                                   BCPR 875
C     STOP                                                              BCPR 880
C                                                                       BCPR 885
  390    CONTINUE                                                       BCPR 890
C                                                                       BCPR 895
C1000 FORMAT(1H1,'*** WARNING DURING NON-RAINFALL PERIOD',I3,           BCPR 900
C    1 '-TH VB NODE USING MINIMUM HEAD CONDITION RESULTS IN ',          BCPR 905
C    2 'DARCY FLUX INTO THE REGION'/1X,'THIS SHOULD NOT HAPPEN. ',       BCPR 910
C    3 'HENCE EITHER MINIMUM HEAD IS NOT SMALL ENOUGH OR THE INTERIOR ',BCPR 915
C    4 'NODE PRESSURE IS TOO SMALL: STOP')                              BCPR 920
C                                                                       BCPR 925
      RETURN                                                            BCPR 930
      END                                                               BCPR 935
```

```
      SUBROUTINE ASEMBL(X,Y,Z,IE, CMATRX,RLD,LRN, HW,HP,DTH,AKR,PROP,
     > SOS,MSEL,ISTYP,WSS,NPW,IWTYP, KSS,W,DELT)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO ASSEMBLE THE GLOBAL COEFFICIENT MATRIX AND GLOBAL LOAD
C ------- VECTOR IN COMPRESSED FORM.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: X(NNP), Y(NNP), Z(NNP), IE(NEL,9), LRN(JBAND,NNP),
C --------       HW(NNP), HP(NNP), DTH(8,NEL), AKR(8,NEL),
C -------        PROP(NMPPM,NMAT), SOS(NSPR), MSEL(NSEL), ISTYP(NSEL),
C -------        WSS(NWPR), NPW(NWNP), IWTYP(NWNP), KSS, W, DELT.
C
C ------- OUTPUT: CMATRX(NNP,JBAND), RLD(NNP).
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,KGRAV,NIT,NDTCHG
      COMMON /CS/ MXSPR,MXSDP,MXSEL,NSPR,NSDP,NSEL
      COMMON /CW/ MXWPR,MXWDP,MXWNP,NWPR,NWDP,NWNP
C
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM
C
      DIMENSION X(MAXNP),Y(MAXNP),Z(MAXNP),IE(MAXEL,9)
      DIMENSION CMATRX(MAXNP,JBAND),RLD(MAXNP),LRN(JBAND,MAXNP)
      DIMENSION HW(MAXNP),HP(MAXNP),DTH(8,MAXEL),AKR(8,MAXEL)
      DIMENSION PROP(MXMPPM,MAXMAT)
      DIMENSION SOS(MXSPR), MSEL(MXSEL),ISTYP(MXSEL)
      DIMENSION WSS(MXWPR), NPW(MXWNP),IWTYP(MXWNP)
C
      DIMENSION QA(8,8),QB(8,8),RQ(8)
      DIMENSION DTHG(8),AKXG(8),AKYG(8),AKZG(8),AKXYG(9),AKXZG(8),
     1 AKYZG(8), XQ(8),YQ(8),ZQ(8),IEM(8)
C
C
      AGRAV=DFLOAT(KGRAV)
C
      DELTI=1./DELT
      W1=W
      W2=1.-W
      IF (KSS.GT.0) GO TO 100
      DELTI=0.
      W1=1.
      W2=0.
```

```
C
C ------- INITIATE MATRICES C(NP,IB) AND R(NP)
C
  100 DO 150 NP=1,NNP
      RLD(NP)=0.0
      DO 150 I=1,JBAND
      CMATRX(NP,I)=0.0
  150 CONTINUE
C
C ------- START TO ASSEMBLE OVER ALL ELEMENTS
C
      DO 490 M=1,NEL
C
C ------- COMPUTE MATRICES QA(IQ,JQ), QB(IQ,JQ), AND RQ(IQ) FOR EACH M
C
      MTYP=IE(M,9)
      SAKX=PROP(1,MTYP)
      SAKY=PROP(2,MTYP)
      SAKZ=PROP(3,MTYP)
      SAKXY=PROP(4,MTYP)
      SAKXZ=PROP(5,MTYP)
      SAKYZ=PROP(6,MTYP)
C
      DO 210 IQ=1,8
      NP=IE(M,IQ)
      IEM(IQ)=NP
      XQ(IQ)=X(NP)
      YQ(IQ)=Y(NP)
      ZQ(IQ)=Z(NP)
  210 CONTINUE
C
      DO 220 KG=1,8
      DTHG(KG)=DTH(KG,M)
      AKXG(KG)=SAKX*AKR(KG,M)
      AKYG(KG)=SAKY*AKR(KG,M)
      AKZG(KG)=SAKZ*AKR(KG,M)
      AKXYG(KG)=SAKXY*AKR(KG,M)
      AKXZG(KG)=SAKXZ*AKR(KG,M)
      AKYZG(KG)=SAKYZ*AKR(KG,M)
  220 CONTINUE
C
      SOSM=0.0
      IF(NSEL.EQ.0) GO TO 260
      DO 240 I=1,NSEL
      MP=MSEL(I)
      IF(MP.NE.M) GO TO 240
      ITYP=ISTYP(I)
      SOSM=SOS(ITYP)
```

```
      GO TO 260
  240 CONTINUE
  260 CONTINUE
C
      CALL Q8(QA,QB,RQ, DTHG,AKXG,AKYG,AKZG,AKXYG,AKXZG,AKYZG,
     >        XQ,YQ,ZQ, SOSM,AGRAV)
C
C ------- ASSEMBLE QA(IQ,JQ) AND QB(IQ,JQ) INTO THE GLOBAL MATRIX
C ------- C(NP,IB) - B + A/DELT AND FORM THE GLOBAL LOAD VECTOR R(NP).
C
      DO 390 IQ-1,8
      NI-IEM(IQ)
      RLD(NI)-RLD(NI)-RQ(IQ)
      DO 340 JQ-1,8
      NJ-IEM(JQ)
      QA(IQ,JQ)-QA(IQ,JQ)*DELTI
      RLD(NI)-RLD(NI)+(QA(IQ,JQ)-W2*QB(IQ,JQ))*HP(NJ)
      DO 325 I-1,JBAND
      LNODE-LRN(I,NI)
      IF(NJ.EQ.LNODE) GO TO 330
  325 CONTINUE
C
      WRITE(6,1000) NI,M,JQ
      STOP
C
  330 CMATRX(NI,I)-CMATRX(NI,I)+QA(IQ,JQ)+W1*QB(IQ,JQ)
C
  340 CONTINUE
  390 CONTINUE
  490 CONTINUE
C
C ------- INCORPORATE WELL SOURCE/SINK
C
  700 IF(NWNP.EQ.0) GO TO 910
      DO 790 I-1,NWNP
      NI-NPW(I)
      ITYP-IWTYP(I)
      RLD(NI)-RLD(NI)+WSS(ITYP)
  790 CONTINUE
C
  910 CONTINUE
C
 1000 FORMAT(1H1/5X,'*** WARNING: NONE OF THE LOWER-LEFT NODE IN EQUATIO
     1N',I3,/5X,'***  IS CORRESPONDING TO ',I5,'-TH ELEMENT-S',I2,
     2'-TH NODE; STOP  ****')
C
      RETURN
      END
```

```
      SUBROUTINE Q8(QA,QB,RQ, DTHG,AKXG,AKYG,AKZG,AKXYG,AKXZG,AKYZG,
     > XQ,YQ,ZQ,SOSM,AGRAV)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO COMPUTE ELEMENT MATRICES AND ELEMENT LOAD VECTORS.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: XQ(8), YQ(8), ZQ(8), DTHG(8), AKXG(8), AKYG(8),
C -------        AKZG(8), AKXYG(8), AKXZG(8), AKYZG(8), SOSM, AGRAV.
C
C ------- OUTPUT: QA(8,8), QB(8,8), RQ(8).
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N(8)
C
      DIMENSION QA(8,8),QB(8,8),RQ(8),DTHG(8)
      DIMENSION AKXG(8),AKYG(8),AKZG(8),AKXYG(8),AKXZG(8),AKYZG(8)
      DIMENSION XQ(8),YQ(8),ZQ(8)
C
      DIMENSION DNX(8),DNY(8),DNZ(8)
      DIMENSION S(8),T(8),U(8)
C
      DATA P / 0.577350269189626D0/
      DATA S/-1.0D0,1.0D0,1.0D0,-1.0D0, -1.0D0,1.0D0,1.0D0,-1.0D0/
      DATA T/-1.0D0,-1.0D0,1.0D0,1.0D0, -1.0D0,-1.0D0,1.0D0,1.0D0/
      DATA U/-1.0D0,-1.0D0,-1.0D0,-1.0D0, 1.0D0,1.0D0,1.0D0,1.0D0/
C
C ------- INITIATE MATRICES QA, QB, AND RQ
C
      DO 110 IQ-1,8
      RQ(IQ)-0.0
      DO 110 JQ-1,8
      QA(IQ,JQ)-0.0
      QB(IQ,JQ)-0.0
  110 CONTINUE
C
      DO 490 KG-1,8
C
C ------- DETERMINE LOACAL COORDINATE  OF GAUSSIAN POINT KG
C
      SS-P*S(KG)
      TT-P*T(KG)
      UU-P*U(KG)
```

```
C
C ------- CALCULATE VALUES OF BASIS FUNCTIONS N(IQ) AND THEIR
C ------- DERIVATIVES DNX(IQ), DNY(IQ), & DNZ(IQ) W.R.T. X, Y, & Z,
C ------- RESPECTIVELY AND THE DETERMINANT OF THE JACOBIAN.
C
          CALL BASE(N,DNX,DNY,DNZ,DJAC,XQ,YQ,ZQ,SS,TT,UU)
C
      AKXQP=AKXG(KG)*DJAC
      AKYQP=AKYG(KG)*DJAC
      AKZQP=AKZG(KG)*DJAC
      AKXYQP=AKXYG(KG)*DJAC
      AKXZQP=AKXZG(KG)*DJAC
      AKYZQP=AKYZG(KG)*DJAC
      DTHQP=DTHG(KG)*DJAC
      SOSMQP=SOSM*DJAC
C
C ------- ACCUMULATE THE SUMS TO OBTAIN THE MATRIX INTEGRALS QA(IQ,JQ),
C ------- QB(IQ,JQ), AND RQ(IQ).
C
      DO 390 IQ=1,8
      RQ(IQ)=RQ(IQ)+AGRAV*(DNX(IQ)*AKXZQP+AKYZQP*DNY(IQ)+AKZQP*DNZ(IQ))+
     1 N(IQ)*SOSMQP
      DO 390 JQ=1,8
      QA(IQ,JQ)=QA(IQ,JQ) + DTHQP*N(IQ)*N(JQ)
      QB(IQ,JQ)=QB(IQ,JQ) + DNX(IQ)*(AKXQP*DNX(JQ)+AKXYQP*DNY(JQ)+
     1 AKXZQP*DNZ(JQ)) + DNY(IQ)*(AKXYQP*DNX(JQ)+AKYQP*DNY(JQ)+AKYZQP*
     2 DNZ(JQ)) + DNZ(IQ)*(AKXZQP*DNX(JQ)+AKYZQP*DNY(JQ)+AKZQP*DNZ(JQ))
  390 CONTINUE
C
  490 CONTINUE
  690 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE BASE(N,DNX,DNY,DNZ,DJAC,XQ,YQ,ZQ,SS,TT,UU)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO COMPUTE THE BASIS AND WEIGHTING FUNCTIONS, THEIR
C ------- DERIVATIVES WITH RESPECT TO X, Y, Z, AND THE JACOBIAN
C ------- AT A GAUSSIAN POINT.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: XQ(8), YQ(8), ZQ(8), SS, TT, UU.
C
C ------- OUTPUT: N(8), DNX(8), DNY(8), DNZ(8), DJAC.
C
C ------- LOCALLY DEFINED WORKING ARRAYS: DNSS(8), DNTT(8), DNUU(8).
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N(8)
C
      DIMENSION DNX(8),DNY(8),DNZ(8), XQ(8),YQ(8),ZQ(8)
      DIMENSION DNSS(8),DNTT(8),DNUU(8)
C
      SM=1.0D0-SS
      SP=1.0D0+SS
      TM=1.0D0-TT
      TP=1.0D0+TT
      UM=1.0D0-UU
      UP=1.0D0+UU
C
      N(1)=.125D0*SM*TM*UM
      N(2)=.125D0*SP*TM*UM
      N(3)=.125D0*SP*TP*UM
      N(4)=.125D0*SM*TP*UM
      N(5)=.125D0*SM*TM*UP
      N(6)=.125D0*SP*TM*UP
      N(7)=.125D0*SP*TP*UP
      N(8)=.125D0*SM*TP*UP
C
      DNSS(1)=-.125D0*TM*UM
      DNSS(2)=.125D0*TM*UM
      DNSS(3)=.125D0*TP*UM
      DNSS(4)=-.125D0*TP*UM
      DNSS(5)=-.125D0*TM*UP
      DNSS(6)=.125D0*TM*UP
      DNSS(7)=.125D0*TP*UP
      DNSS(8)=-.125D0*TP*UP
```

```
C
      DNTT(1)=-.125D0*SM*UM
      DNTT(2)=-.125D0*SP*UM
      DNTT(3)=.125D0*SP*UM
      DNTT(4)=.125D0*SM*UM
      DNTT(5)=-.125D0*SM*UP
      DNTT(6)=-.125D0*SP*UP
      DNTT(7)=.125D0*SP*UP
      DNTT(8)=.125D0*SM*UP
C
      DNUU(1)=-.125D0*SM*TM
      DNUU(2)=-.125D0*SP*TM
      DNUU(3)=-.125D0*SP*TP
      DNUU(4)=-.125D0*SM*TP
      DNUU(5)= .125D0*SM*TM
      DNUU(6)= .125D0*SP*TM
      DNUU(7)= .125D0*SP*TP
      DNUU(8)= .125D0*SM*TP
C
      SUM1=0.0
      SUM2=0.0
      SUM3=0.0
      SUM4=0.0
      SUM5=0.0
      SUM6=0.0
      SUM7=0.0
      SUM8=0.0
      SUM9=0.0
C
      DO 290 I=1,8
      SUM1=SUM1+XQ(I)*DNSS(I)
      SUM2=SUM2+YQ(I)*DNSS(I)
      SUM3=SUM3+ZQ(I)*DNSS(I)
      SUM4=SUM4+XQ(I)*DNTT(I)
      SUM5=SUM5+YQ(I)*DNTT(I)
      SUM6=SUM6+ZQ(I)*DNTT(I)
      SUM7=SUM7+XQ(I)*DNUU(I)
      SUM8=SUM8+YQ(I)*DNUU(I)
      SUM9=SUM9+ZQ(I)*DNUU(I)
  290 CONTINUE
C
      DJAC=SUM1*(SUM5*SUM9-SUM6*SUM8) + SUM2*(SUM6*SUM7-SUM4*SUM9) +
     1 SUM3*(SUM4*SUM8-SUM5*SUM7)
C
      DJACI=1.0D0/DJAC
```

```
C                                                                       BASE 470
      SUMI1=DJACI*(SUM5*SUM9-SUM6*SUM8)                                 BASE 475
      SUMI2=DJACI*(SUM3*SUM8-SUM2*SUM9)                                 BASE 480
      SUMI3=DJACI*(SUM2*SUM6-SUM3*SUM5)                                 BASE 485
      SUMI4=DJACI*(SUM6*SUM7-SUM4*SUM9)                                 BASE 490
      SUMI5=DJACI*(SUM1*SUM9-SUM3*SUM7)                                 BASE 495
      SUMI6=DJACI*(SUM3*SUM4-SUM1*SUM6)                                 BASE 500
      SUMI7=DJACI*(SUM4*SUM8-SUM5*SUM7)                                 BASE 505
      SUMI8=DJACI*(SUM2*SUM7-SUM1*SUM8)                                 BASE 510
      SUMI9=DJACI*(SUM1*SUM5-SUM2*SUM4)                                 BASE 515
C                                                                       BASE 520
      DO 390 I=1,8                                                      BASE 525
      DNX(I)=SUMI1*DNSS(I) + SUMI2*DNTT(I) + SUMI3*DNUU(I)              BASE 530
      DNY(I)=SUMI4*DNSS(I) + SUMI5*DNTT(I) + SUMI6*DNUU(I)              BASE 535
      DNZ(I)=SUMI7*DNSS(I) + SUMI8*DNTT(I) + SUMI9*DNUU(I)              BASE 540
  390 CONTINUE                                                          BASE 545
C                                                                       BASE 550
      RETURN                                                            BASE 555
      END                                                               BASE 560
```

```
      SUBROUTINE BC(CMATRX,RLD,LRN, IE,X,Y,Z, AKR,PROP, DCOSB,ISB,NPBB, BC   005
     1 QCB,ISC,ICTYP, QNB,ISN,INTYP, FLX,HCON,HMIN,NPFLX,NPCON,NPMIN,   BC   010
     2 HDB,IDTYP,NPDB, KSS)                                             BC   015
C                                                                       BC   020
C********1*********2*********3*********4*********5*********6*********7**BC   025
C                                                                       BC   030
C ------- TO APPLY CAUCHY, NEUMANN, VARIABLE, AND DIRICHLET BOUNDARY    BC   035
C ------- CONDITIONS.                                                   BC   040
C                                                                       BC   045
C********1*********2*********3*********4*********5*********6*********7**BC   050
C                                                                       BC   055
C ------- INPUT: X(NNP), Y(NNP), Z(NNP), IE(NEL,9), LRN(JBAND,NNP),     BC   060
C -------        AKR(8,NEL), PROP(NMPPM,NMAT), DCOSB(3,NBES),           BC   065
C -------        ISB(6,NBES), NPBB(NBNP), QCB(NCPR), ISC(5,NCES),       BC   070
C -------        ICTYP(NCES), QNB(NNPR), ISN(5,NNES), INTYP(NNES),      BC   075
C -------        FLX(NVNP), HCON(NVNP), HMIN(NVNP), NPFLX(NVNP),        BC   080
C  ------        NPCON(NVNP), NPMIN(NVNP), HDB(NDPR), IDTYP(NDNP),      BC   085
C -------        NPDB(NDNP), KSS.                                       BC   090
C                                                                       BC   095
C ------- OUTPUT: MODIFIED CMATRX(NNP,JBAND), RLD(NNP).                 BC   100
C                                                                       BC   105
C********1*********2*********3*********4*********5*********6*********7**BC   110
C                                                                       BC   115
      IMPLICIT REAL*8(A-H,O-Z)                                          BC   120
```

```
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI,NDTCHG
C
      COMMON /CCBC/ MXCNP,MXCES,MXCPR,MXCDP,NCNP,NCES,NCPR,NCDP
      COMMON /CNBC/ MXNNP,MXNES,MXNPR,MXNDP,NNNP,NNES,NNPR,NNDP
      COMMON /CVBC/ MXVES,MXVNP,MXRPR,MXRDP,NVES,NVNP,NRPR,NRDP
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP
C
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM
CC
      DIMENSION CMATRX(MAXNP,JBAND),RLD(MAXNP),LRN(JBAND,MAXNP)
      DIMENSION X(MAXNP),Y(MAXNP),Z(MAXNP),IE(MAXEL,9)
      DIMENSION AKR(8,MAXEL),PROP(MXMPPM,MAXMAT)
      DIMENSION DCOSB(3,MAXBES),ISB(6,MAXBES),NPBB(MAXBNP)
C
      DIMENSION FLX(MXVNP),HCON(MXVNP),NPCON(MXVNP),NPFLX(MXVNP)
      DIMENSION HMIN(MXVNP),NPMIN(MXVNP)
C
      DIMENSION HDB(MXDPR),IDTYP(MXDNP),NPDB(MXDNP)
      DIMENSION QCB(MXCPR),ICTYP(MXCES),ISC(5,MXCES)
C
      DIMENSION QNB(MXNPR),INTYP(MXNES),ISN(5,MXNES)
C
      DIMENSION R1Q(4),R2Q(4),XQ(4),YQ(4),ZQ(4),F1Q(4),F2Q(4)
C
      DIMENSION KGB(4,6)
C
      DATA KGB/1,4,8,5, 1,2,6,5, 2,3,7,6, 4,3,7,8, 1,2,3,4, 5,6,7,8/
C
      AGRAV=DFLOAT(KGRAV)
C
C ******* APPLY CAUCHY BOUNDARY CONDITIONS
C
      IF(NCES.LE.0) GO TO 300
      DO 260 MP=1,NCES
      ITYP=ICTYP(MP)
      QCBMP=QCB(MP)
C
      MPB=ISC(5,MP)
      LS=ISB(5,MPB)
      M=ISB(6,MPB)
C
      DO 210 IQ=1,4
      I=KGB(IQ,LS)
      NI=IE(M,I)
      XQ(IQ)=X(NI)
      YQ(IQ)=Y(NI)
      ZQ(IQ)=Z(NI)
```

```
      F1Q(IQ)=QCBMP
      F2Q(IQ)=0.0
  210 CONTINUE
C
      CALL Q4S(R1Q,R2Q,XQ,YQ,ZQ,F1Q,F2Q)
C
      DO 230 IQ=1,4
      I=KGB(IQ,LS)
      NI=IE(M,I)
      RLD(NI)=RLD(NI)-R1Q(IQ)
  230 CONTINUE
  260 CONTINUE
C
C ******* APPLY NEUMANN BOUNDARY CONDITIONS
C
  300 IF(NNES.EQ.0) GO TO 500
      DO 390 MP=1,NNES
      ITYP=INTYP(MP)
      QNBMP=QNB(MP)
C
      MPB=ISN(5,MP)
      LS=ISB(5,MPB)
      M=ISB(6,MPB)
C
      MTYP=IE(M,9)
      AKZX=PROP(5,MTYP)
      AKZY=PROP(6,MTYP)
      AKZZ=PROP(3,MTYP)
C
      DO 310 IQ=1,4
      I=KGB(IQ,LS)
      NI=IE(M,I)
      XQ(IQ)=X(NI)
      YQ(IQ)=Y(NI)
      ZQ(IQ)=Z(NI)
      F1Q(IQ)=QNBMP
      F2Q(IQ)=AKR(IQ,M)*(DCOSB(1,MPB)*AKZX+DCOSB(2,MPB)*AKZY+
     1 DCOSB(3,MPB)*AKZZ)
  310 CONTINUE
C
      CALL Q4S(R1Q,R2Q,XQ,YQ,ZQ,F1Q,F2Q)
C
C -------- MODIFY LOAD VECTOR DUE TO NEUMANN FLUX AND GRAVITY TERM.
C
      DO 360 IQ=1,4
      I=KGB(IQ,LS)
      NI=IE(M,I)
      RLD(NI)=RLD(NI)-R1Q(IQ)+R2Q(IQ)
  360 CONTINUE
```

```
C
  390 CONTINUE
C
C ******* APPLY VARIABLE (RAINFALL-SEEPAGE) BOUNDARY CONDITIONS
C
  500 IF(NVES.EQ.0) GO TO 600
C
C -------- CAUCHY PART OF VARIABLE BOUNDARY CONDITIONS
C
      DO 410 NPP=1,NVNP
         NP=NPFLX(NPP)
         IF (NP.EQ.0) GO TO 410
         RLD(NP)=RLD(NP)-FLX(NPP)
  410 CONTINUE
C
C -------- DIRICHLET PART OF VARIABLE BOUNDARY CONDITIONS
C
      DO 490 NPP=1,NVNP
      NI=NPCON(NPP)
      IF(NI.NE.0) GO TO 450
      NI=NPMIN(NPP)
      IF(NI.NE.0) GO TO 460
      GO TO 490
  450 BB=HCON(NPP)
      GO TO 470
  460 BB=HMIN(NPP)
C
  470 RLD(NI)=BB
      DO 480 I=1,JBAND
      CMATRX(NI,I)=0.0
      IB=LRN(I,NI)
      IF(IB.EQ.NI) CMATRX(NI,I)=1.0D0
  480 CONTINUE
  490 CONTINUE
C
C ******* APPLY DIRICHLET BOUNDARY CONDITIONS
C
  600 IF(NDNP.EQ.0) GO TO 900
      DO 740 NPP=1,NDNP
      NP=NPDB(NPP)
      NI=NPBB(NP)
      ITYP=IDTYP(NPP)
      BB=HDB(ITYP)-Z(NI)*AGRAV
      RLD(NI)=BB
      DO 710 I=1,JBAND
      CMATRX(NI,I)=0.0
      IB=LRN(I,NI)
      IF(IB.EQ.NI) CMATRX(NI,I)=1.0D0
  710 CONTINUE
  740 CONTINUE
```

```
C                                                                       BC   865
  900 CONTINUE                                                          BC   870
C                                                                       BC   875
      RETURN                                                            BC   880
      END                                                               BC   885
```

```
      SUBROUTINE Q4S(R1Q,R2Q,XQ,YQ,ZQ,F1Q,F2Q)                          Q4S  005
C                                                                       Q4S  010
C********1*********2*********3*********4*********5*********6*********7**Q4S  015
C                                                                       Q4S  020
C ------- TO COMPUTE BOUNDARY SURFACE LOAD VECTOR OVER A BOUNDARY       Q4S  025
C ------- SURFACE.                                                      Q4S  030
C                                                                       Q4S  035
C********1*********2*********3*********4*********5*********6*********7**Q4S  040
C                                                                       Q4S  045
C ------- INPUT: XQ(4), YQ(4), ZQ(4),F1Q(4), F2Q(4).                    Q4S  050
C                                                                       Q4S  055
C ------- OUTPUT: R1Q(4), R2Q(4).                                       Q4S  060
C                                                                       Q4S  065
C********1*********2*********3*********4*********5*********6*********7**Q4S  070
C                                                                       Q4S  075
      IMPLICIT REAL*8 (A-H,O-Z)                                         Q4S  080
      REAL*8 N(4)                                                       Q4S  085
C                                                                       Q4S  090
      DIMENSION R1Q(4),R2Q(4),XQ(4),YQ(4),ZQ(4),F1Q(4),F2Q(4)           Q4S  095
      DIMENSION S(4),T(4),DNSS(4),DNTT(4)                               Q4S  100
C                                                                       Q4S  105
      DATA P/ 0.577350269189626D0/, S/-1.0D+00, 1.0D+00, 1.0D+00,       Q4S  110
     >- 1.0D+00/, T/-1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00/                Q4S  115
C                                                                       Q4S  120
C ------- INITIATE MATRICES RQ(IQ)                                      Q4S  125
C                                                                       Q4S  130
      DO 100 IQ=1,4                                                     Q4S  135
      R1Q(IQ)=0.0                                                       Q4S  140
  100 R2Q(IQ)=0.0                                                       Q4S  145
C                                                                       Q4S  150
C ------- SUMMATION OF THE INTEGRAND OVER THE GAUSSIAN POINTS           Q4S  155
C                                                                       Q4S  160
      DO 490 KG=1,4                                                     Q4S  165
C                                                                       Q4S  170
C ------- DETERMINE LOACAL COORDINATE OF GAUSSIAN POINT KG              Q4S  175
C                                                                       Q4S  180
      SS=P*S(KG)                                                        Q4S  185
      TT=P*T(KG)                                                        Q4S  190
      SM=1.0D0-SS                                                       Q4S  195
      SP=1.0D0+SS                                                       Q4S  200
```

```
      TM=1.0D0-TT                                                       Q4S  205
      TP=1.0D0+TT                                                       Q4S  210
C                                                                       Q4S  215
      N(1)=0.25D0*SM*TM                                                 Q4S  220
      N(2)=0.25D0*SP*TM                                                 Q4S  225
      N(3)=0.25D0*SP*TP                                                 Q4S  230
      N(4)=0.25D0*SM*TP                                                 Q4S  235
      DNSS(1)=-0.25D0*TM                                                Q4S  240
      DNSS(2)= 0.25D0*TM                                                Q4S  245
      DNSS(3)= 0.25D0*TP                                                Q4S  250
      DNSS(4)=-0.25D0*TP                                                Q4S  255
      DNTT(1)=-0.25D0*SM                                                Q4S  260
      DNTT(2)=-0.25D0*SP                                                Q4S  265
      DNTT(3)= 0.25D0*SP                                                Q4S  270
      DNTT(4)= 0.25D0*SM                                                Q4S  275
C                                                                       Q4S  280
      DXDSS=0.0D0                                                       Q4S  285
      DYDSS=0.0D0                                                       Q4S  290
      DZDSS=0.0D0                                                       Q4S  295
      DXDTT=0.0D0                                                       Q4S  300
      DYDTT=0.0D0                                                       Q4S  305
      DZDTT=0.0D0                                                       Q4S  310
      DO 290 IQ=1,4                                                     Q4S  315
      DXDSS=DXDSS+XQ(IQ)*DNSS(IQ)                                       Q4S  320
      DYDSS=DYDSS+YQ(IQ)*DNSS(IQ)                                       Q4S  325
      DZDSS=DZDSS+ZQ(IQ)*DNSS(IQ)                                       Q4S  330
      DXDTT=DXDTT+XQ(IQ)*DNTT(IQ)                                       Q4S  335
      DYDTT=DYDTT+YQ(IQ)*DNTT(IQ)                                       Q4S  340
      DZDTT=DZDTT+ZQ(IQ)*DNTT(IQ)                                       Q4S  345
  290 CONTINUE                                                          Q4S  350
C                                                                       Q4S  355
      DETZ=DXDSS*DYDTT-DYDSS*DXDTT                                      Q4S  360
      DETY=-DXDSS*DZDTT+DZDSS*DXDTT                                     Q4S  365
      DETX=DYDSS*DZDTT-DZDSS*DYDTT                                      Q4S  370
      DET=DSQRT(DETX*DETX+DETY*DETY+DETZ*DETZ)                          Q4S  375
C                                                                       Q4S  380
C ------- ACCUMULATE THE SUMS TO OBTAIN THE MATRIX INTEGRALS RQ(IQ)     Q4S  385
C                                                                       Q4S  390
      F1K=0.0D0                                                         Q4S  395
      F2K=0.0                                                           Q4S  400
      DO 350 IQ=1,4                                                     Q4S  405
      F1K=F1K+F1Q(IQ)*N(IQ)                                             Q4S  410
      F2K=F2K+F2Q(IQ)*N(IQ)                                             Q4S  415
  350 CONTINUE                                                          Q4S  420
C                                                                       Q4S  425
      DO 390 IQ=1,4                                                     Q4S  430
      R1Q(IQ)=R1Q(IQ)+N(IQ)*F1K*DET                                     Q4S  435
      R2Q(IQ)=R2Q(IQ)+N(IQ)*F2K*DET                                     Q4S  440
  390 CONTINUE                                                          Q4S  445
```

```
C                                                                       Q4S  450
  490 CONTINUE                                                          Q4S  455
C                                                                       Q4S  460
      RETURN                                                            Q4S  465
      END                                                               Q4S  470
```

```
      SUBROUTINE BLKITR(C,CW, CMTRXL,RLDL, CMTRXG,RLDG, GNLR,LNOJCN,    BLKI 005
     1 NNPLR,LMAXDF, TOLB,NITER,IBUG,KPR,OME)                           BLKI 010
C                                                                       BLKI 015
C********1*********2*********3*********4*********5*********6*********7**BLKI 020
C                                                                       BLKI 025
C ------- TO SOLVE THE MATRIX EQUATION WITH BLOCK ITERATION.  FIRST,    BLKI 030
C ------- THE BLOCK MATRIX EQUATION IS ASSEMBLED OUT OF THE GLOBAL      BLKI 035
C ------- MATRIX EQUATION.  THEN THE INTRA-BOUNDARY CONDITIONS ARE      BLKI 040
C ------- IMPLEMENTED.  FINALLY, THE BLOCK MATRIX EQUATION IS SOLVED    BLKI 045
C ------- WITH DIRECT BAND MATRIX SOLVER.                               BLKI 050
C                                                                       BLKI 055
C********1*********2*********3*********4*********5*********6*********7**BLKI 060
C                                                                       BLKI 065
C ------- INPUT: CW(NNP), CMTRXG(NNP,JBAND), RLDG(NNP),                 BLKI 070
C -------        GNLR(NTNNP,NREGN), LNOJCN(JBAND,NNPLR(K),NREGN),       BLKI 075
C -------        NNPLR(NREGN), LMAXDF(NREGN), TOLB, NITER, IBUG, KPR,   BLKI 080
C -------        OME.                                                   BLKI 085
C                                                                       BLKI 090
C ------- OUTPUT: C(NNP), CW(NNP).                                      BLKI 095
C                                                                       BLKI 100
C-------- INPUTING WORKING ARRAYS: CMTRXL(LMXNP,LMXBW), RLDL(LMXNP).    BLKI 105
C                                                                       BLKI 110
C********1*********2*********3*********4*********5*********6*********7**BLKI 115
C                                                                       BLKI 120
      IMPLICIT REAL*8(A-H,O-Z)                                          BLKI 125
      INTEGER*4 GNLR                                                    BLKI 130
C                                                                       BLKI 135
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC      BLKI 140
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI,NDTCHG                 BLKI 145
      COMMON /LGEOM/ LTMXNP,LMXNP,LMXBW,MXREGN,NREGN                    BLKI 150
C                                                                       BLKI 155
      DIMENSION C(MAXNP),CW(MAXNP)                                      BLKI 160
      DIMENSION CMTRXL(LMXNP,LMXBW),RLDL(LMXNP)                         BLKI 165
      DIMENSION CMTRXG(MAXNP,JBAND),RLDG(MAXNP)                         BLKI 170
      DIMENSION GNLR(LTMXNP,MXREGN),LNOJCN(JBAND,LMXNP,MXREGN)          BLKI 175
      DIMENSION NNPLR(MXREGN),LMAXDF(MXREGN)                            BLKI 180
```

```
C
C ------ PUT ZERO-TH ITERATE INTO ARRAY C
C
      DO 110 NP=1,NNP
  110 C(NP)=CW(NP)
C
C ****** START INTERATION LOOP
C
      IF(IBUG.NE.0 .AND. KPR.NE.0) PRINT 1000
C
      DO 690 IT=1,NITER
C
C ------ FOR EACH ITERATION, SOLVE FOR NREGN REGIONS
C
      DO 590 K=1,NREGN
C
      LHALFB=LMAXDF(K)
      LIHBP=LHALFB+1
      IBAND=2*LHALFB+1
      LNNP=NNPLR(K)
C
C ------- PUT GLOBAL LOAD VECTOR INTO CORRESPONDING LOCAL LOAD VECTOR
C ------- AND INITIATE THE LOCAL MATRIX CMTRXL(LNNP,IBAND) WHERE
C
      DO 210 LI=1,LNNP
      NP=GNLR(LI,K)
      RLDL(LI)=RLDG(NP)
      DO 210 J=1,IBAND
      CMTRXL(LI,J)=0.0
  210 CONTINUE
C
C ------ ASSEMBLE LOCAL COEFFICIENT MATRIX CMTRXL(LMXNP,LMXBW) FROM THE
C ------ GLOBAL COEFFICIENT MATRIX CMTRXG(MAXNP,JBAND) AND INCORPORATE
C ------ INTERFACIAL DIRICHLET BOUNDARY CONDITIONS INTO RLDL(LMXNP).
C
      DO 490 LI=1,LNNP
      NI=GNLR(LI,K)
C
      DO 390 J=1,JBAND
      LJ=LNOJCN(J,LI,K)
      NJ=GNLR(LJ,K)
C
      IF(LJ.LE.0) GO TO 390
C
      IF(LJ.GT.LNNP) GO TO 380
      LJB=LJ-LI+LIHBP
      CMTRXL(LI,LJB)=CMTRXG(NI,J)
      GO TO 390
```

```
C
  380 RLDL(LI)=RLDL(LI)-CMTRXG(NI,J)*C(NJ)
C
  390 CONTINUE
C
  490 CONTINUE
C
C ------ SOLVE THE BLOCK EQUATIONS
C
      CALL SOLVE(1,CMTRXL,RLDL,LNNP,LHALFB,LMXNP,LMXBW)
      CALL SOLVE(2,CMTRXL,RLDL,LNNP,LHALFB,LMXNP,LMXBW)
C
C ------ PUT THE NEWLY OBTAINED BLOCK SOLTUION INTO THE GLOBAL SOLUTION.
C
      DO 560 LI=1,LNNP
      NP=GNLR(LI,K)
  560 C(NP)=RLDL(LI)*OME + (1.0D0-OME)*C(NP)
C
  590 CONTINUE
C
C ------ CHECK IF THE CONVERGENT SOLUTION IS OBTAINED?
C
      DIFMAX=0.0
      NOCCUR=1
      DO 660 NP=1,NNP
      DIF=C(NP)-CW(NP)
      DIF=DABS(DIF)
      IF(DIF.LE.DIFMAX) GO TO 660
      DIFMAX=DIF
      NOCCUR=NP
  660 CONTINUE
C
C ------- UP DATA THE ITERATE
C
      DO 680 NP=1,NNP
  680 CW(NP)=C(NP)
C
C------- PRINT ITERATION INFORMATION
C
      IF(IBUG.NE.0 .AND. KPR.NE.0) PRINT 1100, IT,DIFMAX,TOLB,NOCCUR
C
      IF(IT.EQ.1) GO TO 690
C
      IF(DIFMAX.LT.TOLB) GO TO 990
C
  690 CONTINUE
```

```
C
      PRINT 2000, IT,NITER,DIFMAX,TOLB,NOCCUR
C
C
  990 CONTINUE
C
 1000 FORMAT(1H0//75X,'   IT   DIFMAX       TOLB        NOCCUR'/75X,
     1 '   --   ------        ----         ------')
 1100 FORMAT(1H ,75X,I5,2D12.4,I10)
 2000 FORMAT(1H0/60X,' *** WARNING: NO CONVERGENCE IN BLKI AFTER  ',I4,
     1 '    ITERATIONS'/60X,'  NITER =',I4,'  DIFMAX =',D11.4,
     2 '    TOLB  =',D11.4,'  NOCCUR =',I4)
C
      RETURN
      END
```

```
      SUBROUTINE SOLVE(KKK,C,R,NNP,IHALFB,MAXNP,MAXBW)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO SOVE A MATRIX EQUATION WITH BAND MATRIX SOLVER.
C
C********1*********2*********3*********4*********5*********5*********7**
C
C ------- INPUT: C(NEQ,NBAND), R(NEQ), NNP, IHALFB, KKK,
C -------          WHERE NNP=NEQ AND IHALFB=(NBAND-1)/2
C
C ------- OUTPUT: C(NEQ).
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION C(MAXNP,MAXBW),R(MAXNP)
C
      IHBP=IHALFB+1
C
C  IF KKK = 1, THEN TRIANGULARIZE THE BAND MATRIX C(NP,IB), BUT
C  IF KKK = 2, THEN SIMPLY SOLVE WITH THE RIGHT-HAND SIDE R(NP)
C
      IF (KKK.EQ.2) GO TO 50
```

```
C
C  TRIANGULARIZE MATRIX C(NP,IB)
C
      NU-NNP-IHALFB
      DO 20 NI-1,NU
         PIVOTI-1.0D0/C(NI,IHBP)
         NJ-NI+1
         IB-IHBP
         NK-NI+IHALFB
         DO 10 NL-NJ,NK
            IB-IB-1
            A--C(NL,IB)*PIVOTI
            C(NL,IB)-A
            JB-IB+1
            KB-IB+IHALFB
            LB-IHBP-IB
            DO 10 MB-JB,KB
               NB-LB+MB
   10          C(NL,MB)-C(NL,MB)+A*C(NI,NB)
   20    CONTINUE
      NR-NU+1
      NU-NNP-1
      NK-NNP
      IF(NR.GT.NU) RETURN
      DO 40 NI-NR,NU
         PIVOTI-1.0D0/C(NI,IHBP)
         NJ-NI+1
         IB-IHBP
         DO 30 NL-NJ,NK
            IB-IB-1
            A--C(NL,IB)*PIVOTI
            C(NL,IB)-A
            JB-IB+1
            KB-IB+IHALFB
            LB-IHBP-IB
            DO 30 MB-JB,KB
               NB-LB+MB
   30          C(NL,MB)-C(NL,MB)+A*C(NI,NB)
   40    CONTINUE
      RETURN
C
C  MODIFY LOAD VECTOR R(NP)
C
   50 NU-NNP+1
      IBAND-2*IHALFB+1
      DO 70 NI-2,IHBP
         IB-IHBP-NI+1
         NJ-1
         SUM-0.0
```

```
         DO 60 JB-IB,IHALFB                                             SOLV 375
            SUM-SUM+C(NI,JB)*R(NJ)                                      SOLV 380
   60       NJ-NJ+1                                                     SOLV 385
   70    R(NI)-R(NI)+SUM                                                SOLV 390
      IB-1                                                              SOLV 395
      NL-IHBP+1                                                         SOLV 400
      DO 90 NI-NL,NNP                                                   SOLV 405
         NJ-NI-IHBP+1                                                   SOLV 410
         SUM-0.0                                                        SOLV 415
         DO 80 JB-IB,IHALFB                                             SOLV 420
            SUM-SUM+C(NI,JB)*R(NJ)                                      SOLV 425
   80       NJ-NJ+1                                                     SOLV 430
   90    R(NI)-R(NI)+SUM                                                SOLV 435
C                                                                       SOLV 440
C  BACK SOLVE                                                           SOLV 445
C                                                                       SOLV 450
      R(NNP)-R(NNP)/C(NNP,IHBP)                                         SOLV 455
      DO 110 IB-2,IHBP                                                  SOLV 460
         NI-NU-IB                                                       SOLV 465
         NJ-NI                                                          SOLV 470
         MB-IHALFB+IB                                                   SOLV 475
         SUM-0.0                                                        SOLV 480
         DO 100 JB-NL,MB                                                SOLV 485
            NJ-NJ+1                                                     SOLV 490
  100       SUM-SUM+C(NI,JB)*R(NJ)                                      SOLV 495
  110    R(NI)-(R(NI)-SUM)/C(NI,IHBP)                                   SOLV 500
      MB-IBAND                                                          SOLV 505
      DO 130 IB-NL,NNP                                                  SOLV 510
         NI-NU-IB                                                       SOLV 515
         NJ-NI                                                          SOLV 520
         SUM-0.0                                                        SOLV 525
         DO 120 JB-NL,MB                                                SOLV 530
            NJ-NJ+1                                                     SOLV 535
  120       SUM-SUM+C(NI,JB)*R(NJ)                                      SOLV 540
  130    R(NI)-(R(NI)-SUM)/C(NI,IHBP)                                   SOLV 545
      RETURN                                                            SOLV 550
      END                                                               SOLV 555
```

```
      SUBROUTINE SFLOW(X,Y,Z,IE, H,HP,VX,VY,VZ,TH,DTH,
     1 BFLX,BFLXP, DCOSB,ISB,NPBB, MSEL,SOS,ISTYP,NPW,WSS,IWTYP,
     2 NPVB,NPDB,NPCB,NPNB,   DELT, KFLOW)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO COMPUTE WATER FLUXES, INCREMENTAL FLOW, AND ACCUMULATED
C ------- FLOW THROUGH ALL TYPES OF BOUNDARIES AND CHANGE OF WATER
C ------- STORED IN THE REGION OF INTEREST.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: X(NNP), Y(NNP), Z(NNP), IE(NEL,9), H(NNP), HP(NNP),
C -------        VX(NNP), VY(NNP), VZ(NNP), TH(8,NEL), DTH(8,NEL),
C -------        DCOSB(3,NBES), ISB(6,NBES), NPBB(NBNP), MSEL(NSEL),
C -------        SOS(NSPR), ISTYP(NSEL), NPW(NWNP), WSS(NWPR),
C -------        IWTYP(NWNP), NPVB(NVNP), NPDB(NDNP), NPCB(NCNP),
C -------        NPNB(NNNP), DELT, KFLOW.
C
C ------- OUTPUT: FRATE(10), FLOW(10), TFLOW(10).
C
C ------- WORKING ARRAYS: BFLX(NBNP), BFLXP(NBNP).
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,GRAV,NTI
C
      COMMON /CS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP
      COMMON /CW/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP
C
      COMMON /CVBC/ MXVES,MXVNP,MXRPR,MXRDP,NVES,NVNP,NRPR,NRDP
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP
      COMMON /CCBC/ MXCNP,MXCES,MXCPR,MXCDP,NCNP,NCES,NCPR,NCDP
      COMMON /CNBC/ MXNNP,MXNES,MXNPR,MXNDP,NNNP,NNES,NNPR,NNDP
C
      COMMON /CFLOW/ FRATE(10),FLOW(10),TFLOW(10)
C
      DIMENSION X(MAXNP),Y(MAXNP),Z(MAXNP),IE(MAXEL,9)
C
      DIMENSION H(MAXNP),HP(MAXNP),VX(MAXNP),VY(MAXNP),VZ(MAXNP)
      DIMENSION TH(8,MAXEL),DTH(8,MAXEL)
C
      DIMENSION BFLX(MAXBNP),BFLXP(MAXBNP)
      DIMENSION DCOSB(3,MAXBES),ISB(6,MAXBES),NPBB(MAXBNP)
C
      DIMENSION SOS(MXSPR),MSEL(MXSEL),ISTYP(MAXEL)
      DIMENSION WSS(MXWPR),NPW(MXWNP),IWTYP(MXWNP)
```

```
C
      DIMENSION NPVB(MXVNP),NPDB(MXDNP),NPCB(MXCNP),NPNB(MXNNP)
C
      DIMENSION XQ(8),YQ(8),ZQ(8),DHQ(8),THG(8)
      DIMENSION R1Q(4),R2Q(4),XXQ(4),YYQ(4),ZZQ(4),F1Q(4),F2Q(4)
C
      DIMENSION KGB(4,6)
C
      DATA KGB/1,4,8,5, 1,2,6,5, 2,3,7,6, 4,3,7,8, 1,2,3,4, 5,6,7,8/
      DATA QSOS/0.0D0/
C
      DO 110 NP=1,NBNP
      BFLXP(NP)=BFLX(NP)
  110 BFLX(NP)=0.0
C
C ******* CALCULATE VOLUMETRIC FLOW RATE THROUGH ALL BOUNDARY NODES.
C
      DO 170 MP=1,NBES
C
      LS=ISB(5,MP)
      M=ISB(6,MP)
C
      DO 120 IQ=1,4
      I=KGB(IQ,LS)
      NI=IE(M,I)
      XXQ(IQ)=X(NI)
      YYQ(IQ)=Y(NI)
      ZZQ(IQ)=Z(NI)
      F1Q(IQ)=DCOSB(1,MP)*VX(NI)+DCOSB(2,MP)*VY(NI)+DCOSB(3,MP)*VZ(NI)
      F2Q(IQ)=0.0
  120 CONTINUE
C
      CALL Q4S(R1Q,R2Q,XXQ,YYQ,ZZQ,F1Q,F2Q)
C
      DO 140 IQ=1,4
      NII=ISB(IQ,MP)
      BFLX(NII)=BFLX(NII)+R1Q(IQ)
  140 CONTINUE
  170 CONTINUE
C
      IF (KFLOW.GT.0) GO TO 200
      DO 180 NP=1,NBNP
  180 BFLXP(NP)=BFLX(NP)
C
      DO 190 I=1,9
  190 TFLOW(I)=0.0
```

```
C                                                                       SFLO 485
C ******* DETERMINE TOTAL FLOWS AND TOTAL FLOW RATES THROUGH VARIOUS     SFLO 490
C ******* TYPES OF BOUNDARIES, STARTING WITH THE NET FLOWS THROUGH THE   SFLO 495
C ******* ENTIRE BOUNDARY.                                              SFLO 500
C                                                                       SFLO 505
  200 SUM-0.                                                            SFLO 510
      SUMP-0.                                                           SFLO 515
      DO 210 NP-1,NBNP                                                  SFLO 520
      SUM-SUM+BFLX(NP)                                                  SFLO 525
  210 SUMP-SUMP+BFLXP(NP)                                               SFLO 530
C                                                                       SFLO 535
      FRATE(7)-SUM                                                      SFLO 540
      FLOW(7)-0.5DO*(SUM+SUMP)*DELT                                     SFLO 545
C                                                                       SFLO 550
C ******* THE DIRICHLET BOUNDARY                                        SFLO 555
C                                                                       SFLO 560
      FRATE(1)-0.                                                       SFLO 565
      FLOW(1)-0.                                                        SFLO 570
      IF(NDNP.LE.0) GO TO 400                                           SFLO 575
      SUM-0.                                                            SFLO 580
      SUMP-0.                                                           SFLO 585
      DO 330 NPP-1,NDNP                                                 SFLO 590
      NII-NPDB(NPP)                                                     SFLO 595
      SUM-SUM+BFLX(NII)                                                 SFLO 600
      SUMP-SUMP+BFLXP(NII)                                              SFLO 605
  330 CONTINUE                                                          SFLO 610
      FRATE(1)-SUM                                                      SFLO 615
      FLOW(1)-0.5DO*(SUM+SUMP)*DELT                                     SFLO 620
C                                                                       SFLO 625
C                                                                       SFLO 630
C ******* THE CAUCHY BOUNDARY                                           SFLO 635
C                                                                       SFLO 640
  400 FRATE(2)-0.0                                                      SFLO 645
      FLOW(2)-0.0                                                       SFLO 650
      IF(NCNP.LE.0) GO TO 500                                           SFLO 655
      SUM-0.0                                                           SFLO 660
      SUMP-0.0                                                          SFLO 665
      DO 430 NPP-1,NCNP                                                 SFLO 670
      NII-NPCB(NPP)                                                     SFLO 675
      SUM-SUM+BFLX(NII)                                                 SFLO 680
      SUMP-SUMP+BFLXP(NII)                                              SFLO 685
  430 CONTINUE                                                          SFLO 690
      FRATE(2)-SUM                                                      SFLO 695
      FLOW(2)-0.5DO*(SUM+SUMP)*DELT                                     SFLO 700
```

```
C
C
C ******* THE NEUMANN BOUNDARY
C
  500 FRATE(3)-0.
      FLOW(3)-0.
      IF(NNNP.LE.0) GO TO 600
      SUM-0.
      SUMP-0.
      DO 530 NPP-1,NNNP
      NII-NPNB(NPP)
      SUM-SUM+BFLX(NII)
      SUMP-SUMP+BFLXP(NII)
  530 CONTINUE
      FRATE(3)-SUM
      FLOW(3)-0.5D0*(SUM+SUMP)*DELT
C
C ******* THE RAINFALL-SEEPAGE BOUNDARY
C
  600 FRATE(4)-0.
      FLOW(4)-0.
      FRATE(5)-0.
      FLOW(5)-0.
      IF(NVNP.LE.0) GO TO 700
      SUMS-0.
      SUMSP-0.
      SUMR-0.
      SUMRP-0.
      DO 640 NPP-1,NVNP
      NII-NPVB(NPP)
      BFLXA-BFLX(NII)
      IF (BFLXA.LT.0.D0) GO TO 630
      SUMS-SUMS+BFLX(NII)
      SUMSP-SUMSP+BFLXP(NII)
      GO TO 640
  630 SUMR-SUMR+BFLX(NII)
      SUMRP-SUMRP+BFLXP(NII)
  640 CONTINUE
      FRATE(4)-SUMS
      FLOW(4)-0.5D0*(SUMS+SUMSP)*DELT
      FRATE(5)-SUMR
      FLOW(5)-0.5D0*(SUMR+SUMRP)*DELT
```

```
C
C ******* THE UNSPECIFIED BOUNDARY, I. E. BOUNDARY WITH ZERO TOTAL FLUX
C
  700 SUM=0.
      SUMP=0.
      DO 710 I=1,5
      SUM=SUM+FRATE(I)
  710 SUMP=SUMP+FLOW(I)
      FRATE(6)=FRATE(7)-SUM
      FLOW(6)=FLOW(7)-SUMP
C
C ******* CALCULATE THE INCREASE IN THE WATER CONTENT AND THE SOURCE
C
      QSOSP=QSOS
      QSOS=0.0
      QTH=0.
      DO 850 M=1,NEL
C
      SOURCE=0.0
      IF(NSEL.EQ.0) GO TO 830
      DO 810 MP=1,NSEL
      MS=MSEL(MP)
      IF(MS.NE.M) GO TO 810
      ITYP=ISTYP(MP)
      SOURCE=SOS(ITYP)
      GO TO 830
  810 CONTINUE
C
  830 DO 840 IQ=1,8
      NP=IE(M,IQ)
      XQ(IQ)=X(NP)
      YQ(IQ)=Y(NP)
      ZQ(IQ)=Z(NP)
      DHQ(IQ)=H(NP)-HP(NP)
      IF(KFLOW.LE.0) DHQ(IQ)=1.0D0
      THG(IQ)=TH(IQ,M)
      IF(KFLOW.GT.0) THG(IQ)=DTH(IQ,M)
  840 CONTINUE
C
      CALL Q8TH(QTHM,QSOSM,DHQ,THG,XQ,YQ,ZQ,SOURCE)
C
      QSOS=QSOS-QSOSM
      QTH=QTH+QTHM
  850 CONTINUE
C
      IF(NWNP.EQ.0) GO TO 870
      DO 860 I=1,NWNP
      ITYP=IWTYP(I)
  860 QSOS=QSOS-WSS(ITYP)
  870 CONTINUE
```

```
C
      IF(KFLOW.GT.0) GO TO 880
      QSOSP-QSOS
C
  880 FRATE(8)-QSOS
      FLOW(8)-0.5D0*(QSOS+QSOSP)*DELT
      FLOW(9)-QTH
      FRATE(9)-FLOW(9)/DELT
      IF(KFLOW.LE.0) FRATE(9)--(FRATE(7)+FRATE(8))
C
      DO 910 I-1,9
  910 TFLOW(I)-TFLOW(I)+FLOW(I)
C
      RETURN
      END
```

```
      SUBROUTINE Q8TH(QTHM,QSOSM,DHQ,THG,XQ,YQ,ZQ,SOURCE)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO COMPUTE WATER CONTENT INTEGRATION AND ELEMENT SOURCE
C ------- INTEGRATION OVER AN ELEMENT.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: DHQ(8), THG(8), XQ(8), YQ(8), ZQ(8), SOURCE.
C
C ------- OUTPUT: QTHM, QSOSM.
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N(8)
C
      DIMENSION DHQ(8),THG(8), XQ(8),YQ(8),ZQ(8)
      DIMENSION PJAB(3,3),DNSS(8),DNTT(8),DNUU(8)
      DIMENSION S(8),T(8),U(8)
C
      DATA P / 0.577350269189626D0/
      DATA S/-1.0D0,1.0D0,1.0D0,-1.0D0, -1.0D0,1.0D0,1.0D0,-1.0D0/
      DATA T/-1.0D0,-1.0D0,1.0D0,1.0D0, -1.0D0,-1.0D0,1.0D0,1.0D0/
      DATA U/-1.0D0,-1.0D0,-1.0D0,-1.0D0, 1.0D0,1.0D0,1.0D0,1.0D0/
```

```
C
      QSOSM=0.0
      QTHM=0.
      DO 490 KG=1,8
C
C ------- DETERMINE LOACAL COORDINATE OF GAUSSIAN POINT KG
C
      SS=P*S(KG)
      TT=P*T(KG)
      UU=P*U(KG)
C
      SM=1.0D0-SS
      SP=1.0D0+SS
      TM=1.0D0-TT
      TP=1.0D0+TT
      UM=1.0D0-UU
      UP=1.0D0+UU
C
C ------- CALCULATE VALUES OF BASIS FUNCTIONS N(IQ).
C
      N(1)=0.125D0*SM*TM*UM
      N(2)=0.125D0*SP*TM*UM
      N(3)=0.125D0*SP*TP*UM
      N(4)=0.125D0*SM*TP*UM
      N(5)=0.125D0*SM*TM*UP
      N(6)=0.125D0*SP*TM*UP
      N(7)=0.125D0*SP*TP*UP
      N(8)=0.125D0*SM*TP*UP
C
      DNSS(1)=-0.125D0*TM*UM
      DNSS(2)= 0.125D0*TM*UM
      DNSS(3)= 0.125D0*TP*UM
      DNSS(4)=-0.125D0*TP*UM
      DNSS(5)=-0.125D0*TM*UP
      DNSS(6)= 0.125D0*TM*UP
      DNSS(7)= 0.125D0*TP*UP
      DNSS(8)=-0.125D0*TP*UP
C
      DNTT(1)=-0.125D0*SM*UM
      DNTT(2)=-0.125D0*SP*UM
      DNTT(3)= 0.125D0*SP*UM
      DNTT(4)= 0.125D0*SM*UM
      DNTT(5)=-0.125D0*SM*UP
      DNTT(6)=-0.125D0*SP*UP
      DNTT(7)= 0.125D0*SP*UP
      DNTT(8)= 0.125D0*SM*UP
```

```
C
      DNUU(1)--0.125D0*SM*TM
      DNUU(2)--0.125D0*SP*TM
      DNUU(3)--0.125D0*SP*TP
      DNUU(4)--0.125D0*SM*TP
      DNUU(5)- 0.125D0*SM*TM
      DNUU(6)- 0.125D0*SP*TM
      DNUU(7)- 0.125D0*SP*TP
      DNUU(8)- 0.125D0*SM*TP
C
      DO 210 J-1,3
      DO 210 I-1,3
  210 PJAB(I,J)-0.0
C
      DO 220 I-1,8
      PJAB(1,1)-PJAB(1,1)+XQ(I)*DNSS(I)
      PJAB(1,2)-PJAB(1,2)+YQ(I)*DNSS(I)
      PJAB(1,3)-PJAB(1,3)+ZQ(I)*DNSS(I)
      PJAB(2,1)-PJAB(2,1)+XQ(I)*DNTT(I)
      PJAB(2,2)-PJAB(2,2)+YQ(I)*DNTT(I)
      PJAB(2,3)-PJAB(2,3)+ZQ(I)*DNTT(I)
      PJAB(3,1)-PJAB(3,1)+XQ(I)*DNUU(I)
      PJAB(3,2)-PJAB(3,2)+YQ(I)*DNUU(I)
      PJAB(3,3)-PJAB(3,3)+ZQ(I)*DNUU(I)
  220 CONTINUE
C
      DJAC-PJAB(1,1)*(PJAB(2,2)*PJAB(3,3)-PJAB(2,3)*PJAB(3,2)) -
     1      PJAB(1,2)*(PJAB(2,1)*PJAB(3,3)-PJAB(2,3)*PJAB(3,1)) +
     2      PJAB(1,3)*(PJAB(2,1)*PJAB(3,2)-PJAB(2,2)*PJAB(3,1))
C
C ------- INTERPOLATE TO OBTAIN WATER CONTENT AT THE GAUSSIAN POINT KG
C
      DHQP-0.0
      DO 390 IQ-1,8
      DHQP-DHQP+DHQ(IQ)*N(IQ)
  390 CONTINUE
C
      THQP-DHQP*THG(KG)
C
C ------- ACCUMULATE THE SUM TO EVALUATE THE INTEGRAL
C
      QSOSM-QSOSM+SOURCE*DJAC
      QTHM-QTHM+THQP*DJAC
  490 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE PRINTT(VX,VY,VZ,H,HT,TH, NPBB,BFLX, NPVB,DCYFLX,NPCON,
     1 NPFLX,NPMIN, SUBHD, TIME,DELT, KPR,KOUT,KDIAG,ITIM)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO OUTPUT FLOWS, PRESSURE HEAD, TOTAL HEAD, WATER CONTENT,
C ------- AND DARCY'S VELOCITY AS SPECIFIED BY THE PARAMETER KPR.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: VX(NNP), VY(NNP), VZ(NNP), H(NNP), HT(NNP), TH(8,NEL),
C -------        NPBB(NBNP), DCYFLX(NVNP), NPCON(NVNP), NPFLX(NVNP),
C -------        NPMIN(NVNP), SUBHD(8), TIME, DELT, KPR, KOUT, KDIAG,
C -------        ITIM.
C
C ------- OUTPUT: LINE PRINT ALL INPUTS IF NEEDED EXCEPT FOR KPR.
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 SUBHD
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI
      COMMON /CVBC/ MXVES,MXVNP,MXRPR,MXRDP,NVES,NVNP,NRPR,NRDP
C
      COMMON /CFLOW/ FRATE(10),FLOW(10),TFLOW(10)
C
      DIMENSION VX(MAXNP),VY(MAXNP),VZ(MAXNP),H(MAXNP),HT(MAXNP)
      DIMENSION TH(8,MAXEL)
      DIMENSION NPBB(MAXBNP),BFLX(MAXBNP)
      DIMENSION NPVB(MXVNP),NPCON(MXVNP),NPFLX(MXVNP),NPMIN(MXVNP)
      DIMENSION DCYFLX(MXVNP)
C
      DIMENSION SUBHD(8)
C
      IF(KPR.EQ.0) RETURN
C
C -------PRINT DIAGNOSTIC FLOW INFORMATION
C
      KDIAG=KDIAG+1
      KDIA=KDIAG-1
      PRINT 1000, KDIA,TIME,DELT,ITIM,(FRATE(I),FLOW(I),TFLOW(I),I=1,9)
      IF (NVNP.EQ.0) GO TO 130
      DO 120 NPP=1,NVNP
      NKK=NPVB(NPP)
  120 DCYFLX(NPP)=BFLX(NKK)
      PRINT 1100
      PRINT 1110, (DCYFLX(NPP),NPP=1,NVNP)
      PRINT 1120, (NPCON(NPP),NPP=1,NVNP)
```

```
      PRINT 1125, (NPMIN(NPP),NPP-1,NVNP)                               PRIN 255
      PRINT 1130,  (NPFLX(NPP),NPP-1,NVNP)                              PRIN 260
  130 IF (KPR.EQ.1) RETURN                                              PRIN 265
C                                                                       PRIN 270
C ------- PRINT PRESSURE HEADS                                          PRIN 275
C                                                                       PRIN 280
      KOUT-KOUT+1                                                       PRIN 285
      KLINE--1                                                          PRIN 290
      PRINT 2000, KOUT,TIME,DELT,JBAND,ITIM,(SUBHD(I),I-1,8)            PRIN 295
      DO 210 NI-1,NNP,8                                                 PRIN 300
         NJMN-NI                                                        PRIN 305
         NJMX-MINO(NI+7,NNP)                                            PRIN 310
         KLINE-KLINE+1                                                  PRIN 315
         IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) PRINT 2000, KOUT,TIME, PRIN 320
     >     DELT,JBAND,ITIM,(SUBHD(I),I-1,8)                             PRIN 325
  210    PRINT 2100, NI,(H(NJ),NJ-NJMN,NJMX)                            PRIN 330
      IF (KPR.EQ.2) RETURN                                              PRIN 335
C                                                                       PRIN 340
C ------- PRINT TOTAL HEADS                                             PRIN 345
C                                                                       PRIN 350
      KOUT-KOUT+1                                                       PRIN 355
      KLINE--1                                                          PRIN 360
      PRINT 3000, KOUT,TIME,DELT,JBAND,ITIM,(SUBHD(I),I-1,8)            PRIN 365
      DO 310 NI-1,NNP,8                                                 PRIN 370
         NJMN-NI                                                        PRIN 375
         NJMX-MINO(NI+7,NNP)                                            PRIN 380
         KLINE-KLINE+1                                                  PRIN 385
         IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) PRINT 3000, KOUT,TIME, PRIN 390
     >     DELT,JBAND,ITIM,(SUBHD(I),I-1,8)                             PRIN 395
  310    PRINT 2100, NI,(HT(NJ),NJ-NJMN,NJMX)                           PRIN 400
      IF(KPR.EQ.3) RETURN                                               PRIN 405
C                                                                       PRIN 410
C ------- PRINT WATER CONTENTS                                          PRIN 415
C                                                                       PRIN 420
      KOUT-KOUT+1                                                       PRIN 425
      KLINE--1                                                          PRIN 430
      PRINT 4000, KOUT,TIME,DELT,JBAND,ITIM,(SUBHD(I),I-1,8)            PRIN 435
      DO 410 M-1,NEL                                                    PRIN 440
      KLINE-KLINE+1                                                     PRIN 445
      IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) PRINT 4000, KOUT,TIME,    PRIN 450
     >     DELT,JBAND,ITIM,(SUBHD(I),I-1,8)                             PRIN 455
  410 PRINT 4100, M,(TH(IQ,M),IQ-1,8)                                   PRIN 460
      IF (KPR.EQ.4) RETURN                                              PRIN 465
```

```
C
C ------- PRINT DARCY VELOCITIES
C
      KOUT=KOUT+1
      KLINE=-1
      PRINT 5000, KOUT,TIME,DELT,JBAND,ITIM,(SUBHD(I),I=1,8)
      DO 510 NP=1,NNP,3
      KLINE=KLINE+1
      IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) PRINT 5000, KOUT,TIME,
     >     DELT,JBAND,ITIM,(SUBHD(I),I=1,8)
      NJMN=NP
      NJMX=MIN0(NP+2,NNP)
  510 PRINT 5100,  (NJ,VX(NJ),VY(NJ),VZ(NJ),NJ=NJMN,NJMX)
C
      RETURN
C
 1000 FORMAT(1H1,' TABLE OF SYSTEM-FLOW PARAMETERS',2X,'TABLE: ',I4,
     > '.. AT TIME =',1PD12.4,' ,(DELT =',1PD12.4,')',' ITIM=',I4//5X,
     > ' TYPE OF FLOW',35X,'RATE(L**3/T  )',3X,'INC. FLOW(L**3  )',4X,
     > 'TOTAL FLOW(L**3  )'/5X,
     1 '  1. FLOW THROUGH DIRICHLET NODES .. . .',3(E12.4,10X)/5X,
     2 '  2. FLOW THROUGH CAUCHY NODES . . . . .',3(E12.4,10X)/5X,
     3 '  3. FLOW THROUGH NEUMANN NODES .  . . .',3(E12.4,10X)/5X,
     4 '  4. FLOW THROUGH SEEPAGE NODES .. . . .',3(E12.4,10X)/5X,
     5 '  5. FLOW THROUGH INFILTRATION NODES . .',3(E12.4,10X)/5X,
     6 '  6. FLOW THROUGH UNSPECIFIED NODES  . .',3(E12.4,10X)/5X,
     7 '  7. NET FLOW THROUGH ENTIRE BOUNDARY  .',3(E12.4,10X)/5X,
     8 '  8. ARTIFICIAL SOURCES/SINKS    . . .  ',3(E12.4,10X)/5X,
     9 '  9. INCREASE IN WATER CONTENT . . . . .',3(E12.4,10X)/5X,
     A ' *** NOTE: (+) = OUT FROM, (-) = INTO THE REGION. '//)
 1100 FORMAT(/' RAINFALL-SEEPAGE NODAL FLOWS (L**3/T)')
 1110 FORMAT(8D15.4)
 1120 FORMAT(1H0,' VALUES OF NPCON'/(8I15))
 1125 FORMAT(1H0,' VALUES OF NPMIN'/(8I15))
 1130 FORMAT(1H0,' VALUES OF NPFLX'/(8I15))
 2000 FORMAT('1 OUTPUT TABLE',I4,'.. PRESSURE HEADS(L) AT TIME =',
     1 1PD12.4,' ,(DELT =',1PD12.4,'),(BAND WIDTH =',I4,')',' IT =',
     2 I5//1X,8A4/1X,' NODE I',5X,'PRESSURE HEAD (L) OF NODES I,I+1,..',
     3 ',I+7 '/)
 2100 FORMAT(I7,8(1PD15.4))
 3000 FORMAT('1 OUTPUT TABLE',I4,'. TOTAL HEADS(L) AT TIME =',1PD12.4,
     1 ' ,(DELT =',1PD12.4,'),(BAND WIDTH =',I4,')',' IT =',I5//1X,8A4,
     2 /1X,' NODE I',5X,'TOTAL HEAD (L) OF NODES I,I+1,...,I+7'/)
 4000 FORMAT('1 OUTPUT TABLE',I4,'. WATER CONTENT(L**3/L**3) AT TIME =',
     1 1PD12.4,' ,(DELT =',1PD12.4,'),(BAND WIDTH =',I4,')',' IT =',
     3 I5//1X,8A4//51X,' GAUSSIAN POINTS'/17X,'1',14X,'2',14X,'3',14X,
     4 '4',14X,'5',14X,'6',14X,'7',14X,'8'/1X,'ELEMENT',2X,'********',
     5 '**************************************************',
     6 '*****************************************'/)
```

```
 4100 FORMAT(1H ,I7,2X,8D15.7)                                          PRIN 715
 5000 FORMAT('1 OUTPUT TABLE',I4,'.. DARCY VELOCITIES (L/T) AT TIME -', PRIN 720
     1 1PD12.4,' ,(DELT -',1PD12.4,'),(BAND WIDTH -',I4,')',' IT -',    PRIN 725
     2 I5//1X,8A4//1X,3(' NODE     VX          VY          VZ',         PRIN 730
     3 '    ')/1X,3('****************************************')/)      PRIN 735
 5100 FORMAT(1H ,3(I5,3D11.3))                                          PRIN 740
      END                                                               PRIN 745
```

```
      SUBROUTINE STORE(X,Y,Z,IE,H,HT,TH,VX,VY,VZ, DCOSB,ISB,NPBB,       STOR 005
     1 NNPLR,GNLR, TITLE, TIME, NPROB)                                  STOR 010
C                                                                       STOR 015
C*******1*********2*********3*********4*********5*********6*********7**STOR 020
C                                                                       STOR 025
C ------- TO STORE PERTINENT QUANTITIES ON AUXILIARY DEVICES FOR FUTURE STOR 030
C ------- USES, E. G., FOR PLOTTING.  WHAT DEVICE IS TO BE USED MUST BE STOR 035
C ------- SPECIFIED IN THE JCL.                                         STOR 040
C                                                                       STOR 045
C*******1*********2*********3*********4*********5*********6*********7**STOR 050
C                                                                       STOR 055
C ------- INPUT: X(NNP), Y(NNP), Z(NNP), IE(NEL,9), H(NNP), HT(NNP),    STOR 060
C -------        TH(8,NEL), VX(NNP), VY(NNP), VZ(NNP), DCOSB(3,NBES),   STOR 065
C -------        ISB(6,NBES),NPBB(NBNP), NNPLR(MXREGN),                 STOR 070
C -------        GNLR(LTMXNP,MXREGN), TITLE,TIME,NPROB.                 STOR 075
C                                                                       STOR 080
C ------- OUTPUT: STORE ALL INPUTS IN LOGICAL UNIT 1.                   STOR 085
C                                                                       STOR 090
C*******1*********2*********3*********4*********5*********6*********7**STOR 095
C                                                                       STOR 100
      IMPLICIT REAL*8(A-H,O-Z)                                          STOR 105
      INTEGER*4 GNLR                                                    STOR 110
C                                                                       STOR 115
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI              STOR 120
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI                        STOR 125
      COMMON /LGEOM/ LTMXNP,LMXNP,LMXBW,MXREGN,NREGN                    STOR 130
C                                                                       STOR 135
      DIMENSION TITLE(9)                                                STOR 140
      DIMENSION X(MAXNP),Y(MAXNP),Z(MAXNP),IE(MAXEL,9)                  STOR 145
      DIMENSION H(MAXNP),HT(MAXNP),VX(MAXNP),VY(MAXNP),VZ(MAXNP)        STOR 150
      DIMENSION TH(8,MAXEL)                                             STOR 155
      DIMENSION DCOSB(3,MAXBES),ISB(6,MAXBES),NPBB(MAXBNP)              STOR 160
      DIMENSION NNPLR(MXREGN),GNLR(LTMXNP,MXREGN)                       STOR 165
C                                                                       STOR 170
      DATA NPPROB/-1/                                                   STOR 175
```

```
C
      IF (NPPROB.EQ.(-1)) REWIND 1
      IF (NPPROB.EQ.NPROB) GO TO 110
      WRITE(1) (TITLE(I),I-1,9),NPROB,NNP,NEL,NBNP,NBES,NTI
      WRITE(1) (X(N),N-1,NNP),(Y(N),N-1,NNP),(Z(N),N-1,NNP),
     1 ((IE(M,I),M-1,NEL),I-1,9),((DCOSB(I,M),I-1,3),M-1,NBES),
     2 ((ISB(I,M),I-1,6),M-1,NBES),(NPBB(N),N-1,NBNP),(NNPLR(N),N-1,
     3 NREGN),((GNLR(N,I),N-1,LTMXNP),I-1,NREGN)
      NPPROB-NPROB
C
  110 WRITE(1) TIME,(H(N),N-1,NNP),(HT(N),N-1,NNP),((TH(I,M),I-1,8),M-1,
     1 NEL),(VX(N),N-1,NNP),(VY(N),N-1,NNP),(VZ(N),N-1,NNP)
      RETURN
      END
```

```
      SUBROUTINE INTERP(FALL,TRF,RF,TIME,MXPR,MXDP,NPR,NDP)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO INTERPOLATE FOR THE VALUE FROM THE TABLE INPUT.  IF THE
C ------- NUMBER IS OUT OF RANGE, THE FUNCTIONAL VALUE IS SET TO ZERO.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: TRF(NPD,NPR), RF(NPD,NPR), TIME, NPR, AND NPD.
C
C ------- OUTPUT: FALL(NPR).
C
C********1*********1*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION FALL(MXPR),TRF(MXDP,MXPR),RF(MXDP,MXPR)
C
      DO 160 I-1,NPR
      DO 140 J-2,NDP
      IF(TRF(J-1,I).LE.TIME .AND. TIME.LE.TRF(J,I)) GO TO 120
      GO TO 140
  120 RFJM1-RF(J-1,I)
      TRFJM1-TRF(J-1,I)
      RFJ-RF(J,I)
      RFJ-TRF(J,I)
      ABC-RFJ-RFJM1
      ABCD-TRFJ-TRFJM1
      FALL(I)-RFJM1+(TIME-TRFJM1)*ABC/ABCD
```

```
      GO TO 160
  140 CONTINUE
      FALL(I)=0.0
  160 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE READR(F,MAXNOD,NNP)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO AUTOMATICALLY GENERATE REAL NUMBER INPUT.
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION F(MAXNOD)
C
      NODES=0
  150 READ 40, NI,NSEQ,NAD,FNI,FAD,FRD
      IF(NI.EQ.0) GO TO 170
      NJ=NI+NSEQ
      DO 160 N=NI,NJ
      NODES=NODES+1
      I=NI+(N-NI)*NAD
      IF(FRD.NE.0.0) GO TO 155
      F(I)=FNI+FAD*DFLOAT(N-NI)
      GO TO 160
  155 I1=I-NAD
      IF(N.EQ.NI) F(I)=FNI
      IF(N.EQ.NI) DINC=1.0D0
      IF(N.GT.NI) DINC=DINC*(1.0D0+FRD)
      IF(N.GT.NI) F(I)=F(I1)+FAD*DINC
  160 CONTINUE
      GO TO 150
  170 IF(NODES.EQ.NNP) GO TO 180
      PRINT 1100
      STOP
  180 IF(NNP.LE.MAXNOD) GO TO 190
      PRINT 1200
      STOP
  190 CONTINUE
```

```
C                                                                       REDR 185
   40 FORMAT(3I5,5X,4D10.3)                                             REDR 190
 1100 FORMAT(1H1/1H ,64H *** ERROR IN EXECUTING READR SINCE NODES  .NE. REDR 195
     > NNP:   STOP ***)                                                 REDR 200
 1200 FORMAT(1H1/1H ,46H *** NNP .GT. MAXNOD  IN EXECUTING READR: STOP) REDR 205
C                                                                       REDR 210
      RETURN                                                            REDR 215
      END                                                               REDR 220
```

```
      SUBROUTINE READN(INDTYP,MXTYP,NTYPE)                              REDN 005
C                                                                       REDN 010
C********1*********2*********3*********4*********5*********6*********7**REDN 015
C                                                                       REDN 020
C ------- TO AUTOMATICALLY GENERATE INTEGER INPUT.                      REDN 025
C                                                                       REDN 030
C********1*********2*********3*********4*********5*********6*********7**REDN 035
C                                                                       REDN 040
      IMPLICIT REAL*8(A-H,O-Z)                                          REDN 045
C                                                                       REDN 050
      DIMENSION INDTYP(MXTYP)                                           REDN 055
C                                                                       REDN 060
      NTYPES=0                                                          REDN 065
  110 READ 30,NI,NSEQ,NAD,NITYP,NTYPAD                                  REDN 070
      IF(NI.EQ.0) GO TO 130                                             REDN 075
      NJ=NI+NSEQ                                                        REDN 080
      DO 120 N=NI,NJ                                                    REDN 085
      I=NI+(N-NI)*NAD                                                   REDN 090
      INDTYP(I)=NITYP + (N-NI)*NTYPAD                                   REDN 095
      NTYPES=NTYPES+1                                                   REDN 100
  120 CONTINUE                                                          REDN 105
      GO TO 110                                                         REDN 110
  130 IF(NTYPES.EQ.NTYPE) GO TO 140                                     REDN 115
      PRINT 1100                                                        REDN 120
      STOP                                                              REDN 125
  140 IF(NTYPE.LE.MXTYP) GO TO 150                                      REDN 130
      PRINT 1200                                                        REDN 135
      STOP                                                              REDN 140
  150 CONTINUE                                                          REDN 145
C                                                                       REDN 150
   30 FORMAT(5I5)                                                       REDN 155
 1100 FORMAT(1H1/1H ,64H *** ERROR IN EXECUTING READN SINCE NTYPES .NE. REDN 160
     > NTYPE: STOP ***)                                                 REDN 165
 1200 FORMAT(1H1/1H ,46H *** NTYPE .GT. MXTYP IN EXECUTING READN: STOP) REDN 170
C                                                                       REDN 175
      RETURN                                                            REDN 180
      END                                                               REDN 185
```

APPENDIX B

Listing of FORTRAN Source Program of 3DFEMWATER Code PTI (Point Iteration Version)

```
      SUBROUTINE GW3D(X,Y,Z,IE, CMATRX,GNOJCN,RLD,RI,RL, H,HP,HW,HT,
     1 VX,VY,VZ,  TH,DTH,AKR, NPCNV, DCOSB,ISB,NPBB, BFLX,BFLXP,
     2 SOS,SOSF,TSOSF,ISTYP,MSEL, WSS,WSSF,TWSSF,IWTYP,NPW,
     3 QCB,QCBF,TQCBF,ICTYP,ISC,NPCB, QNB,QNBF,TQNBF,INTYP,ISN,NPNB,
     4 RFALL,RF,TRF,IRTYP,ISV,NPVB, DCYFLX,FLX,HCON,HMIN,
     5 NPFLX,NPCON,NPMIN, HDB,HDBF,THDBF,IDTYP,NPDB,
     6 PROP,THPROP,AKPROP, KPR,KDSK,TDTCH)
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 SUBHD
      INTEGER*4 GNOJCN
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC
      COMMON /CGEOM/ NNP,NEL,NBNP,NPES,KGRAV,NTI,NDTCHG
      COMMON /CINTE/ NCYL,NITER,KSTF ,KPRO,KDSKO,KSS,NPITER,IGEOM
      COMMON /CREAL/ DELT,CHNG,DELMAX,TMAX,DELTO,TOLA,TOLB,W,OME,OMI
C
      COMMON /CS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP
      COMMON /CW/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP
C
      COMMON /CCBC/ MXCNP,MXCES,MXCPR,MXCDP,NCNP,NCES,NCPR,NCDP
      COMMON /CNBC/ MXNNP,MXNES,MXNPR,MXNDP,NNNP,NNES,NNPR,NNDP
      COMMON /CVBC/ MXVES,MXVNP,MXRPR,MXRDP,NVES,NVNP,NRPR,NRDP
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP
C
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM
      COMMON /CMTL/ NMAT,NMPPM,NSPPM
C
      COMMON /CFLOW/ FRATE(10),FLOW(10),TFLOW(10)
C
      DIMENSION X(MAXNP),Y(MAXNP),Z(MAXNP),IE(MAXEL,9)
      DIMENSION CMATRX(MAXNP,JBAND),GNOJCN(JBAND,MAXNP),RLD(MAXNP)
      DIMENSION RI(MAXNP),RL(MAXNP)
      DIMENSION H(MAXNP),HP(MAXNP),HW(MAXNP),HT(MAXNP)
      DIMENSION VX(MAXNP),VY(MAXNP),VZ(MAXNP)
      DIMENSION TH(8,MAXEL),DTH(8,MAXEL),AKR(8,MAXEL), NPCNV(MAXNP)
C
      DIMENSION DCOSB(3,MAXBES),ISB(6,MAXBES), NPBB(MAXBNP)
      DIMENSION BFLX(MAXBNP),BFLXP(MAXBNP)
C
      DIMENSION SOS(MXSPR),SOSF(MXSDP,MXSPR),TSOSF(MXSDP,MXSPR)
      DIMENSION ISTYP(MXSEL),MSEL(MXSEL)
      DIMENSION WSS(MXWPR),WSSF(MXWDP,MXWPR),TWSSF(MXWDP,MXWPR)
      DIMENSION IWTYP(MXWNP),NPW(MXWNP)
C
      DIMENSION QCB(MXCPR),QCBF(MXCDP,MXCPR),TQCBF(MXCDP,MXCPR)
      DIMENSION ICTYP(MXCES),ISC(5,MXCES), NPCB(MXCNP)
C
      DIMENSION QNB(MXNPR),QNBF(MXNDP,MXNPR),TQNBF(MXNDP,MXNPR)
      DIMENSION INTYP(MXNES),ISN(5,MXNES), NPNB(MXNNP)
```

```
C
      DIMENSION RFALL(MXRPR),RF(MXRDP,MXRPR),TRF(MXRDP,MXRPR)
      DIMENSION IRTYP(MXVES),ISV(5,MXVES), NPVB(MXVNP)
      DIMENSION DCYFLX(MXVNP),FLX(MXVNP),HCON(MXVNP),HMIN(MXVNP)
      DIMENSION NPFLX(MXVNP),NPCON(MXVNP),NPMIN(MXVNP)
C
      DIMENSION HDB(MXDPR),HDBF(MXDDP,MXDPR),THDBF(MXDDP,MXDPR)
      DIMENSION IDTYP(MXDNP),NPDB(MXDNP)
C
      DIMENSION PROP(MXMPPM,MAXMAT)
      DIMENSION THPROP(MXSPPM,MAXMAT),AKPROP(MXSPPM,MAXMAT)
C
      DIMENSION KPR(MAXNTI),KDSK(MAXNTI),TDTCH(MXNDTC)
C
      DIMENSION TITLE(9)
      DIMENSION SUBHD(8,3)
C
      DATA SUBHD/4HINPU,4HT IN,4HITIA,4HL CO,4HNDIT,4HIONS,2*4H
     > 4HSTEA,4HDY-S,4HTATE,4H INI,4HTIAL,4H CON,4HDITI,4HONS , 8*
     > 4H    /
C
C ******* DATA SET 1: PROBLEM IDENTIFICATION AND DESCRIPTION
C
  100 READ 10, NPROB,(TITLE(I),I-1,9),IGEOM,IBUG,ICHNG
C
      IF (NPROB.LE.0) GO TO 990
C
      PRINT 1000, NPROB,(TITLE(I),I-1,9),IGEOM,IBUG,ICHNG
C
C ------- READ AND PRINT INPUT DATA BY CALLING DATAIN
C
      KOUT-0
      TIME-0.0
C
C ******* DATA SETS 2 THROUGH 16 WILL BE READ IN DATAIN
C
      CALL DATAIN(TITLE,NPROB, KPR,KDSK,TDTCH,
     1 PROP,THPROP,AKPROP, X,Y,Z,IE,H, DCOSB,ISB,NPBB,
     2 SOSF,TSOSF,ISTYP,MSEL, WSSF,TWSSF,IWTYP,NPW,
     3 QCBF,TQCBF,ICTYP,ISC,NPCB,  QNBF,TQNBF,INTYP,ISN,NPNB,
     4 RF,TRF,IRTYP,ISV,NPVB, HCON,HMIN,  HDBF,THDBF,IDTYP,NPDB,
     5 ISTOP)
C
      IF(IGEOM.LE.3) CALL PAGEN(GNOJCN,IE)
      REWIND 4
      IF(IGEOM.LE.3) WRITE(4) ((GNOJCN(J,I),J-1,JBAND),I-1,NNP)
      IF(IGEOM.GT.3) READ(4) ((GNOJCN(J,I),J-1,JBAND),I-1,NNP)
C
      KDIG-0
      IF (ISTOP.GT.0) GO TO 990
```

```
C
C ------- PREPARE INITIAL OR PRE-INITIAL VARIABLES
C
      IF(NSEL.NE.0) CALL INTERP(SOS,TSOSF,SOSF,TIME,
     1                MXSPR,MXSDP,NSPR,NSDP)
      IF(NWNP.NE.0) CALL INTERP(WSS,TWSSF,WSSF,TIME,
     1                MXWPR,MXWDP,NWPR,NWDP)
C
      IF(NCES.NE.0) CALL INTERP(QCB,TQCBF,QCBF,TIME,
     1                MXCPR,MXCDP,NCPR,NCDP)
      IF(NNES.NE.0) CALL INTERP(QNB,TQNBF,QNBF,TIME,
     1                MXNPR,MXNDP,NNPR,NNDP)
      IF(NVES.NE.0) CALL INTERP(RFALL,TRF,RF,TIME,
     1                MXRPR,MXRDP,NRPR,NRDP)
      IF(NDNP.NE.0) CALL INTERP(HDB,THDBF,HDBF,TIME,
     1                MXDPR,MXDDP,NDPR,NDDP)
C
C ------- PUT DIRICHLET BOUNDARY VALUES TO INITIAL CONDITIONS
C
      DO 130 I-1,NDNP
      NI-NPDB(I)
      NP-NPBB(NI)
      ITYP-IDTYP(I)
      H(NP)-HDB(ITYP)-Z(NP)*DFLOAT(KGRAV)
  130 CONTINUE
C
      CALL SPROP(TH,DTH,AKR, IE,H,THPROP,AKPROP)
C
      CALL VELT(VX,VY,VZ,CMATRX,X,Y,Z,IE,H,HT,AKR,PROP)
      KFLOW--1
C
      CALL SFLOW(X,Y,Z,IE, H,HP,VX,VY,VZ,TH,DTH,
     1 BFLX,BFLXP,DCOSB,ISB,NPBB, MSEL,SOS,ISTYP, NPW,WSS,IWTYP,
     2 NPVB,NPDB,NPCB,NPNB,   DELT, KFLOW)
C
      DO 140 I-1,9
      IF(I.EQ.9) GO TO 140
      FLOW(I)-0.0
      TFLOW(I)-0.0
  140 CONTINUE
      FLOW(9)-0.0
C
C ------- PRINT INITIAL OR PRE-INITIAL VARIABLES
C
      KDIAG-0
C
      CALL PRINTT(VX,VY,VZ,H,HT,TH, NPBB,BFLX, NPVB,DCYFLX,NPCON,
     1NPFLX,NPMIN, SUBHD(1,1), TIME,DELT,KPRO,KOUT,KDIAG,-1)
```

```
C                                                                       GW3D 745
      IF(KSTR.EQ.1 .AND. KSS.EQ.1  .AND. KDSKO.EQ.1)                    GW3D 750
     > CALL STORE(X,Y,Z,IE, H,HT,TH,VX,VY,VZ,DCOSB,ISB,NPBB,            GW3D 755
     1 TITLE, TIME, NPROB)                                              GW3D 760
C                                                                       GW3D 765
      IF (KSS.NE.0) GO TO 500                                           GW3D 770
C                                                                       GW3D 775
C $$$$$$$                                                               GW3D 780
C $$$$$$$ PERFORM STEADY-STATE CALCULATION                              GW3D 785
C $$$$$$$                                                               GW3D 790
C                                                                       GW3D 795
      IF (NVES.EQ.0) GO TO 170                                          GW3D 800
C                                                                       GW3D 805
      DO 150 NPP-1,NVNP                                                 GW3D 810
      NI-NPVB(NPP)                                                      GW3D 815
      NPCON(NPP)-NPBB(NI)                                               GW3D 820
      NPMIN(NPP)-0                                                      GW3D 825
  150 NPFLX(NPP)-0                                                      GW3D 830
C                                                                       GW3D 835
      NCHG--1                                                           GW3D 840
      CALL BCPREP(IE,X,Y,Z,H,VX,VY,VZ,DCOSB,ISB,NPVB,ISV,DCYFLX,FLX,    GW3D 845
     > HCON,HMIN,NPFLX,NPCON,NPMIN, IRTYP,RFALL, NCHG)                  GW3D 850
C                                                                       GW3D 855
  170 DO 180 NP-1,NNP                                                   GW3D 860
  180 HP(NP)-H(NP)                                                      GW3D 865
C                                                                       GW3D 870
      KDIG-KDIG+1                                                       GW3D 875
      IF(IBUG.NE.0) PRINT 10400,KDIG,TIME,DELT                          GW3D 880
C                                                                       GW3D 885
C ------- ITERATION LOOP ON SEEPAGE-RAINFALL BOUNDARY CONDITIONS BEGINS GW3D 890
C                                                                       GW3D 895
      EPS-0.5D0*TOLA                                                    GW3D 900
C                                                                       GW3D 905
      DO 390 ICY-1,NCYL                                                 GW3D 910
C                                                                       GW3D 915
      DO 210 NP-1,NNP                                                   GW3D 920
      HW(NP)-OME*H(NP)+(1.0D0-OME)*HP(NP)                               GW3D 925
      RI(NP)-HW(NP)                                                     GW3D 930
  210 CONTINUE                                                          GW3D 935
C                                                                       GW3D 940
C ------- ITERATION LOOP ON THE NON-LINEAR EQUATION BEGINS              GW3D 945
C                                                                       GW3D 950
      IF(IBUG.NE.0) PRINT 10401, ICY                                    GW3D 955
C                                                                       GW3D 960
C ------- PUT DIRICHLET BOUNDARY VALUES OF THE VARIABLE BOUNDARY        GW3D 965
C ------- INTO H, RI, HW, AND RL                                        GW3D 970
C                                                                       GW3D 975
      IF(NVES.EQ.0) GO TO 250                                           GW3D 980
      DO 230 NPP-1,NVNP                                                 GW3D 985
      NI-NPMIN(NPP)                                                     GW3D 990
```

```
      IF(NI.EQ.0) GO TO 220
      H(NI)=HMIN(NPP)
      RI(NI)=HMIN(NPP)
      HW(NI)=HMIN(NPP)
      RL(NI)=HMIN(NPP)
      GO TO 230
  220 NI=NPCON(NPP)
      IF(NI.EQ.0) GO TO 230
      H(NI)=HCON(NPP)
      RI(NI)=HCON(NPP)
      HW(NI)=HCON(NPP)
      RL(NI)=HCON(NPP)
  230 CONTINUE
  250 CONTINUE
C
      DO 350 IT=1,NITER
C
C ------- EVALUATE SOIL PROPERTIES FOR PREVIOUS ITERATE
C
      CALL SPROP(TH,DTH,AKR, IE,HW,THPROP,AKPROP)
C
C ------- ASSEMBLE STEAD-STATE ELEMENT MATRICES QA AND QB INTO THE
C ------- GLOBAL MATRIX C AND CONSTRUCT GLOBAL LOAD VECTOR R FROM
C ------- ELEMENT LOAD VECTOR RQ.
C
      CALL ASEMBL(X,Y,Z,IE, CMATRX,RLD,GNOJCN,HW,HP,DTH,AKR,PROP,
     > SOS,MSEL,ISTYP,WSS,NPW,IWTYP, KSS,W,DELT)
C
C ------- APPLY STEADY-STATE BOUNDARY CONDITIONS
C
      CALL BC(CMATRX,RLD,GNOJCN, IE,X,Y,Z, AKR,PROP, DCOSB,ISB,NPBB,
     1 QCB,ISC,ICTYP, QNB,ISN,INTYP, FLX,HCON,HMIN,NPFLX,NPCON,NPMIN,
     2 HDB,IDTYP,NPDB, KSS)
C
C ------- SOLVE THE MATRIX EQUATION BY POINTWISE ITERATION
C
      CALL PISS(RL,RI,RLD,CMATRX,GNOJCN,OMI,EPS,NNP,NPITER,MAXNP,JBAND,
     1 IBUG,KPRO)
C
C ------- OBTAIN MAXIMUM RELATIVE DEVIATION FROM PREVIOUS ITERATE
C
      NPP=0
      RD=-1.0D0
      RES=-1.0D0
      DO 320 NP=1,NNP
      RESNP=DABS(RL(NP)-H(NP))
      RES=DMAX1(RES,RESNP)
      IF(H(NP).NE.0.0D0) RD=DMAX1(RD,DABS(RESNP/H(NP)))
      IF(RESNP .LE. TOLA) GO TO 320
      NPP=NPP+1
```

```
      NPCNV(NPP)=NP                                                     GW3D1245
  320 CONTINUE                                                          GW3D1250
C                                                                       GW3D1255
      NNCVN=NPP                                                         GW3D1260
C                                                                       GW3D1265
C ------- UPDATE PRESSURE WITH CURRENT ITERATE                          GW3D1270
C                                                                       GW3D1275
      DO 330 NP=1,NNP                                                   GW3D1280
      H(NP)=OME*RL(NP)+(1.0D0-OME)*H(NP)                                GW3D1285
      RI(NP)=H(NP)                                                      GW3D1290
      HW(NP)=H(NP)                                                      GW3D1295
  330 CONTINUE                                                          GW3D1300
C                                                                       GW3D1305
C ------- ESCAPE FROM ITERATION LOOP IF THE MAXIMUM RESIDUAL IS         GW3D1310
C ------- SUFFICIENTLY SMALL                                            GW3D1315
C                                                                       GW3D1320
      IF(IBUG.NE.0) PRINT 10200, IT,RES,RD,NNCVN                        GW3D1325
      IF(IT.EQ.1) GO TO 350                                             GW3D1330
      IF(RES.LT.TOLA) GO TO 360                                         GW3D1335
C                                                                       GW3D1340
  350 CONTINUE                                                          GW3D1345
C                                                                       GW3D1350
C ------- END OF ITERATION LOOP ON THE NON-LINEAR EQUATION              GW3D1355
C                                                                       GW3D1360
      PRINT 10210, ICY,IT,NITER,RES,RD,NNCVN                            GW3D1365
C                                                                       GW3D1370
C ------- PRINT NONCONVERGENING NODES                                   GW3D1375
C                                                                       GW3D1380
      IF(IBUG.EQ.0) GO TO 360                                           GW3D1385
      PRINT 10500                                                       GW3D1390
      PRINT 10600, (NPCNV(NPP),NPP=1,NNCVN)                             GW3D1395
C                                                                       GW3D1400
  360 IF(ICHNG.EQ.0) GO TO 380                                          GW3D1405
      IF(NVES.EQ.0) GO TO 380                                           GW3D1410
C                                                                       GW3D1415
C ------- PRINT RAINFALL-SEEPAGE B. C. CHANGE INFORMATION               GW3D1420
C                                                                       GW3D1425
      PRINT 10402, ICY                                                  GW3D1430
      DO 370 I=1,NVNP                                                   GW3D1435
      NI=NPVB(I)                                                        GW3D1440
      NP=NPBB(NI)                                                       GW3D1445
      PRINT 10403, I,NP,NPCON(I),HCON(I),NPMIN(I),HMIN(I),NPFLX(I),     GW3D1450
     1   FLX(I),DCYFLX(I)                                               GW3D1455
  370 CONTINUE                                                          GW3D1460
C                                                                       GW3D1465
C ------- CALCULATE DARCY'S VELOCITY                                    GW3D1470
C                                                                       GW3D1475
  380 CALL SPROP(TH,DTH,AKR, IE,H,THPROP,AKPROP)                        GW3D1480
C                                                                       GW3D1485
      CALL VELT(VX,VY,VZ,CMATRX,X,Y,Z,IE,H,HT,AKR,PROP)                 GW3D1490
```

```
C
      IF(NVES .EQ. 0) GO TO 440
C
C ------- PREPARE BOUNDARY CONDITIONS ON THE VARIABLE-TYPE BOUNDARY FOR
C ------- NEXT CYCLE COMPUTATIONS.
C
      CALL BCPREP(IE,X,Y,Z, H,VX,VY,VZ, DCOSB,ISB, NPVB,ISV,DCYFLX,FLX,
     > HCON,HMIN,NPFLX,NPCON,NPMIN,IRTYP,RFALL, NCHG)
C
      IF(NCHG.EQ.0) GO TO 440
  390 CONTINUE
C
C ------- END OF ITERATION LOOP ON SEEPAGE-RAINFALL BOUNDARY CONDITIONS
C
      PRINT 10610, ICY,IT,NCYL,NITER,RES,RD,NNCVN
C
  440 IF(NNCVN.EQ.0) GO TO 445
      PRINT 10610, ICY,IT,NCYL,NITER,RES,RD,NNCVN
C
C ------- COMPUTE FLUXES THROUGH ALL TYPES OF BOUNDARIES.
C
  445 KFLOW-0
      CALL SFLOW(X,Y,Z,IE, H,HP,VX,VY,VZ TH,DTH,
     1 BFLX,BFLXP,DCOSB,ISB,NPBB, MSEL,SOS,ISTYP,NPW,WSS,IWTYP,
     2 NPVB,NPDB,NPCB,NPNB,   DELT, KFLOW)
C
      DO 450 I-1,9
      IF(I.EQ.9) GO TO 450
      FLOW(I)-0.0
      TFLOW(I)-0.0
  450 CONTINUE
      FLOW(9)-0.0
C
C ------- PRINT STEADY-STATE VARIABLES
C
      CALL PRINTT(VX,VY,VZ,H,HT,TH, NPBB,BFLX, NPVB,DCYFLX,NPCON,
     1 NPFLX,NPMIN, SUBHD(1,2), TIME,DELT, KPRO,KOUT,KDIAG,0)
C
      IF(KSTR.EQ.1 .AND. KDSKO.EQ.1) CALL STORE(X,Y,Z,IE,
     > H,HT,TH,VX,VY,VZ,DCOSB,ISB,NPBB, TITLE,TIME,NPROB)
C
      IF (NTI.EQ.0) GO TO 100
C
      KSS-1
C
C $$$$$$$
C $$$$$$$ PERFORM TRANSIENT-STATE CALCULATION
C $$$$$$$
C
  500 IF (NVES.EQ.0) GO TO 550
```

```
C                                                                       GW3D1745
      DO 510 NPP=1,NVNP                                                 GW3D1750
      NI=NPVB(NPP)                                                      GW3D1755
      NPCON(NPP)=NPBB(NI)                                               GW3D1760
      NPMIN(NPP)=0                                                      GW3D1765
  510 NPFLX(NPP)=0                                                      GW3D1770
C                                                                       GW3D1775
      NCHG=-1                                                           GW3D1780
C                                                                       GW3D1785
  550 TIME=TIME+DELT                                                    GW3D1790
      W1=W                                                              GW3D1795
      W2=1.0D0-W                                                        GW3D1800
      KFLOW=1                                                           GW3D1805
      TFLOW(9)=0.0                                                      GW3D1810
C                                                                       GW3D1815
C ------- BEGIN THE TIME-MARCHING LOOP                                  GW3D1820
C                                                                       GW3D1825
      EPS=0.5D0*TOLB                                                    GW3D1830
      IDELT=0                                                           GW3D1835
      DO 890 ITM=1,NTI                                                  GW3D1840
      ITMITM=ITM                                                        GW3D1845
C                                                                       GW3D1850
C ------- PREPARE TRANSIENT BOUNDARY CONDITIONS AND SOURCE FOR THE STEP GW3D1855
C                                                                       GW3D1860
      IF(NSEL.NE.0) CALL INTERP(SOS,TSOSF,SOSF,TIME,MXSPR,MXSDP,        GW3D1865
     1 NSPR,NSDP)                                                       GW3D1870
      IF(NWNP.NE.0) CALL INTERP(WSS,TWSSF,WSSF,TIME,MXWPR,MXWDP,        GW3D1875
     1 NWPR,NWDP)                                                       GW3D1880
C                                                                       GW3D1885
      IF(NCES.NE.0) CALL INTERP(QCB,TQCBF,QCBF,TIME,                    GW3D1890
     1                       MXCPR,MXCDP,NCPR,NCDP)                     GW3D1895
      IF(NNES.NE.0) CALL INTERP(QNB,TQNBF,QNBF,TIME,                    GW3D1900
     1                       MXNPR,MXNDP,NNPR,NNDP)                     GW3D1905
      IF(NVES.NE.0) CALL INTERP(RFALL,TRF,RF,TIME,                      GW3D1910
     1                       MXRPR,MXRDP,NRPR,NRDP)                     GW3D1915
      IF(NDNP.NE.0) CALL INTERP(HDB,THDBF,HDBF,TIME,                    GW3D1920
     1                       MXDPR,MXDDP,NDPR,NDDP)                     GW3D1925
C                                                                       GW3D1930
      IF(NVES.EQ.0) GO TO 560                                           GW3D1935
      NCHG=-1                                                           GW3D1940
      CALL BCPREP(IE,X,Y,Z, H,VX,VY,VZ, DCOSB,ISB, NPVB,ISV,DCYFLX,FLX, GW3D1945
     1 HCON,HMIN,NPFLX,NPCON,NPMIN, IRTYP,RFALL, NCHG)                  GW3D1950
C                                                                       GW3D1955
  560 DO 570 NP=1,NNP                                                   GW3D1960
      RL(NP)=H(NP)                                                      GW3D1965
      HP(NP)=H(NP)                                                      GW3D1970
  570 CONTINUE                                                          GW3D1975
C                                                                       GW3D1980
      KDIG=KDIG+1                                                       GW3D1985
      IF(IBUG.NE.0 .AND. KPR(ITM).NE.0) PRINT 10400, KDIG,TIME,DELT     GW3D1990
```

```
C                                                                       GW3D1995
C ------- BEGIN ITERATION LOOP ON SEEPAGE-RAINFALL BOUNDARY CONDITIONS  GW3D2000
C                                                                       GW3D2005
      DO 690 ICY-1,NCYL                                                 GW3D2010
      IF(IBUG.NE.0 .AND. KPR(ITM).NE.0) PRINT 10401, ICY                GW3D2015
C                                                                       GW3D2020
      DO 580 NP-1,NNP                                                   GW3D2025
      H(NP)-OME*RL(NP)+(1.0DO-OME)*H(NP)                                GW3D2030
      RI(NP)-H(NP)                                                      GW3D2035
      HW(NP)-W1*(OME*H(NP)+(1.0DO-OME)*HP(NP))+W2*HP(NP)                GW3D2040
  580 CONTINUE                                                          GW3D2045
C                                                                       GW3D2050
C ------- BEGIN ITERATION LOOP ON THE NON-LINEAR EQUATION               GW3D2055
C                                                                       GW3D2060
C ------- PUT DIRICHLET BOUNDARY VALUES OF THE VARIABLE BOUNDARY        GW3D2065
C ------- INTO H, RI, HW, AND RL                                        GW3D2070
C                                                                       GW3D2075
      IF(NVES.EQ.0) GO TO 595                                           GW3D2080
      DO 590 NPP-1,NVNP                                                 GW3D2085
      NI-NPMIN(NPP)                                                     GW3D2090
      IF(NI.EQ.0) GO TO 585                                             GW3D2095
      H(NI)-HMIN(NPP)                                                   GW3D2100
      RI(NI)-HMIN(NPP)                                                  GW3D2105
      HW(NI)-HMIN(NPP)                                                  GW3D2110
      RL(NI)-HMIN(NPP)                                                  GW3D2115
      GO TO 590                                                         GW3D2120
  585 NI-NPCON(NPP)                                                     GW3D2125
      IF(NI.EQ.0) GO TO 590                                             GW3D2130
      H(NI)-HCON(NPP)                                                   GW3D2135
      RI(NI)-HCON(NPP)                                                  GW3D2140
      HW(NI)-HCON(NPP)                                                  GW3D2145
      RL(NI)-HCON(NPP)                                                  GW3D2150
  590 CONTINUE                                                          GW3D2155
  595 CONTINUE                                                          GW3D2160
C                                                                       GW3D2165
      DO 650 IT-1,NITER                                                 GW3D2170
C                                                                       GW3D2175
C  ------- EVALUATE SOIL PROPERTIES FOR PREVIOUS ITERATE                GW3D2180
C                                                                       GW3D2185
      CALL SPROP(TH,DTH,AKR, IE,HW,THPROP,AKPROP)                       GW3D2190
C                                                                       GW3D2195
C ------- ASSEMBLE ELEMENT MATRICES QA AND QB INTO THE GLOBAL MATRIX C  GW3D2200
C ------- AND CONSTRUCT THE GLOBAL LOAD VECTOR R FROM ELEMENT LOAD      GW3D2205
C ------- VECTOR RQ.                                                    GW3D2210
C                                                                       GW3D2215
      CALL ASEMBL(X,Y,Z,IE, CMATRX,RLD,GNOJCN,HW,HP,DTH, AKR,PROP,      GW3D2220
     > SOS,MSEL,ISTYP,WSS,NPW,IWTYP, KSS,W,DELT)                        GW3D2225
```

```
C
C ------- APPLY BOUNDARY CONDITIONS TO MODIFY THE GLOBAL MATRIX C AND
C ------- THE LOAD VECTOR R.
C
      CALL BC(CMATRX,RLD,GNOJCN, IE,X,Y,Z, AKR,PROP, DCOSB,ISB,NPBB,
     1 QCB,ISC,ICTYP, QNB,ISN,INTYP, FLX,HCON,HMIN,NPFLX,NPCON,NPMIN,
     2 HDB,IDTYP,NPDB, KSS)
C
C ------- SOLVE THE MATRIX EQUAITON BY POINTWISE ITERATION
C
      CALL PISS(RL,RI,RLD,CMATRX,GNOJCN,OMI,EPS,NNP,NPITER,MAXNP,JBAND,
     1 IBUG,KPR(ITM))
C
C ------- OBTAIN MAXIMUM RELATIVE DEVIATION FROM PREVIOUS ITERATE
C
      NPP=0
      RD=-1.0D0
      RES=-1.0D0
      DO 620 NP=1,NNP
      RESNP=DABS(RL(NP)-H(NP))
      RES=DMAX1(RES,RESNP)
      IF(H(NP) .NE. 0.0D0) RD=DMAX1(RD,DABS(RESNP/H(NP)))
      IF(RESNP .LE. TOLB) GO TO 620
      NPP=NPP+1
      NPCNV(NPP)=NP
  620 CONTINUE
C
      NNCVN=NPP
C
C ------- UPDATE PRESSURE WITH CURRENT ITERATE
C
      DO 630 NP=1,NNP
      H(NP)=OME*RL(NP)+(1.0D0-OME)*H(NP)
      RI(NP)=H(NP)
      HW(NP)=W1*H(NP)+W2*HP(NP)
  630 CONTINUE
C
C ------- ESCAPE FROM ITERATION LOOP IF THE MAXIMUM RESIDUAL IS
C ------- SUFFICIENTLY SMALL.
C
      IF(IBUG.NE.0 .AND. KPR(ITM).NE.0) PRINT 10200, IT,RES,RD,NNCVN
      IF(IT.EQ.1 .AND. ITM.EQ.1) GO TO 650
      IF(RES.LT.TOLB) GO TO 660
C
  650 CONTINUE
C
C ------- END THE ITERATION LOOP ON THE NON-LINEAR EQUATION
C
      PRINT 10710, ITM,ICY,IT,NITER,RES,RD,NNCVN
```

```
C                                                                       GW3D2475
      IF(IBUG.EQ.0 .OR. KPR(ITM).EQ.0) GO TO 66C                        GW3D2480
C                                                                       GW3D2485
C ------- PRINT NONCONVERGING NODES                                     GW3D2490
C                                                                       GW3D2495
      PRINT 10500                                                       GW3D2500
      PRINT 10600, (NPCNV(NPP),NPP-1,NNCVN)                             GW3D2505
C                                                                       GW3D2510
  660 IF(ICHNG.EQ.0 .OR. KPR(ITM).EQ.0) GO TO 680                       GW3D2515
      IF(NVES.EQ.0) GO TO 680                                           GW3D2520
C                                                                       GW3D2525
C ------- PRINT RAINFALL-SEEPAGE BOUNDARY CONDITION CHANGE INFORMATION  GW3D2530
C                                                                       GW3D2535
      PRINT 10402, ICY                                                  GW3D2540
      DO 670 I-1,NVNP                                                   GW3D2545
      NI-NPVB(I)                                                        GW3D2550
      NP-NPBB(NI)                                                       GW3D2555
      PRINT 10403, I,NP,NPCON(I),HCON(I),NPMIN(I),HMIN(I),NPFLX(I),     GW3D2560
     1    FLX(I),DCYFLX(I)                                              GW3D2565
  670 CONTINUE                                                          GW3D2570
C                                                                       GW3D2575
C ------- CALCULATE DARCY'S VELOCITY                                    GW3D2580
C                                                                       GW3D2585
  680 CALL SPROP(TH,DTH,AKR, IE,H,THPROP,AKPROP)                        GW3D2590
C                                                                       GW3D2595
      CALL VELT(VX,VY,VZ,CMATRX,X,Y,Z,IE,H,HT,AKR,PROP)                 GW3D2600
C                                                                       GW3D2605
      IF(NVES.EQ.0) GO TO 710                                           GW3D2610
C                                                                       GW3D2615
      CALL BCPREP(IE,X,Y,Z,H,VX,VY,VZ, DCOSB,ISB, NPVB,ISV,DCYFLX,FLX,  GW3D2620
     > HCON,HMIN,NPFLX,NPCON,NPMIN,IRTYP,RFALL, NCHG)                   GW3D2625
C                                                                       GW3D2630
      IF(NCHG.EQ.0) GO TO 710                                           GW3D2635
C                                                                       GW3D2640
  690 CONTINUE                                                          GW3D2645
C                                                                       GW3D2650
C ------- END ITERATION LOOP ON SEEPAGE-RAINFALL BOUNDARY CONDITIONS     GW3D2655
C                                                                       GW3D2660
      PRINT 10810, ITM,ICY,IT,NCYL,NITER,RES,RD,NNCVN                   GW3D2665
  710 IF(NNCVN.EQ.0) GO TO 740                                          GW3D2670
      PRINT 10810, ITM,ICY,IT,NCYL,NITER,RES,RD,NNCVN                   GW3D2675
C                                                                       GW3D2680
C ------- COMPUTE FLUXES THROUGH ALL TYPES OF BOUNDARIES                GW3D2685
C                                                                       GW3D2690
  740 CALL SFLOW(X,Y,Z,IE, H,HP,VX,VY,VZ,TH,DTH,                        GW3D2695
     1 BFLX,BFLXP,DCOSB,ISB,NPBB, MSEL,SOS,ISTYP,NPW,WSS,IWTYP,         GW3D2700
     2 NPVB,NPDB,NPCB,NPNB,  DELT, KFLOW)                               GW3D2705
```

```
C                                                                       GW3D2710
C ------- PRINT VARIABLES AT EACH TIME STEP                             GW3D2715
C                                                                       GW3D2720
      CALL PRINTT(VX,VY,VZ,H,HT,TH, NPBB,BFLX, NPVB,DCYFLX,NPCON,       GW3D2725
     1 NPFLX,NPMIN,SUBHD(1,3), TIME,DELT, KPR(ITM),KOUT,KDIAG,ITMITM)   GW3D2730
C                                                                       GW3D2735
      IF(KSTR.EQ.1 .AND.KDSK(ITM).EQ.1) CALL STORE(X,Y,Z,IE,            GW3D2740
     > H,HT,TH,VX,VY,VZ, DCOSB,ISB,NPBB, TITLE, TIME,NPROB)             GW3D2745
C                                                                       GW3D2750
C ------- PREPARE FOR NEXT TIME STEP                                    GW3D2755
C                                                                       GW3D2760
      IF(TIME.GT.TMAX) GO TO 100                                        GW3D2765
      DELT=DELT*(1.0D0+CHNG)                                            GW3D2770
      DELT=DMIN1(DELT,DELMAX)                                           GW3D2775
      IF(IDELT.EQ.0) GO TO 880                                          GW3D2780
      IF(TIME.EQ.TDTCH(IDELT)) DELT=DELT0                               GW3D2785
  880 TIME=TIME+DELT                                                    GW3D2790
      IF(TIME.LT.TDTCH(IDELT+1)) GO TO 890                              GW3D2795
      IDELT=IDELT+1                                                     GW3D2800
      TIME=TIME-DELT                                                    GW3D2805
      DELT=TDTCH(IDELT)-TIME                                            GW3D2810
      IF(DELT.LE.0.0) DELT=DELT0                                        GW3D2815
      TIME=TIME+DELT                                                    GW3D2820
  890 CONTINUE                                                          GW3D2825
C                                                                       GW3D2830
C ------- END OF TIME-MARCHING LOOP                                     GW3D2835
C                                                                       GW3D2840
      GO TO 100                                                         GW3D2845
C                                                                       GW3D2850
  990 RETURN                                                            GW3D2855
C                                                                       GW3D2860
   10 FORMAT(I5,9A8,3I1)                                                GW3D2865
 1000 FORMAT('1 PROBLEM',I5,'.. ',9A8,1X,3I1/)                          GW3D2870
10200 FORMAT(5X,I10,3X,E12.4,3X,E12.4,15X,I10)                          GW3D2875
10400 FORMAT('1','****************************************************',GW3D2880
     1 '**************************************************************',GW3D2885
     2 '*****'///' DIAGNOSTIC TABLE',I4,'.. AT TIME =',1PD12.4,         GW3D2890
     3 ' ,(DELT =', 1PD12.4,')')                                        GW3D2895
10401 FORMAT(//' TABLE OF ITERATIVE PARAMETERS FOR',I3,'-TH CYCLE'//6X, GW3D2900
     1 'ITERATION',7X,'RESIDUAL',6X,'DEVIATION',6X,                     GW3D2905
     2 'NO. NON-CONV. NODES')                                           GW3D2910
10402 FORMAT(//' TABLE OF RAINFALL/EVAPORATION-SEEPAGE B. C. USED FOR', GW3D2915
     1 I3,'-TH CYCLE'//7X,' I  NPVB  NPCON     HCON     NPMIN     HMIN',GW3D2920
     2 '     NPFLX       FLX         DCYFLX FROM PREVIOUS CYCLE'/7X,    GW3D2925
     3       ' - ---- -----     ----      -----     ----',              GW3D2930
     4 '     -----       ---         ---------------------------')      GW3D2935
10403 FORMAT(1H ,I8,I6,I7,D12.4,I7,D12.4,I7,D12.4,12X,D12.4)            GW3D2940
10500 FORMAT(//' TABLE OF NON-CONVERGING NODES')                        GW3D2945
10600 FORMAT(/(5X,20I5))                                                GW3D2950
```

```
10210 FORMAT(1H0,'WARING: NON-CONVERGENCE OCCUR DURING STEADY STATE SOLUGW3D2955
     1TION AT',I3,' -TH CYCLE'/1H ,'IT - ',I3,'  .GT.  MAXIT - ',I3,    GW3D2960
     2 ',  RES -',D12.5,',  RD -',D12.4,',  NNCVN -',I4)                GW3D2965
10610 FORMAT(1H0,'ABSOLUTELY WARNING: STEADY STATE SOLUTION IS NG'/1H , GW3D2970
     > 'ICY - ',I3,'  IT - ',I3,'  MAXCY - ',I3,'  MAXIT - ',I3,        GW3D2975
     > ',  RES -',D12.4,',  RD -',D12.4,',  NNCVN -',I4)                GW3D2980
10710 FORMAT(1H0,'WARNING: NON-CONVERGENCE OCCUR AT',I5,' -TH TIME STEP'GW3D2985
     >,I3,' -TH CYCLE'/1H ,'IT - ',I3,'  .GT.  MAXIT - ',I3,2D12.4,I5)  GW3D2990
10810 FORMAT(1H0,'ABSOLUTELY WARNING: TRANSIENT SOLUTION IS NG AT ',I5, GW3D2995
     > ' -TH TIME STEP'/1H ,'ICY - ',I3,'  IT - ',I3,'  MAXCY - ',I3,   GW3D3000
     > '  MAXIT - ',I3,',  RES -',D12.4,',  RD -',D12.4,',  NNCVN =',I4)GW3D3005
      END                                                               GW3D3010
```

```
      SUBROUTINE DATAIN(TITLE,NPROB, KPR,KDSK,TDTCH,                    DATA 005
     1 PROP,THPROP,AKPROP, X,Y,Z,IE,H, DCOSB,ISB,NPBB,                  DATA 010
     2 SOSF,TSOSF,ISTYP,MSEL, WSSF,TWSSF,IWTYP,NPW,                     DATA 015
     3 QCBF,TQCBF,ICTYP,ISC,NPCB,  QNBF,TQNBF,INTYP,ISN,NPNB,           DATA 020
     4 RF,TRF,IRTYP,ISV,NPVB, HCON,HMIN,  HDBF,THDBF,IDTYP,NPDB,        DATA 025
     5 ISTOP)                                                           DATA 030
C                                                                       DATA 035
C********1*********2*********3*********4*********5*********6*********7**DATA 040
C ------- TO READ AND PRINT SYSTEM PARAMETERS, GEOMETRY, BOUNDARY AND   DATA 045
C ------- INITIAL CONDITIONS, AND PROPERTIES OF THE MEDIA.              DATA 050
C********1*********2*********3*********4*********5*********6*********7**DATA 055
C                                                                       DATA 060
      IMPLICIT REAL*8(A-H,O-Z)                                          DATA 065
      REAL*4 PMAT,THPAR,AKPAR                                           DATA 070
C                                                                       DATA 075
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC      DATA 080
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI,NDTCHG                 DATA 085
      COMMON /CINTE/ NCYL,NITER,KSTR,KPRO,KDSKO,KSS,NPITER,IGEOM        DATA 090
      COMMON /CREAL/ DELT,CHNG,DELMAX,TMAX,DELTO,TOLA,TOLB,W,OME,OMI    DATA 095
C                                                                       DATA 100
      COMMON /CS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP                      DATA 105
      COMMON /CW/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP                      DATA 110
C                                                                       DATA 115
      COMMON /CCBC/ MXCNP,MXCES,MXCPR,MXCDP,NCNP,NCES,NCPR,NCDP         DATA 120
      COMMON /CNBC/ MXNNP,MXNES,MXNPR,MXNDP,NNNP,NNES,NNPR,NNDP         DATA 125
      COMMON /CVBC/ MXVES,MXVNP,MXRPR,MXRDP,NVES,NVNP,NRPR,NRDP         DATA 130
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP                    DATA 135
C                                                                       DATA 140
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM                                DATA 145
      COMMON /CMTL/ NMAT,NMPPM,NSPPM                                    DATA 150
C                                                                       DATA 155
      DIMENSION TITLE(9)                                                DATA 160
```

```
C
      DIMENSION KPR(MAXNTI),KDSK(MAXNTI),TDTCH(MXNDTC)
C
      DIMENSION PROP(MXMPPM,MAXMAT)
      DIMENSION THPROP(MXSPPM,MAXMAT),AKPROP(MXSPPM,MAXMAT)
C
      DIMENSION X(MAXNP),Y(MAXNP),Z(MAXNP),IE(MAXEL,9), H(MAXNP)
      DIMENSION DCOSB(3,MAXBES),ISB(6,MAXBES), NPBB(MAXBNP)
C
      DIMENSION MSEL(MXSEL),ISTYP(MXSEL)
      DIMENSION SOSF(MXSDP,MXSPR),TSOSF(MXSDP,MXSPR)
      DIMENSION NPW(MXWNP),IWTYP(MXWNP)
      DIMENSION WSSF(MXWDP,MXWPR),TWSSF(MXWDP,MXWPR)
C
      DIMENSION QCBF(MXCDP,MXCPR),TQCBF(MXCDP,MXCPR),ICTYP(MXCES)
      DIMENSION ISC(5,MXCES),NPCB(MXCNP)
      DIMENSION QNBF(MXNDP,MXNPR),TQNBF(MXNDP,MXNPR),INTYP(MXNES)
      DIMENSION ISN(5,MXNES),NPNB(MXNNP)
      DIMENSION RF(MXRDP,MXRPR),TRF(MXRDP,MXRPR), IRTYP(MXVES)
      DIMENSION ISV(5,MXVES),NPVB(MXVNP)
      DIMENSION HCON(MXVNP),HMIN(MXVNP)
      DIMENSION HDBF(MXDDP,MXDPR),THDBF(MXDDP,MXDPR),IDTYP(MXDNP)
      DIMENSION NPDB(MXDNP)
C
      DIMENSION NIMI(4),NJMJ(4),IEM(8)
      DIMENSION PMAT(3,6),AKPAR(3,8),THPAR(3,8)
C
      DATA PMAT/4H   S,4HAT K,4HXX  ,4H   S,4HAT K,4HYY  ,4H   S,
     > 4HAT K,4HZZ  ,4H   S,4HAT K,4HXY  ,4H   S,4HAT K,4HXZ  ,4H   S,
     >4HAT K,4HYZ  /
C
      DATA THPAR/4H    ,4H TH1,4H    ,4H    ,4H TH2,4H    ,4H    ,
     > 4H TH3,4H    ,4H    ,4H TH4,4H    ,4H    ,4H TH5,4H    ,4H    ,
     > 4H TH6,4H    ,4H    ,4H TH7,4H    ,4H    ,4H TH8,4H    /
C
      DATA AKPAR/4H    ,4H  K1,4H    ,4H    ,4H  K2,4H    ,4H    ,
     > 4H  K3,4H    ,4H    ,4H  K4,4H    ,4H    ,4H  K5,4H    ,
     > 4H    ,4H  K6,4H    ,4H    ,4H  K7,4H    ,4H    ,4H  K8,4H    /
C
      ISTOP-0
C
C ******* DATA SET 2: BASIC INTEGER PARAMETERS
C
      READ 10, NNP,NEL,NMAT,NCM,NTI,KSS,NSPPM,NMPPM,KSTR,
     1 KCP,KGRAV,NITER,NCYL,NDTCHG,NPITER,NREGN
      IF(NDTCHG.LE.0) NDTCHG-1
      PRINT 1000, NNP,NEL,NMAT,NCM,NTI,KSS,NSPPM,NMPPM,KSTR,
     1 KCP,KGRAV,NITER,NCYL,NDTCHG,NPITER,NREGN
```

```
C                                                                   DATA 405
C ******* DATA SET 3: BASIC REAL PARAMETERS                         DATA 410
C                                                                   DATA 415
      READ 20, DELT,CHNG,DELMAX,TMAX,TOLA,TOLB,RHO,GRAV,VISC,W,     DATA 420
     1 OME,OMI                                                      DATA 425
C                                                                   DATA 430
      DELTO-DELT                                                    DATA 435
      IF(TMAX.LE.0.0) TMAX-1.0D38                                   DATA 440
C                                                                   DATA 445
      PRINT 1100, DELT,CHNG,DELMAX,TMAX,TOLA,TOLB,RHO,GRAV,VISC,W,  DATA 450
     1 OME,OMI                                                      DATA 455
C                                                                   DATA 460
C ******* DATA SET 4: LINE PRINT CONTROL AND DISK STORE CONTROL     DATA 465
C                                                                   DATA 470
      READ 30, KPRO,(KPR(ITM),ITM-1,NTI)                            DATA 475
      READ 30, KDSKO,(KDSK(ITM),ITM-1,NTI)                          DATA 480
      READ 20, (TDTCH(I),I-1,NDTCHG)                                DATA 485
C                                                                   DATA 490
      PRINT 1200                                                    DATA 495
      PRINT 1210, KPRO,(KPR(ITM),ITM-1,NTI)                         DATA 500
      PRINT 1300                                                    DATA 505
      PRINT 1210,  KDSKO,(KDSK(ITM),ITM-1,NTI)                      DATA 510
      PRINT 1350,  (TDTCH(I),I-1,NDTCHG)                            DATA 515
C                                                                   DATA 520
C ******* DATA SET 5: MATERIAL PROPERTIES                           DATA 525
C                                                                   DATA 530
      PRINT 1400, ((PMAT(I,J),I-1,3),J-1,NMPPM)                     DATA 535
      DO 100 I-1,NMAT                                               DATA 540
      READ 20, (PROP(J,I),J-1,NMPPM)                                DATA 545
  100 PRINT 1410, I,(PROP(J,I),J-1,NMPPM)                           DATA 550
C                                                                   DATA 555
C ******* DATA SET 6: SOIL PROPERTIES                               DATA 560
C                                                                   DATA 565
C                                                                   DATA 570
C -------- READ AND PRINT PARAMETERS REQUIRE TO COMPUTE             DATA 575
C -------- MOISTURE CONTENT GIVEN THE PRESSURE HEAD                 DATA 580
C                                                                   DATA 585
      IF (NSPPM.EQ.0) GO TO 200                                     DATA 590
      PRINT 1500, ((THPAR(I,J),I-1,3),J-1,8)                        DATA 595
      DO 105 I-1,NMAT                                               DATA 600
      READ 20, (THPROP(J,I),J-1,NSPPM)                              DATA 605
      PRINT 1510, I,(THPROP(J,I),J-1,NSPPM)                         DATA 610
  105 CONTINUE                                                      DATA 615
C                                                                   DATA 620
C ------- READ AND PRINT PARAMETERS REQUIRED TO COMPUTE RELATIVE    DATA 625
C ------- HYDRAULIC CONDUCTIVITY                                    DATA 630
C                                                                   DATA 635
      PRINT 1600, ((AKPAR(I,J),I-1,3),J-1,8)                        DATA 640
      DO 110 I-1,NMAT                                               DATA 645
      READ 20, (AKPROP(J,I),J-1,NSPPM)                              DATA 650
```

```
      PRINT 1510, I,(AKPROP(J,I),J-1,NSPPM)
  110 CONTINUE
      IF (KCP.EQ.0) GO TO 200
C
C ------- CONVERT FROM SATURATED PERMEABILITY TO SATURATED CONDUCTIVITY
C
      DO 190 I-1,NMAT
      PKCF-RHO*GRAV/VISC
      DO 190 J-1,6
      PROP(J,I)-PROP(J,I)*PKCF
  190 CONTINUE
C
C ******* DATA SET 7: NODE COORDINATES
C
C ------- READ NODAL POINT COORDINATES
C
  200 NPI-0
  210 READ 40, NI,NSEQ,NIAD,XI,YI,ZI,XIAD,YIAD,ZIAD
      IF(NI.EQ.0) GO TO 240
      NJ-NI+NSEQ
      DO 220 NP-NI,NJ
      I-NI+NIAD*(NP-NI)
      X(I)-XI+XIAD*DFLOAT(NP-NI)
      Y(I)-YI+YIAD*DFLOAT(NP-NI)
      Z(I)-ZI+ZIAD*DFLOAT(NP-NI)
      NPI-NPI+1
  220 CONTINUE
      GO TO 210
  240 IF(NPI.EQ.NNP) GO TO 250
      PRINT 2000
      STOP
  250 CONTINUE
C
C ------- PRINT NODAL POINT COORDINATES
C
      IF(MOD(IGEOM,2).EQ.0) GO TO 270
      LINE-0
      DO 265 NP-1,NNP,3
      NJMN-NP
      NJMX-MIN0(NP+2,NNP)
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 2100
      PRINT 2110, (NJ,X(NJ),Y(NJ),Z(NJ),NJ-NJMN,NJMX)
  265 CONTINUE
  270 CONTINUE
C
C ******* DATA SET 8: ELEMENT DATA
C
C ------- READ ELEMENT INDICES AND COMPUTE MAXIMUM NODAL DIFFERENCE FOR
C ------- EACH ELEMENT
```

```
C                                                                       DATA 905
      MMP-0                                                             DATA 910
  300 READ 10, MI,NSEQ,MIAD,(IEM(I),I-1,8),IEMAD                        DATA 915
      IF(MI.EQ.0) GO TO 330                                             DATA 920
C                                                                       DATA 925
      MJ-MI+NSEQ                                                        DATA 930
      DO 320 MP-MI,MJ                                                   DATA 935
      M-MI+(MP-MI)*MIAD                                                 DATA 940
      DO 310 IQ-1,8                                                     DATA 945
      NI-IEM(IQ)+(MP-MI)*IEMAD                                          DATA 950
  310 IE(M,IQ)-NI                                                       DATA 955
      MMP-MMP+1                                                         DATA 960
  320 CONTINUE                                                          DATA 965
      GO TO 300                                                         DATA 970
C                                                                       DATA 975
  330 IF(MMP.EQ.NEL) GO TO 350                                          DATA 980
      PRINT 3000, MMP,NEL                                               DATA 985
      STOP                                                              DATA 990
C                                                                       DATA 995
  350 DO 360 M-1,NEL                                                    DATA1000
  360 IE(M,9)-1                                                         DATA1005
C                                                                       DATA1010
C ******* DATA SET 9: MATERIAL CORRECTIONS                              DATA1015
C                                                                       DATA1020
      IF (NCM.LE.0) GO TO 405                                           DATA1025
      CALL READN(IE(1,9),MAXEL,NCM)                                     DATA1030
C                                                                       DATA1035
C ------- PRINT ELEMENT INCIDENCE AND MATERIAL TYPES FOR EACH ELEMENT   DATA1040
C                                                                       DATA1045
  405 CONTINUE                                                          DATA1050
      IF(MOD(IGEOM,2).EQ.0) GO TO 415                                   DATA1055
      LINE-0                                                            DATA1060
      DO 410 NI-1,NEL,2                                                 DATA1065
      NJMN-NI                                                           DATA1070
      NJMX-MIN0(NI+1,NEL)                                               DATA1075
      LINE-LINE+1                                                       DATA1080
      IF(MOD(LINE-1,50).EQ.0) PRINT 2700                                DATA1085
      PRINT 3100, (NJ,(IE(NJ,K),K-1,9),NJ-NJMN,NJMX)                    DATA1090
  410 CONTINUE                                                          DATA1095
  415 CONTINUE                                                          DATA1100
C                                                                       DATA1105
C ------- CHECK IF MATERIAL TYPE FOR EACH ELEMENT IS CORRECT            DATA1110
C                                                                       DATA1115
      DO 420 M-1,NEL                                                    DATA1120
      MTYP-IE(M,9)                                                      DATA1125
      IF(MTYP.GT.0 .AND. MTYP.LE.NMAT) GO TO 420                        DATA1130
      PRINT 4200, M                                                     DATA1135
      ISTOP-ISTOP+1                                                     DATA1140
  420 CONTINUE                                                          DATA1145
```

```
C
      IF(ISTOP.EQ.0) GO TO 430
      PRINT 4300, ISTOP
      STOP
C
C ------- IDENTIFY BOUNDARY ELEMENTS AND COMPUTE DIRECTIONAL COSINES
C
  430 CONTINUE
      IF(IGEOM.LE.1) CALL SURF(X,Y,Z,IE,DCOSB,ISB,NPBB)
      REWIND 3
      IF(IGEOM.LE.1) WRITE(3) NBES,NBNP,((DCOSB(J,I),J-1,3),I-1,NBES),
     1 ((ISB(J,I),J-1,6),I-1,NBES),(NPBB(I),I-1,NBNP)
      IF(IGEOM.GT.1) READ(3) NBES,NBNP,((DCOSB(J,I),J-1,3),I-1,NBES),
     1 ((ISB(J,I),J-1,6),I-1,NBES),(NPBB(I),I-1,NBNP)
C
C ******* DATA SET 10: INITIAL CONDITIONS
C
C
C ------- READ INITIAL OR PRE-INITIAL CONDITIONS VIA CARDS
C
      CALL READR(H,MAXNP,NNP)
C
C ******* DATA SET 11: INTEGERS CONTROLLING SOURCES AND B.C.
C
      READ 10, NSEL,NSPR,NSDP, NWNP,NWPR,NWDP, NCES,NCNP,NCPR,NCDP,
     1 NNES,NNNP,NNPR,NNDP, NVES,NVNP,NRPR,NRDP, NDNP,NDPR,NDDP
C
      PRINT 5100, NSEL,NSPR,NSDP, NWNP,NWPR,NWDP
      PRINT 5150, NCES,NCNP,NCPR,NCDP,NNES,NNNP,NNPR,NNDP,
     1 NVES,NVNP,NRPR,NRDP, NDNP,NDPR,NDDP
C
C ******* DATA SET 12: SOURCE DATA
C
      IF(NSEL.EQ.0) GO TO 560
      PRINT 5300
      DO 510 I-1,NSPR
      READ 20, (TSOSF(J,I),SOSF(J,I),J-1,NSDP)
      PRINT 5500, I
      PRINT 5510, (TSOSF(J,I),SOSF(J,I),J-1,NSDP)
  510 CONTINUE
C
C ------- READ SOURCE TYPE ASSIGNED TO EACH ELEMENT
C
      READ 10, (MSEL(M),M-1,NSEL)
      CALL READN(ISTYP,MXSEL,NSEL)
C
C ------- PRINT ELEMENT SOURCE/SINK PROFILES AND TYPE
C
      LINE-0
```

```
      DO 520 I-1,NSEL,5
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 5600
      NJMN-I
      NJMX-MINO(I+4,NSEL)
      PRINT 5650, (J,MSEL(J),ISTYP(J),J-NJMN,NJMX)
  520 CONTINUE
C
C -------  READ AND WRITE WELL SOURCE/SINK PROFILES
C
  560 IF(NWNP.EQ.0) GO TO 600
      PRINT 5700
      DO 570 I-1,NWPR
      READ 20, (TWSSF(J,I),WSSF(J,I),J-1,NWDP)
      PRINT 5710, I
      PRINT 5510, (TWSSF(J,I),WSSF(J,I),J=1,NWDP)
  570 CONTINUE
C
C ------- READ WELL SOURCE/SINK NODES AND TYPE OF PROFILES ASSIGNED TO
C ------- EACH OF NWNP NODES.
C
      READ 10, (NPW(I),I-1,NWNP)
      CALL READN(IWTYP,MXWNP,NWNP)
C
C ------- PRINT GLOBAL WELL NODE NUMBERS AND PROFILE TYPE OF WELL NODE
C
      LINE-0
      DO 590 I-1,NWNP,5
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 5800
      NJMN-I
      NJMX-MINO(I+4,NWNP)
      PRINT 5850, (J,NPW(J),IWTYP(J),J-NJMN,NJMX)
  590 CONTINUE
C
C ******* DATA SET 13: RAINFALL/EVAPORATION-SEEPAGE BOUNDARY CONDITIONS
C
  600 IF(NVES.EQ.0) GO TO 700
C
      PRINT 6000
C
C ------- READ AND WRITE RAINFALL (+)/EVAPORATION (-) PROFILES
C
      PRINT 6100
      DO 610 I-1,NRPR
         READ 20, (TRF(J,I),RF(J,I),J-1,NRDP)
      PRINT 6150, I
      PRINT 5510, (TRF(J,I),RF(J,I),J-1,NRDP)
  610 CONTINUE
```

```
C                                                                       DATA1640
C ------- READ RAINFALL/EVAPORATION TYPE ASSIGNED TO EACH RS SIDE       DATA1645
C                                                                       DATA1650
      CALL READN(IRTYP,MXVES,NVES)                                      DATA1655
C                                                                       DATA1660
C ------- READ FOUR GLOBAL NODE NUMBER FOR EACH OF ALL VARIABLE         DATA1665
C ------- BOUNDARY ELEMENT SIDES.                                       DATA1670
C                                                                       DATA1675
      MPI=0                                                             DATA1680
  620 READ 10, MI,NSEQ,MIAD,I1,I2,I3,I4,I1AD,I2AD,I3AD,I4AD             DATA1685
      IF(MI.EQ.0) GO TO 630                                             DATA1690
      MJ=MI+NSEQ                                                        DATA1695
      DO 625 MP=MI,MJ                                                   DATA1700
      I=MI+(MP-MI)*MIAD                                                 DATA1705
      ISV(1,I)=I1+(MP-MI)*I1AD                                          DATA1710
      ISV(2,I)=I2+(MP-MI)*I2AD                                          DATA1715
      ISV(3,I)=I3+(MP-MI)*I3AD                                          DATA1720
      ISV(4,I)=I4+(MP-MI)*I4AD                                          DATA1725
      MPI=MPI+1                                                         DATA1730
  625 CONTINUE                                                          DATA1735
      GO TO 620                                                         DATA1740
  630 IF(MPI.EQ.NVES) GO TO 635                                         DATA1745
      PRINT 6300                                                        DATA1750
      STOP                                                              DATA1755
C                                                                       DATA1760
C ------- PRINT INPUTTED GLOBAL NODAL NUMBER AND RAINFALL TYPES OF ALL  DATA1765
C ------- VARIABLE BOUNDARY ELEMENT SIDES.                              DATA1770
C                                                                       DATA1775
  635 LINE=0                                                            DATA1780
      DO 640 MP=1,NVES,3                                                DATA1785
      LINE=LINE+1                                                       DATA1790
      IF(MOD(LINE-1,50).EQ.0) PRINT 6400                                DATA1795
      NJMN=MP                                                           DATA1800
      NJMX=MIN0(MP+2,NVES)                                              DATA1805
      PRINT 6450, (J,(ISV(I,J),I=1,4),IRTYP(J),J=NJMN,NJMX)             DATA1810
  640 CONTINUE                                                          DATA1815
C                                                                       DATA1820
C ------- READ GLOBAL NODAL NUMBER FOR EACH OF ALL VARIABLE NODES.      DATA1825
C                                                                       DATA1830
      CALL READN(NPVB,MXVNP,NVNP)                                       DATA1835
C                                                                       DATA1840
C ------- READ PONDING DEPTH AND MINIMUM HEAD FOR EACH OF ALL RS NODES  DATA1845
C                                                                       DATA1850
      CALL READR(HCON,MXVNP,NVNP)                                       DATA1855
      CALL READR(HMIN,MXVNP,NVNP)                                       DATA1860
C                                                                       DATA1865
C ------- PRINT GLOBAL NODAL NUMBER, PONDING DEPTH AND MINIMUM PRESURE  DATA1870
C ------- RESSURE HEAD FOR ALL VARIABLE BOUNDARY NODES                  DATA1875
C                                                                       DATA1880
      LINE=0                                                            DATA1885
```

```
      DO 645 I=1,NVNP,3
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 6500
      NJMN=I
      NJMX=MIN0(I+2,NVNP)
      PRINT 6550, (J,NPVB(J),HCON(J),HMIN(J),J=NJMN,NJMX)
  645 CONTINUE
C
C ------- COMPUTE BOUNDARY SIDE NUMBER FOR EACH OF ALL VARIABLE
C ------- BOUNDARY SIDES.
C
      DO 659 MI=1,NVES
      DO 651 IQ=1,4
  651 NIMI(IQ)=ISV(IQ,MI)
C
      DO 657 MJ=1,NBES
      DO 652 JQ=1,4
      IJ=ISB(JQ,MJ)
  652 NJMJ(JQ)=NPBB(IJ)
      IEQ=0
      DO 656 IQ=1,4
      NI=NIMI(IQ)
      DO 653 JQ=1,4
      NJ=NJMJ(JQ)
      IF(NJ.EQ.NI) GO TO 655
  653 CONTINUE
      GO TO 657
  655 IEQ=IEQ+1
  656 CONTINUE
      IF(IEQ.EQ.4) GO TO 658
  657 CONTINUE
C
      PRINT 6570, MI
      STOP
  658 ISV(5,MI)=MJ
C
  659 CONTINUE
C
C ------- CHANGE NPVB FROM CONTAINING GLOBAL NODAL NUMBER TO
C ------- CONTAINING BOUNDARY NODAL NUMBER.
C
      DO 669 NP=1,NVNP
      NI=NPVB(NP)
C
      DO 665 I=1,NBNP
      NJ=NPBB(I)
      IF(NJ.NE.NI) GO TO 665
      NII=I
      GO TO 667
  665 CONTINUE
```

```
C
      PRINT 6670, NP
      STOP
  667 NPVB(NP)-NII
C
  669 CONTINUE
C
C --------- PRINT COMPUTED BOUNDARY NODAL NUMBER FOR ALL VB NODES
C
      LINE-0
      DO 670 I-1,NVNP,10
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 6700
      NJMN-I
      NJMX-MINO(I+9,NVNP)
      PRINT 6750, (J,NPVB(J),J-NJMN,NJMX)
  670 CONTINUE
C
C ------- CHANGE ISV(I,MP) I-1,4 FROM CONTAINING GLOBAL NODAL
C ------- NUMBER TO CONTAINING BOUNDARY NODAL NUMBER.
C
      DO 690 MP-1,NVES
      MPB-ISV(5,MP)
      DO 685 IQ-1,4
      NB-ISB(IQ,MPB)
      DO 675 I-1,NVNP
      NI-NPVB(I)
      IF(NI.NE.NB) GO TO 675
      NII-I
      GO TO 680
  675 CONTINUE
      PRINT 6751, IQ,MP
      STOP
  680 ISV(IQ,MP)-NII
  685 CONTINUE
  690 CONTINUE
C
C ------- PRINT COMPUTED BOUNDARY NODAL NUMBER & SIDE NUMBER AND
C ------- RAINFALL TYPES FOR ALL VB SIDES
C
      LINE-0
      DO 695 MP-1,NVES,3
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 6900
      NJMN-MP
      NJMX-MINO(MP+2,NVES)
      PRINT 6950, (J,(ISV(I,J),I-1,5),IRTYP(J),J-NJMN,NJMX)
  695 CONTINUE
```

```
C                                                                       DATA2380
C ******* DATA SET 14: DIRICHLET BOUNDARY CONDITIONS                     DATA2385
C                                                                       DATA2390
C ------- READ AND PRINT TOTAL DIRICHLET HEAD PROFILES                  DATA2395
C                                                                       DATA2400
  700 IF(NDNP.EQ.0) GO TO 800                                           DATA2405
      PRINT 7000                                                        DATA2410
      DO 710 I=1,NDPR                                                   DATA2415
      READ 20, (THDBF(J,I),HDBF(J,I),J=1,NDDP)                          DATA2420
      PRINT 7100, I                                                     DATA2425
      PRINT 5510, (THDBF(J,I),HDBF(J,I),J=1,NDDP)                       DATA2430
  710 CONTINUE                                                          DATA2435
C                                                                       DATA2440
C ------- READ GLOBAL NODAL NUMBER OF ALL DIRICHLET NODES AND           DATA2445
C ------- THE TYPE OF TOTAL HEAD ASSIGNED TO EACH OF THEM.              DATA2450
C                                                                       DATA2455
      READ 10, (NPDB(I),I=1,NDNP)                                       DATA2460
      CALL READN(IDTYP,MXDNP,NDNP)                                      DATA2465
C                                                                       DATA2470
C --------- PRINT GLOBAL NODAL NUMBER AND PROFILE OF DIRICHLET NODES    DATA2475
C                                                                       DATA2480
      LINE=0                                                            DATA2485
      DO 720 I=1,NDNP,5                                                 DATA2490
      LINE=LINE+1                                                       DATA2495
      IF(MOD(LINE-1,50).EQ.0) PRINT 7200                                DATA2500
      NJMN=I                                                            DATA2505
      NJMX=MIN0(I+4,NDNP)                                               DATA2510
      PRINT 7250, (J,NPDB(J),IDTYP(J),J=NJMN,NJMX)                      DATA2515
  720 CONTINUE                                                          DATA2520
C                                                                       DATA2525
C ------- CHANGE NPDB FROM CONTAINING GLOBAL NODAL NUMBER TO            DATA2530
C ------- CONTAINING BOUNDARY NODAL NUMBER.                             DATA2535
C                                                                       DATA2540
      DO 769 NP=1,NDNP                                                  DATA2545
      NI=NPDB(NP)                                                       DATA2550
C                                                                       DATA2555
      DO 765 I=1,NBNP                                                   DATA2560
      NJ=NPBB(I)                                                        DATA2565
      IF(NJ.NE.NI) GO TO 765                                            DATA2570
      NII=I                                                             DATA2575
      GO TO 767                                                         DATA2580
  765 CONTINUE                                                          DATA2585
C                                                                       DATA2590
      PRINT 7670, NP                                                    DATA2595
      STOP                                                              DATA2600
  767 NPDB(NP)=NII                                                      DATA2605
C                                                                       DATA2610
  769 CONTINUE                                                          DATA2615
```

```
C
C --------- PRINT COMPUTED BOUNDARY NODAL NUMBER AND TYPE OF PROFILES
C --------- FOR DIRICHLET BOUNDARY NODES
C
      LINE=0
      DO 770 I=1,NDNP,5
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 7700
      NJMN=I
      KJMX=MIN0(I+4,NDNP)
      PRINT 7750, (J,NPDB(J),IDTYP(J),J=NJMN,NJMX)
  770 CONTINUE
C
C ******* DATA SET 15: CAUCHY BOUNDARY CONDITIONS
C
C ------- READ AND PRINT CAUCHY FLUX PROFILES
C
  800 IF(NCES.EQ.0) GO TO 900
      PRINT 8000
      DO 810 I=1,NCPR
      READ 20, (TQCBF(J,I),QCBF(J,I),J=1,NCDP)
      PRINT 8100, I
      PRINT 5510, (TQCBF(J,I),QCBF(J,I),J=1,NCDP)
  810 CONTINUE
C
C ------- READ CAUCHY FLUX TYPE ASSIGNED TO EACH CAUCHY SIDE
C
      CALL READN(ICTYP,MXCES,NCES)
C
C ------- READ FOUR GLOBAL NODE NUMBER FOR EACH OF ALL CAUCHY SIDES
C
      MPI=0
  820 READ 10, MI,NSEQ,MIAD,I1,I2,I3,I4,I1AD,I2AD,I3AD,I4AD
      IF(MI.EQ.0) GO TO 830
      MJ=MI+NSEQ
      DO 825 MP=MI,MJ
      I=MI+(MP-MI)*MIAD
      ISC(1,I)=I1+(MP-MI)*I1AD
      ISC(2,I)=I2+(MP-MI)*I2AD
      ISC(3,I)=I3+(MP-MI)*I3AD
      ISC(4,I)=I4+(MP-MI)*I4AD
      MPI=MPI+1
  825 CONTINUE
      GO TO 820
  830 IF(MPI.EQ.NCES) GO TO 835
      PRINT 8300
      STOP
```

```
C
C ------- PRINT INPUTTED GLOBAL NODAL NUMBER AND CAUCHY FLUX TYPES
C ------- FOR ALL CAUCHY BOUNDARY ELEMENT SIDES.
C
  835 LINE-0
      DO 840 MP-1,NCES,3
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 8400
      NJMN-MP
      NJMX-MINO(MP+2,NCES)
      PRINT 8450, (J,(ISC(I,J),I-1,4),ICTYP(J),J-NJMN,NJMX)
  840 CONTINUE
C
C ------- READ GLOBAL NODAL NUMBER FOR EACH OF ALL CAUCHY NODES.
C
      READ 10, (NPCB(I),I-1,NCNP)
C
C ------- PRINT GLOBAL NODAL NUMBER FOR ALL CAUCHY NODES
C
      LINE-0
      DO 845 I-1,NCNP,10
      LINE-LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 8500
      NJMN-I
      NJMX-MINO(I+9,NCNP)
      PRINT 8550, (J,NPCB(J),J-NJMN,NJMX)
  845 CONTINUE
C
C ------- COMPUTE BOUNDARY SIDE NUMBER FOR ALL CAUSHY SIDES
C
      DO 859 MI-1,NCES
      DO 851 IQ-1,4
  851 NIMI(IQ)-ISC(IQ,MI)
C
      DO 857 MJ-1,NBES
      DO 852 JQ-1,4
      IJ-ISB(JQ,MJ)
  852 NJMJ(JQ)-NPBB(IJ)
      IEQ-0
      DO 856 IQ-1,4
      NI-NIMI(IQ)
      DO 853 JQ-1,4
      NJ-NJMJ(JQ)
      IF(NJ.EQ.NI) GO TO 855
  853 CONTINUE
      GO TO 857
  855 IEQ-IEQ+1
  856 CONTINUE
      IF(IEQ.EQ.4) GO TO 858
  857 CONTINUE
```

```
C                                                                       DATA3105
      PRINT 8570, MI                                                    DATA3110
      STOP                                                              DATA3115
  858 ISC(5,MI)=MJ                                                      DATA3120
C                                                                       DATA3125
  859 CONTINUE                                                          DATA3130
C                                                                       DATA3135
C ------- CHANGE NPCB FROM CONTAINING GLOBAL NODAL NUMBER TO            DATA3140
C ------- CONTAINING BOUNDARY NODAL NUMBER.                             DATA3145
C                                                                       DATA3150
      DO 869 NP=1,NCNP                                                  DATA3155
      NI=NPCB(NP)                                                       DATA3160
C                                                                       DATA3165
      DO 865 I=1,NBNP                                                   DATA3170
      NJ=NPBB(I)                                                        DATA3175
      IF(NJ.NE.NI) GO TO 865                                            DATA3180
      NII=I                                                             DATA3185
      GO TO 867                                                         DATA3190
  865 CONTINUE                                                          DATA3195
C                                                                       DATA3200
      PRINT 8670, NP                                                    DATA3205
      STOP                                                              DATA3210
  867 NPCB(NP)=NII                                                      DATA3215
C                                                                       DATA3220
  869 CONTINUE                                                          DATA3225
C                                                                       DATA3230
C --------- PRINT COMPUTED BOUNDARY NODAL NUMBER FOR ALL CAUCHY NODES   DATA3235
C                                                                       DATA3240
      LINE=0                                                            DATA3245
      DO 870 I=1,NCNP,10                                                DATA3250
      LINE=LINE+1                                                       DATA3255
      IF(MOD(LINE-1,50).EQ.0) PRINT 8700                                DATA3260
      NJMN=I                                                            DATA3265
      NJMX=MIN0(I+9,NCNP)                                               DATA3270
      PRINT 8750, (J,NPCB(J),J=NJMN,NJMX)                               DATA3275
  870 CONTINUE                                                          DATA3280
C                                                                       DATA3285
C ******* DATA SET 16:  NEUMANN BOUNDARY CONDITIONS                     DATA3290
C                                                                       DATA3295
C ------- READ AND PRINT NEUMANN FLUX PROFILES                          DATA3300
C                                                                       DATA3305
  900 IF(NNES.EQ.0) GO TO 999                                           DATA3310
      PRINT 9000                                                        DATA3315
      DO 910 I=1,NNPR                                                   DATA3320
      READ 20, (TQNBF(J,I),QNBF(J,I),J=1,NNDP)                          DATA3325
      PRINT 9100, I                                                     DATA3330
      PRINT 5510, (TQNBF(J,I),QNBF(J,I),J=1,NNDP)                       DATA3335
  910 CONTINUE                                                          DATA3340
```

```
C                                                                       DATA3345
C ------- READ NEUMANN FLUX TYPE ASSIGNED TO EACH NEUMANN SIDE          DATA3350
C                                                                       DATA3355
      CALL READN(INTYP,MXNES,NNES)                                      DATA3360
C                                                                       DATA3365
C ------- READ FOUR GLOBAL NODE NUMBER FOR EACH OF ALL NEUMANN SIDES    DATA3370
C                                                                       DATA3375
      MPI-0                                                             DATA3380
  920 READ 10, MI,NSEQ,MIAD,I1,I2,I3,I4,I1AD,I2AD,I3AD,I4AD             DATA3385
      IF(MI.EQ.0) GO TO 930                                             DATA3390
      MJ-MI+NSEQ                                                        DATA3395
      DO 925 MP-MI,MJ                                                   DATA3400
      I-MI+(MP-MI)*MIAD                                                 DATA3405
      ISN(1,I)-I1+(MP-MI)*I1AD                                          DATA3410
      ISN(2,I)-I2+(MP-MI)*I2AD                                          DATA3415
      ISN(3,I)-I3+(MP-MI)*I3AD                                          DATA3420
      ISN(4,I)-I4+(MP-MI)*I4AD                                          DATA3425
      MPI-MPI+1                                                         DATA3430
  925 CONTINUE                                                          DATA3435
      GO TO 920                                                         DATA3440
  930 IF(MPI.EQ.NNES) GO TO 935                                         DATA3445
      PRINT 9300                                                        DATA3450
      STOP                                                              DATA3455
C                                                                       DATA3460
C ------- PRINT INPUTTED GLOBAL NODAL NUMBER AND NEUMANN FLUX TYPES     DATA3465
C ------- FOR ALL NEUMANN BOUNDARY ELEMENT SIDES.                       DATA3470
C                                                                       DATA3475
  935 LINE-0                                                            DATA3480
      DO 940 MP-1,NNES,3                                                DATA3485
      LINE-LINE+1                                                       DATA3490
      IF(MOD(LINE-1,50).EQ.0) PRINT 9400                                DATA3495
      NJMN-MP                                                           DATA3500
      NJMX-MINO(MP+2,NNES)                                              DATA3505
      PRINT 9450, (J,(ISN(I,J),I-1,4),INTYP(J),J-NJMN,NJMX)             DATA3510
  940 CONTINUE                                                          DATA3515
C                                                                       DATA3520
C ------- .EAD GLOBAL NODAL NUMBER FOR EACH OF ALL NEUMANN NODES.       DATA3525
C                                                                       DATA3530
      READ 10, (NPNB(I),I-1,NNNP)                                       DATA3535
C                                                                       DATA3540
C ------- PRINT GLOBAL NODAL NUMBER FOR ALL NEUMANN NODES               DATA3545
C                                                                       DATA3550
      LINE-0                                                            DATA3555
      DO 945 I-1,NNNP,10                                                DATA3560
      LINE-LINE+1                                                       DATA3565
      IF(MOD(LINE-1,50).EQ.0) PRINT 9500                                DATA3570
      NJMN-I                                                            DATA3575
      NJMX-MINO(I+9,NNNP)                                               DATA3580
      PRINT 9550, (J,NPNB(J),J-NJMN,NJMX)                               DATA3585
  945 CONTINUE                                                          DATA3590
```

```
C
C ------- COMPUTE BOUNDARY SIDE NUMBER FOR EACH OF NEUMANN
C ------- BOUNDARY SIDES.
C
      DO 959 MI-1,NNES
      DO 951 IQ-1,4
  951 NIMI(IQ)-ISN(IQ,MI)
C
      DO 957 MJ-1,NBES
      DO 952 JQ-1,4
      IJ-ISB(JQ,MJ)
  952 NJMJ(JQ)-NPBB(IJ)
      IEQ-0
      DO 956 IQ-1,4
      NI-NIMI(IQ)
      DO 953 JQ-1,4
      NJ-NJMJ(JQ)
      IF(NJ.EQ.NI) GO TO 955
  953 CONTINUE
      GO TO 957
  955 IEQ-IEQ+1
  956 CONTINUE
      IF(IEQ.EQ.4) GO TO 958
  957 CONTINUE
C
      PRINT 9570, MI
      STOP
  958 ISN(5,MI)-MJ
C
  959 CONTINUE
C
C ------- CHANGE NPNB FROM CONTAINING GLOBAL NODAL NUMBER TO
C ------- CONTAINING BOUNDARY NODAL NUMBER.
C
      DO 969 NP-1,NNNP
      NI-NPNB(NP)
C
      DO 965 I-1,NBNP
      NJ-NPBB(I)
      IF(NJ.NE.NI) GO TO 965
      NI1-I
      GO TO 967
  965 CONTINUE
C
      PRINT 9670, NP
      STOP
  967 NPNB(NP)-NI1
C
  969 CONTINUE
```

```
C
C --------- PRINT COMPUTED BOUNDARY NODAL NUMBER FOR ALL NEUMANN NODES
C
      LINE=0
      DO 970 I=1,NNNP,10
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ 0) PRINT 9700
      NJMN=I
      NJMX=MIN0(I+9,NNNP)
      PRINT 9750, (J,NPNB(J),J=NJMN,NJMX)
  970 CONTINUE
C
  999 CONTINUE
C
      RETURN
C
   10 FORMAT(16I5)
   20 FORMAT(8D10.3)
   30 FORMAT(80I1)
   40 FORMAT(3I5,5X,6D10.3)
C
 1000 FORMAT(40H0      **** BAISC INTEGER PARAMETERS ****//5X,
     1 40H NUMBER OF NODAL POINTS. . . . . . . . . .,I5/ 5X,
     2 40H NUMBER OF ELEMENTS. . . . . . . . . . . .,I5/ 5X,
     3 40H NUMBER OF DIFFERENT MATERIALS . . . . . .,I5/ 5X,
     4 40H NUMBER OF CORRECTION MATERIALS. . . . . .,I5/ 5X,
     5 40H NUMBER OF TIME INCREMENTS . . . . . . . .,I5//5X,
     6 40H STEADY-STATE I.C. CONTROL . . . . . . . .,I5/ 5X,
     8 40H NUMBER OF SOIL PARAMETERS . . . . . . . .,I5/ 5X,
     9 40H NUMBER OF MATERIAL PROPERTIES . . . . . .,I5//5X,
     A 40H AUXILIARY STORAGE CONTROL . . . . . . . .,I5/ 5X,
     B 40H CONDUCTIVITY-PERMEABILITY CONTROL . . . .,I5/ 5X,
     C 40H GRAVITY CONTROL . . . . . . . . . . . . .,I5/ 5X,
     E 40H NO. OF ITERATIONS PER CYCLE . . . . . . .,I5/ 5X,
     F 40H NO. OF CYCLES PER TIME STEP . . . . . . .,I5/ 5X,
     G 40H NO. OF TIMES TO RESET TIME STEP SIZE  . .,I5/ 5X,
     H 40H NO. OF BLOCKWISE ITERATIONS ALLOWED .  .,I5/ 5X,
     I 40H NO. OF SUBREGIONS . . . . . . . . . . . .,I5/)
 1100 FORMAT(5X,40H TIME INCREMENT. . . . . . . . . . . . . . . .,E15.6/ 5X,
     1 40H MULTIPLIER FOR INCREASING DELT. . . . . .,E15.6/ 5X,
     2 40H MAXIMUM VALUE OF DELT . . . . . . . . . .,E15.6/ 5X,
     3 40H MAXIMUM VALUE OF TIME . . . . . . . . . .,E15.6//5X,
     5 40H STEADY-STATE TOLERANCE. . . . . . . . . .,E15.6/ 5X,
     6 40H TRANSIENT-STATE TOLERANCE . . . . . . . .,E15.6//5X,
     7 40H DENSITY OF WATER. . . . . . . . . . . . .,E15.6/ 5X,
     8 40H ACCELERATION OF GRAVITY . . . . . . . . .,E15.6/ 5X,
     9 40H VISCOSITY OF WATER. . . . . . . . . . . .,E15.6//5X,
     A 40H TIME-INTEGRATION PARAMETER. . . . . . . .,E15.6/ 5X,
     B 40H ITERATION PARAMETER FOR NONLINEAR EQ.   ,E15.6/ 5X,
     C 40H RELAXATION PARAMETER FOR POINTWISE SOL. ,E15.6//)
```

```
 1200 FORMAT(//6X,14HOUTPUT CONTROL)                                     DATA4090
 1210 FORMAT(10X,50I2)                                                   DATA4095
 1300 FORMAT(//6X,19HDISK OUTPUT CONTROL)                                DATA4100
 1350 FORMAT(1H0,6X,'TIME OF CHANGING DELT'/(10X,8D12.4))                DATA4105
 1400 FORMAT(36H1       **** MATERIAL PROPERTIES **** // 9H MAT. NO., 9( DATA4110
     > 3A4))                                                             DATA4115
 1410 FORMAT(I8,9D12.4)                                                  DATA4120
 1500 FORMAT(44H1INPUT TABLE 3.  MOISTURE-CONTENT PARAMETERS//           DATA4125
     > 9H MAT. NO.,8(3A4))                                               DATA4130
 1510 FORMAT(I8,9D12.4/(8X,9D12.4))                                      DATA4135
 1600 FORMAT(40H1       **** CONDUCTIVITY PARAMETERS **** // 9H MAT. NO., DATA4140
     > 8(3A4))                                                           DATA4145
 2000 FORMAT(1H0/5X,'*** ERROR IN READING COORDINATE STOP ***'/)         DATA4150
 2100 FORMAT(38H1       **** NODAL COORDINATE DATA **** //1X,            DATA4155
     > 3(1X,5H NODE,11H     X     ,11H     Y     ,11H     Z     ,1X)/1X,DATA4160
     > 3(1X,5H ****,33H *********  *********  ********* ,1X))           DATA4165
 2110 FORMAT(1H ,3(1X,I5,3D11.3,1X))                                     DATA4170
 2700 FORMAT(62H1       **** ELEMENT DATA: GLOBAL INDICES OF ELEMENT NODESDATA4175
     1 **** //2(5X,5H  ELM,5H NOD1,5H NOD2,5H NOD3,5H NOD4,5H NOD5,      DATA4180
     2 5H NOD6,5H NOD7,5H NOD8,5H MTYP)/2(5X,5H  ---,5H ----,5H ----,    DATA4185
     3 5H ----,5H ----,5H ----,5H ----,5H ----,5H ----,5H ----))         DATA4190
 3000 FORMAT(////'ERROR IN READING IE, MMP =',I5,' NEL =',I5,' STOP')    DATA4195
 3100 FORMAT(2(5X,10I5))                                                 DATA4200
 4200 FORMAT(////40H ERROR IN MATERIAL TYPE CODE FOR ELEMENT,I5///)      DATA4205
 4300 FORMAT(////28H EXECUTION HALTED BECAUSE OF,I5,13H FATAL ERRORS///)DATA4210
 5100 FORMAT('1       **** TRANSIENT INTEGERS **** '// 5X,               DATA4215
     1 ' NO. OF SOURCE ELEMENTS . . . . . . . . . ',I5/ 5X,              DATA4220
     2 ' NO. OF SOURCE PROFILES  . . . . . . . . .',I5/ 5X,              DATA4225
     3 ' NO. OF DATA POINTS FOR EACH SOURCE PROF',I5/ 5X,                DATA4230
     4 ' NO. OF WELL SOURCES/SINKS NODES . . . .',I5/ 5X,                DATA4235
     5 ' NO. OF WELL SOURCE PROFILES . . . . . .',I5/ 5X,                DATA4240
     6 ' NO. OF DATA POINTS IN EACH WELL PROF. .',I5/)                   DATA4245
 5150 FORMAT(1H /5X,                                                     DATA4250
     1 40H NO. OF CAUCHY SIDES . . . .  . . . . . . .,I5/ 5X,            DATA4255
     2 40H NO. OF CAUCHY NODES . . . . . . . . . . .,I5/ 5X,             DATA4260
     3 40H NO. OF CAUCHY FLUX PROFILES . . . . . .,I5/ 5X,               DATA4265
     4 40H NO. OF DATA POINTS IN EACH CAUCHY PROF.,I5/ 5X,               DATA4270
     5 40H NO. OF NEUMANN SIDES . . . . . . . . . .,I5/ 5X,              DATA4275
     6 40H NO. OF NEUMANN NODES . . . . . . . . . .,I5/ 5X,              DATA4280
     7 40H NO. OF NEUMANN FLUXES . . . . . . . . . .,I5/ 5X,             DATA4285
     8 40H NO. OF DATA POINTS IN NEUMANN PROF. . .,I5/ 5X,               DATA4290
     9 40H NO. OF VARIABLE BOUNDARY SIDES. . . . .,I5/ 5X,               DATA4295
     A 40H NO. OF VARIABLE BOUNDARY NODES. . . . .,I5/ 5X,               DATA4300
     B 40H NO. OF RAINFALL PROFILES . . . . . . . ,I5/ 5X,               DATA4305
     C 40H NO. OF DATA POINTS IN RAINFALL PROF. . ,I5/ 5X,               DATA4310
     D 40H NO. OF DIRICHLET NODES . . . . . . . . . ,I5/ 5X,             DATA4315
     E 40H NO. OF DIRICHLET TOTAL HEAD PROF. . . ..,I5/ 5X,              DATA4320
     F 40H NO. OF DATA POINTS IN DIRICHLET PROF. .,I5/)                  DATA4325
 5300 FORMAT(1H1/5X,27H *** SOURCE INFORMATION ***)                      DATA4330
```

```
5500 FORMAT(1H0/5X,12H PROFILE NO.,I2,/ 5(4X,4HTIME,6X,6HSOURCE,2X)/   DATA4335
    > 5(4X,4H----,6X,6H------,2X))                                      DATA4340
5510 FORMAT(1H ,5(2D11.3))                                              DATA4345
5600 FORMAT(1H0//10X,65H ELEMENT NUMBER AND PROFILE TYPES OF ELEMENT   DATA4350
    1                    //5X,5(5H     I,5H MSEL,5H STYP,5X))          DATA4355
5650 FORMAT(1H ,4X,5(3I5,5X))                                           DATA4360
5700 FORMAT(1H0///5X,37H *** WELL SOURCE/SINK INFORMATION ***)          DATA4365
5710 FORMAT(1H0/5X,12H PROFILE NO.,I2/ 4(4X,4HTIME,6X,6HSOURCE,2X)/    DATA4370
    1  4(4X,4H----,6X,6H -----,2X))                                     DATA4375
5800 FORMAT(1H0//10X,65H GLOBAL NODAL NUMBER AND PROFILE TYPE OF WELLS DATA4380
    1OUCE/SINK NODES   //5X,5(5H     I,5H  NPW,5H WTYP,5X))             DATA4385
5850 FORMAT(1H ,4X,5(3I5,5X))                                           DATA4390
6000 FORMAT(1H1/5X,46H **** RAINFALL-SEEPAGE BOUNDARY CONDITIONS ***)  DATA4395
6100 FORMAT(1H0///10X,25H --- RAINFALL PROFILE ---)                     DATA4400
6150 FORMAT(1H0/5X,12H PROFILE NO.,I2,//5(4X,4HTIME,6X,6H RAINS,2X)/   DATA4405
    > 5(4X,4H----,6X,6H------,2X))                                      DATA4410
6300 FORMAT(1H0,10X,61H *** ERROR IN READING RAINFALL-SEEPAGE ELEMENT  DATA4415
    1SIDE STOP ***)                                                     DATA4420
6400 FORMAT(1H0/10X,36H --- INPUTTED VARIABLE SIDE DATA ---//5X,       DATA4425
    2 3(5H   MP,5H  GN1,5H  GN2,5H  GN3,5H  GN4,5H RTYP,5X)/5X,        DATA4430
    3 3(30H    --   ---   ---   ---   --- ----,5X))                     DATA4435
6450 FORMAT(1H ,4X,3(6I5,5X))                                           DATA4440
6500 FORMAT(1H0/10X,36H --- INPUTTED VARIABLE NODE DATA ---//1X,       DATA4445
    1 3(1X,5H     I,5H NPVB,12H     HCON    ,12H      HMIN    ,1X)/1X, DATA4450
    2 3(1X,5H     -,5H ----,12H     ----    ,12H      ----    ,1X))    DATA4455
6550 FORMAT(1H ,3(1X,2I5,2D12.4,1X))                                    DATA4460
6570 FORMAT(1H1/5X,44H CANNOT FIND A BOUNDARY SIDE COINCIDING WITH,    DATA4465
    1 I3,36H-TH VARIABLE BOUNDARY SIDE: STOP ***)                       DATA4470
6670 FORMAT(1H1/5X,44H *** CANNOT FIND A BOUNDARY NODAL NUMBER FOR,    DATA4475
    1 I3,31H-TH VARIABLE BOUNDARY NODE STOP)                            DATA4480
6700 FORMAT(1H0//10X,47H COMPUTED BOUNDARY NODAL NUMBER OF ALL VB NODESDATA4485
    1 //5X,10(5H     I,5H NPVB,2X)/5X,10(5H     -,5H ----,2X))          DATA4490
6750 FORMAT(1H ,4X,10(2I5,2X))                                          DATA4495
6751 FORMAT(1H0,5X,42H *** CAN NOT FIND A COMPRESSED RS NODE FOR,      DATA4500
    1 I2,12H-TH POINT OF,I4,20H-TH RS SIDE STOP ***)                    DATA4505
6900 FORMAT(1H0/10X,30H --- COMPUTED VB SIDE DATA ---,//1X,            DATA4510
    1 3(5H   MP,5H CNP1,5H CNP2,5H CNP3,5H CNP4,5H  MPB,5H RTYP,1X)/1X,DATA4515
    2 3(5H   --,5H ----,5H ----,5H ----,5H ----,5H ----,5H ----,1X))   DATA4520
6950 FORMAT(1H ,3(7I5,1X))                                              DATA4525
7000 FORMAT(1H1/5X,40H **** DIRICHLET BOUNDARY CONDITIONS ****)        DATA4530
7100 FORMAT(1H0/5X,12H PROFILE NO.,I2,/ 4(4X,4HTIME,6X,6H HEAD ,2X)/   DATA4535
    > 4(4X,4H----,6X,6H ---- ,2X))                                      DATA4540
7200 FORMAT(1H0//10X,65H GLOBAL NODAL NUMBER AND PROFILE TYPE OF DIRICHDATA4545
    1LET BOUNDARY NODES//5X,5(5H     I,5H NPDB,5H TYPE,5X)/             DATA4550
    1 5X,5(5H     -,5H ----,5H ----,5X))                                DATA4555
7250 FORMAT(1H ,4X,5(3I5,5X))                                           DATA4560
7670 FORMAT(1H1/5X,44H *** CANNOT FIND A BOUNDARY NODAL NUMBER FOR,    DATA4565
    1 I3,32H-TH DIRICHLET BOUNDARY NODE STOP)                           DATA4570
```

```
 7700 FORMAT(1H0//10X,66H COMPUTED BOUNDARY NODAL NUMBER & TYPE OF DIRICDATA4575
     1HLET BOUNDARY NODES//5X,5(5H    I,5H NPDB,5H TYPE,5X)/              DATA4580
     1 5X,5(5H    -,5H ----,5H ----,5X))                                  DATA4585
 7750 FORMAT(1H ,4X,5(3I5,5X))                                            DATA4590
 8000 FORMAT(1H1/5X,38H **** CAUCHY  BOUNDARY CONDITIONS ****)            DATA4595
 8100 FORMAT(1H0/5X,12H PROFILE NO.,I2/ 4(4X,4HTIME,6X,6H FLUX ,2X)/      DATA4600
     > 4(4X,4H----,6X,6H ---- ,2X))                                       DATA4605
 8300 FORMAT(1H0,10X,61H *** ERROR IN READING  CAUCHY  BOUNDARY ELEMEN    DATA4610
     1T SIDE STOP *)                                                      DATA4615
 8400 FORMAT(1H0/10X,34H --- INPUTTED CAUCHY SIDE DATA ---//5X,           DATA4620
     1 3(5H   MP,5H  GN1,5H  GN2,5H  GN3,5H  GN4,5H CTYP,5X)/5X,          DATA4625
     3 3(30H   --  ---  ---  ---  --- ----,5X))                           DATA4630
 8450 FORMAT(1H ,4X,3(6I5,5X))                                            DATA4635
 8500 FORMAT(1H0/10X,34H --- INPUTTED CAUCHY NODE DATA ---//5X,           DATA4640
     1 10(5H    I,5H NPCB,2X)/5X,10(5H    -,5H ----,2X))                  DATA4645
 8550 FORMAT(1H ,4X,10(2I5,2X))                                           DATA4650
 8570 FORMAT(1H1/5X,44H CANNOT FIND A BOUNDARY SIDE COINCIDING WITH,      DATA4655
     1 I3,36H-TH CAUCY BOUNDARY SIDE:    STOP ***)                        DATA4660
 8670 FORMAT(1H1/5X,44H *** CANNOT FIND A BOUNDARY NODAL NUMBER FOR,      DATA4665
     1 I3,31H-TH CAUCHY BOUNDARY NODE:  STOP)                             DATA4670
 8700 FORMAT(1H0/10X,34H --- COMPUTED CAUCHY NODE DATA ---//5X,           DATA4675
     1 10(5H    I,5H NPCB,2X)/5X,10(5H    -,5H ----,2X))                  DATA4680
 8750 FORMAT(1H ,4X,10(2I5,2X))                                           DATA4685
 9000 FORMAT(1H1/5X,38H **** NEUMANN BOUNDARY CONDITIONS ****)            DATA4690
 9100 FORMAT(1H0/5X,12H PROFILE NO.,I2/ 4(4X,4HTIME,6X,6H FLUX ,2X)/      DATA4695
     > 4(4X,4H----,6X,6H ---- ,2X))                                       DATA4700
 9300 FORMAT(1H0,10X,61H *** ERROR IN READING  NEUMANN BOUNDARY ELEMEN    DATA4705
     >T SIDE STOP *)                                                      DATA4710
 9400 FORMAT(1H0/10X,35H --- INPUTTED NEUMANN SIDE DATA ---//5X,          DATA4715
     1 3(5H   MP,5H  GN1,5H  GN2,5H  GN3,5H  GN4,5H NTYP,5X)/5X,          DATA4720
     2 3(30H   --  ---  ---  ---  ---  ---,5X))                           DATA4725
 9450 FORMAT(1H ,4X,3(6I5,5X))                                            DATA4730
 9500 FORMAT(1H0/10X,35H --  INPUTTED NEUMANN NODE DATA ---//5X,          DATA4735
     1 10(5H    I,5H NPNB,2X)/5X,10(5H    -,5H ----,2X))                  DATA4740
 9550 FORMAT(1H ,4X,10(2I5,2X))                                           DATA4745
 9570 FORMAT(1H1/5X,44H CANNOT FIND A BOUNDARY SIDE COINCIDING WITH,      DATA4750
     1 I3,37H-TH NEUMANN BOUNDARY SIDE:    STOP ***)                      DATA4755
 9670 FORMAT(1H1/5X,44H *** CANNOT FIND A BOUNDARY NODAL NUMBER FOR,      DATA4760
     1 I3,31H-TH NEUMANN BOUNDARY NODE: STOP)                             DATA4765
 9700 FORMAT(1H0/10X,35H --- COMPUTED NEUMANN NODE DATA ---//5X,          DATA4770
     1 10(5H    I,5H NPNB,2X)/5X,10(5H    -,5H ----,2X))                  DATA4775
 9750 FORMAT(1H ,4X,10(2I5,2X))                                           DATA4780
C                                                                         DATA4785
      END                                                                 DATA4790


      SUBROUTINE SURF(X,Y,Z,IE, DCOSB,ISB,NPBB)                           SURF 005
```

(Same as those in Appendix A)

```
      SUBROUTINE PAGEN(GNOJCN, IE)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO GENERATE POINTER ARRAYS.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: IE(NEL,9)
C
C ------- OUTPUT: GNOJCN(JBAND,NNP)
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 GNOJCN
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI,MXNDTC
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI,NDTCHG
      COMMON /CINTE/ NCYL,NITER,KSTR,KPRO,KDSKO,KSS,NPITER,IGEOM
C
      DIMENSION GNOJCN(JBAND,MAXNP),IE(MAXEL,9)
      DIMENSION IEM(8)
C
C ******* GENERATE GNOJCN(JBAND,MAXNP) BASED ON IE(MAXEL,9)
C
      DO 490 NP=1,NNP
C
      DO 110 I=1,JBAND
  110 GNOJCN(I,NP)=0
C
      NLNOD=0
      KOUNT=0
C
      DO 390 M=1,NEL
C
      DO 160 IQ=1,8
  160 IEM(IQ)=IE(M,IQ)
C
      DO 170 IQ=1,8
      NI=IEM(IQ)
      IF(NI.NE.NP) GO TO 170
      GO TO 220
  170 CONTINUE
      GO TO 390
C
  220 DO 290 IQ=1,8
      NI=IE(M,IQ)
```

```
C
C ------- COMPRESS THE NODES TO 1 TO JBAND
C
      NLNOD=NLNOD+1
      IF(NLNOD.GT.1) GO TO 230
C
C ------- THE FIRST NODE IS ENCOUNTERED
C
      KOUNT=KOUNT+1
      GNOJCN(KOUNT,NP)=NI
      GO TO 290
C
C ------- IF NLNOD IS GREATER THAN 1, WE HAVE TO CHECK IF NI IS THE
C ------- NODE ALREADY COMPRESSED? IF YES, SKIP. IF NOT INCREASE THE
C ------- KOUNT.
C
  230 DO 240 J=1,KOUNT
      NJ=GNOJCN(J,NP)
      IF(NI.EQ.NJ) GO TO 290
  240 CONTINUE
C
C ------- THE NODE NI HAS NOT BEEN COMPRESSED YET.  HENCE COMPRESS IT.
C
      KOUNT=KOUNT+1
      GNOJCN(KOUNT,NP)=NI
C
  290 CONTINUE
C
  390 CONTINUE
C
      IF(KOUNT.LE.JBAND) GO TO 490
      KONT=KOUNT-1
      JBND=JBAND-1
      WRITE(6,1000) NP,KONT,JBND
      STOP
C
  490 CONTINUE
C
C ******* PRINT GENERATED ARRAY GNOJCN
C
      IF(MOD(IGEOM,2).EQ.0) GO TO 595
      LINE=0
      DO 590 NP=1,NNP
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 5000
      PRINT 5100, NP,(GNOJCN(I,NP),I=1,JBAND)
  590 CONTINUE
  595 CONTINUE
```

```
C                                                                      PAGE 485
 1000 FORMAT(1H0//5X,' ***',I4,'-TH NODE HAS ',I4,' NODES SURROUNDING', PAGE 490
     1 ' IT, WHICH IS MORE THAN JBAND - 1 -',I5,'  STOP ***')          PAGE 495
 5000 FORMAT(1H1/5X,' ** GENERATED SURROUNDING NODES OF ALL NODES *'// PAGE 500
     1 5X,'  NP   1   2   3   4   5   6   7   8',                      PAGE 505
     2 '    9  10  11  12  13  14  15  16  17',                        PAGE 510
     3 '  18  19  20  21  22  23  24  25  26  27',                     PAGE 515
     4 /5X,28('  --'))                                                 PAGE 520
 5100 FORMAT(1H ,4X,28I4)                                              PAGE 525
C                                                                      PAGE 530
      RETURN                                                           PAGE 535
      END                                                              PAGE 540
```

```
      SUBROUTINE VELT(VX,VY,VZ, CMATRX, X,Y,Z,IE,H,HT,AKR, PROP)        VELT 005
```

(Same as those in Appendix A)

```
      SUBROUTINE Q8DV(QB,QRX,QRY,QRZ, XQ,YQ,ZQ,HTQ,AKXG,AKYG,AKZG,      Q8DV 005
     1 AKXYG,AKXZG,AKYZG)                                               Q8DV 010
```

(Same as those in Appendix A)

```
      SUBROUTINE SPROP(TH,DTH,AKR, IE,H,THPROP,AKPROP)                  SPRO 005
```

(Same as those in Appendix A)

```
      SUBROUTINE BCPREP(IE,X,Y,Z,  H,VX,VY,VZ, DCOSB,ISB, NPVB,ISV,     BCPR 005
     > DCYFLX,FLX,HCON,HMIN,NPFLX,NPCON,NPMIN, IRTYP,RFALL, NCHG)       BCPR 010
```

(Same as those in Appendix A)

```
      SUBROUTINE ASEMBL(X,Y,Z,IE, CMATRX,RLD,LRN, HW,HP,DTH,AKR,PROP,   ASEM 005
     > SOS,MSEL,ISTYP,WSS,NPW,IWTYP, KSS,W,DELT)                        ASEM 010
```

(Same as those in Appendix A)

```
      SUBROUTINE Q8(QA,QB,RQ, DTHG,AKXG,AKYG,AKZG,AKXYG,AKXZG,AKYZG,    Q8   005
     > XQ,YQ,ZQ,SOSM,AGRAV)                                             Q8   010
```

(Same as those in Appendix A)

```
      SUBROUTINE BASE(N,DNX,DNY,DNZ,DJAC,XQ,YQ,ZQ,SS,TT,UU)             BASE 005
```

(Same as those in Appendix A)

```
      SUBROUTINE BC(CMATRX,RLD,LRN, IE,X,Y,Z, AKR,PROP, DCOSB,ISB,NPBB, BC   005
     1 QCB,ISC,ICTYP, QNB,ISN,INTYP, FLX,HCON,HMIN,NPFLX,NPCON,NPMIN,   BC   010
     2 HDB,IDTYP,NPDB, KSS)                                             BC   015
```

(Same as those in Appendix A)

```
      SUBROUTINE Q4S(R1Q,R2Q,XQ,YQ,ZQ,F1Q,F2Q)                          Q4S  005
```

(Same as those in Appendix A)

```
      SUBROUTINE PISS(R,RI,RL,C,LRN,OMI,EPS,NNP,NITER.MAXNP,JBAND,IBUG,
     1 KTIM)
C
C*******1*********2*********3*********4*********5*********6*********7**
C
C ------- TO SOLVE A MATRIX EQUATION WITH POINTWISE ITERATION SOLUTION
C ------- STRATEGIES.
C
C*******1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: RI(NNP), RL(NNP), C(NNP,JBAND), LRN(JBAND,NNP),
C -------               OMI, EPS, NNP, NITER, MAXNP, JBAND, IBUG, AND KTIM.
C
C ------- OUTPUT: R(NNP) - FINAL SOLUTION; AND RI(NNP)-ITERATE.
C
C*******1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION R(MAXNP),RI(MAXNP)
      DIMENSION RL(MAXNP),C(MAXNP,JBAND),LRN(JBAND,MAXNP)
C
C ------- PRINT ITERATION INFORMATION IF DESIRED.
C
      IF(IBUG.NE.0 .AND. KTIM.NE.0) PRINT 1000
C
      DO 290 IT=1,NITER
C
C ------- FOR EACH ITERATION, PUT THE LOAD VECTOR INTO R(MAXNP).
C
      DO 210 NP=1,NNP
  210 R(NP)=RL(NP)
C
C ------- THE MATRIX C = L + I + U, WHERE L IS THE LOWER TRIANGULAR
C ------- MATRIX AND U IS THE UPPER TRIANGULAR MATRIX AND I IS THE
C ------- DIAGONAL MATRIX.
C ------- NOW ADD U.RI TO THE RIGHT HAND SIDE, WHERE RI IS THE PREVIOUS
C ------- ITERATE.
C
      DO 230 NP=1,NNP
      DO 220 I=1,JBAND
      LNODE=LRN(I,NP)
      IF(LNODE.EQ.0) GO TO 220
      IF(LNODE.LE.NP) GO TO 220
      R(NP)=R(NP)-C(NP,I)*RI(LNODE)
  220 CONTINUE
  230 CONTINUE
```

```
C
C ------- START TO COMPUTE NEW ITERATE WITH POINTWISE ITERATIONS.
C
      DO 250 NP-1,NNP
      DO 240 I-1,JBAND
      LNODE-LRN(I,NP)
      IF(LNODE.EQ.0) GO TO 240
      IF(LNODE.GE.NP) GO TO 240
      R(NP)-R(NP)-C(NP,I)*R(LNODE)
  240 CONTINUE
      DO 245 I-1,JBAND
      LNODE-LRN(I,NP)
      IF(LNODE.NE.NP) GO TO 245
      R(NP)-OMI*(R(NP)/C(NP,I)) + (1.0D0-OMI)*RI(NP)
  245 CONTINUE
  250 CONTINUE
C
C ------- CHECK IF A CONVERGENT SOLUTION IS OBTAINED?
C
      NNCVN-0
      RELERR--1.0D0
      ABSERR--1.0D0
      DO 260 NP-1,NNP
      ABSDIF-DABS(R(NP)-RI(NP))
      ABSERR-DMAX1(ABSERR,ABSDIF)
      IF(RI(NP).NE.0.0D0) RELERR-DMAX1(RELERR,DABS(ABSDIF/RI(NP)))
      IF(ABSERR .LE. EPS) GO TO 260
      NNCVN-NNCVN+1
  260 CONTINUE
C
C ------- UPDATE THE ITERATE.
C
      DO 280 NP-1,NNP
  280 RI(NP)-R(NP)
C
C -------  PRINT ITERATION INFORMATION IF DESIRED.
C
      IF(IBUG.NE.0 .AND. KTIM.NE.0) PRINT 1100, IT,ABSERR,RELERR,NNCVN
C
      IF(IT.EQ.1) GO TO 290
      IF(ABSERR .LE. EPS) GO TO 990
C
  290 CONTINUE
C
      PRINT 2000, IT,NITER,ABSERR,RELERR,NNCVN
C
  990 CONTINUE
```

```
C
 1000 FORMAT(1H0/75X,5H   IT,12H   ABS ERR   ,12H   REL ERR   ,
     1 13H NO. OF NODES/75X,5H   --,12H   --- ---   ,12H   --- ---   ,
     2 13H --- -- -----)
 1100 FORMAT(75X,I5,2D12.4,5X,I5)
 2000 FORMAT(1H0/70X,43H ::: WARNING: NO CONVERGENCE IN PISS AFTER ,
     1 I4,11H ITERATIONS/75X,9H NPITER =,I4,11H   ABSERR =,D12.4/75X,
     2 11H   RELERR =,D12.4,11H   NNCVN =,I4)
C
      RETURN
      END
```

```
      SUBROUTINE SFLOW(X,Y,Z,IE, H,HP,VX,VY,VZ,TH,DTH,
     1 BFLX,BFLXP, DCOSB,ISB,NPBB, MSEL,SOS,ISTYP,NPW,WSS,IWTYP,
     2 NPVB,NPDB,NPCB,NPNB,   DELT, KFLOW)
```

(Same as those in Appendix A)

```
      SUBROUTINE Q8TH(QTHM,QSOSM,DHQ,THG,XQ,YQ,ZQ,SOURCE)
```

(Same as those in Appendix A)

```
      SUBROUTINE PRINTT(VX,VY,VZ,H,HT,TH, NPBB,BFLX, NPVB,DCYFLX,NPCON,
     1 NPFLX,NPMIN, SUBHD, TIME,DELT, KPR,KOUT,KDIAG,ITIM)
```

(Same as those in Appendix A)

```
      SUBROUTINE STORE(X,Y,Z,IE,H,HT,TH,VX,VY,VZ, DCOSB,ISB,NPBB,
     1 TITLE, TIME, NPROB)
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- TO STORE PERTINENT QUANTITIES ON AUXILIARY DEVICES FOR FUTURE
C ------- USES, E. G., FOR PLOTTING.  WHAT DEVICE IS TO BE USED MUST BE
C ------- SPECIFIED IN THE JCL.
C
C********1*********2*********3*********4*********5*********6*********7**
C
C ------- INPUT: X(NNP), Y(NNP), Z(NNP), IE(NEL,9), H(NNP), HT(NNP),
C -------        TH(8,NEL), VX(NNP), VY(NNP), VZ(NNP), DCOSB(3,NBES),
C -------        ISB(6,NBES), NPBB(NBNP), TITLE, TIME, NPROB.
C
C ------- OUTPUT: STORE ALL INPUTS IN LOGICAL UNIT 1.
C
C********1*********2*********3*********4*********5*********6*********7**
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBES,MAXBNP,JBAND,MAXNTI
      COMMON /CGEOM/ NNP,NEL,NBNP,NBES,KGRAV,NTI
C
      DIMENSION TITLE(9)
      DIMENSION X(MAXNP),Y(MAXNP),Z(MAXNP),IE(MAXEL,9)
      DIMENSION H(MAXNP),HT(MAXNP),VX(MAXNP),VY(MAXNP),VZ(MAXNP)
      DIMENSION TH(8,MAXEL)
      DIMENSION DCOSB(3,MAXBES),ISB(6,MAXBES),NPBB(MAXBNP)
C
      DATA NPPROB/-1/
C
      IF (NPPROB.EQ.(-1)) REWIND 1
      IF (NPPROB.EQ.NPROB) GO TO 110
      WRITE(1) (TITLE(I),I-1,9),NPROB,NNP,NEL,NBNP,NBES,NTI
      WRITE(1) (X(N),N-1,NNP),(Y(N),N-1,NNP),(Z(N),N-1,NNP),
     1 ((IE(M,I),M-1,NEL),I-1,9),((DCOSB(I,M),I-1,3),M-1,NBES),
     2 (NPBB(N),N-1,NBNP)
      NPPROB-NPROB
C
  110 WRITE(1) TIME,(H(N),N-1,NNP),(HT(N),N-1,NNP),((TH(I,M),I-1,8),M-1,
     1 NEL),(VX(N),N-1,NNP),(VY(N),N-1,NNP),(VZ(N),N-1,NNP)
      RETURN
      END
```

```
      SUBROUTINE INTERP(FALL,TRF,RF,TIME,MXPR,MXDP,NPR,NDP)             INTE 005
          (Same as those in Appendix A)


      SUBROUTINE READR(F,MAXNOD,NNP)                                    REDR 005
          (Same as those in Appendix A)


      SUBROUTINE READN(INDTYP,MXTYP,NTYPE)                              REDN 005
          (Same as those in Appendix A)
```

INTERNAL DISTRIBUTION

1. T. L. Ashwood
2. S. I. Auerbach
3. W. J. Boegly, Jr.
4. R. O. Chester
5. R. B. Clapp
6. N. H. Cutshall
7. E. C. Davis
8. R. B. Dreier
9. T. O. Early
10. C. W. Francis
11. C. S. Haase
12. S. G. Hildebrand
13. D. D. Huff
14. D. C. Kocher
15. D. W. Lee
16. S. Y. Lee
17. F. R. Mynatt
18-27. T. E. Myrick
28. P. T. Owen
29. W. W. Pitt
30-31. D. E. Reichle
32. C. R. Richmond
33. T. H. Row
34. F. E. Sharples
35. D. K. Solomon
36. S. H. Stow
37. L. E. Stratton
38. J. Switek
39. J. R. Trabalka
40. S. D. Van Hoesen
41. L. D. Voorhees
42-46. G. T. Yeh
47. Central Research Library
48-62. ESD Library
63-64. Laboratory Records Dept.
65. Laboratory Records, ORNL-RC
66. ORNL Patent Section
67. ORNL Y-12 Technical Library

EXTERNAL DISTRIBUTION

68. V. Dean Adams, Tennessee Technological University, Cookeville, TN 38501
69. M. U. Ahmad, Department of Geology, Ohio University, Athens, OH 45701
70. L. R. Ahuja, Water Quality and Watershed Research Laboratory, P.O. Box 1430, Agricultural Research Services, Southern Plains Area, USDA, Durant, OK 74702-1430
71. F. R. Baddour, Eastern Research Group, Inc., 6 Whittemore Street, Arlington, MA 02174
72. Andrew Barry, Virginia Polytechnic Institute and State University, Department of Agronomy, Blacksburg, VA 24061
73. Paul Beck, Ministry of the Environment, 135 St. Clair Ave. West, Toronto Ontario CANADA M4V 1P5
74. D. Bedcock, Gannett Fleming Engineers, P.O. Box 1963, Harrisburg, PA 17105
75. Beniwal Bowser and Morner, P.O. Box 51, Dayton, Ohio 45401
76. T. F. Bergen, Computer Application Engineers, Inc., P.O. Box 408, Exton, PA 19341
77. M. P. Bergeron, USGS, Water Resources Division, 521 West Seneca Street, Ithaca, NY 14850
78. R. Bibby, Atomic Energy Establishment, Winfrigh, Dorchester Dorset DT2 8DH

79. Julian Blonley, Middle South Services, P.O. Box 6100,
New Orleans, LA 70161
80. Efthimios Bouloutas, MIT, Dept. of Civil Engineering,
Cambridge, MA 02139
81. Luis Bravo, Radian Corporation, 3200 E. Chapel Hill Rd.,
Research Triangle Park, NC 27709
82. Eshel Bresler, Soil Physicist, Agricultural Research
Organization, The Volcanic Center, Institute of Soil and Water,
P.O. Box 6, Bet Dagan, 50-250, Israel
83. John Broadway, Eastern Environmental Radiation Facility,
P.O. Box 3009, Montgomery, AL 36193
84. S. R. Brown, Department of Hydrology, University of Arizona,
Tucson, AZ 85721
85. T. W. Broyd, Atkins Research and Development, Woodcote Grove
Ashley Rd., Epsom Surrey KT18 5BW England
86. W. Brutsaert, School of Civil Engineering, Cornell University,
Ithaca, NY 14850
87. A. Burges, Golder Associates, 4104 148th Avenue, N.E., Redmond,
WA 98052
88. Christopher B. Burke, Harza Engineering Company, 150 South
Wacker Drive, Chicago, IL 60606
89. J. Thomas Callahan, Associate Director, Ecosystem Studies
Program, Room 336, 1800 G Street, NW, National Science
Foundation, Washington, DC 20550
90. G. N. Camp, Jr., 810 Smiley Ave., Cincinnati, OH 45240
91. D. B. Chambers, Executive Vice President, SENES Consultants
Limited, 499 McNicoll Ave., Willowdale, Ontario, CANADA M2H 2C9
92. Jui-Lin Chang, College of Mineral and Energy Resources,
West Virginia University, Morgantown, WV 26506-6070
93. Shoou-Yuh Chang, Dept. of Civil Engineering, University of
Missouri-Rolla, Rolla, MO 65401
94. S. T. Chau, Alberta Environmental Protection Service,
14th Floor, Standard Life Center, 10405 Jasper Ave., Edmonton,
Alberta, Canada
95. T. C. Chee, R&D and Byproducts Division, DP-123 (GTN),
U.S. Department of Energy, Washington, DC 20545
96. Yee-Ping P. Chia, Argonne National Laboratory, 9700 South Cass
Avenue, Argonne, IL 60439
97. James Chieh, Ecology and Environment, Inc., 195 Sugg Road,
P.O. Box D, Buffalo, NY 14225
98. S. W. Childs, Department of Soil Science, Oregon State
University, Corvallis, OR 97331
99. A. T. Clark, Jr., Advanced Fuel and Spent Fuel Licensing
Branch, Division of Fuel Cycling and Material Safety, 396-SS,
U.S. Nuclear Regulatory Commission, 7915 Eastern Avenue,
Silver Spring, MD 20910
100. Royce Cline, North Dakota State Water Commission, 900 East
Blvd., Bismarck, ND 58505
101. C. C. Clyde, Utah Water Resources Laboratory, Utah State
University, Logan, UT 84322

102. C. R. Cole, Pacific Northwest Laboratory, P.O. Box 999, Richland, WA 99352
103. M. A. Collins, Illinois State Water Survey, 2204 Griffith Drive, Champaign, IL 61820
104. E. F. Conti, Office of Nuclear Regulatory Research, Nuclear Regulatory Commission, MS-1130-SS, Washington, DC 20555
105. D. Cooper, School of Public Health, Harvard University, 665 Huntington Ave., Boston, MA 02115
106. Ashok S. Damle, Research Triangle Institute, P.O. Box 12194, Research Triangle Park, NC 27709
107. S. R. Davidson, SCS Engineers, 11260 Roger Bacon Drive, Reston, VA 22090
108. D. G. deMarsily, Ecole Nationale, Superieure des Mines, Centre d'Informatique Geologigue, 35 Rue Saint-Honore, 77305 Fontainebleau, France
109. A. K. Deb, Roy F. Weston, Inc., Weston Way, West Center, PA 19380
110. D. B. Dunbar, Woodward-Clyde Consultants, Three Embarcadero Center, Suite 700, San Francisco, CA 94111
111. Amos Eddy, Director, OCS, Oklahoma Climatological Survey, 710 Asp, Suite 8, Norman, OK 73019
112. B. E. Ehleringer, Geosciences, Inc., San Diego, CA 92123
113. Aly El-Kadi, International Groundwater Modeling Center, Holcomb Research Institute, Butler University, 4600 Sunset Ave., Indianapolis, IN 46208
114. G. L. Evavold, RERRM, Inc. 408 Board of Trade Bldg., 301 West First Street, Duluth, MN 55802
115. Greg Faber, DPRA Development Planning and Research Associates, Inc., 200 Research Drive, P.O. Box 727, Manhattan, KS 66502
116. Pat Fennessy, Golder Associates, 4104 148th Avenue, N.E., Redmond, WA 98052
117. G. J. Foley, Office of Environmental Process and Effects Research, U.S. Environmental Protection Agency, 401 M Street, SW, RD-682, Washington, DC 20460
118. W. A. Frankhouser, Director, Operations and Traffic Division, DP-122 (GTN), U.S. Department of Energy, Washington, DC 20545
119. R. A. Freeze, Dept. of Geological Sciences, University of British Columbia, Vancouver, B.C. Canada
120. Jesus Lopez Garcia, Division of Recursos Hidraulicos, INTECSA, Francisco Gervas, 6, 28020 Madrid, Spain
121. W. J. Gburek, U.S. Department of Agriculture, Northwest Watershed Research Center, 110 Research Building A, University Park, PA 16802
122. Ferruccio Gera, ISMES SPA, sede de Roma, via dei Crociferi 44-00187 ROMA
123. C. G. Glyde, Utah Water Resources Laboratory, Utah State University, Logan, UT 84222
124. C. R. Goldman, Professor of Limnology, Director of Tahoe Research Group, Division of Environmental Studies, University of California, Davis, CA 94616

125. Dan Goode, Environmental Engineering Dept., GCA Corporation,
213 Burlington Road, Bedford, MA 01730
126. Stephen Gorelick, USGS, Water Resources Division, MS-21,
345 Middlefield Road, Menlo Park, CA 94205
127. Jennifer B. Gorrell, Water and Earth Science Associates, LTD,
Box 430, CARP, Ontario KOA 1L0
128. A. Berge Gureghian, Project Management Division, Performance
Analysis Department, Battelle Columbus, 505 King Avenue,
Columbus, OH 43201
129. D. A. Hamilton, Waste Management Division, Dept. of Natural
Resources, Stevens T. Mason Building, P.O. Box 30028, Lansing,
MI 48909
130. J. M. Hamlett, Dept. of Agriculture Engineering, Iowa State
University, Ames, IA 50011
131. Keith Hansen, Law Environmental Services, 2749 Delk Road,
Marietta, GA 30067
132. J. E. Hawley, Chen and Associates, 96 South Zuni Street,
Denver, CO 80223
133. M. E. Herman, USGS, Water Resources Division, Rm. 227, Federal
Bldg., 705 North Plaza Street, Carson City, NV 89701
134. L. A. Hermann, Dept. of Geosciences, Louisiana Tech University,
Ruston, LA 71272
135. Beverly Herzog, Illinois State Geological Survey, Natural
Resources Bldg., 615 E. Peabody Drive, Champaign, IL 61820
136. Francois E. Heuze, Lawrence Livermore National Laboratory,
Lawrence Livermore, CA 94550
137. Richard Heystee, Ontario Hydro, 700 University Avenue, Toronto,
Ontario M5G 1X6 Canada
138. Jimmy Ho, NUS Corporation, Park West Two, Cliff Mine Road,
Pittsburgh, PA 15275
139. J. A. Hoopes, 1261B Engineering Bldg., Department of Civil and
Environmental Engineering, University of Wisconsin-Madison,
Madison, WI 53706
140. J. W. Hopmans, Dept of Hydraulics and Catchment Hydrology,
Agricultural University, Wagenigen, The Netherlands
141. J. Horyna, Department of Radiation Protection, Nuclear Research
Institute 250-68 Rez., Prague, Czechoslovakia
142. Sharon M. Hotz, Harding Lawson Associates, 7655 Redwood Blvd.,
Novato, CA 94948
143. Tai-Dan Hso, WRC Engineering, Inc., 1660 South Albion Street,
Suite 500, Denver, CO 80222
144. J. W. Huckabee, Manager, Ecological Studies Program, Electric
Power Research Institute, 3412 Hillview Avenue, P.O. Box 10412,
Palo Alto, CA 94303
145. Paul Hwang, ES Engineering Science, Two Flint Hill,
10521 Rosehaven Street, Fairfax, VA 22030-2899
146. Judith Jeffcoate, Scicon Computer Services Limited, Brick Close
Kiln Farm, Milton Keynes, Great Britain MK1 3EJ
147. Cady Johnson, Mifflin & Associates, Inc., 2700 Sunset,
Suite B-13, Las Vegas, NV 89120

148. R. Johnson, Department of Environmental Sciences, The Oregon Graduate Center, 19600 NW Walker Road, Beaverton, OR 97006
149. A. Jonasson, Department of Geology, Chalmers University of Technology, S-412, 96 Gothenburg, Sweden
150. David B. Jones, O'brien & Gere, Box 4873, 1304 Buckley Road, Syracuse, NY 13231
151. George Y. Jordy, Director, Office of Program Analysis, Office of Energy Research, ER-30, G-226, U.S. Department of Energy, Washington, DC 20545
152. Kyla S. Juvonen, Woodward-Clyde Consultants, Stanford Place 3, Suite 100, 4582 South Ulster Street Parkway, Denver, CO 80237
153. J. L. Kauschinger, Dept. of Civil Engineering, Tuft University, Medford, MA 02155
154. D. Kent, Whitehurst Hall, Oklahoma University, Stillwater, OK 74078
155. Raz Khaleel, New Mexico Tech, Dept. of Geosciences, Socorro, NM 87801
156. Anne Klock, Earth Technology Corporation, 3777 Long Beach Blvd., Long Beach, CA 90807
157. W. J. Kutz, HARZA Engineering Company, 150 South Wacker Drive, Chicago, IL 60606
158. David Langseth, Metcalf & Eddy, Inc., 50 Stanford Street, Boston, MA 02114
159. T. H. Larson, State Geological Survey Division, Illinois Department of Energy and Natural Resources, Natural Resources Bldg., 615 East Peabody Drive, Champaign, IL 61820
160. R. T. Lawson, Environmental Sciences Division, Australian Atomic Energy Commission, New Illanarra Rd., Lucas Heights, N.S.W., Australia
161. Denis R. LeBlanc, USGS, Water Resources Division, 150 Causeway Street, Suite 1001, Boston, MA 02114
162. D. B. Leclaire, Director, Office of Defense Waste and Transportation Management, DP-12 (GTN), U.S. Department of Energy, Washington, DC 20545
163. T. C. Lederman, Radiological Sciences Institute, State of New York, Dept. of Health, The Governor Nelson A. Rockefeller Empire State Plaza, Albany, NY 12201
164. Tien-chang Lee, University of California-Riverside, Dept. of Earth Sciences, Riverside, CA 92521
165. Joan Lefler, Environmental Review Division, Office of Testing and Evaluation Office of Toxic Substances, Environmental Protection Agency, 401 M Street, SW, Washington, DC 20460
166. J. F. Lentz, Applied Physics Laboratory, The Johns Hopkins University, Johns Hopkins Rd., Laurel, MD 20810
167. C. C. Leser, Dept. of Civil Engineering, University of Notre Dame, Notre Dame, IN 46556
168. J. T. Lin, United Industries Corp, Executive Plaza, Suite 211, 12835 Bell-Red Road, Bellevue, WA 98005
169. Stanley E. Logan, S. E. Logan and Associates, Inc., Santa Fe, NM 87501

170. G. V. Loganathan, Virginia Polytechnic Institute and State University, Dept. of Civil Engineering, Blacksburg, VA 24061
171. Y. C. Lu, Dept. of Civil Engineering, National Taiwan University, Taipei, Taiwan
172. M. A. Luckett, GURR & Associates, Inc., 2000 East Edgewood Drive, Suite 1, Lakeland, FL 33803
173. P. R. Manam, Sargent & Lundy Engineers, 55 East Monroe Street, Chicago, IL 60603
174. C. J. Mankin, Director, Oklahoma Geological Survey, The University of Oklahoma, 830 Van Vleet Oval, Room 163, Norman, OK 73019
175. A. Marples, Safety and Medical Services, British Nuclear Fuels Limited, Sellafield Seascale, Cumbria CA20 1PG, England
176. M. L. Maslia, USGS, 6481 Peachtree Industrial Blvd., Doraville, CA 30360
177. Helen McCammon, Director, Ecological Research Division, Office of Health and Environmental Research, Office of Energy Research, MS-E201, ER-75, Room E-233, Department of Energy, Washington, DC 20545
178. Barry McGuire, University of Pittsburgh, Pittsburgh, PA 15005
179. D. C. McKinney, Department of Environmental Resources Engineering, Hornbondt State University, Cacata, CA 95521
180. C. F. McLane III, Dept. of Environmental Sciences, University of Virginia, Charlottesville, VA 22903
181. G. L. Meyer, Environmental Protection Agency, 401 M Street SW, MS-ANR459, Washington, DC 20460
182. W. B. Miller, Sandia National Laboratory, P.O. Box 5800, Albuquerque, NM 87185
183. M. T. Mills, CorStar, 7315 Wisconsin, North Tower #702, Bethesda, MD 20814
184. Jan Mockler, Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439
185. David E. Mohr, Jacobs Engineering Group, Inc., 5301 Central Avenue, Suite 1700, Albuquerque, NM 87108
186. Fred H. Molz, Director, School of Engineering and Engineering Experiment Station, Auburn University, Auburn, AL 36849
187. H. A. Mooney, Department of Biological Sciences, Stanford University, Stanford, CA 94305
188. G. E. Moore, System Applications Inc., 101 Lucas Valley Road, San Rafael, CA 94903
189. Sharon M. Motz, Harding Lawson Associates, 7655 Redwood Blvd., Novato, CA 94948
190. Fabio Musmeci, ENEA CASACCIA, PAS/ISP/INGSITI, Via Anguillarese 301, 006 Roma Italy
191. J. D. Nasson, NEBG Installation Representative, Mail Code 311, General Electric Company, 175 Curtner Ave, San Jose, CA 95125
192. Wan Mokhtar Nawang, Universiti Tecknologi Malaysia, Kuala Lumpur, Malaysia
193. R. Obrist, Eidg. Techn. Hochschule, Institut fur Hydromechanik and Wasserwirtschaft, CH-8093 Zurich-Honggerberg, Switzerland

194. Edward O'Donnell, Division of Radiation Programs and Earth Sciences, U.S. Nuclear Regulatory Commission, Mail Stop 1130 SS, Washington, DC 20555
195. P. M. Ornstein, MS-697, U.S. Nuclear Regulatory Commission, 7915 Eastern Ave., Silver Spring, MD 20910
196. C. A. Oster, Energy Systems Department, Pacific Northwest Laboratories, Richland, WA 99352
197. Frank Parker, Dept. of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN 37235
198. J. C. Parker, Department of Agronomy, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061
199. D. V. Patel, Southern Company Services, Birmingham, AL 35202
200. Jack Paul, PEDCO, Inc., 1006 N. Bowen Road, Arlington, TX 76012
201. G. F. Pinder, Dept. of Civil and Environmental Engineering, Princeton University, Princeton, NJ 08544
202. James Pittman, State of Maryland, Environmental Health Administration, 201 West Preston Street, Baltimore, MD 21201
203. Wilfred Pon, Environsphere Company, Two World Trade Center, New York, NY 10048
204. Susan Power, ES Engineering-Science, 290 Elwood Davis Road, Liverpool, NY 13088
205. David Prudic, USGS, Water Resources Division, Room 229, Federal Building, Carson City, NV 89701
206. Ellen J. Quinn, Northeast Utilities, P.O. Box 270, Hartford, CT 06141
207. P.A.C. Raats, Institute for Soil Fertility, P.O. Box 30003, 9750 Ra Haren (Gr), Netherlands
208. M. S. Rahman, North Carolina State University, Dept. of Civil Engineering, Raleigh, NC 27695
209. Vishna Ranganathan, Dept. of Geology, Louisiana State University, Baton Rouge, LA 70803
210. Gregory Reed, Department of Civil Engineering, The University of Tennessee, Knoxville, TN 37916
211. Ed Repa, JRB Associates, 8400 West Park Drive, McLean, VA 22102
212. Antonio Marozzi Righetto, Universidade de Sao Paulo, Depto. de Hidraulica e Saneamento, Escola de Engenharia de S. Carlos, 13.560 - Sao Carlos, SP, Brazil
213. James W. Roberts, 1824 Arcadia Drive, Lafayette, IN 47905
214. John C. Rodgers, Group LS-6, Environmental Sciences Group, Los Alamos National Laboratory, Los Alamos, NM 87545
215. William L. Rohrer, Pope-Reid Associates, Inc., 24 E. 6th Street, Suite 813, St. Paul, MN 55101
216. E. M. Romney, University of California, Los Angeles, 900 Veteran Avenue, Los Angeles, CA 90024
217. Wyn Ross, Desert Research Institute, University of Nevada, Reno, NV 89506
218. Philip Rosten, Jr., ERM-Southeast, Inc., P.O. Box 881, Brentwood, TN 37027
219. Dauglas Rumbaugh, Woodward-Clyde Consultants, 5120 Butler Pike, Plymouth Meeting, PA 19462

220. Lisa A. Sabol, IT Corporation, 10 Duff Road, Pittsburgh, PA 15235
221. Z. Saleem, Ebasco Services, Inc., 2211 West Meadowview Road, Greensboro, NC 27407
222. Karen A. Savinda, D'Appolonia, 10 Duff Road, Pittsburgh, PA 15235
223. J. L. Schnoor, Division of Energy Engineering, The University of Iowa, Iowa City, IA 52242
224. John E. Sevee, Sevee & Maher Engineers, P.A., 668 Main Street, Westbrook, ME 04092
225. Thomas R. Schultz, Wahler Associates, 12477 W. Cedar Drive, Suite 206, Lakewood, CO 80228
226. Joanna I. Scott, TASC-The Analytical Sciences Corporation, One Jacob Way, Reading, MA 01867
227. Maurice Sell, 1980 Philips Drive, Northglenn, CO 80233
228. Jacob Shapiro, Radiation Protection Officer, University Health Services, Harvard University, Cambridge, MA 02138
229. Derek Snowball, SCICON, Scicon Limited, Brick Close, Kiln Farm, Milton Keynes MK11 3EJ England
230. Marios Sophocleous, Kansas Geological Survey, 1930 Avenue A, Campus West, The University of Kansas, Lawrence, KS 66044
231. Eveerett P. Spring, NW Watershed Research Center, USDA, Suite 116, 1175 South Orchard, Boise, ID 83705
232. Jack Sosebee, Environmental Sciences and Engineering, Inc., 7332 S. Alton Way, Suite H-I, Englewood, CO 80110
233. R. H. Stacks, Department of Pollution Control and Ecology, State of Arkansas 8001 National Drive, P.O. Box 9583, Little Rock, AR 72209
234. R. J. Starmer, HLW Technical Development Branch, Office of Nuclear Material Safety and Safeguards, Nuclear Regulatory Commission, Room 427-SS, Washington, DC 20555
235. F. Stauffer, Institute fur Hydromechanik and Wasserwirtschaft, ETH-Honggerberg, CH-8093 Zurich, Switzerland
236. J. G. Steger, Environmental Sciences Group, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545
237. D. B. Stephens, New Mexico Tech, Department of Geosciences, Socorro, NM 87801
238. R. J. Stern, Director, Office of Environmental Compliance, MS PE-25, FORRESTAL, U.S. Department of Energy, 1000 Independence Avenue, SW, Washington, DC 20585
239. M. T. Stewart, University of South Florida, Tampa, FL 33620
240. C. John Suen, Brookhaven National Laboratory, Nuclear Waste and Materials Technology Division, Upton, Long Island, NY 11973
241. Angelo Tagliacozzo, Gibbs & Hill, Inc., 11 Penn Plaza, New York, NY 10001
242. Robert J. Telford, ACRES Consulting Limited, 5259 Dorchester Road, P.O. Box 1001, Niagara Falls, Ontarion L2E 6W1 CANADA
243. Khanh T. Tran, AMI Applied Modeling Inc., Warner Financial Plaza, 6355 Topanga Canyon Blvd., Suite 433, Woodland Hills, CA 91367

244. Howard Trussell, 1261 Engineering Bldg., University of Wisconsin-Madison, Madison, WI 53706
245. Steve Y. Tsai, Division of Environmental Impact Studies, Argonne National Laboratory, 9700 South Case Avenue, Argonne, IL 60439
246. S. B. Upchurch, University of South Florida, Tampa, FL 33620
247. M. van Genuchten, U.S. Salinity Laboratory, 4500 Glenwood Drive, Riverside, CA 92501
248. John J. Vasconcelos, James M. Montgomery Engineers, 250 North Madison Avenue, Pasadena, CA 91101
249. Monique Villars, EWA, Environment and Water Resources Management, 12800 Industrial Park Blvd., Minneapolis, MN 55441
250. Jan Wagner, School of Chemical Engineering, Oklahoma State University, Stillwater, OK 74078
251. Ken Walker, Department of Geology, The University of Tennessee, Knoxville, TN 37916
252. Terry Warner, Rio Algom Mining Corp., La Sal Route, Moab, UT 84532
253. Leonard H. Weinstein, Program Director of Environmental Biology, Cornell University, Boyce Thompson Institute for Plant Research, Ithaca, NY 14853
254. T. C. Wells, D'Appolania Consulting Engineers, Inc., 10 Duff Road, Pittsburgh, PA 15235
255. D. E. Wessol, Reactor Physics Branch, EG&G Idaho, Inc., P.O. Box 1625, Idaho Falls, ID 83415
256. T. E. White, School of Civil Engineering, Purdue University, West Lafayette, IN 47907
257. S. L. Whiteside, Geotechnical Engineers, Inc., 1017 Main Street, Winchester, MA 01890
258. D. Widmayer, U.S. Miclear Regulatory Commission, 7915 Eastern Ave., Silver Spring, MD 20920
259. P. J. Wierenga, Department of Agronomy, New Mexico State University, P.O. Box 3Q, Las Cruces, NM 88003
260. Raymond G. Wilhour, Chief, Air Pollution Effects Branch, Corvallis Environmental Research Laboratory, U.S. Environmental Protection Agency, 200 SW 35th Street, Corvallis, OR 97330
261. Patrick J. Withkowski, Dept. of Civil Engineering, Princeton University, Princeton, NJ 08544
262. Frank J. Wobber, Ecological Research Division, Office of Health and Environmental Research, Office of Energy Research, MS-E201, Department of Energy, Washington, DC 20545
263. M. Gordon Wolman, The Johns Hopkins University, Department of Geography and Environmental Engineering, Baltimore, MD 21218
264. J. R. Yearsley, U.S. Environmental Protection Agency, REGION 10, 1200 Sixth Avenue, Seattle, WA 98101
265. Jim Yeh, Dept. of Environmental Sciences, University of Virginia, Charlottesville, VA 22903
266. Young S. Yoon, Boyle Engineering Corporation, Suite 200, 165 South Union, Lakewood, CO 80228
267. H. H. Zehner, U.S. Geological Survey-Water Resources Division, 1013 N. Broadway, Knoxville, TN 37917

268. H. D. Yeh, Department of Civil Engineering, National Chiao-Tung University, Hsinchu, Taiwan
269. Paolo Zannetti, AeroVironment Inc., 825 Myrtle Ave., Monrovia, CA 91016-3424
270. Feng-quing Zhuang, Institute of Mechanics, Chinese Academy of Sciences, Beijing, Peoples' Republic of China
271. P. B. Zielinski, Water Resources Institute, Clemson University, Clemson, SC 29631
272. Office of Assistant Manager for Energy Research and Development, Oak Ridge Operations, P.O. Box E, U.S. Department of Energy, Oak Ridge, TN 37831
273-302. Technical Information Center, Oak Ridge, TN 37831

☆U.S. GOVERNMENT PRINTING OFFICE: 1987-748-168/60075