START

0 000085 my

UR 0307-3

30 9-24-95

(2) I-31935

(AD) (5)

ORNL-5567/R1

ornl

OAK RIDGE NATIONAL LABORATORY

FEMWATER: A Finite Element Model of WATER Flow through Saturated-Unsaturated Porous Media—First Revision

G. T. Yeh

Environmental Sciences Division

Publication No. 2943

BEST AVAILABLE COPY

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

9 2 1 2 4 5 8 0 4 6 0

LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

1

9 2 1 1 2 4 5 8 0 4 6 1

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: A11; Microfiche: A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

9 2 1 2 4 5 8 0 4 6 2

ORNL--5567/R1

DE87 014966

FEMWATER: A Finite Element Model of WATER Flow through Saturated-Unsaturated Porous Media - First Revision

G. T. Yeh

Environmental Sciences Division
Publication No. 2943

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

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

9 2 1 2 4 5 8 0 4 6 3

CONTENTS

	Page
LIST OF FIGURES	v
LIST OF TABLES	vii
ABSTRACT	ix
1. INTRODUCTION	1
2. MATHEMATICAL STATEMENTS	7
2.1 Derivation of the Governing Equations	7
2.2 Initial and Boundary Conditions	17
3. NUMERICAL APPROXIMATIONS	23
3.1 Finite Element Approximation in Space	25
3.2 Finite Difference Approximation in Time	39
3.3 Numerical Implementation of Boundary Conditions	41
3.4 Solution of the Matrix Equations	47
3.5 Computation of Mass Balance	50
4. DESCRIPTION OF THE COMPUTER PROGRAM	53
4.1 Purpose of FEMWATER	53
4.2 Program Structure	56

Page

5. USER'S MANUAL 71

5.1 Specification of Array Dimensions 71

5.2 Soil Property Function Specifications 82

5.3 Data Input Guide 84

6. SAMPLE PROBLEMS 119

6.1 One-dimensional Column Problem 119

6.2 Two-Dimensional Drainage Problem 126

6.3 Seepage Pond Problem 134

7. REFERENCES 149

APPENDIX A: LISTING OF FORTRAN SOURCE CODE 155

LIST OF FIGURES

Figure		Page
1	An elemental volume containing constant amount of solids	8
2	Region of interest and its boundaries	19
3	Example discretization of a region by 80 nodes and 69 elements (with 59 quadrilateral and 10 triangular elements)	26
4	Relationship between globally defined and elementwise defined basis functions	27
5	Global coordinate vs local coordinate for a typical quadrilateral element	30
6	Global coordinate vs area coordinate for a typical triangular element	32
7	Program structure of FEMWATER	56
8	Flow chart of Subroutine GW2DXZ	58
9	Numbering of element incidence	95
10	Problem definition for the one-dimensional transient flow in a soil column	120
11	Simulated pressure head profiles at various time during (a) infiltration and (b) evaporation for the one-dimensional column problem	125

Figure		Page
12	Problem definition for the two-dimensional steady state flow to the parallel drains . . .	127
13	Simulated steady state profiles of water table and base pressure head for the two-dimensional drainage problem	133
14	Problem definition for the seepage pond problem	135
15	Hydraulic property functions used for the seepage pond problem	136
16	Finite element discretization for the seepage pond problem	137
17	Flow variables of the seepage pond as simulated by the revised FEMWATER, (a) Darcy's velocity plot, (b) distribution of pressure head, and (c) distribution of total head	147

LIST OF TABLES

Table		Page
1	Input data for the one-dimensional column problem	122
2	Input data for the two-dimensional drainage problem	129
3	Input data for the seepage pond problem	139

9/2/4 5 8 0 4 6 9

ABSTRACT

YEH, G. T. 1987. FEMWATER: A Finite Element Model of WATER Flow through Saturated-Unsaturated Porous Media - First Revision, ORNL-5567/R1, Oak Ridge National Laboratory, Oak Ridge, TN 37831. 248 pp.

This report presents the first revision of the "FEMWATER: A Finite Element Model of WATER flow through Saturated-Unsaturated Porous Media (ORNL-5567)." The special features of the revised FEMWATER are its flexibility and versatility in modeling as wide a range of real world problems as possible. The model is designed: (1) to treat heterogeneous and anisotropic media consisting as many geologic formation as desired, (2) to consider spatially and temporally distributed as well as point sources/sinks, (3) to accept the prescribed initial conditions or to obtain them by simulating steady state version of the system under consideration, (4) to deal with transient head distributed over the Dirichlet boundary, (5) to handle time dependent fluxes due to pressure gradient varying along the Neumann boundary, (6) to treat time-dependent total fluxes (i.e., the sum of gravitational fluxes and pressure-gradient fluxes) distributed over the Cauchy boundary, (7) to automatically determine variable boundary conditions of evaporation, infiltration, or seepage on the soil-air interface, (8) to include the off-diagonal hydraulic conductivity components in the modified Richards' equations for

dealing with cases when the coordinate system does not coincide with the principal directions of the hydraulic conductivity tensor, (9) to provide two options of treating the mass matrix - consistent and lumping, (10) to provide three alternatives of approximating the time derivative term- Crank-Nicolson central difference, backward difference, and mid-difference, (11) to give three options (exact relaxation, under-and over-relaxation) for estimating the nonlinear matrix, (12) to include two options (direct solution with Gaussian elimination method and successive point iterations) for solving the linearized matrix equations, (13) to include both quadrilateral and triangular elements to facilitate the discretization of the region, (14) to automatically reset time step size when boundary conditions or source/sinks change abruptly, (15) to check the mass balance computation over the entire region for every time step, and (16) to consider cases of axis-symmetrical problems. The model is verified with other numerical models for three examples.

9 2 1 2 4 5 8 0 4 7 0

1. INTRODUCTION

To study the transport of dissolved constituents in a subsurface flow 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 eta 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 of these model lie mainly in the number of space dimension, the utilization of numerical methods, the treatment of initial and boundary conditions, and the consideration of source/sink.

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

9 2 1 2 4 5 8 0 4 7 1

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) aquifer heterogeneity and anisotropy are easily handled, (4) fewer node points are sometimes required to represent the aquifer to the same level of accuracy, resulting in savings of computational time and computer storage, and (4) flux types of boundary conditions result naturally from the integral formulation. Hence the FEM is used in this report 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 eta al, 1981). 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 (Duguid and Reeves 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

9 2 1 2 4 5 8 0 4 7 2

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 has not included all these 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 ate time-dependent either gradually or abruptly and vary along the boundary. On the treatment of source/sinks, existing computational codes fare much better since quite a few indeed included transient, point as well as distributed source/sink (Neumann et al 1974).

9 2 1 2 4 5 8 0 4 7 3

In the real-world problems, the recharge and pumping due to rainfall and artificial withdrawals are likely to be distributed over the region of interest and time dependent. The given boundary pressure head are, with rare exception, varying with time and along Dirichlet segments at the surface water bodies which inevitably have their water levels fluctuations along the soil-water interfaces. The prescribed pressure gradient 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 the revised FEMWATER also includes several special features. By including the off-diagonal hydraulic conductivity components, the model can then applied to anisotropic and nonuniform media. For an anisotropic uniform medium, one can imbed the coordinate system to coincide with the principle directions of the hydraulic

9 2 1 2 4 5 8 0 4 7 4

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 varies with space, prohibiting the imbedance of a unique coordinate system.

The Richards' equation is a nonlinear one, its numerical solution exhibits fluctuations in iterations for many situations. Under-relaxation is thus provided in 3DFEMWATER to dump the fluctuations. For certain sets of system parameters such as hydraulic retention and hydraulic conductivity, the rate of numerical convergence is very slow. Hence, over-relaxation 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 employment of direct elimination solution of the matrix equation becomes impractical because of excessive requirements on the CPU memory. For example, a 400 x 50 node discretization in the two-dimensional problems would need a CPU memory of about 8.3 megabytes for the banded coefficient matrix alone. These plus other overhead CPU memory make the problem excessively large in terms of CPU memory requirement. To alleviate this problem, basic point iteration methods such as the Gauss-

9 2 1 2 4 5 8 0 4 7 5

Seidel, successive under-relaxation, and successive over-relaxation schemes are built in the revised FEMWATER computer program as alternative strategies in obtaining the solutions.

When irregular curved boundaries are encountered, the use of quadrilateral elements alone to approximate such boundaries often require elaborate and tedious efforts. Hence, to facilitate the discretization of compound regions with complex curved boundaries, both quadrilateral and triangular elements are included in the revised FEMWATER.

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 in FEMWATER to facilitate the computations. This alleviates the troublesome external manipulations.

9 2 1 2 4 5 8 0 4 7 6

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). Basing on (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 combined governing equation for the distribution of pressure head in the porous media. The derivation of this combined equation will be described below.

The continuity of fluid is expressed in integral form as:

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

where

t = time (T),

$D(\ )/Dt$ = material 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),

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).

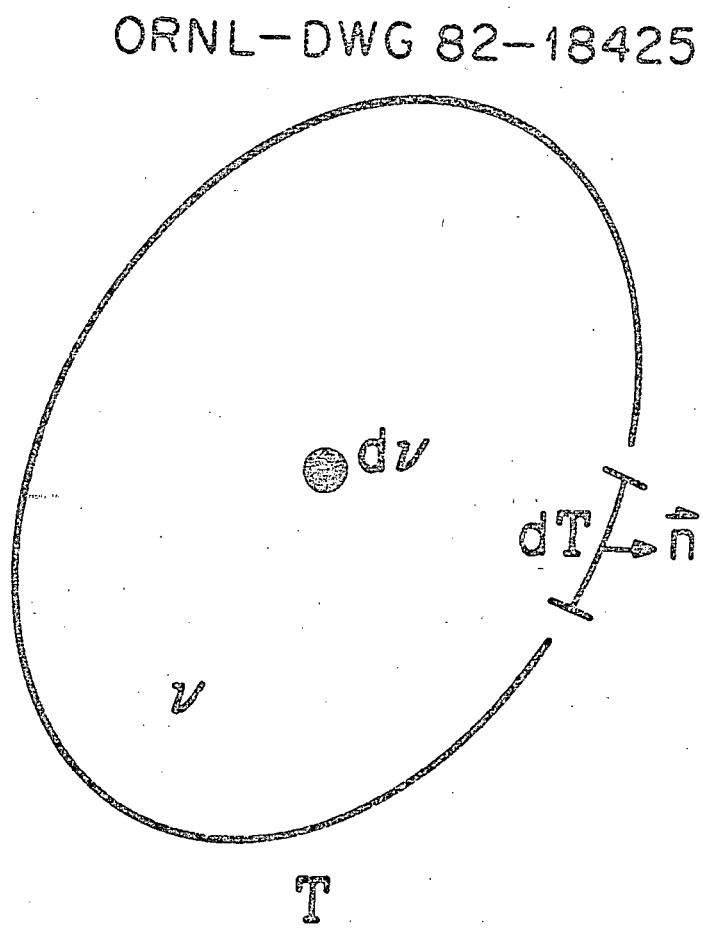


Fig. 1. An elemental volume containing constant amount of solids

By the Reynold's transport theorem (Owczarek, 1964), Eq. (1) can be written as:

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

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 ξ is arbitrary, one can obtain the following continuity equation of fluid in differential form:

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

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:

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

where

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

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

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

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

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

However, since the solid grains are incompressible, we have:

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

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

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

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 proportionally 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 ($ML/T^2/L^2/T$).

9 2 1 2 4 5 8 6 0 4 8 1

g = acceleration of gravity, (L/T^2).

A new state variable h in Eqs. (9) through (11) 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, i.e.,

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

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:

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

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 constitute relationship, we shall assume that the moisture content, θ, is a unique function of the pressure head. This function is termed the retention curve in the soil physics literature and expressed mathematically as

$$\theta = n_e S = \theta(h) , \tag{21}$$

where θ is the moisture content.

Equations (3), (8), (9), (13), (20), and (21) (representing the continuity of fluid, the continuity of solids, the Darcy's law, the equation of state, the vertical consolidation of the solid matrix, and the retention characteristics of the media) 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:

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

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

and

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

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

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

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

$$\beta = \frac{1}{\rho_f} \frac{d\rho_f}{dp} \quad . \tag{25}$$

Rewriting Eq. (25) in the following form,

$$d\rho_f = \rho_f \beta dp \quad , \tag{26}$$

and using Eq. (13), we will have

$$\frac{\partial \rho_f}{\partial t} = \rho_f \beta \frac{\partial p}{\partial t} = \rho_f \beta' \frac{\partial h}{\partial t} \quad , \tag{27}$$

where

$$\beta' = \rho_f g \beta \quad , \tag{28}$$

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

$$\frac{d\theta}{dh} = n_e \frac{dS}{dh} \quad , \tag{29}$$

from which

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

9 2 1 2 4 5 8 0 4 8 6

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 third 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:

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

in which

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

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:

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

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 soil-air interface, or soil-soil interface, or soil-water interface. From the mathematical point of view, it may be treated as Dirichlet boundary on which the functional values are prescribed, or Neumann boundary on which the gradients of the function are known, or 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 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).

No matter from which point of view, all boundary conditions must be eventually 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 4 parts, B_D, B_N, B_C, and B_V (Fig. 2),

ORNL-DWG 87M-9564

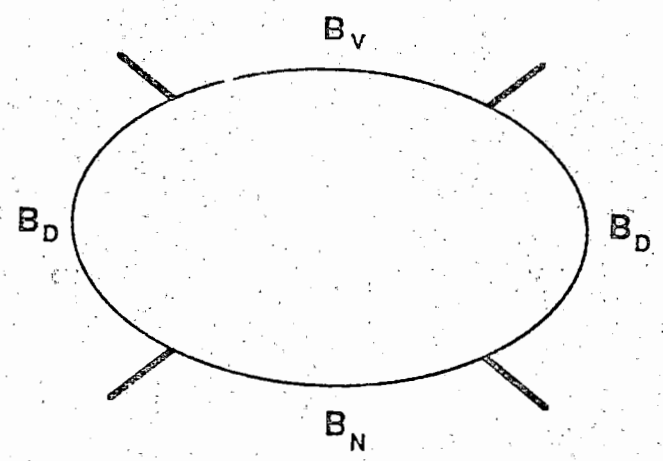


Fig. 2. Region of interest and its boundaries.

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

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

9 2 1 2 4 5 8 0 4 8 9

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

and

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

where (x_b, z_b) is the spatial coordinate on the boundary; 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, treated separately for precipitation and nonprecipitation periods. During precipitation periods, we impose

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

or

$$-n \cdot (\underline{K} \cdot \nabla H) = q_p(x_b, z_b, t) \qquad \text{on} \quad B_V \,, \tag{39}$$

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 prior. Such a case would arise at the ground surface where either Dirichlet (ponding) or Cauchy (infiltration) conditions could prevail. The change over from Dirichlet condition specified by Eq. (38) to Cauchy condition 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 will be treated in section 3.3.

During non-precipitation period, we imposed

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

or

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

or

$$-n \cdot K \cdot (\nabla H) = q_e(x_b, z_b, t) \quad \text{on} \quad B_V , \tag{42}$$

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 will have to be devised to solve the problem. The finite element method (FEM) rather than the finite difference method (FDM) 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 regular grid system and to interpolate the flux-type boundary conditions.

9 2 1 2 4 5 8 0 4 9 1

3. NUMERICAL APPROXIMATIONS

Eqs. (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 tool 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 (FDM) and finite element methods (FEM) (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; Reisenauer et al 1981) or the integrated compartment method (ICM) (Yeh and Luxmoore 1983), have been employed to deal with special cases of the Richards' equations. Only the finite differences and finite elements can be applied to the most generalized form of the transport equations. The advantages of FEM are their inherent ability to discretize complex boundaries, easiness to deal with flux-type boundary conditions, and flexibility to include cross-derivative terms. The most disadvantages of 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 provides

9 2 1 2 4 5 8 0 4 9 2

flexibility of solving the resulting matrix equation with various iteration methods. However, it suffers from the facts that the regular rectangular grid system has to be used, the flux-type boundary conditions have to 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 over-relaxation schemes to solve the matrix equation (Yeh and Luxmoore 1983). In light of these discussions and the facts that a 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 modified Richards' equation.

Because applications of finite element methods to groundwater flow have been fully addressed (Pinder and Gray 1977), the theoretical basis

9 2 1 2 4 5 B 0 4 9 3

of the method will not repeated, only the numerical procedures are summarized. The application of finite element methods to groundwater flow generally consists of nine steps: (1) subdivide the region into a number of elements interconnected by a number of nodes, (2) define basis functions (interpolating functions), (3) approximate the solution by linear interpolation, (4) define weighting functions, (5) define a residual, (6) set the residual to zero to obtain a set of ordinary differential equations with respect to time, (7) time integration to derive the coefficient matrix, (8) incorporate boundary conditions into coefficient matrix, and (9) solve the matrix equation. These procedures are addressed below.

3.1 Finite Element Approximations in Space

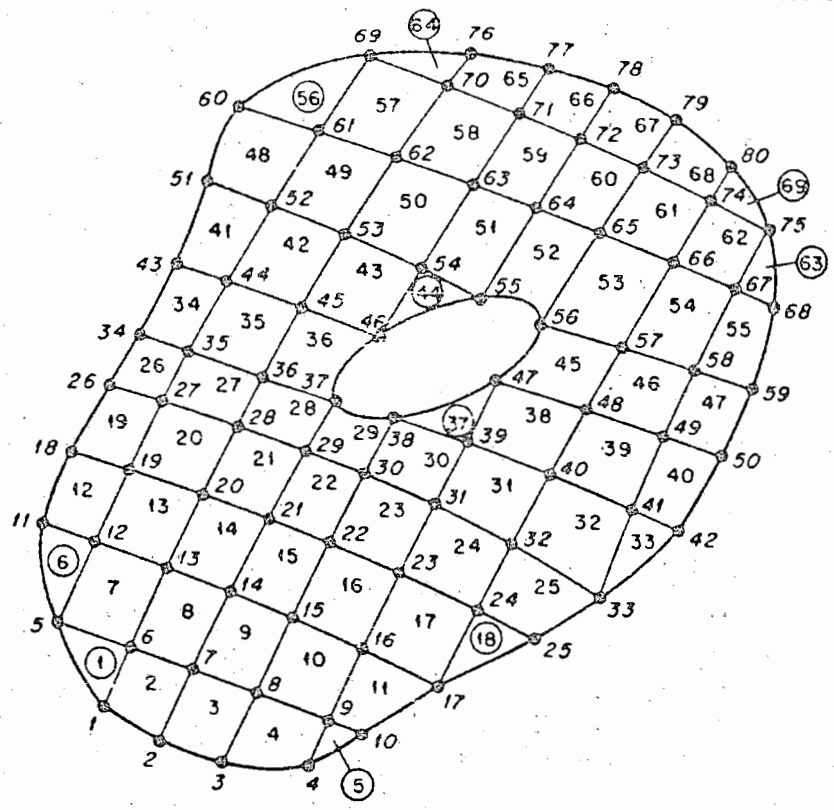
Step 1 - The first step is to subdivide the region into an assemblage of M subdomains called elements interconnected by N vertices called nodes. An element can either be quadrilateral or triangular in this revised FEMWATER. For example, in Fig. 3, the region is divided into 69 elements ($M = 69$) with 59 quadrilateral elements and 10 triangular elements. These elements are connected by 80 nodes ($N = 80$).

Step 2 - According to finite element method, the solution over the entire region is approximated by interpolating the values at N nodes. Thus, the second step is to define the interpolating functions (basis functions). Let $N_j(x,z)$ be the basis function of node point j in the

9 2 1 2 4 5 8 0 4 9 4

9 2 1 2 4 5 8 0 4 9 5

ORNL-DWG 82-20016



2 NODE NUMBER
(5) TRIANGULAR ELEMENT
2 QUADRILATERAL ELEMENT

Fig. 3. Example discretization of a region by 80 nodes and 69 elements (with 59 quadrilateral and 10 triangular elements)

two-dimensional space. The basis function, $N_j(x,z)$, will have a value of 1.0 at the nodal point j and a value of 0.0 at all other nodal points, f, g, h, i, k, □, and m (Fig. 4). Furthermore, $N_j(x,z)$ will be given nonzero values only over those elements that have one nodal point coinciding with point j and zero values over all other elements in the domain.

ORNL-DWG 82-10307B

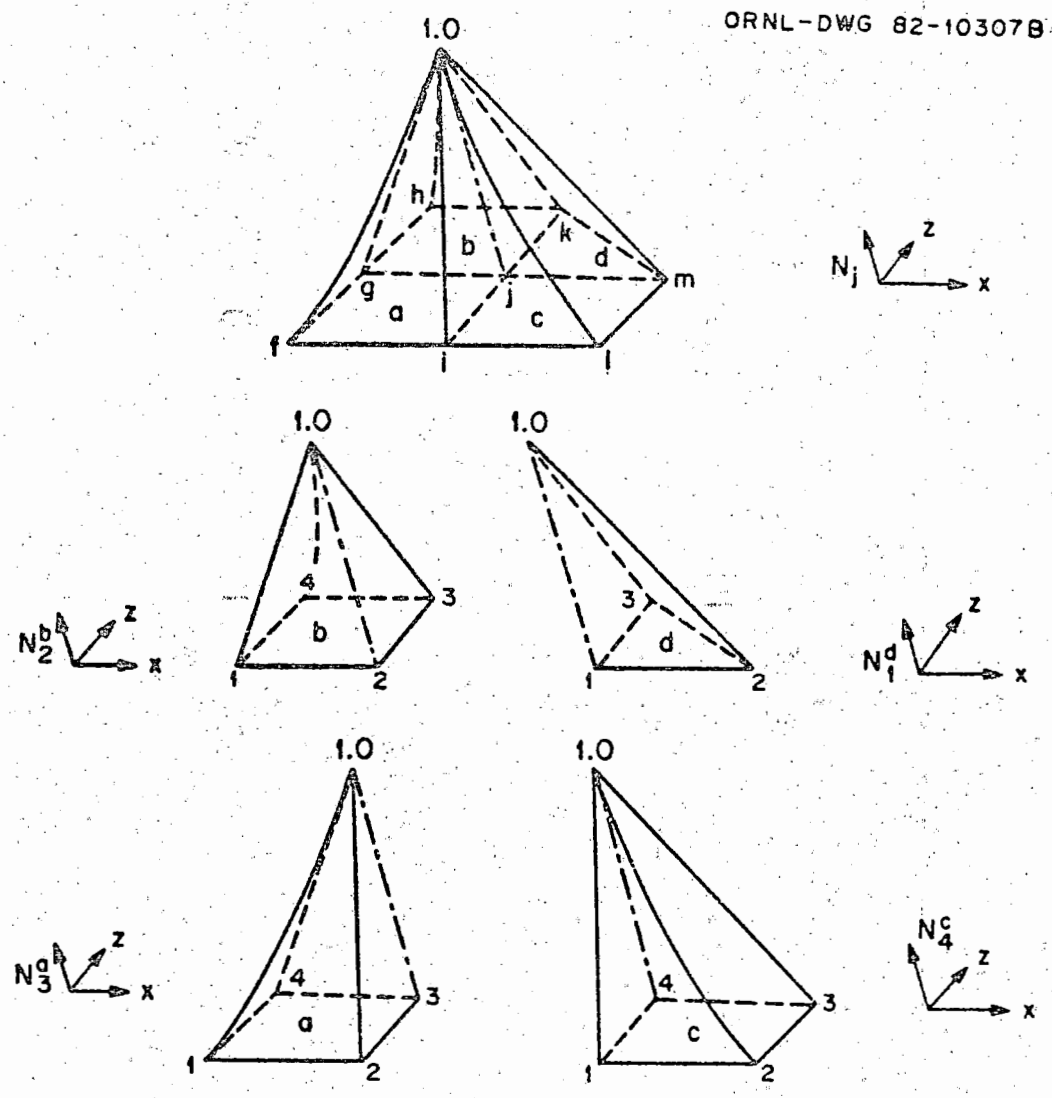


Fig. 4. Relationship between globally defined and elementwise defined basis functions

9 2 1 2 4 5 8 0 4 9 6

In the finite element methods, one normally constructs element matrices first, then assembles them to form the global matrix. Hence, for each linear quadrilateral element, one has to define four basis functions for its four nodes and, for each linear triangular element, three basis functions for its three nodes. This can be done simply by equating element basis functions to the corresponding global basis functions. For example, it can be seen from Fig. 4:

$$N_3^a = N_j \qquad \text{over element a ,} \tag{43a}$$

$$N_2^b = N_j \qquad \text{over element b ,} \tag{43b}$$

$$N_4^c = N_j \qquad \text{over element c ,} \tag{43c}$$

$$N_1^d = N_j \qquad \text{over element d ,} \tag{43d}$$

where

N_3^a = the basis function for node 3 of element a,

N_2^b = the basis function for node 2 of element b,

N_4^c = the basis function for node 4 of element c,

N_1^d = the basis function for node 1 of element d.

Similarly, we may obtain:

$$N_1^a = N_f, \quad N_2^a = N_i, \quad \text{and} \quad N_4^a = N_g \qquad \text{over element a ,} \tag{43e}$$

9 2 1 2 4 5 8 0 4 9 7

$$N_1^b = N_g, \quad N_3^b = N_k, \quad \text{and} \quad N_4^b = N_h \qquad \text{over element b} , \tag{43f}$$

$$N_1^c = N_i, \quad N_2^c = N_\square , \quad \text{and} \quad N_3^c = N_m \qquad \text{over element c} , \tag{43g}$$

$$N_2^d = N_m \quad \text{and} \quad N_3^d = N_k \qquad \text{over element d} , \tag{43h}$$

Thus, to define basis functions element-wise is equivalent to defining them globally and vice versa.

For a linear quadrilateral element e (Fig. 5), the following four basis function are employed:

$$\begin{aligned} N_1^e &= (1-\xi)(1-\eta)/4, \qquad & N_2^e &= (1+\xi)(1-\eta)/4 , \\ N_3^e &= (1+\xi)(1+\eta)/4, \qquad & N_4^e &= (1-\xi)(1+\eta)/4 , \end{aligned} \tag{44}$$

where N_e^j (j = 1, 2, 3, or 4) is the basis function of node j in terms of local coordinate (ρ,θ). The transformation from the local coordinate (ξ,η) to the global coordinate (x,z) is achieved by:

$$x = \sum_{j=1}^{4} x_j N_j^e(\xi,\eta) \quad \text{and} \quad z = \sum_{j=1}^{4} z_j N_j^e(\xi,\eta) . \tag{45}$$

Figure 5 illustrates a typical element in global and local coordinates. Note that the element is square in the local coordinate system regardless of the shape of the quadrilateral element in the global coordinate system.

ORNL-DWG 74-3899R

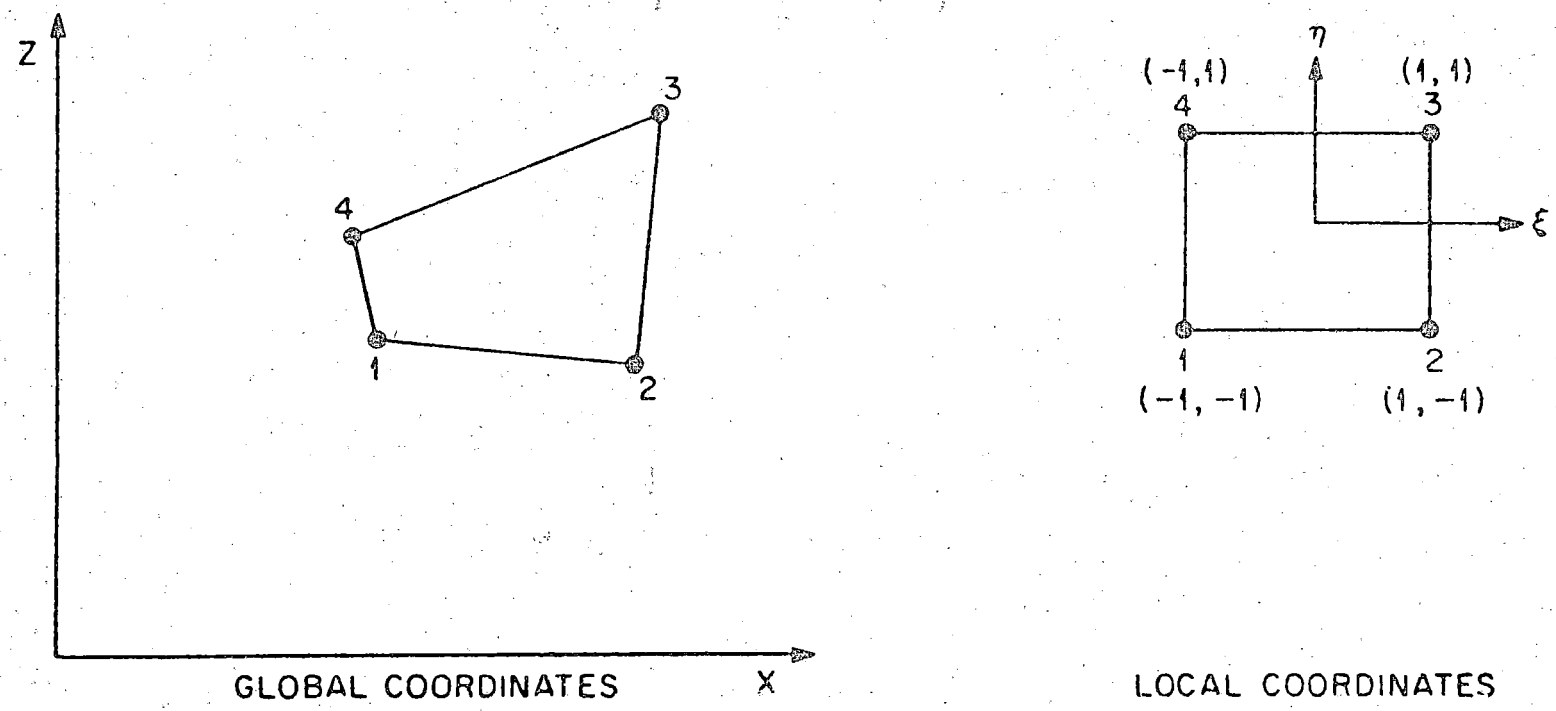


Fig. 5. Global coordinate vs local coordinate for a typical quadrilateral element

Since the basis functions are used for coordinate transformation, this element formulation is term isoparametric.

For a linear triangular element e (Fig. 6), the three basis functions for its three nodes are given by:

$$N_1^e = L_1, \quad N_2^e = L_2, \quad \text{and} \quad N_3^e = L_3 , \tag{46}$$

where N_j^e $(j = 1, 2, \text{or } 3)$ is the basis function of node j in terms of the local coordinate $(L_1, L_2, L3)$ (Fig. 6). The local coordinates are also called area coordinates for a reason to be explained later. The global coordinate $(x, z,)$ and the area coordinate (L_1, L_2, L_3) are related by:

$$L_1 = (a_1 + b_1 x + c_1 z)/(2A) , \tag{47a}$$

$$L_2 = (a_2 + b_2 x + c_2 z)/(2A) , \tag{47b}$$

$$L_3 = (a_3 + b_3 x + c_3 z)/(2A) , \tag{47c}$$

where

$$a_1 = x_2 z_3 - x_3 z_2, \quad b_1 = z_2 - z_3, \quad c_1 = x_3 - x_2, \tag{47d}$$

$$a_2 = x_3 z_1 - x_1 z_3, \quad b_2 = z_3 - z_1, \quad c_2 = x_1 - x_3, \tag{47e}$$

$$a_3 = x_1 z_2 - x_2 z_1, \quad b_3 = z_1 - z_2, \quad c_3 = x_2 - x_1, \tag{47f}$$

ORNL-DWG 82-20017

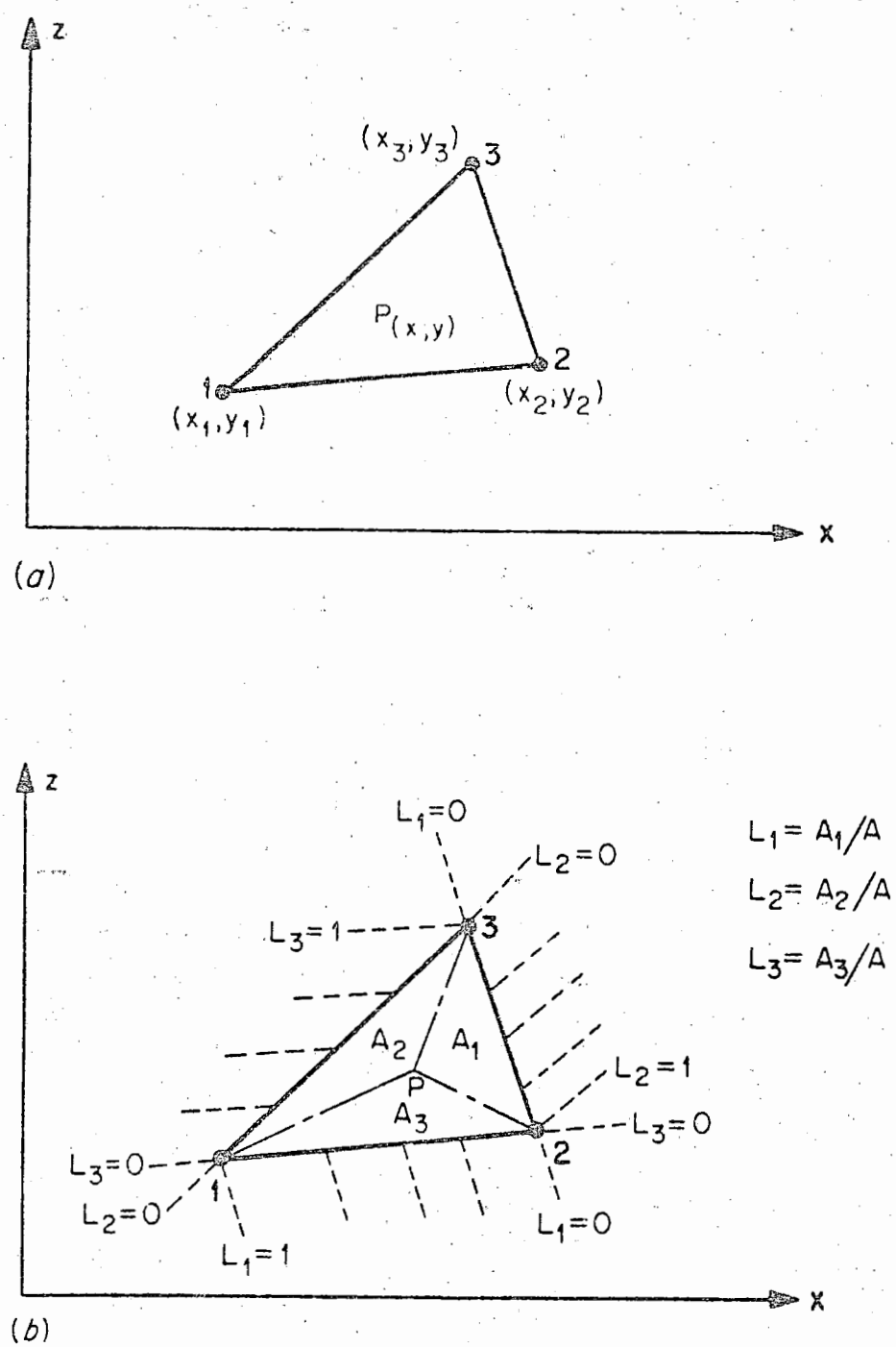


Fig. 6. Global coordinate vs area coordinate for a typical triangular element.

and A is the area of the triangle. It is seen that from Eqs. (47a) through (47f) that:

$$L_1 + L_2 + L_3 = 0 \ . \tag{48}$$

Thus, it is clear that only two of the three area coordinates, L_1, L_2, and L_3, can be independent, just as in the original coordinate system, where there are only two independent coordinates, x and z. Furthermore, a little algebraic manipulation will reveal that the coordinates L_1, L_2, and L_3 are in fact the ratios of the areas A_1, A_2, and A_3, respectively, to the triangular area A (Fig. 6). This is why they are called area coordinates.

Step 3 - The third step of the finite element method is to approximate the hydraulic head h by:

$$h \approx h = \sum_{j=1}^{N} h_j(t) N_j(x,z) \ , \tag{49}$$

where N is the total number of nodes in the region and N_j and h_j are the basis function and the amplitude of h, respectively, at nodal point j. It is interesting to note that the fundamental distinction between finite element methods (FEMs) and finite difference methods (FDMs) is that the former is based directly upon approximation of the function as in Eq. (49), whereas the latter is based on approximation of derivative (Hilderbrand, 1968).

Step 4 - The fourth step of the FEM is to choose a set of weighting functions. We will choose the Galerkin weighting functions that are the set of basis functions, i.e.:

$$W_i(x,z) = N_i(x,z), \quad i = 1, 2, \ldots, N , \tag{50}$$

where W_i is the weighting function for nodal point i.

Step 5 - Since h is only an approximate solution of Eq. (32), it will not satisfy the equation. Thus, the fifth step of the FEM is to define a residual R_r by:

$$R = F \frac{\partial h}{\partial t} - \nabla \cdot [K \cdot (\nabla H)] - q . \tag{51}$$

Step 6 - According to the principle of the FEM, we choose the coefficient h_j's in Eq. (49) such that the residual in Eq. (51) weighted by the set of weighting functions is zero. In other words, the sixth step of the FEM applied to this problem is:

$$\int_R W_i (F \frac{\partial h}{\partial t} - \nabla \cdot [K \cdot (\nabla H)] - q) \, dR, \quad i = 1, 2, \ldots, N . \tag{52}$$

Substituting Eqs. (46) and (50) into Eq. (52) and integrating by part, we obtain:

$$[\int_R N_i F N_j dR] \frac{dh_j}{dt} + [\int_R (\nabla N_i)\cdot K\cdot(\nabla N_j)\ dR]\ h_j = \int_R N_i q\ dR$$

$$- \int_R (\nabla N_i)\cdot K\cdot(\nabla z)\ dR + \int_B n\cdot K\cdot(\nabla H) N_i\ dB\ , \tag{53}$$

$$i = 1,\ 2,\ \ldots,\ N\ .$$

Equation (53) written in matrix form is:

$$[M]\{\frac{dh}{dt}\} + [S]\{h\} = \{G\} + \{Q\} + \{B\}\ , \tag{54}$$

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 stiff matrix resulting from the action of conductivity; $\{G\}$, $\{Q\}$ 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

$$M_{ij} = \sum_{e\epsilon M_e} \int_{R_e} N^e_\alpha F N^e_\beta\ dR\ , \tag{55}$$

$$S_{ij} = \sum_{e\epsilon M_e} \int_{R_e} [(\nabla N^e_\alpha)\cdot K\cdot(\nabla N^e_\beta)\ dR\ , \tag{56}$$

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

$$G_i = - \sum_{e \epsilon M_e} \int_{R_e} (\nabla N_\alpha^e) \cdot K \cdot \nabla z \, dR \, , \tag{57}$$

$$Q_i = \sum_{e \epsilon M_e} \int_{R_e} N_\alpha^e q \, dR \, , \tag{58}$$

and

$$B_i = - \sum_{e \epsilon N_{se}} \int_{B_e} N_\alpha^e \cdot [-K \cdot (\nabla H)] \, dB \, , \tag{59}$$

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 set of basis functions as the set of weighting functions for all terms. Hence the Galerkin finite element method results.

The reduction of the partial differential equation (PDE), Eq. (32),

to the set of ordinary differential equations (ODE), Eq. (54), simplifies to the evaluation of integrals on the right hand side of Eqs. (55) through (59) for every element or boundary segment e 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:

$$M^e_{\alpha\beta} = \int_{R_e} N^e_\alpha F N^e_\beta \, dR \ , \tag{60}$$

$$S^e_{\alpha\beta} = \int_{R_e} (\nabla N^e_\alpha)\cdot K\cdot(\nabla N^e_\beta) \, dR \ , \tag{61}$$

$$G^e_\alpha = -\int_{R_e} (\nabla N^e_\alpha)\cdot K\cdot\nabla z \, dR \ , \tag{62}$$

$$Q^e_\alpha = \int_{R_e} N^e_\alpha q \, dR \ , \tag{63}$$

and

$$B^e_\alpha = -\int_{B_e} N^e_\alpha n\cdot[-K\cdot(\nabla H)] \, dB \ , \tag{64}$$

where the superscript or subscript e denote the element and α, β = 1, 2, 3, or 4 for linear quadrilateral elements or α, β = 1, 2, or 3 for linear triangular elements.

For a quadrilateral element, Eqs. (60) through (63) are computed by Gaussian quadrature (Conte, 1965) because it is not easy to solve Eq. (42) for ξ and η in terms of x and z. On the other hand, for a triangular element, the transformation from the global coordinate (x,z) to the local coordinate (L_1, L_2, L_3) is already given explicitly by Eqs. (44a) through (44f). This fact and the following integration identity,

$$\int_{R_e} L_1^n L_2^m L_3^k dR = 2A \frac{n!\ m!\ k!}{(m+n+k+2)!} \quad , \tag{65}$$

make the evaluation of Eqs. (60) through (63) for linear triangular element obtainable analytically. With the element matrices $[M^e]$ and $[S^e]$ and the element column vectors $\{G^e\}$ and $\{Q^e\}$ computed, the global matrices [M] and [S] and the global column vectors {G} and {Q} are then assembled element by element.

Referring to [M], one may recall that this is a unit matrix if the finite difference formulation is used in spatial discretization. Hence, by proper scaling, the mass matrix can be reduced to the finite-difference equivalent by lumping (Clough 1971). In many cases, the lumped mass matrix would result in better solution, in particular, if it is used in conjunction with the central or backward-difference time marching (Yeh and Ward 1980). Under such circumstances, it is preferred to the consistent mass matrix (mass matrix without lumping). Therefore, options are provided for the lumping of the matrix [M]. More explicitly, [M] will be lumped according to:

$$M_{ij} = \sum_{e \epsilon M_e} \left(\sum_{\beta=1}^{4} \int_{R_e} N_\alpha^e F N_\beta^e \, dR \right) \qquad \text{if} \quad j = i \,, \tag{66a}$$

and

$$M_{ij} = 0 \qquad \text{if} \quad j \neq i \tag{66b}$$

3.2 Finite Difference Approximation in Time

Step 7 - The seventh step of the FEM is to derive a matrix equation by integrating Eq. (54). For the time integration of Eq. (54), the load vector {B} will be ignored. This load vector will be discussed in the next section on the numerical implementation of boundary conditions. An important advantage in finite element approximation over the finite difference approximation is the 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. Three time-marching methods are adopted in the present water flow model. In the first one, the central or Crank-Nicolson formulation may be written as:

$$[M](\{h\}_{t+\Delta t} - \{h\}_t)/\Delta t + \frac{1}{2}[S](\{h\}_{t+\Delta t} + \{h\}_t) = \{G\} + \{Q\} \,, \tag{67}$$

where [M], [S], {G}, and {Q} are evaluated at $t + \Delta t/2$.

In the second method, the backward difference formulation may be written as:

$$[M]\{\{h\}_{t+\Delta t} - \{h\}_t\}/\Delta t + [S]\{h\}_{t+\Delta t} = \{G\} + \{Q\} , \tag{68}$$

where [M], [S], {G}, and {Q} are evaluated at $t + \Delta t$.

In the third optional method, the values of unknown variables are assumed to vary linearly with time during the time interval, Δt. In this mid-difference method, the recurrence formula is written as:

$$\left(\frac{1}{\Delta t}[M] + [S]\right)\{h\}_{t+\Delta t/2} - \frac{1}{\Delta t}[M]\{h\}_t = \{G\} + \{Q\} , \tag{69a}$$

and

$$\{h\}_{t+\Delta t} = 2\{h\}_{t+\Delta t/2} - \{h\}_t , \tag{69b}$$

where [M], [S], {G}, and {Q} are evaluated at $(t + \Delta t/2)$.

Equations (67) through (69) can be written as a matrix equation

$$[T]\{h\} = \{Y\} , \tag{70}$$

where [T] is the matrix, {h} is the unknown vector to be found and represents the values of discretized pressure field at new time, and {Y} is the load vector. Take for example, Eq. (67), [T] and {Y} represent the following:

$$[T] = [M]/\Delta t + [S] , \tag{71a}$$

and

92124580509

$$\{Y\} = [M]\{h\}_t/\Delta t + \{G\} + \{Q\} \ , \tag{71b}$$

where $\{h\}_t$ is the vector of the discretized pressure field at previous time.

3.3 Numerical Implementation of Boundary Conditions

Step 8 - The matrix equation, Eq. (70), is singular. To make the problem uniquely defined, boundary conditions [Eqs. (35) through (42)] must be implemented. Hence, the eighth step of the FEM is the incorporation of boundary conditions into the matrix equation [Eq. (70)].

For the Cauchy boundary condition given by Eq. (37), we simply substitute Eq. (37) into Eq. (64) to yield a boundary-element column vector $\{B_c^e\}$ for a Cauchy segment:

$$\{B_c^e\} = \{q_c^e\} \ , \tag{72}$$

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

$$q_{c\alpha}^e = - \int_{B_e} N_\alpha^e q_C \, dB, \qquad \alpha = 1 \text{ or } 2 \ , \tag{73}$$

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

For the Neumann boundary condition given by Eq. (36), we substitute

Eq. (36) into Eq. (64) to yield a boundary-element column vector $\{B_n^e\}$ for a Neumann segment:

$$\{B_n^e\} = \{q_n^e\} , \tag{74}$$

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

$$q_{n\alpha}^e = \int_{B_e} (N_\alpha^e n \cdot K \cdot \nabla z - N_\alpha^e q) \, dB \; ; \quad \alpha = 1 \text{ or } 2 , \tag{75}$$

which is independent of the pressure head.

The implementation of variable-type boundary condition is more involved. It is normally occurs at air-soil interface. During precipitation period, we will assume that only seepage or infiltration can occur for any point on the air-soil interface. No evapotranspiration is allowed. If seepage happens, 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 property and throughfall rate qp 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), two much water has been forced into the region through the node. In other words, the throughfall rate is greater than that which the media 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 that q_p in Eq. (39) or result in seepage. If the computed head is less than the ponding depth, the media 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's 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's 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 change-over of boundary condition is encountered along the entire boundary.

Similarly, during non-precipitation 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

9 2 1 2 4 5 8 0 5 1 2

condition) must specified at the node. On the other hand, if evapotranspiration happens, either Eq. (41) (Dirichlet bourdary 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 the Darcy's flux. When the computed Darcy's flux is going out of the region, the existing boundary condition is consistent and no change on boundary condition is necessary. When the Darcy's 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 generates 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's flux. Since the minimum pressure is prescribed on the boundary, it is unlikely that this computed Darcy's flux will be directed into the region. Thus, when the computed outgoing Darcy's 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's 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

9 2 1 2 4 5 8 0 5 1 3

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 two 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. (64) to yield a boundary element column vector $\{B_v^e\}$ for a variable boundary segment:

$$\{B_v^e\} = \{q_v^e\} , \tag{76}$$

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

$$q_{v\alpha}^e = -\int_{B_e} N_\alpha^e q_p \, dB, \quad \text{or} \quad q_{v\alpha} = -\int_{B_e} N_\alpha^e q_e \, dB; \quad \alpha = 1 \text{ or } 2 , \tag{77}$$

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

$$\{B\} = \{q\} , \tag{78}$$

in which

$$\{q\} = \sum_{e \epsilon N_{ne}} \{q_n^e\} + \sum_{e \epsilon N_{ce}} \{q_c^e\} + \sum_{e \epsilon N_{ve}} \{q_v^e\} , \tag{79}$$

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. The boundary flux {B} given by Eqs. (78) and (79) should be added to the right hand side {Y} in Eq. (70) during the incorporation of boundary conditions.

At nodes where Dirichlet boundary conditions are applied, an identity equation is generated for each node and included in the matrices of Eq. (70). 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), or (40), or (41) is applied.

After incorporating the boundary conditions described in this section, Eq. (70) becomes

$$[C]\{h\} = \{R\} , \tag{80}$$

where [C] is the coefficient matrix obtained by incorporating Dirichlet boundary conditions into matrix [T] and {R} is the load vector of the matrix equation, Eq. (80). {R} is obtained by by adding {B} in Eq. (78) to {Y} in Eq. (70) and then modifying it with Dirichlet boundary condition.

3.4 Solution of the Matrix Equations

Step 9 - Equation (80) is in general a banded sparse matrix equation. It may be solved numerically by either direct method or iteration methods. In direct methods, a sequence of operation is performed only once. This would result in each solution except for round-off error. In this method, one is concerned with the efficiency and magnitude of round-off error associated with the sequence of operations. On the other hand, in an iterative method, one attempts to the solution by a process of successive approximations. This involve in making an initial guess, then improving the guess by some iterative process until an error criterion is obtained. Therefore, in this technique, one must be concerned with convergence, and the rate of convergence. The round-off errors tend to be self-corrected.

For practical purposes, the most advantages of direct method are: (1) the efficient computation when the bandwidth of the matrix [C] is small, and (2) the fact that no problem of convergency is encountered when the matrix equation is linear or less severity in convergence than iterative methods even when the matrix equation is nonlinear. The most disadvantages of direct methods are the excessive requirements on CPU storage and CPU time when a large number of nodes is needed for discretization. On the other hand, the most advantages of iteration methods are efficiencies in terms of CPU storage and CPU time when large problems are encountered. Their most disadvantages are the requirements that the matrix [C] must be well conditioned to grantee a convergent solution. Hence, this report provides two options to solve the matrix

equation so that as wide a range of problems as possible can be dealt with. One is the Gaussian direct elimination method and the other is the successive point iteration methods. The former should be used when the size of the problems is such that the resulting bandwidth of matrix [C] is not large, say no greater than 50; whereas the latter should be used if the bandwidth of the matrix [C] is large.

The matrix equation, Eq. (80), 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. (80) 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:

$$\{h^{(k+1)}\} = \{h\} + (1-\omega)\{h^k\} , \tag{81}$$

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 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 over-relaxation. The under-relaxation

option should be used to overcome cases when nonconvergency or the slow convergent rate is due to fluctuation rather than due to "blowup" computations. Over-relaxation should be used to speed up convergent rate when the rate of convergency decrease monotically.

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

In summary, the revised FEMWATER includes 36 optional numerical schemes to deal with as wide a range of problems as possible. These are the combinations of: (1) two ways of treating the mass matrix (lumping and no-lumping); (2) three ways of approximating the time derivatives (central difference, backward difference, and mid-difference), and (3) six ways of solving the resulting matrix equation just mentioned above.

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):

$$F_V = \int_R (F \frac{\partial h}{\partial t} - q)\, dR \,, \tag{82}$$

and

$$F_B = \int_B F_n dB \,, \tag{83}$$

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:

$$F_n = - n \cdot K \cdot \nabla H \,, \tag{84}$$

Having obtained the pressure head field by solving Eq. (80), one can integrate Eqs. (82) and (83) 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. (84) is broken into two parts:

$$F_P = \int_R F \frac{\partial h}{\partial t} dR \,, \tag{85a}$$

and

$$F_S = \int_R q\, dR \,, \tag{85b}$$

where F_P and F_S represent the volumetric rates due to head change and artificial sources, respectively. Similarly, the integral in Eq. (84) is broken into six parts:

$$F_D = \int_{B_D} F_n \, dB \, , \tag{86a}$$

$$F_C = \int_{B_C} F_n dB \, , \tag{86b}$$

$$F_N = \int_{B_N} F_n dB \, , \tag{86c}$$

$$F_{VO} = \int_{B_{VO}} F_n dB \, , \tag{86d}$$

$$F_{VI} = \int_{B_{VI}} F_n dB \, , \tag{86e}$$

and

$$F_L = \int_{B-B_D-B_C-B_N-B_V} F_n dB \, , \tag{86f}$$

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 $B-B_D-B_C-B_N-B_V$.

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.

9 2 1 2 4 5 8 0 5 2 1

4. DESCRIPTION OF THE COMPUTER PROGRAM

4.1 Purpose of 3DFEMWATER

To refresh our memory, we will restate the governing equation and initial and boundary conditions for which this revised FEMWATER is designed to solve:

Governing Equation

$$F \frac{\partial h}{\partial t} = \nabla \cdot [K_s K_r \cdot (\nabla h + \nabla z)] + q \ , \tag{87}$$

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 storage coefficient given by:

$$F = \alpha' \frac{\theta}{n_e} + \beta' \theta + d\theta/dh \ , \tag{88}$$

where θ is the water capacity, n_e is the effective porosity, α' is the modified compressibility of the media, and β' is the modified compressibility of water.

Initial Conditions

$$h = h_i(x,z) \qquad \text{in} \qquad R \ , \tag{89}$$

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. (87).

Boundary Conditions

Dirichlet Conditions:

$$h = h_d(x_b, z_b, t) \quad \text{on} \quad B_d \,, \tag{90}$$

Neumann Conditions:

$$-n \cdot K_s K_r \cdot \nabla h = q_n(x_b, z_b, t) \quad \text{on} \quad B_n \,, \tag{91}$$

Cauchy Conditions:

$$-n \cdot (K_s K_r \cdot \nabla h + K_s K_r \cdot \nabla z) = q_c(x_b, z_b, t) \quad \text{on} \quad B_c \,, \tag{92}$$

Variable Conditions - During Precipitation Period:

$$h = h_p(x_b, z_b, t) \quad \text{on} \quad B_v \,, \tag{93a}$$

or

$$-n \cdot (K_s K_r \cdot \nabla h + K_s K_r \cdot \nabla z) = q_p(x_b, z_b, t) \quad \text{on} \quad B_v \,, \tag{93b}$$

Variable Conditions - During Non-precipitation Period:

$$h = h_p(x_b, z_b, t) \qquad \text{on} \qquad B_v , \tag{93c}$$

or

$$h = h_m(x_b, z_b, t) \qquad \text{on} \qquad B_v , \tag{93d}$$

or

$$-n \cdot K_s K_r \cdot (\nabla h + \nabla z) = q_e(x_b, z_b, t) \qquad \text{on} \qquad B_v , \tag{93e}$$

where (x_b, z_b) is the spatial coordinate on the boundary; 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. (93a) through (93e) is used at any point on the variable boundary at any time instant.

4.2 Program Structure

The revised FEMWATER consists of a MAIN program, a section of BLOCK DATA, and 25 subroutines. The MAIN is utilized to specify the sizes of all arrays. The control and coordinate activity are performed by the

subroutine GW2DXZ. Figure 7 shows the structure of the program. The functions of these subroutine are described below.

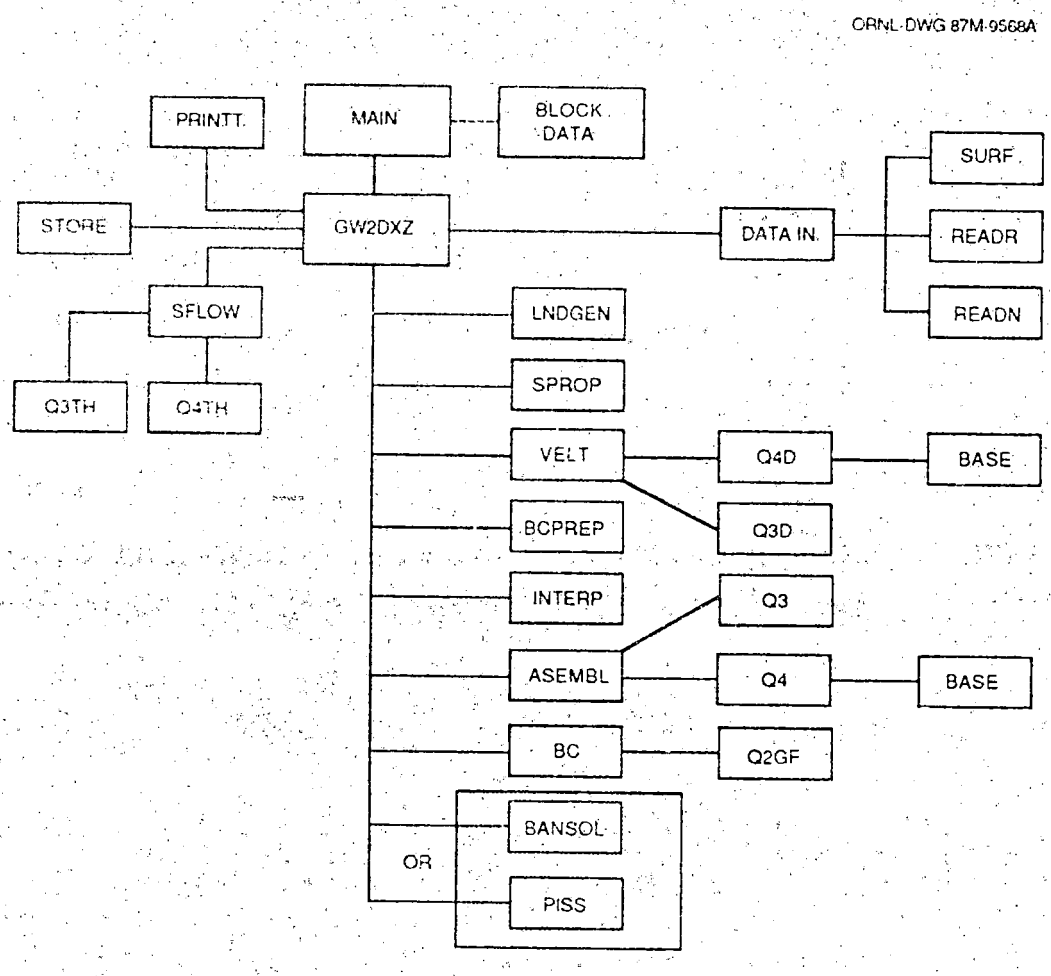


Fig. 7. Program structure of FEMWATER

Subroutine GW2DXZ

The subroutine GW2DXZ 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 GW2DXZ 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 Figure 9.

GW2DXZ calls to subroutine DATAIN to read and print input data; subroutine LNDGEN to generate the needed relationship between node number and equation number when pointwise iteration solution strategies are used; 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 BANSOL or Subroutine PISS to solve the resulting matrix equations with direct elimination method or pointwise iteration methods;

9 2 1 2 4 5 8 0 5 2 6

subroutine SFLOW to calculate flux through all types of boundaries and water accumulated in the media; subroutine PRINIT to print out the results; and subroutine STORE to store the flow variables for input to a revised FEMWASTE or for plotting (Figure 8).

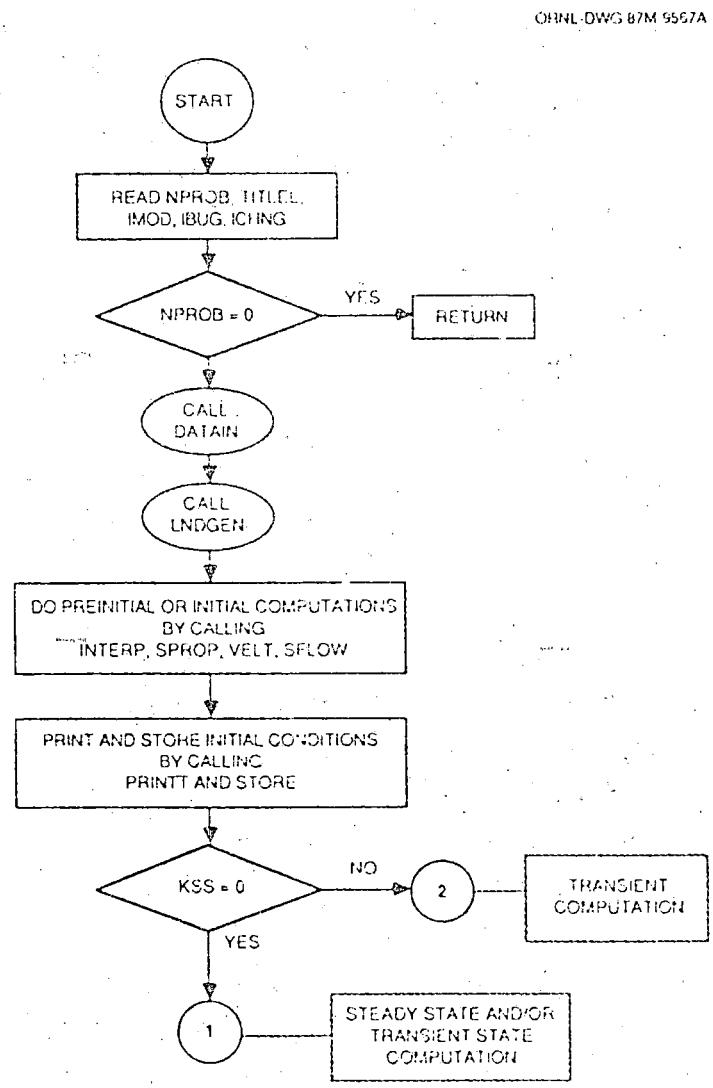


Fig. 8 Flow chart of subroutine GW2DXZ (1 of 3)

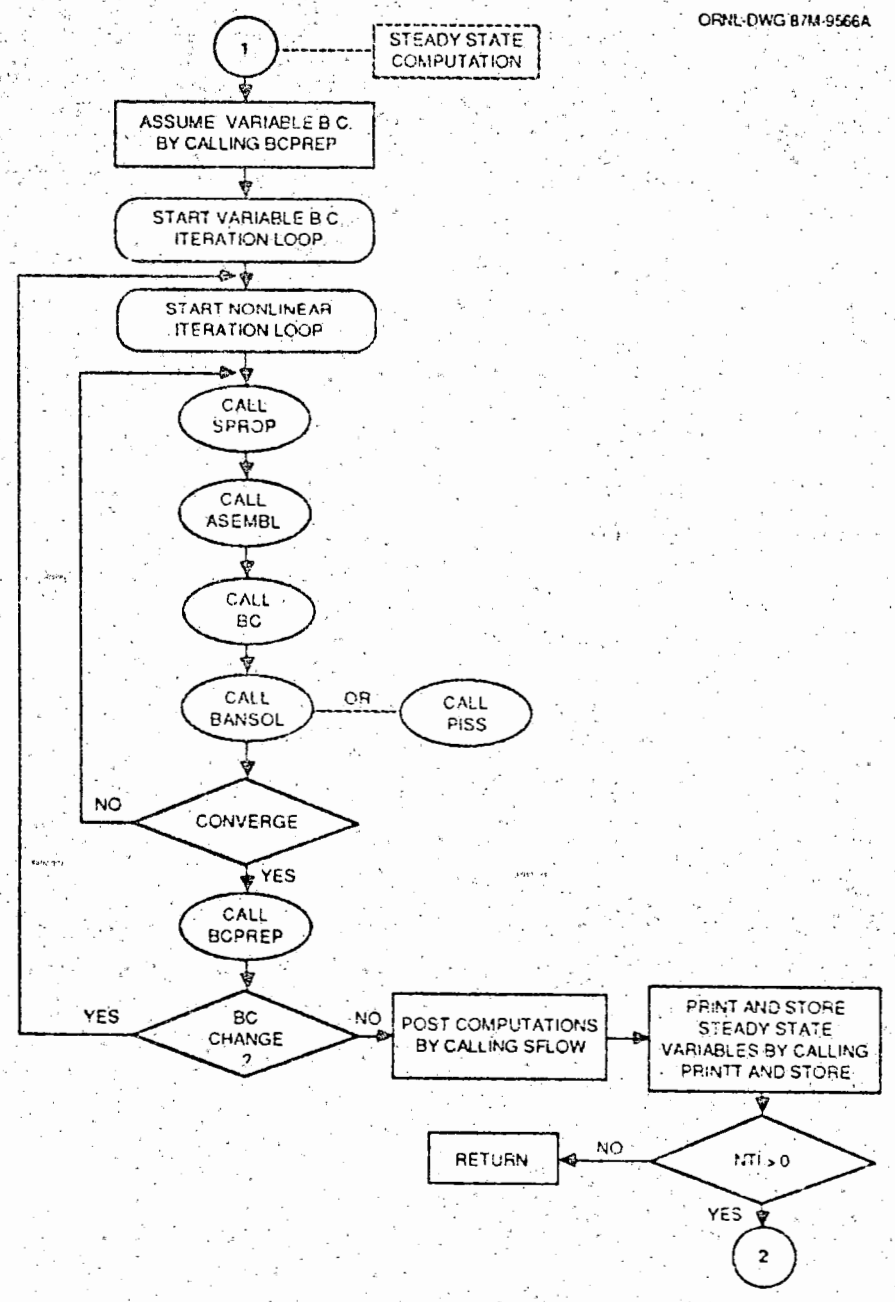


Fig. 8 Flow chart of subroutine GW2DXZ (2 of 3).

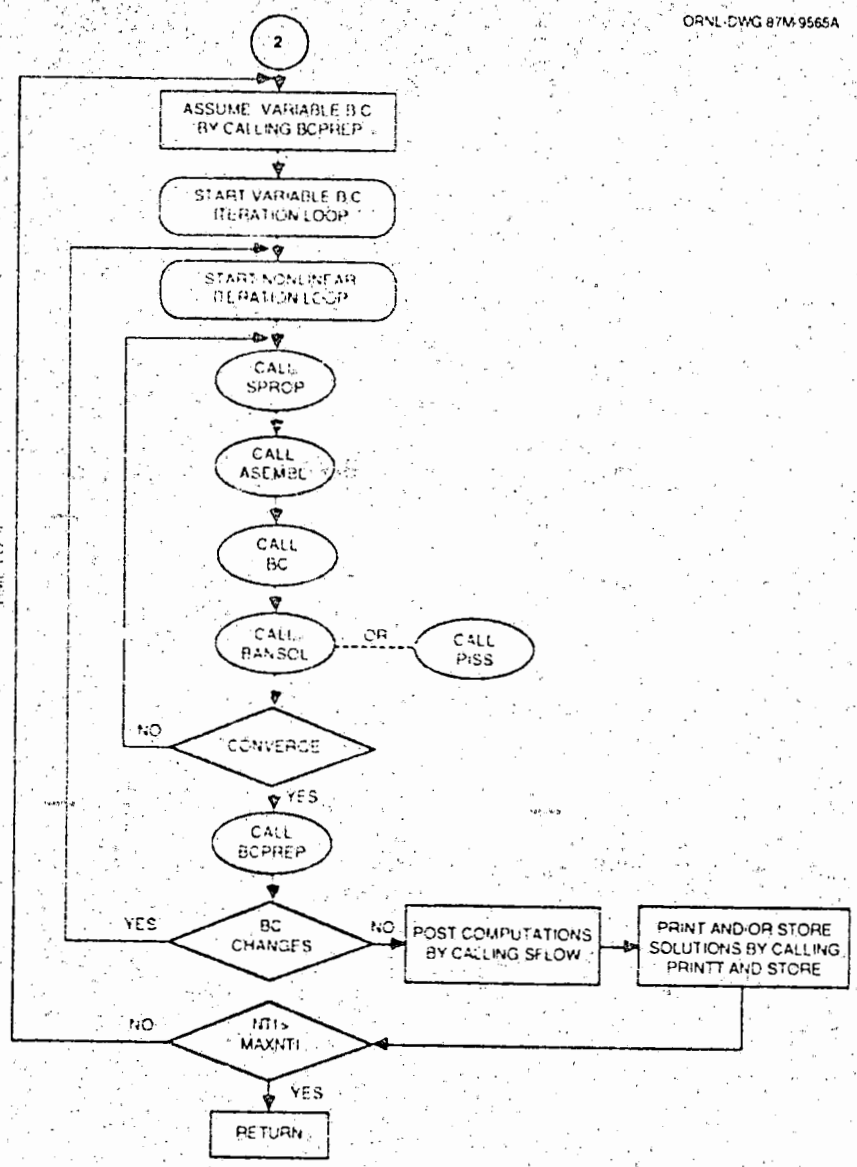


Fig. 8 Flow chart of subroutine GW2DXZ (3 of 3).

Subroutine DATAIN

Subroutine DATAIN reads all data input described in section 5.3 except the card group 1. It also prints all the input information, and calls subroutine SURF to identify the boundary segments 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 mapping from boundary nodes to global nodes are stored in NPB(I) (where NPB(I) is the global node number of i-th boundary node). The element number associated with the boundary sides are stored in NBE. The length and directional cosines for each side are stored in DLB and DCOSXB and DCOSZB, respectively. The local and global nodal number of two nodes of each side are stored in ISB. The information containing in NPB, NBE, ISB, DCOSXB, and DCOSZB along with the number of boundary nodes and the number of boundary sides are 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, 10, 13(d), 13(e), 13(f) of Section 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 integers Data Sets (9), 12(1)(c), 12(2)(c), 13(c), 14(c), 15(c), and 16(c) of Section 5.3

Subroutine LNDGEN

This subroutine is called by the controlling subroutine GW2DXZ to preprocess the pointer array that is needed to assemble the global matrix in compressed form when pointwise solution methods are used. The pointer array automatically generated in this subroutine is the global node connectivity (stencil) GNOJCN(J,N). Here GNOJCN(J,N) is the global node number of J-th node connected to the global node N. This pointer array is generated based on the element connectivity IE(M,J). Here IE(M,J) is the global node number of J-th node of element M.

Subroutine INTERP

This subroutine is called by the subroutine GW2DXZ 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. Two options are available

One is to interpolate the values from the tabular input. The other is to compute the value with analytical functions. When the analytical option is used, the users must supply the functional form.

Subroutine VELT

This subroutine calls Q4D or Q3D 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. Subsequently, it calls Subroutine BANSOL to yield the solution. To save computational time, an option of lumping the matrix is included. The velocity components can thus be solved point by point. The computed velocity field is then returned to GW2DXZ through the argument. This velocity field is also passed to subroutine BCPREP to evaluate the Darcy flux across the seepage-infiltration-evapotranspiration surfaces.

Subroutine Q4D and Q3D

Subroutine Q4D and Q3D are called by the subroutine VELT to compute the element matrices given by:

$$QB(I,J) = \int_{R_e} N_i^e N_j^e \, dR \, , \tag{94a}$$

where N_i^e and N_j^e are the basis functions for nodal point i and j of element e, respectively. Subroutines Q4D and Q3D also evaluate the element load vector:

$$QRX(I) = -\int_{R_e} N_i^e \mathbf{i} \cdot K_s K_r \cdot (\nabla N_j^e) H_j dR \ , \tag{94b}$$

$$QRZ(I) = -\int_{R_e} N_i^e \mathbf{k} \cdot K_s K_r \cdot (\nabla N_j^e) H_j dR \ , \tag{94c}$$

where

H_j = the total head at nodal point j;

$\mathbf{i}$ = the unit vector along the x-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 GW2DXZ to prepare the infiltration-seepage boundary conditions during rainfall period or the seepage-evaportranspiration 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 form 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 Q4 and Q3 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 Q4

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

$$QA(I,J) = \int_{R_e} N_i^e F \, N_j^e dR \, , \tag{95a}$$

$$QB(I,J) = \int_{R_e} (\nabla N_i^e) \cdot K_s K_r \cdot (\nabla N_j^e) \, dR \, , \tag{95b}$$

where F is the soil property function. Q4 is to compute Eq. (95) for quadrilateral elements, while Q3 is to compute those for triangular elements. Subroutines Q4 and Q3 also calculate the element load vector given by:

$$RQ(I) = \int_{R_e} [(\nabla N_i^e) \cdot K_s K_r \cdot (\nabla z) - N_i^e q] \, dR \, , \tag{95c}$$

where q is the source/sink.

Subroutine BASE

This subroutine is called by subroutines Q4D, Q4TH, and Q4 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 added to the load vector. For a Neumann surface, the integrations of the gradient fluxes is added to the load vector. It then calls Q2GF to obtain gravity fluxes. These gravity fluxes are added to load vector. The subroutine BC also implements the variable boundary conditions. First, it checks over all infiltration-evapotransipration-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 over 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 Q2GF

This subroutine is called by the subroutine BC to compute the gravity flux through Neumann boundary given by:

$$RQ(I) = \int_{R_e} N_i^e n \cdot K_s K_r \cdot \nabla z \, dB \ , \tag{96}$$

Subroutine BANSOL

This subroutine is called by the subroutines VELT and GW2DXZ to solve for the matrix equation of the type:

$$[C]\{x\} = \{y\} \ , \tag{97}$$

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 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, the revised FEMWASTE or for plotting. The information stored includes region geometry, subregion data, and hydrological variables such as pressure head, total 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. FRATE(7) is to store the flux through the whole boundary enclosing the region of interest. It is given by

$$\text{FRATE}(7) = \int_B (V_x n_x + V_y n_y + V_z n_z)\, dB \,, \tag{98}$$

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 B_S, and infiltration boundary B_R, respectively, and are given by:

$$\text{FRATE}(1) = \int_{B_D} (V_x n_x + V_y n_y + V_z n_z)\, dB \,, \tag{99a}$$

$$\text{FRATE}(2) = \int_{B_C} (V_x n_x + V_y n_y + V_z n_z)\, dB \,, \tag{99b}$$

$$\text{FRATE}(3) = \int_{B_N} (V_x n_x + V_y n_y + V_z n_z)\, dB \,, \tag{99c}$$

$$\text{FRATE}(4) = \int_{B_S} (V_x n_x + V_y n_y + V_z n_z)\, dB \,, \tag{99d}$$

$$\text{FRATE}(5) = \int_{B_R} (V_x n_x + V_y n_y + V_z n_z)\, dB \,. \tag{99e}$$

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

$$\text{FRATE}(6) = \text{FRATE}(7) - \sum_{I=1}^{5} \text{FRATE}(I) \,. \tag{100}$$

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

$$\text{FRATE}(8) = -\int_R q\, dR \,, \tag{101}$$

and

$$\text{FRATE}(9) = \int_R F \frac{\partial h}{\partial t}\, dR \,. \tag{102}$$

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

$$\text{FRATE(9)} = -\ [\text{FRATE(7)} + \text{FRATE(8)}] \ , \tag{103}$$

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 source/sink is equal to the rate of water accumulated in the region.

Subroutine Q4TH and Q3TH

These subroutines are used to compute the contribution of water content increasing rate from an element e:

$$\text{QTHP} = \int_{R_e} (\alpha' \frac{\theta}{n_e} + \beta' \theta + \frac{d\theta}{dh}) \frac{\partial h}{\partial t} dR \ , \tag{104}$$

where α' and β' are the modified compressibilities of the soil matrix and liquid fluid, respectively, n_e is the effective porosity, and θ is the moisture content. The computation of the above integration is straightforward. Q4Th is used to evaluate the integration for quadrilateral elements and Q3TH for triangular elements.

5. USER'S MANUAL

The following describes what one should do about the revised FEMWATER code for each site-specific applications and data input guide.

5.1 Specifications of Array Dimensions

The subscripted variable arrays have to be dimensioned for each site-specific problem in the MAIN program. Eighty five (85) arrays variables contained between lines MAIN 145 and MAIN 300 in the source code in Appendix A should be dimensioned according to the indicated subscripts (a total of 32). The listing and definitions of these parameters and their associated subscripts (maximum-control numbers) are given below:

<u>Arrays for Node Coordinates and Element Connectivity</u>

X(MAXNP), Z(MAXNP), IE(MAXEL,5);

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

X(NP) = x-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 4, integer to indicate the material type of the M-th element if I is equal to 5.

<u>Arrays to Store the Global Matrix, Global Stencil, and Load Vector</u>

C(MAXNP,MAXBHP), LLNODE(JBAND,MAXNP), R(MAXNP), RI(MAXNP), RL(MAXNP)

JBAND = maximum number of non-zero elements in any row of the matrix,
MAXBHP = maximum number of half bandwidth plus 1.

C(NP,I) = an array to store the assembled globle matrix,
LLNODE(I,NP) = global node number of I-th node connected to node NP,
R(NP) = an array to store the assembled global load vector,
RI(NP) = pressure head iterate in subroutine PISS,
RL(NP) = a working array to contain final solution for the pressure head in subroutine PISS,

Arrays for Hydrological Variables and Nonconvergent Nodes

H(MAXNP), HP(MAXNP), HW(MAXNP), HT(MAXNP),
VX(MAXNP), VZ(MAXNP), NPCNV(MAXNP),
TH(4,MAXEL), DTH(4,MAXEL), AKX(4,MAXEL), AKZ(4,MAXEL),
AKXZ(4,MAXEL);

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,
VZ(NP) - z-component velocity at NP-th node,
NPCNV(NP) - nodal point of NP-th nonconvergent 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,
AKX(I,M) - xx-hydraulic conductivity at I-th Gaussian point of M-th element,
AKZ(I,M) - zz-hydraulic conductivity at I-th Gaussian point of M-th element,
AKXZ(I,M) - xz-hydraulic conductivity at I-th Gaussian point of the M-th element.

Arrays for Boundary Surfaces

DLB(MAXBEL), DCOSX(MAXBEL), DCOSZ(MAXBEL),
NBE(MAXBEL), ISB(4,MAXBEL), NPB(MAXBNP),
BFLX(MAXBNP), BFLXP(MAXBNP);

MAXBEL - maximum number of boundary-element sides,
MAXBNP - maximum number of boundary nodal points;

DLB(I) - length of the I-th boundary element side,
DCOSXB(I) - x-directional cosine of I-th boundary element side,
DCOSZB(I) - z-directional cosine of I-th boundary element side,
NBE(I) - element number to which the I-th boundary element-side belong,
ISB(1,I) - global node number of the first node of I-th boundary element surface,
ISB(2,I) - global node number of the second node of I-th boundary element surface,
ISB(3,I) - element NBE(I)'s local node number of the first node of the I-th boundary element side,
ISB(4,I) - element NBE(I)'s local node number of the second node of the I-th boundary element side,
NPB(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), LES(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 point 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,
LES(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(MXCNP), ISC(2,MXCEL), NPCB(MXCNP);

MXCNP - maximum number of Cauchy nodal points,
MXCEL - maximum number of Cauchy element sides,
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(NP) - type of Cauchy flux vs time profile assigned to NP-th Cauchy boundary node,
NPCB(NP) - global node number of NP-th Cauchy node,
ISC(1,MP) - Compressed Cauchy node number of the first node of MP-th Cauchy boundary side,

ISC(2,MP) = Compressed Cauchy node number of the second node of MP-th Cauchy boundary side,

Arrays for Neumann Boundary Conditions

QNB(MXNPR), QNBF(MXNDP,MXNPR), TQNBF(MXNDP,MXNPR), INTYP(MXNNP), ISN(4,MXNEL), NPNB(MXNNP); DLN(MXNEL), DCOSXN(MXNEL), DCOSZN(MXNEL), MNES(MXNEL)

MXNNP = maximum number of Neumann nodal points,
MXNEL = maximum number of Neumann element sides,
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(NP) = type of Neumann flux vs time profile assigned to NP-th Neumann boundary node,
NPNB(NP) = global node number of NP-th Neumann node,
ISN(1,MP) = Compressed Neumann node number of the first node of MP-th Neumann boundary side,
ISN(2,MP) = Compressed Neumann node number of the second node of MP-th Neumann boundary side,
ISN(3,MP) = Element MNES(I)'s local node number of the first node of MP-th Neumann boundary side,
ISN(4,MP) = Element MNES(I)'s local node number of the second node of MP-th Neumann boundary side,
DLN(MP) = length of the MP-th Neumann element side,
DCOSXN(MP) = x-directional cosine of MP-th Neumann element side,
DCOSZB(MP) = z-directional cosine of MP-th Neumann element side,
MNES(MP) = element number to which the MP-th Neumann element-side belong,

Arrays for Variable Boundary Conditions

DL(MXRSEL), DCOSX(MXRSEL), DCOSZ(MXRSEL), DCYFLX(MXRSNP), FLX(MXRSNP), RSFLX(MXRSNP), HCON(MXRSNP), NRSE(MXRSEL), IS(4,NRSEL), NPRS(MXRSNP), NPCON(MXRSNP), NPFLX(MXRSNP), IRFTYP(MXRSNP), TRF(MXRPAR,MXRFPR), RF(MXRPAR,MXRFPR), RFALL(MXRFPR), HMIN(MXRSNP), NPMIN(MXRSNP)

MXRSEL = maximum number of rainfall-seepage element sides,
MXRSNP = maximum number of rainfall-seepage nodal points,
MXRFPR = maximum number of rainfall profiles,
MXRPAR = maximum number of data point on each rainfall profile;

DL(MP) = length of the MP-th RS element side,
DCOSX(MP) = x-directional cosine of MP-th RS element side,
DCOSZ(MP) = z-directional cosine of MP-th RS element side,
DCYFLX(NP) = Darcy flux through the NP-th RS boundary node,
FLX(NP) = rainfall rate through the NP-th RS boundary node,
RSFLX(NP) = Darcy's velocity at the NP-th RS node,
HCON(NP) = ponding depth allowed for the NP-th RS boundary node,
NRSE(MP) = element number to which the MP-th RS element-side belong,
IS(1,MP) = Compressed RS node number of the first node of the MP-th RS element side,
IS(2,MP) = Compressed RS node number of the second node of the MP-th RS element side,
IS(3,MP) = Element NRSE(MP)'s local node number of the first node of the MP-th RS element side,
IS(4,MP) = Element NRSE(MP)'s local node number of the second node of the MP-th RS element side,
NPRS(NP) = global node number of the NP-th RS node,
NPCON(NP) = ponding depth Dirichlet boundary condition indicator for the NP-th variable node,
NPFLX(NP) = Cauchy boundary condition indicator for the NP-th variable node,
IRFTYP(MP) = type of rainfall profile assigned to MP-th RS boundary side,
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,
RFALL(I) = value of I-th rainfall vis time profile at present time,
HMIN(NP) = minimum pressure head allowed for the NP-th RS boundary node,
NPMIN(NP) = minimum pressure head Dirichlet boundary condition indicator for the NP-th RS boundary node,

<u>Arrays for Dirichlet Boundary Conditions</u>

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

MXRDP = 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,

Arrays for Material Property Array and Material Characteristics

PROP(MXMPPM,MAXMAT), THPROP(MXSPPM,MAXMAT), AKPROP(MXSPPM,MAXMAT); CAPROP(MXSPPM,MAXMAT), HPROP(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 = modified compressibility of the media,
J = 2 = modified compressibility of the water,
J = 3 = effective porosity,
J = 4 = saturated xx-hydraulic conductivity or permeability,
J = 5 = saturated zz-hydraulic conductivity or permeability,
J = 6 = saturated xz-hydraulic conductivity or permeability,

THPROP(J,I) = Moisture content of J-th data point on the soil characteristic curves for I-th material,
AKPROP(J,I) = Relative hydraulic conductivity of J-th data point on the soil characteristic curves for I-th material.
CAPROP(J,I) = Water capacity of J-th data point on the soil characteristic curves for I-th material,
HPROP(J,I) = Pressure head of J-th data point on the soil characteristic curves 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:

```
Real Storage = [(12 + JBAND)*MAXNP] + (20*MAXEL)
     + (3*MAXBEL + 2*MAXBNP) + [(1 + 2*MXSDP)*MXSPR]
     +  [(1 + 2*MXWDP)*MXWPR] + [(1 + 2*MXCDP)*MXCPR]
     + [(1 + 2*MXNDP)*MXNPR + 3*MXNEL]
     + [(1 + 2*MXRPAR)*MXRFPR + 5*MXRSNP + 3*MXRSEL]
     + [(1 + 2*MXDDP)*MXDPR] + [(MXMPPM + 4*MXSPPM)*MAXMAT]        (105)
```

```
Integer Storage = [(1 + JBAND)*MAXNP] + (5*MAXEL) + (5*MAXBEL + 3*MAXBNP)
     + [(2*MXSEL) + (2*MXWNP)] + (2*MXCES + 2*MXCNP)
     + (5*MXNEL + 2*MXNNP) + (5*MXRSEL + 5*MXRSNP)
     + (2*MXDNP) + (2*MAXNTI + MXNDTC)                             (106)
```

The subscripts (maximum-control numbers) of the above 85 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 that 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 will demonstrate how to dimension the above 85 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 61 x 51 nodes and 60 x 50 elements. In other words, we are discretizing the region with 51 nodes along the longitudinal or x-direction, and 51 nodes along the vertical or z-direction. Since we have a total of 61 x 51 = 3,111 nodes, the maximum number of nodes is MAXNP = 3111. The total number of elements is 60 x 50 = 3,000, i.e. MAXEL = 3000. For this simple discretization problem, the maximum connecting number to any of the 3,111

nodes in the region of interest is 9, i.e., JBAND = 9. There will be 60 element-sides each on the right and left sides and 50 element-sides each on the top and bottom sides of the region. Thus, there will be a total of 220 element-sides, i.e., MAXBEL = 220. Similarly, we can compute the boundary nodes to be 220, i.e., MAXBNP = 220. The half bandwidth plus 1 is equal to 53, i.e., MAXHBP = 53. With these maximum-control numbers specified, the arrays related to the global region, global boundary, and hydrologic variables can be dimensioned as:

```
DIMENSION X(3111),Z(3111),IE(3000,5)
DIMENSION C(3111,53),LLNODE(9,3111),R(3111),RI(3111),RL(3111)
DIMENSION H(3111),HP(3111),HW(3111),HT(3111),VX(3111),VZ(3111)
DIMENSION NPCNV(3111), TH(4,3000),DTH(4,3000)
DIMENSION AKX(4,3000),AKZ(4,3000),AKXZ(4,3000)
DIMENSION DLB(220),DCOSXB(220),DCOSZB(220),ISB(4,220),NBE(220)
DIMENSION NPBB(220),BFLX(220),BFLXP(220)
```

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 distinguish point source/sink profiles. Then we will have MXSPR = 3 and MXWPR = 5. Let us further assume four data points are needed to describe the distributed source/sink profiles as function of time and 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),LES(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 side is a variable boundary, i.e., on the air-soil interface either the ponding or the infiltration or the evapotranspiration may take place. On the left side, fluxes from the adjacent aquifer are known. On the right side, the total head is assumed known. On the bottom side, natural drainage is assumed to occur, i.e., the gradient of the pressure head can be assumed zero.

There are 51 nodes on the left face and 50 element-sides; thus MXCNP = 51 and MXCEL = 50. It is further assumed that there two different fluxes going into the region through the left face and each flux can be described by four data points as 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 NPCB(51),ICTYP(51), ISC(2,50)
```

On the bottom side, there are 61 nodes and 60 element sides. 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 = 61, MXNEL = 60, 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 NPNB(61),INTYP(61), ISN(4,60)
DIMENSION DLN(60),DCOSXN(60),DCOSZN(60),MNES(60)
```

On the top side, there will be 61 nodes and 60 element sides. Let us assume there are three different rainfall intensities that might be fall on the air-soil interface, and each rainfall intensity is a function of time and can be described by 24 data points. With these descriptions, we have MXRSNP = 61, MXRSEL = 60, MXRFPR = 3, and MXRPAR = 24; and the variable boundary condition arrays can be specified as:

```
DIMENSION RFALL(3),RF(24,3),TRF(24,3)
DIMENSION NPRS(61),IRFTYP(61), IS(4,60)
DIMENSION DL(60),DCOSX(60),DCOSZ(60),NRSE(60),RSFLX(61),DCYFLX(61)
DIMENSION HCON(61),FLX(61),HMIN(61),NPCON(61),NPFLX(61),NPMIN(61)
```

On the right side, there are 51 nodes. Let us assume that each of these nodes have the same values of the total head. We further assume that this total head can be described by 4 data points as function of time. We then have MXDNP = 51, MXDPR = 1, and MXDDP = 4; and the Dirichlet boundary condition arrays can be dimensioned as:

```
DIMENSION HDB(1),HDBF(4,1),THDBF(4,1)
DIMENSION IDTYP(51),NPDB(51)
```

In this report, we have six material properties 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 52 data points. We then have MAXMAT = 3,

MXMPPM = 6, and MXSPPM = 52; and the material property arrays can be dimensioned as:

```
DIMENSION PROP(6,3)
DIMENSION THPROP(52,3),AKPROP(52,3),HPROP(52,3),CAPROP(52,3)
```

If we assume that we will make a 500 time step simulation and we will re-initiate 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,MAXBEL,MAXBNP,MAXHBP/3000,3111,220,220,53/
DATA JBAND/9/
DATA MAXNTI,MXNDTC/500,20/
DATA MXSEL,MXSPR,MXSDP,MXWNP,MXWPR,MXWDP/11,3,4,10,5,8/
DATA MXCNP,MXCEL,MXCPR,MXCDP/51,50,2,4/
DATA MXNNP,MXNEL,MXNPR,MXNDP/61,60,1,2/
DATA MXRSEL,MXRSNP,MXRFPR,MXRPAR/60,61,3,24/
DATA MXDNP,MXDPR,MXDDP/51,1,4/
DATA MAXMAT,MXSPPM,MXMPPM/3,52,6/
```

If we use the point iteration method to solve the matrix equation in stead of the direct band matrix solver, we should set MAXHBP equal to JBAND.

9 2 1 2 4 5 8 0 5 5 0

5.2 Soil Function Specifications

The revised FEMWATER provides two options to handle the functional relationships of moisture content, water capacity, and relative hydraulic conductivity with the pressure head. One is the tabular input while the other is the analytic function specification. If the former option is used, the user may ignore this section. If the latter is used, the user must supply three functions to compute the moisture content, hydraulic conductivity, and water capacity based on the current value of pressure head. The parameters needed to specify these functional forms are read and stored in AKPROP and THPROP arrays. One example is shown in the subroutine SPROP between line SPRO 635 and SPRO 780 in Appendix A when KSP = 0. In this example, the water content, water capacity, and relative hydraulic conductivity are given by (van Genuchten, 1980):

$$\theta = \theta_r + \frac{\theta_s - \theta_r}{[1 + (\alpha h)^n]^m} \tag{107}$$

$$\frac{d\theta}{dh} = \alpha(n-1)\ [1 - f(\theta)]^m\ [f(\theta)]\ (\theta_s - \theta_r) \tag{108}$$

$$K_r = [(\theta - \theta_s)/(\theta_s - \theta_r)]^{1/2}\ \{1 - [1 - f(\theta)]^m\}^2 \tag{109}$$

in which

$$f(\theta) = [(\theta - \theta_r)/(\theta_s - \theta_r)]^{1/m} \tag{110}$$

and

$$m = 1 - n \tag{111}$$

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. (107) through (111), let us assume that the following Fermi types of functions are used to represent the unsaturated hydraulic properties (Yeh, 1987a):

$$\theta = \theta_r + (\theta_s - \theta_r)/\{1 + \exp[-\alpha(h-h_\theta)]\} \tag{112}$$

$$d\theta/dh = \alpha(\theta_s-\theta_r)\exp[-\alpha(h-h_\theta)]/\{1 + \exp[-\alpha(h-h_\theta)]\}^2 \tag{113}$$

and

$$\log_{10}(K_r) = \epsilon/\{1 + \exp[-\beta(h-h_k)]\} - \epsilon \tag{114}$$

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 635 and SPRO 780 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,NQ
C
C -------- SATURATED CONDITION
C
      IF(HNP.GT.0.0) GO TO 700
      TH(KG,M)=WCS
      DTH(KG,M)=0.0D0
      AKX(KG,M)=SATKX
      AKZ(KG,M)=SATKZ
      AKXZ(KG,M)=SATKZ
      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
      RK=10.0D0**AKRLOG
      AKX(KG,M)=RK*SATKX
      AKZ(KG,M)=RK*SATKZ
      AKXZ(KG,M)=RK*SATKXZ
C
```

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.

   FORMAT(I5,9A8,3I1)

NPROB	TITLE	IMOD	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.

IMOD = Integer indicating if the simulation is to be made, 0 = no, 1 = yes

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
Two cards per problem.

Card 1 - FORMAT(16I5)

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

NMPPM	KSTR	KCP	KGRAV	NSTRT	NITER	NCYL	NDTCHG
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.

KSP = Soil property input control,
0 = analytical input,
1 = Tabular data input.

NSPPM = Number of points in tabular soil property functions or number of parameters to specify analytical soil functions per material.

NMPPM - No. of material properties per material.

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

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

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

NSTRT - Number of logical records to be read via auxiliary storage device for restarting calculation, 0 - no restart.

NITER - Number of iterations allowed for solving the non-linear equation.

NCYL - No. of cycles permitted for iterating rainfall-seepage boundary conditions per time step.

NDTCHG - No. of times to reset time step size to initial time step size.

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.

Card 2 - FORMAT(16I5)

ILUMP	IMID	IPNTS	NPITER	IVML	
5	10	15	20	25	80

ILUMP - Is mass lumping? 0 - no, 1 - yes.

IMID - Is mid-difference? 0 - no, 1 - yes.

IPNTS - Is pointwise iterative method used?
0 - No, 1 - Yes.

NPITER - No. of iterations for pointwise solution.

IVML = Is velocity-solving matrix to be lumped?
0 = No, 1 = yes.

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	FE	TOLA	TOLB	RHO
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).

FE = Angle between coordinate axes and the horizontal-vertical axes, (degree)

TMAX = Maximum simulation time, (T).

TOLA = Steady-state convergence criteria, (L).

TOLB = Transient-state convergence criteria, (L).

RHO = Density of water, (M/L**3).

Card 2 - FORMAT(3D10.3).

GRAV	VISC	W	OME	OMI	CYLIND	
10	20	30	40	50	60	80

GRAV = Acceleration of gravity, (L/T**2).

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.

9 2 1 2 4 5 8 0 5 5 6

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

CYLIND - Cylinder coordinate? 0.0 - no, 1.0 - yes.

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 of cards is [(NTI/80+1)*2 + (NDTCHG/8+1)]. (NTI/80+1) cards 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)

KPR0	KPR(1)	KPR(2)	---	KPR(I)	---	KPR(NTI)
1	2	3				80

KPR0 - 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 Darcy velocity.

KPR(I) - Printer control for I-th time step similar to KPR0.

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

KDSK0	KDSK(1)	KDSK(2)	---	KDSK(I)	---	KDSK(NTI)
1	2	3				80

KDSK0 - 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 KDSK0.

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(3,NMAT)	
10	20		60	80

PROP(1,I) - Modified coefficient of compressibility of the medium I, (1/L).

PROP(2,I) - Modified coefficient of compressibility of water in medium I, (1/L).

PROP(3,I) - Effective porosity of medium I, (dimensionless in decimal point).

PROP(4,I) - xx-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L**2).

PROP(5,I) - zz-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L**2).

PROP(6,I) - xz-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L**2).

6A. ANALYTICAL SOIL PARAMETERS

These cards are input if and only if KSP = 0. 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.

6B. SOIL PROPERTIES IN TABULAR FORM

These cards are input if and only if KSP .NE. 0. Four sets of cards per material -- one each for pressure, water-content, relative conductivity (or relative permeability), and water capacity, respectively. The number of cards per set is determined by the input parameters, NSPPM and NMAT:

FORMAT(8D10.3)

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

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

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

9 2 1 2 4 5 8 0 5 6 0

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

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

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

HPROP(J,K) = Tabular values of pressure head of J-th point for material K, (L).

THPROP(J,K) = Tabular values of moisture-content of J-th point for material K, (L**3/L**3).

AKPROP(J,K) = Tabular values of relative conductivity (or relative permeability) of J-th point for material K, (Decimal point).

CAPROP(J,K) = Tabular values of specific moisture-content capacity of J-th point for material J, (1/L).

7. NODAL POINT COORDINATE

Usually a total of 2*NNP cards are required, NNP cards for the x-coordinate and NNP cards for the z-coordinate. However, if a group of subsequent nodes appear in regular pattern, automatic generation can be made.

(a) Cards for x-Coordinate

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

NI	NSEQ	NAD		XNI	XAD	XRD	
5	10	15	20	30	40	50	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).

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

XRD = Percentage of the increase of the increment over its preceding increment, (Decimal point); If XRD .EQ. 0, all increments XAD's are the same. If XRD .GT. 0, the first increment is XAD*(1+XRD), the second increment is XAD*(1+XRD)**2, the third increment is XAD*(1+XRD)**3, and so on.

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

(b) Cards for z-Coordinate

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

NI	NSEQ	NAD		ZNI	ZAD	ZRD	
5	10	15	20	30	40	50	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.

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

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

ZRD = Percentage of the increase of the increment over its preceding increment, (Decimal point); If ZRD .EQ. 0, all increments ZAD's are the same. If ZRD .GT. 0, the first increment is ZAD*(1+ZRD),

the second increment is ZAD*(1+ZRD)**2, the third increment is ZAD*(1+ZRD)**3, and so on.

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

8. ELEMENT INCIDENCES

Usually a total of NEL cards are needed. However, if a group of elements appear in regular pattern, automatic generation is made by filling the gap between element numbers.

FORMAT(16I5)

MI	IE(MI,1)	IE(MI,2) ---	IE(MI,5)	MODL	NLAY	
5	10	15	30	35	40	80

MI = Global element number.

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) = Material type to be applied to element MI.

MODL = Number of elements in the direction of most rapidly numbered nodes.

NLAY = Number of elements in the direction of least rapidly numbered nodes.

IE(MI,1) -- IE(MI,4) are numbered beginning with the lower left corner and progressing around the element in a counterclockwise direction. For a rectangular block of elements, it is only necessary to specify the first element, the width MODL and the length NLAY, where MODL and NLAY are measured in elements. Figure 9 provides an example. The object is considered to be rectangular since it has width MODL = 3 on two opposite sides and length NLAY = 5 on the

other two opposite sides. To generate definitions of elements 2 through 15 automatically, including both the incidence and material type, only one card is necessary:

1	1	5	6	2	1	3	5	
5	10	15	20	25	30	35	40	80

Although all elements of this example will be assumed to contain the same material type, MTYP = 1, this situation can easily be changed by using material-correction facility.

ORNL-DWG 81-3924 ESD

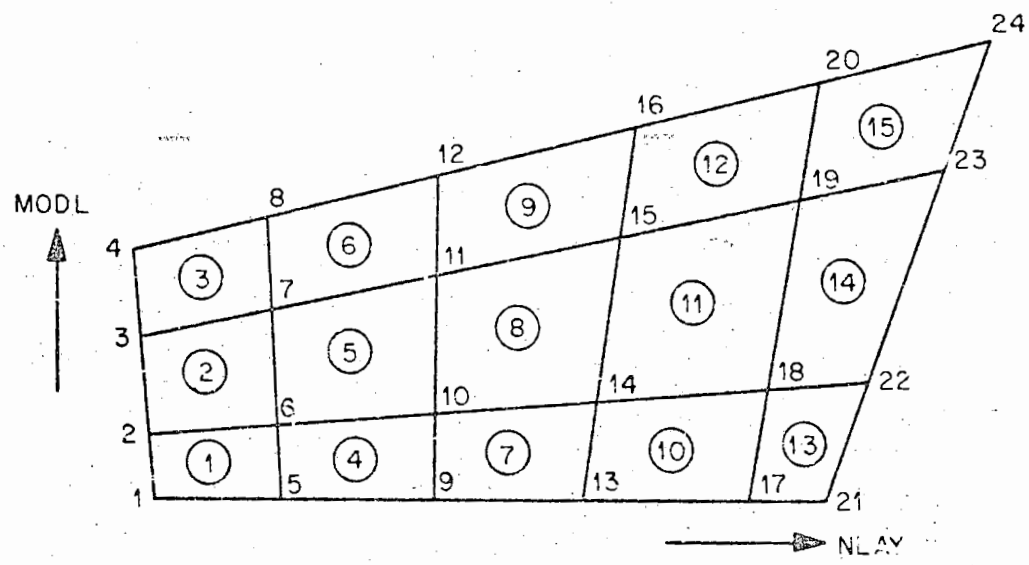


Fig. 9 Numbering of element incidences

9. MATERIAL TYPE CORRECTION
This data set is required only if NCM > 0. Normally, NCM cards are required. However, if a group of elements appear 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.

10. CARD INPUT FOR INITIAL OR PRE-INITIAL CONDITIONS
NNP cards, one card for each node, are needed. However, if a group of nodes appear 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 AND RESTARTING: The initial condition for a transient calculation may be obtained in three different ways: from card input, auxiliary storage input, or steady-state calculation using a 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. Auxiliary storage input is necessary whenever the re-starting facility is being used. That is, pressure distribution for NSTRT different times have been generated and written on disk or magnetic tape. If NSTRT > 0, these distribution will be read from Logical Unit 2, and NSTRT-th distribution will be used as the initial condition for current calculation. If KSTR is greater than zero, the pressure values will be written on a different device as they are read out so that a complete record of the calculations may be kept on one device, Logical Unit 1. If either the first (card input) or the last (steady-state) options are desired, then NSTRT = 0.

NOTE ON AUXILIARY STORAGE UNITS: Logical Unit 1 is used to store output if KSTR > 0, and Logical Unit 2 is used for input NSTRT > 0. Proper identification of these two units must be made in the JCL if either of these two options is used.

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.

9 2 1 2 4 5 8 0 5 6 6

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

FORMAT(16I5)

NSEL	NSPR	NSDP	NRSEL	NRSN	NRFPR	NRPAR	NDNP
5	10	15	20	25	30	35	40

NDPR	NDDP	NCNP	NCEL	NCPR	NCDP	NNNP	NNEL
45	50	55	60	65			80

NNPR	NNDP	NWNP	NWPR	NWDP			
5	10	15	20	25			80

NSEL = No. of source/sink elements.

NSPR = No. of source/sink profiles

NSDP = No. of data points in each of the NSPR source/sink profiles

NRSEL = No. of rainfall-seepage element sides.

NRSN = No. of rainfall-seepage nodes.

NRFPR = No. of rainfall profiles.

NRPAR = No. of rainfall parameters (data points) in each of the NRFPR rainfall profiles.

NDNP = No. of Dirichlet nodal points, should be .GE. 1.

NDPR = No. of total Dirichlet-head profiles, should be .GE. 1.

NDDP = No. of data points in each of the NDNP total Dirichlet-head profiles, should be .GE. 1.

NCNP = No. of Cauchy nodal points.

NCEL = No. of Cauchy boundary element sides.

NCPR = No. of Cauchy-flux profiles.

NCDP = No. of data points in each of the NCPR Cauchy-flux profiles.

9 2 1 2 4 5 8 0 5 6 7

NNNP = No. of Neuamnn nodal points.

NNEL = No. of Neumann boundary element sides.

NNPR = No. of Neumann flux profiles.

NNDP = No. of data points in each of the NNPR Neumann-flux profiles.

NWNP = No. of well or point source/sink nodal points.

NWPR = No. of well or point source/sink strength profiles.

NWDP = No. of data points in each of the NWPR profiles.

12. DISTRIBUTED AND POINT SOURCES/SINKS
Two groups of cards are needed: one group for distributed source/sink and the other for well (point source/sink).

(1) Distributed Source/Sink: This group of data is needed if and only if NSEL .GT. 0.

(a) Sources/sinks Profiles - 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)

First 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**2/L).

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

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. 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) Well (Point) Source/sink: This group of data is needed if and only if NWNP .GT. 0.

(a) Well or point Source/sink Profiles - Number of cards depends on NWPR and NWDP. This sub-data set is read in similar to 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/L).

(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 sub-data set.

9 2 1 2 4 5 8 0 5 7 2

13. RAINFALL/EVAPORATION-SEEPAGE (RES) BOUNDARY CONDITIONS
Six groups of cards are required for this data set if and only if NRSEL .GT. 0. The first group is used to specify the rainfall/evaporation profile, the second group is used to read-in the global nodal numbers of all NRSN RES nodes, the third group is used to assign the type of rainfall/evaporation profile to each of the NRSN RES nodes, the fourth group is used to read-in the ponding depth for each of the NRSN RES nodes, the fifth group is used to read-in the allowed minimum pressure for each of the NRSN RES nodes, and the sixth group is used to specify the NRSEL RES boundary element sides.

(a) Rainfall/Potential Evaporation Profiles - This subdata set is read in similar to that of 12 (1)(a).

FORMAT(8D10.3)
First profile

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

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

Second profile

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

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

I-th profile

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

...	TRF(NRPAR,I)	RF(NRPAR,I)

80

.
.
.

NRFPR-th profile

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

.
.

... TRF(NRPAR,NRFPR)	RF(NRPAR,NRFPR)

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) Global nodal number of all rainfall/evaporation-seepage (RES) nodes - This subdata set is read in similar to that in 12 (1)(b).

FORMAT(5I5)

NI	NSEQ	NIAD	NODE	NODEAD	
5	10	15	20	25	80

NI = Compressed RES nodal number of the first node in a sequence.

NSEQ = NSEQ subsequent RES nodes will be generated automatically.

NIAD = Increment of the compressed RES nodal number for each of the NSEQ subsequent 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 used to signal the end of this subdata set.

(c) Rainfall/evaporation type assigned to each of all RES nodes - This subdata set is read in similar to that in 12(1)(c).

FORMAT(5I5)

NI	NSEQ	NIAD	NITYP	NTYPAD
5	10	15	20	25 ... 80

NI = Compressed RES nodal number of the first node in a sequence.

NSEQ = NSEQ subsequent nodes will be generated automatically.

NIAD = Increment of the compressed RES nodal number for each of the NSEQ subsequent nodes.

NITYP = Type of rainfall/evaporation profile assigned to node NI.

NTYPAD = Increment of the type of rainfall/evaporation profile for each of the NSEQ subsequent nodes.

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

(d) Ponding depth - Normally, NRSN cards are needed, one for each of the NRSN RES nodes. However, if a group of 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 RES nodal number of the first node in a sequence.

NSEQ = NSEQ subsequent nodes will be generated

automatically.

NIAD = Increment of the compressed RES 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.

(e) Minimum pressure head allowed in each RES nodes - This subdata set is read-in similar to the above subdata set.

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

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

NI = Compressed RES nodal number of the first node in a sequence.

NSEQ = NSEQ subsequent nodes will be generated automatically.

NIAD = Increment of the compressed RES nodal 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.

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

FORMAT(16I5)

MI	NSEQ	M	IS1	IS2	MIAD	MAD	IS1AD	IS2AD	
5	10	15	20	25	30	35	40	45	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.

M = Global element number to which MI-th element-side belong to.

IS1 = Compressed RES nodal number of the first node of element side MI.

IS2 = Compressed RES nodal number of the second node of element side MI.

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

MAD = Increment of M for each of the NSEQ subsequent RES element sides.

IS1AD = Increment of IS1 for each of the NSEQ subsequent RES element sides.

IS2AD = Increment of IS2 for each of the NSEQ subsequent RES element sides.

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

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

(a) Dirichlet-head Profiles - This sub-data set is read in similar to that in sub-data 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 sub-data 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, 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 sub-data set.

9 2 1 2 4 5 8 0 5 7 9

15. CAUCHY BOUNDARY CONDITIONS

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

(a) Prescribed Cauchy-flux Profiles - This subdata set is read in similar to that in 12(1)(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) 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.

(c) Type of Cauchy flux profile assigned to Cauchy Nodes - Normally, one card per Cauchy node, i. e. a total of NCNP cards is required. However, if a group of cards appears in regular pattern, automatic generation may be made.

FORMAT(5I5)

NI	NSEQ	NAD	NITYP	NTYPAD
5	10	15	20	25	80

NI = Compressed Cauchy nodal number of the first node in a sequence.

NSEQ = NSEQ subsequent Cauchy nodes will be generated automatically.

NAD = Increment of NI for each of the NSEQ subsequent nodes.

NITYP = Type of Cauchy-flux profile assigned to node NI.

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

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

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

FORMAT(16I5)

MI	NSEQ	IS1	IS2	MIAD	IS1AD	IS2AD	
5	10	15	20	25	30	35	80

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

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

IS1 = Compressed Cauchy nodal number of the first node on the element-side MI.

IS2 = Compressed Cauchy nodal number of the second node on the element side MI.

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

IS1AD = Increment of IS1 for each of the NSEQ subsequent element-sides.

IS2AD = Increment of IS2 for each of the NSEQ subsequent element-sides.

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

16. NEUMANN BOUNDARY CONDITIONS
This data set is required if and only if NNNP > 0.

(a) Prescribed Neumann Flux Profiles - This sub-data set is read in similar to that in sub-data set 12(1)(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)	---
10	20	30	40	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) Neumann Nodes - The number of cards needed in this sub-data 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.

(c) Type of Neumann flux profile assigned to Neumann Nodes - Normally, one card per Neumann node, i. e., a total of NNNP cards is required. However, if a group of cards appears in regular pattern, automatic generation may be made.

FORMAT(4I5)

NI	NSEQ	NAD	NITYP	NTYPAD	
5	10	15	20	25	80

NI = Compressed Neumann node number of the first node in the sequence.

NSEQ = NSEQ subsequent Neumann nodes will be applied to by Neumann flux of type NITYP.

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

NITYP - Type of Neumann flux profile for node NI and its 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 sub-data set.

(d) Neumann Boundary Element Sides - Normally one card per Neumann boundary-element side, i.e., a total of NNEL cards is required.

However, if a group of Neumann boundary-element sides appear in regular pattern, automatic generation may be made.

FORMAT(16I5)

MI	NSEQ	M	IS1	IS2	MIAD	MAD	IS1AD	IS2AD	
5	10	15	20	25	30	35	40	45	80

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

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

M - Global element number to which the MI-th element side belong.

IS1 - Compressed Neumann node number of the first node in the element-side MI.

IS2 - Compressed Neumann node number of the second node in the element-side MI.

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

MAD - Increment of M for each of the NSEQ subsequent element sides.

IS1AD - Increment of the first compressed Neumann node number of element-side MI.

IS2AD - Increment of the second compressed Neumann node number of element-side MI.

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

17. 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.

9 2 1 2 4 5 8 0 5 8 6

6. SAMPLE PROBLEMS

To test the consistency of the revised FEMWATER, three illustrative examples reported elsewhere (Yeh and Ward, 1980; Huyakorn 1986; Yeh 1987b) are presented. The first two examples represent simple one- and two-problems, respectively (Huyakorn 1986; Yeh 1987b). The third example is the seepage pond problem reported in the original FEMWATER (Yeh and Ward 1980).

6.1 One-dimensional Column Problem.

This example is selected to represent the simulation of a one-dimensional problem with the revised FEMWATER. The column is 200 cm long and 50 cm wide (Fig. 10). The column is assumed to contain the soil with a saturated hydraulic conductivity of 10 cm/day, 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:

$$\theta = \theta_s - (\theta_s - \theta_r)\left(\frac{1}{h_b - h_a}\right) \tag{115}$$

and

$$K_r = \frac{\theta - \theta_r}{\theta_s - \theta_r} \tag{116}$$

where h_b and h_a are the parameters used to computer the water content and the relative hydraulic conductivity.

ORNL-DWG 87-9750

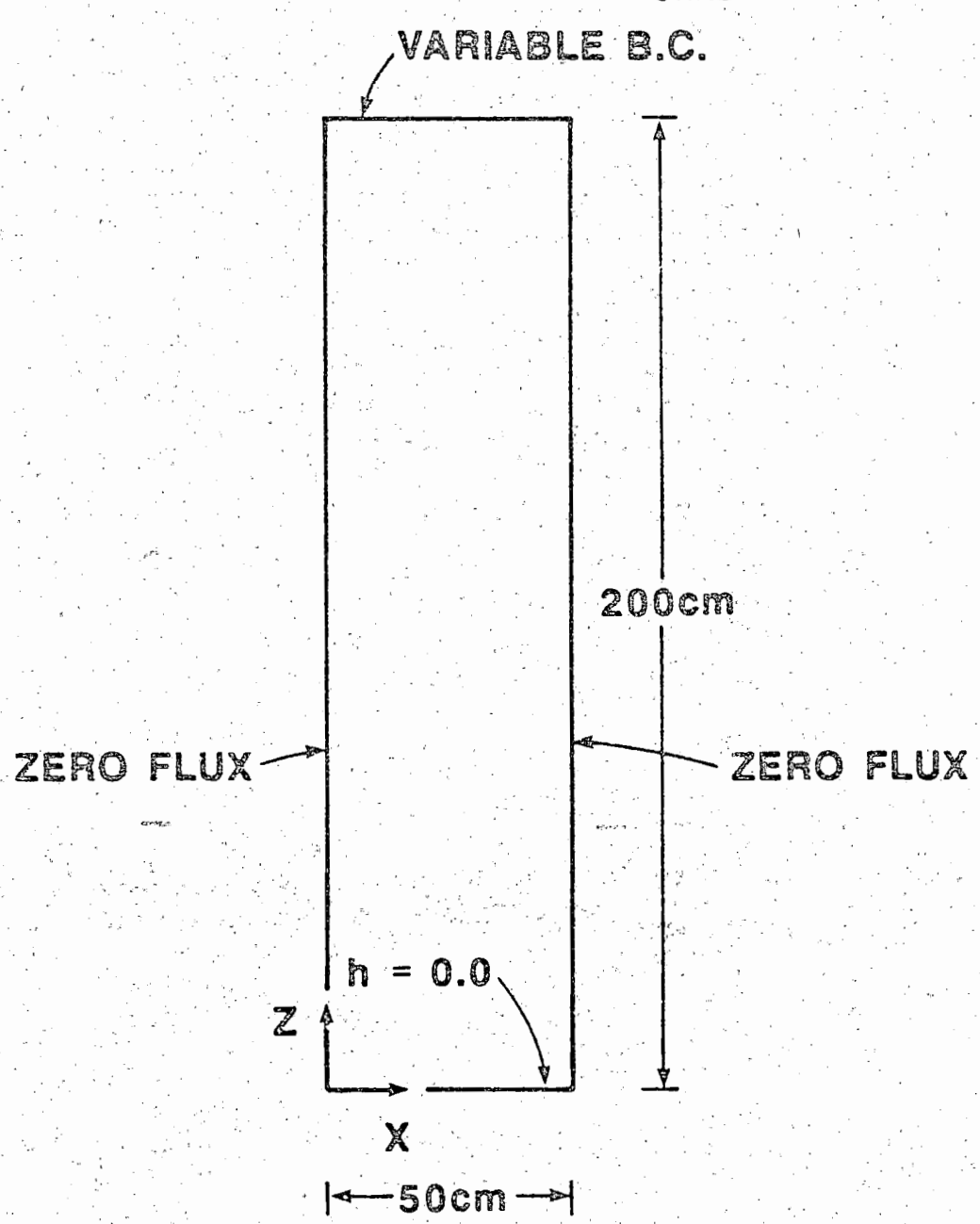


Fig. 10 Problem definition for the one-dimensional transient flow in a soil column.

The initial conditions are assumed: a pressure head of -90 cm is 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 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/day for the first ten days and a potential evaporation of 5 cm/day for the second ten days.

The region of interest, i. e. the whole column, will be discretized with 1 x 40 = 40 elements with element size = 50 x 5 cm, resulting in 2 x 41 = 82 node points. 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 day. Because there is a abrupt change in the flux value from 5 cm/day (infiltration) to -5 cm/day (evaporation) imposed on the top surface at day 10, the time step size is automatically reset to 0.05 day on tenth day. A 20 day simulation will be made with the revised FEMWATER. With the time step size described above, 44 time steps will be needed.

The pressure head tolerance for nonlinear iteration is $2 \cdot 10^{-2}$ cm, and the relaxation factors is set equal to 0.5. With the above descriptions, the input data can be prepared following the instruction of Section 5.3 Data Input Guide. The input data is given in Table 1.

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

```
  1  ONE-D COLUMN  INFILTRATION-EVAPORATION WITH FEMWATER;L=CM, T=DAY, M=G 111
 82   40    1    0   44    1    0    4    6    1    0    1    0   50   20    3
  0    0    0  300    1
  0.05D0     0.2D0      1.0D0     20.0D0     0.0D00     2.0D-2     2.0D-2     1.0D0
 7.316D12  1.1232D2     1.0D0      0.5D0      0.5D0
555151511151115115111155551515111511151151111 5
111010100010001001000011101010001000100100001
  1.0D01  2.0000D1     1.0D38
   0.0D0     0.0D0      0.3D0      0.0D0     10.0D0      0.0D0
 0.150D0   0.450D0    0.00D0     -1.0D2                            THPROP
 0.000D0   0.000D0    0.00D0      0.0D0                            AKPROP
  1   40    2          0.0D0      0.0D0
  2   40    2          5.0D0      0.0D0
                                                            END OF X-CORD
  1   40    2          0.0D0      5.0D0
  2   40    2          0.0D0      5.0D0
                                                            END OF Z-CORD
  1    1    2    4     3    1    1   40
  1    1    1          0.0D0      0.0D0
  3   77    1         -9.70D1     0.0D0
 81    1    1         -9.00D1     0.0D0
                                                                END OF IC
  0    0    0    1     2    1    4    2    1    2    0    0    0    0    0    0
  0    0    0    0     0
   0.0D0     5.0D0     10.0D0      5.0D0   10.001D0    -5.0D0     1.0D38    -5.0D0
  1    1    1   81     1
                                  END OF NPVB
  1    1    1    1     0
                                    END OF IRTYP
  1    1    1          0.0D0      0.0D0
                                                        END OF HCON
  1    1    1         -90.0D0     0.0D0
                                                        END OF HMIN
  1    0   40    1     2    0    0    1    1
                                                    END OF ISV(J,I) J=1,2
   0.0D0     0.0D0     1.0D38      0.0D0
  1    2
  1    1    1    1     0
                                                            END OF IDTYP
                        END OF JOB
```

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

```
DIMENSION X(82),Z(82),IE(40,5),LLNODE(9,82)
DIMENSION C(82,4),R(82),RL(82),RI(82)
DIMENSION H(82),HP(82),HW(82),HT(82),VX(82),VZ(82)
DIMENSION NPCNV(82), TH(4,40),DTH(4,40)
DIMENSION AKX(4,40),AKZ(4,40),AKXZ(4,40)
DIMENSION DLB(82),DCOSXB(82),DCOSZB(82),NBE(82),ISB(4,82)
DIMENSION BFLX(82),BFLXP(82),NPB(82)
DIMENSION SOS(1),SOSF(1,1),TSOSF(1,1),ISTYP(1),LES(1)
DIMENSION WSS(1),WSSF(1,1),TWSSF(1,1),IWTYP(1),NPW(1)
DIMENSION RFALL(1),RF(4,1),TRF(4,1)
DIMENSION NPRS(2),IRFTYP(2), IS(4,1)
DIMENSION DL(1),DCOSX(1),DCOSZ(1),NRSE(1),RSFLX(2),DCYFLX(2)
DIMENSION HCON(2),FLX(2),HMIN(2),NPCON(2),NPFLX(2),NPMIN(2)
DIMENSION HDB(1),HDBF(2,1),THDBF(2,1)
DIMENSION IDTYP(2),NPDB(2)
DIMENSION QCB(1),QCBF(1,1),TQCBF(1,1)
DIMENSION NPCB(1),ICTYP(1), ISC(2,1)
DIMENSION QNB(1),QNBF(1,1),TQNBF(1,1)
DIMENSION NPNB(1),INTYP(1), ISN(4,1)
DIMENSION PROP(6,1)
DIMENSION THPROP(4,1),AKPROP(4,1),HPROP(4,1),CAPROP(4,1)
DIMENSION KPR(44),KDSK(44),TDTCH(3)
```

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

```
DATA MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI/40,82,82,82,4,44/
DATA JBAND/9/
```

```
      DATA MXNDTC/3/
      DATA MXSEL,MXSPR,MXSDP,MXWNP,MXWPR,MXWDP/1,1,1,1,1,1/
      DATA MXRSEL,MXRSNP,MXRFPR,MXRPAR/1,4,1,4/
      DATA MXDNP,MXDPR,MXDDP/4,1,2/
      DATA MXCNP,MXCEL,MXCPR,MXCDP/1,1,1,1/
      DATA MXNNP,MXNEL,MXNPR,MXNDP/1,1,1,1/
      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 635 and SPRO 780 in the Subroutine SPROP must be modified according to:

```
      WCR=THPROP(1,MTYP)
      WCS=THPROP(2,MTYP)
      HAA=THPROP(3,MTYP)
      HAB=THPROP(4,MTYP)
      DO 800 KG=1,NQ
      HNP=HKG(KG)
      HNP=-HNP
C
C ------- SATURATED CONDITION
C
      IF(HNP.GT.0) GO TO 700
      TH(KG,M)=WCS
      DTH(KG,M)=0.0D0
      AKX(KG,M)=SATKX
      AKZ(KG,M)=SATKZ
      AKXZ(KG,M)=SATKXZ
      GO TO 800
C
C ------- UNSATURATED CASE
C
  700 THMKG=WCS-(WCS-WCR)*(-HNP-HAA)/(HAB-HAA)
      TH(KG,M)=THMKG
      RK=(THMKG-WCR)/(WCS-WCR)
      DTH(KG,M)=-(WCS-WCR)/(HAB-HAA)
      AKX(KG,M)=RK*SATKX
      AKZ(KG,M)=RK*SATKZ
      AKXZ(KG,M)=RK*SATKXZ
C
```

The pressure head simulated with the revised FEMWATER is plotted in Fig. 11. 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 that given by Huyakorn (1986) and 3DFEMWATER (Yeh 1987b). Thus, the consistency of the computer code is partially verified.

ORNL-DWG 87C-9550

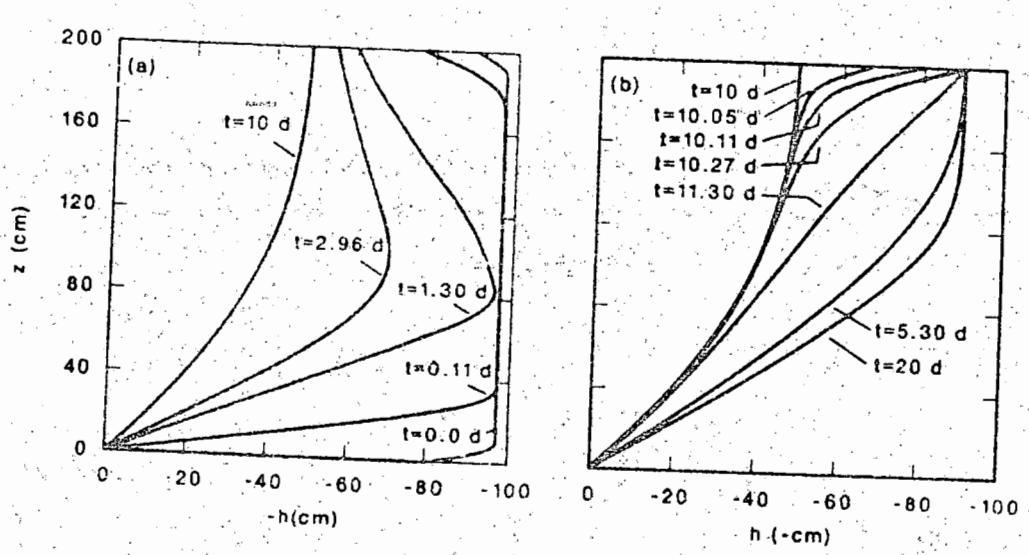


Fig. 11 Simulated pressure head at various time during (a) infiltration and (b) evaporation for the one-dimensional column problem.

6.2 Two-dimensional Drainage Problem.

This example is selected to represent the simulation of a two-dimensional problem with FEMWATER. 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. 12). The distance between two rains is 20 m apart (Fig. 12). The medium is assumed to have a saturated hydraulic conductivity of 0.01 m/day, a porosity of 0.25 and a field capacity of 0.05. The unsaturated characteristic hydraulic properties of the medium are given as:

$$\theta = \theta_s + (\theta_s - \theta_r) \frac{A}{A + |h-h_a|^B} \tag{117}$$

and

$$K_r = \left(\frac{\theta - \theta_r}{\theta_s - \theta_r}\right)^n \tag{118}$$

where h_a, A, and B are the parameters used to computer 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. The boundary conditions are given as: no flux is imposed on the left ($x = 0$) 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

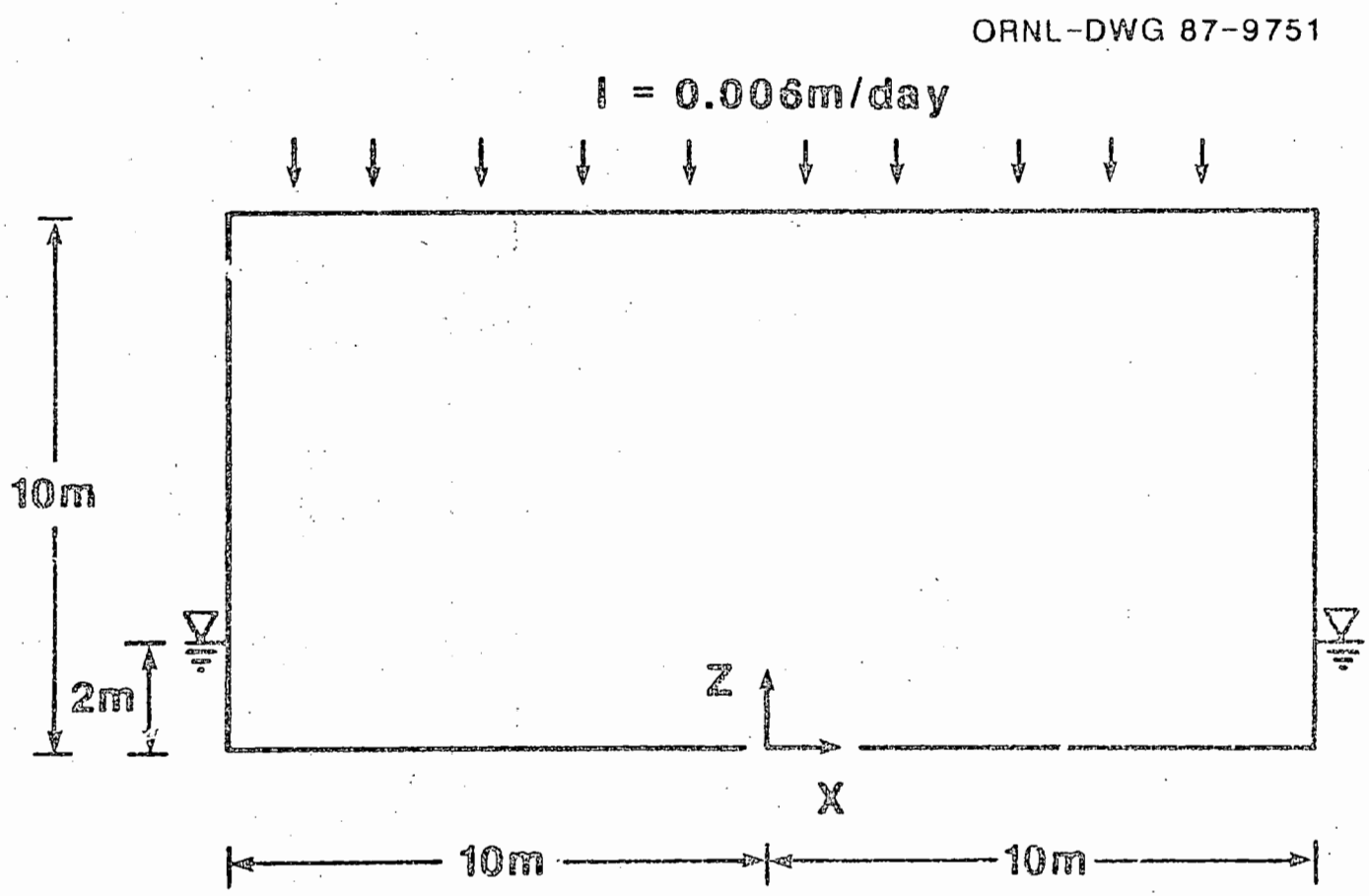


Fig. 12 Problem definition for the two-dimensional steady state flow to the parallel drains.

conditions are used elsewhere. Ponding depth is assumed to be zero m on the whole variable boundary. Fluxes on the top side of the variable boundary are assumed equal to 0.006 m/day 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 10 = 100 elements with element size = 1 x 1 cm, resulting in 11 x 11 = 121 node points. The pressure head tolerance for nonlinear iteration is $2 \cdot 10^{-3}$ m. The relaxation factor the nonlinear iteration is set equal to 0.5.

With the above descriptions, the input data can be prepared following the instruction of Section 5.3 Data Input Guide. The input data is given in Table 2.

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

```
DIMENSION X(121),Z(121),IE(100,9),LLNODE(9,121)
DIMENSION C(121,13),R(121),RL(121),RI(121)
DIMENSION H(121),HP(121),HW(121),HT(121),VX(121),VZ(121)
DIMENSION NPCNV(121), TH(4,100),DTH(4,100)
DIMENSION AKX(4,100),AKZ(4,100),AKXZ(4,100)
DIMENSION DLB(40),DCOSXB(40),DCOSZB(40),NBE(40),ISB(4,40)
DIMENSION BFLX(40),BFLXP(40),NPB(40)
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)
```

Table 2. Input data for two-dimensional drainage problem

```
  2   SIMULATION OF TWO-D STEADY DRAINAGE  WITH FEMWATER, L=M,T=HR,M=KG    111
 121  100    1    0    0    0    0    5    6    1    0    1    0   50   20    1
   0    0    0  100    1
   0.05D0     0.2D0      1.0D0     20.0D0     0.0D0     2.0D-3     2.0D-3     1.0D0
 7.316D10  1.1232D4      1.0D0      0.5D0     0.5D0      0.0D0
55
11
   1.0D38
   0.00D0     0.0D0     0.20D0     0.01D0    0.01D0      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      1.0D0
   2   10   11          0.0D0      1.0D0
   3   10   11          0.0D0      1.0D0
   4   10   11          0.0D0      1.0D0
   5   10   11          0.0D0      1.0D0
   6   10   11          0.0D0      1.0D0
   7   10   11          0.0D0      1.0D0
   8   10   11          0.0D0      1.0D0
   9   10   11          0.0D0      1.0D0
  10   10   11          0.0D0      1.0D0
  11   10   11          0.0D0      1.0D0
                                                   END OF X-COORD
   1   10   11          0.0D0      0.0D0
   2   10   11          1.0D0      0.0D0
   3   10   11          2.0D0      0.0D0
   4   10   11          3.0D0      0.0D0
   5   10   11          4.0D0      0.0D0
   6   10   11          5.0D0      0.0D0
   7   10   11          6.0D0      0.0D0
   8   10   11          7.0D0      0.0D0
   9   10   11          8.0D0      0.0D0
  10   10   11          9.0D0      0.0D0
  11   10   11         10.0D0      0.0D0
                                                   END OF Z-COORD
   1    1   12   13    2    1   10   10          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
```

Table 2 (Continued)

```
 10   10   11          1.0D0      0.0D0
 11   10   11          0.0D0      0.0D0
                                                          END OF IC
  0    0    0   18   19    2    2    3    1    2    0    0    0    0    0    0
  0    0    0    0    0
   0.0D0    6.0D-3    1.0D38     6.0D-3
   0.0D0    0.0D00    1.0D38     0.0D00
  1   10    1   11   11
 12    7    1  120   -1
                                                          END OF NPVB
  1   10    1    1    0
 12    7    1    2    0
                                                          END OF IRTYP
  1   18    1          0.0D0      0.0D0
                                                          END OF HCON
  1   18    1         -90.0D2     0.0D0
                                                          END OF HMIN
  1    9   10    1    2    1   10    1    1
 11    7  100   11   12    1   -1   -1   -1
                                                          END OF ISV(J,I) J=1,4
   0.0D0      2.0D0    1.0D38     2.0D0
111  112  113
  1    2    1    1    0
                                                          END OF IDTYP
                    END OF JOB
```

```
DIMENSION RFALL(2),RF(2,2),TRF(2,2)
DIMENSION NPRS(19),IRFTYP(19), IS(4,18)
DIMENSION DL(18),DCOSX(18),DCOSZ(18),NRSE(18),RSFLX(18),DCYFLX(18)
DIMENSION HCON(19),FLX(19),HMIN(19),NPCON(19),NPFLX(19),NPMIN(19)
DIMENSION HDB(1),HDBF(2,1),THDBF(2,1)
DIMENSION QCB(1),QCBF(1,1),TQCBF(1,1)
DIMENSION NPCB(1),ICTYP(1), ISC(2,1)
DIMENSION QNB(1),QNBF(1,1),TQNBF(1,1)
DIMENSION NPNB(1),INTYP(1), ISN(4,1)
DIMENSION DLN(1),DCOSXN(1),DCOSZN(1),MNES(1)
DIMENSION NPDB(3),IDTYP(3)
DIMENSION PROP(6,1)
DIMENSION THPROP(5,1),AKPROP(5,1),HPROP(5,1),CAPROP(5,1)
DIMENSION KPR(1),KDSK(1),TDTCH(1)
```

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

```
DATA MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI/100,121,40,40,13,1/
DATA JBAND/9/
DATA MXNDTC/1/
DATA MXSEL,MXSPR,MXSDP,MXWNP,MXWPR,MXWDP/1,1,1,1,1,1/
DATA MXRSEL,MXRSNP,MXRFPR,MXRPAR/18,19,2,2/
DATA MXDNP,MXDPR,MXDDP/3,1,2/
DATA MXCNP,MXCEL,MXCPR,MXCDP/1,1,1,1/
DATA MXNNP,MXNEL,MXNPR,MXNDP/1,1,1,1/
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 635 and SPRO 780 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,NQ
      HNP=HKG(KG)
      HNP=-HNP
C
C ------- SATURATED CONDITION
C
      IF(HNP.GT.0) GO TO 700
      TH(KG,M)=WCS
      AKX(KG,M)=SATKX
      AKZ(KG,M)=SATKZ
      AKXZ(KG,M)=SATKXZ
      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
      RK=((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
      AKX(KG,M)=RK*SATKX
      AKZ(KG,M)=RK*SATKZ
      AKXZ(KG,M)=RK*SATKXZ
C
```

The pressure head simulated with FEMWATER is plotted in Figure 13. Numerical predictions of these pressure heads are remarkedly in agreement with those given 3DFEMWATER (Yeh 1987b). This verify the one step steady state solution algorithm provided in the revised FEMWATER.

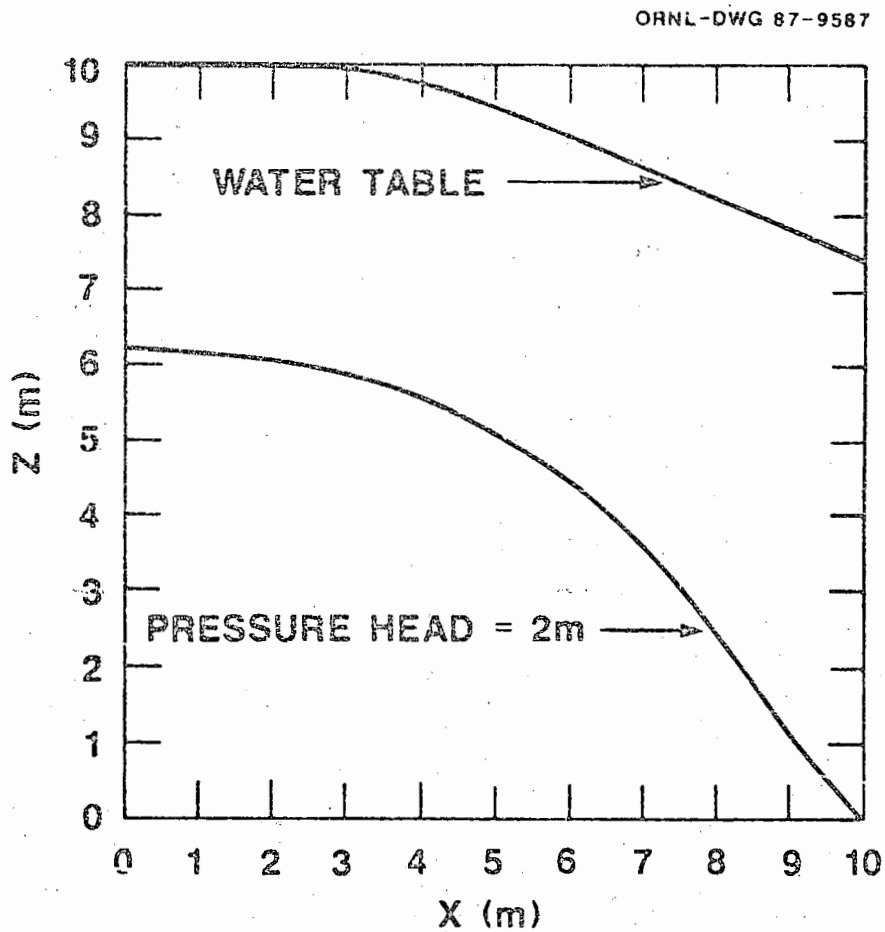


Fig. 13 Simulated steady state profiles of water table and base pressure head for the two-dimensional drainage problem.

6.3 Seepage Pond Problem

This example is selected to compare the simulation results by this revised FEMWATER with those obtained by the original FEMWATER. The problem involved the seepage of water from a pond into the underlying media. The pond near a stream is assumed to be situated entirely in the unsaturated zone above the water table (Fig. 14). This pond provides the source of water which drains into the aquifer. Although the rainfall on the soil surface also provides water sources in the form of infiltration, it is on the average very small compared to the continuous drainage from the pond. After the water reaches the water table, it flows toward a nearby stream as depicted in Fig. 14, which also outlines the surface topography and the extent of the aquifer system. This example typifies a class of problems involving leaching of wastes from storage lagoons. The soil properties are given in Fig. 15. The problem being addressed is to find the spatial distributions of pressure head, total head, moisture content, and Darcy's velocity under steady state condition achieved by a constant drainage rate of 1.44 cm/h.

To use the FEMWATER program for answering the above question, the region of interest in Fig. 14 is discretized by 595 nodes and 528 elements as shown in Fig. 16. For the finite element computation, the seven nodal points on the stream-soil interface are designated as Dirichlet nodes (Fig. 16). Seven nodal points on the bottom of the seepage pond, namely nodal points nos. 152, 164, 172, 180, 188, 196, and 204, are considered as

9 2 1 2 4 5 8 0 6 0 2

9 2 1 2 4 5 8 0 6 0 3

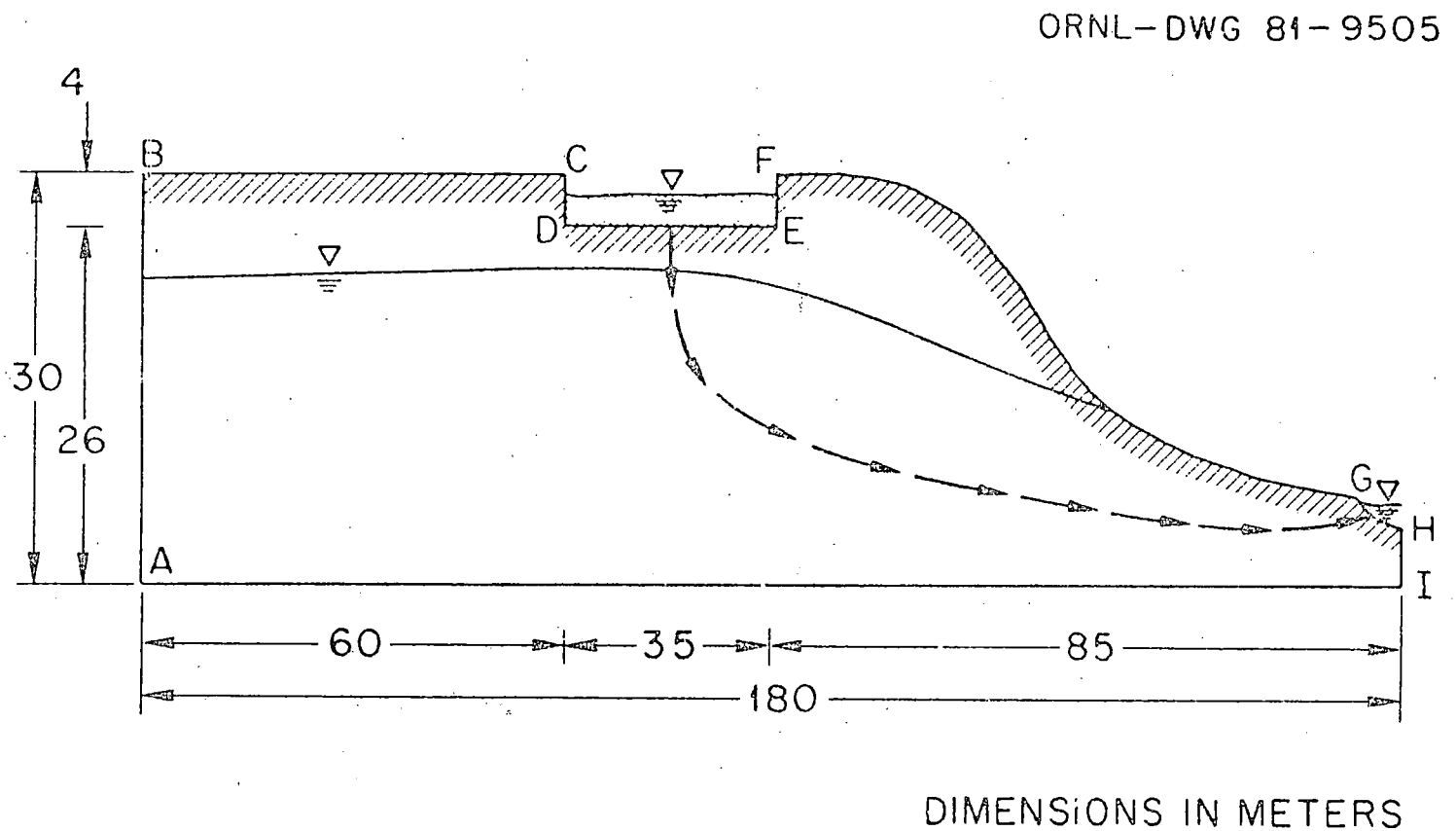


Fig. 14 Problem definition for the seepage pond problem.

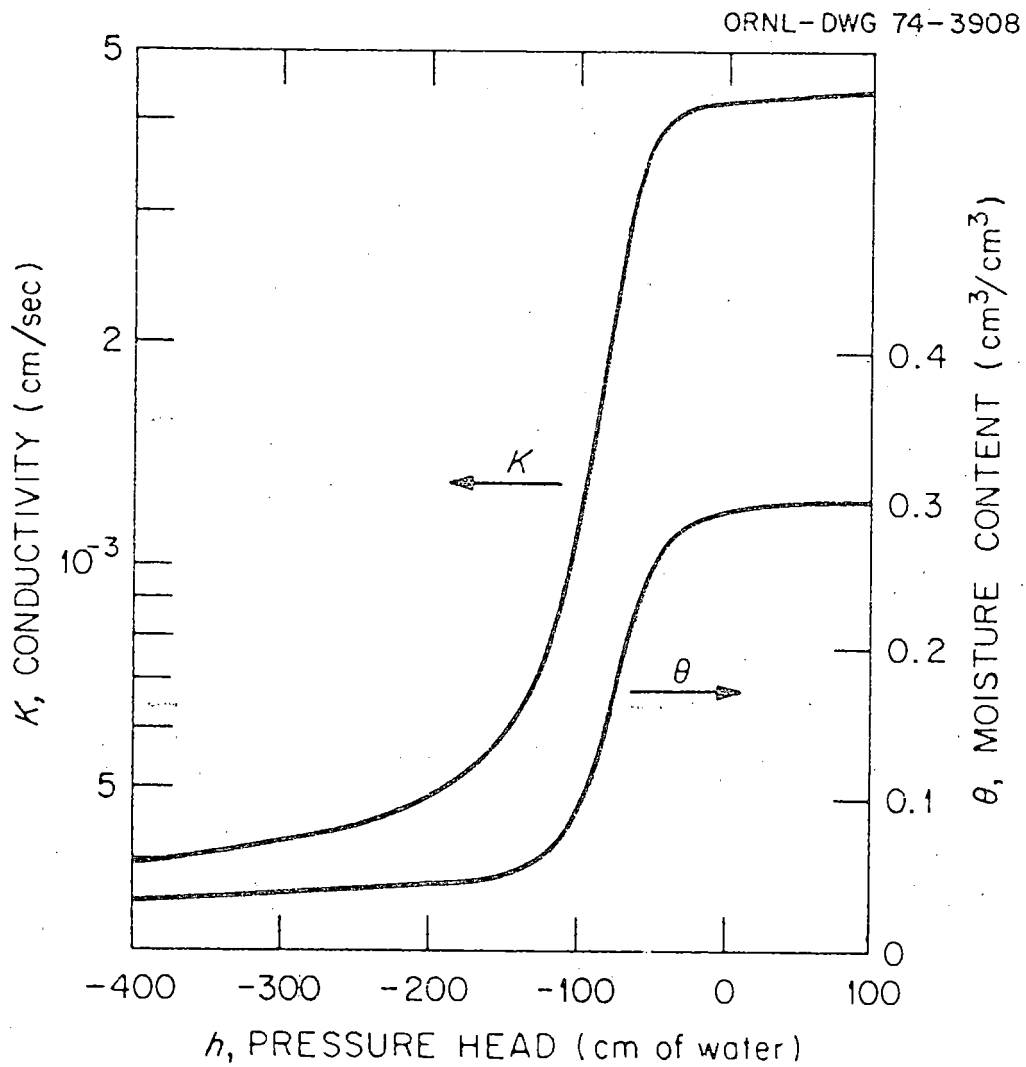


Fig. 15 Hydraulic property functions used for the seepage pond problem.

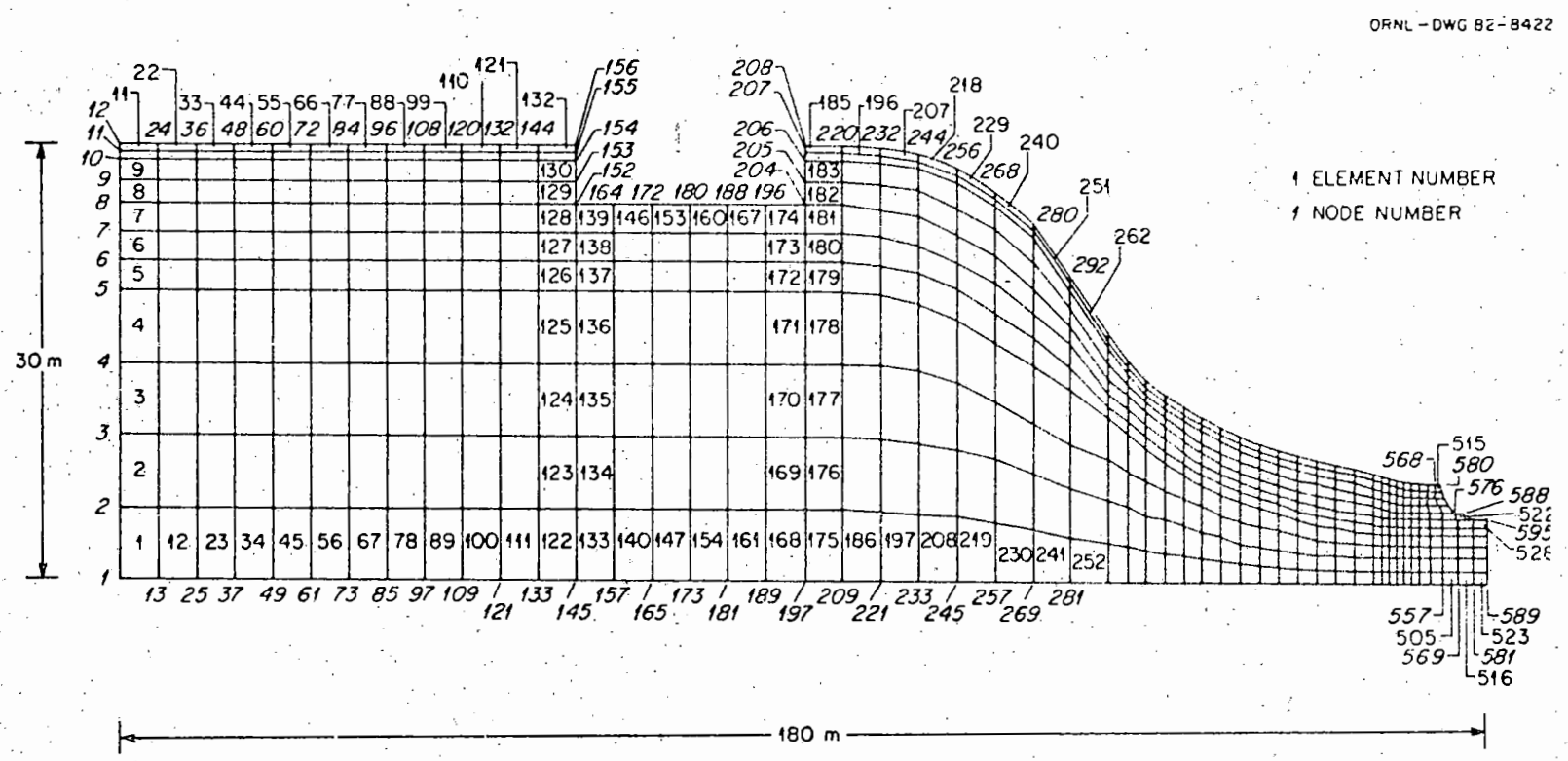


Fig. 16 Finite element discretization for the seepage pond problem.

constant Cauchy flux points and are assigned a constant infiltration rate of 4.0×10^{-4} cm/s. The top side of all elements on the sloping surface, except for the two elements immediately to the right of the seepage pond, are considered the seepage-evaporation-rainfall boundary surface. In other words, the nodal points on this surface are either Dirichlet or Cauchy points with the infiltration rate equal to the throughout rainfall rate, which is assumed zero in this case. All other boundaries are treated as impervious.

With the above descriptions, the input data can be prepared following the instruction of Section 5.3 Data Input Guide. The input data is given in Table 3.

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

```
DIMENSION X(595),Z(595),IE(528,9),LLNODE(9,528)
DIMENSION C(595,14),R(595),RL(595),RI(595)
DIMENSION H(595),HP(595),HW(595),HT(595),VX(595),VZ(595)
DIMENSION NPCNV(595), TH(4,528),DTH(4,528)
DIMENSION AKX(4,528),AKZ(4,528),AKXZ(4,528)
DIMENSION DLB(132),DCOSXB(132),DCOSZB(132),NBE(132)ISB(4,132)
DIMENSION BFLX(132),BFLXP(132),NPB(132)
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)
```

Table 3. Input data for the seepage pond problem

```
   3    SEEPAGE POND, STEADY STATE SIMULATION, WITH FEMWATER-1 CODE          111
 595  528    1    0    0    0    1   16    6    1    1    1    0   20   25    1
   0    0    0  300    0
 300.0      0.5        86400.0        1.0E9 0.0        0.01        0.1        1.0
 980.6      0.013      1.0        1.0        1.0
55
11
   1.0D+38
0.          0.          .3              .58E-7      .58E-7
-800.0     -400.       -200.       -175.       -150.       -125.       -100.       -62.5
 -50.0     -37.5       -25.        -12.5       0.          50.         100.        2000.
.024        .032        .0425       .045        .050        .0625       .09         .21
.25         .275        .285        .290        .2925       .2975       .2995       .3
 0.0758607 0.0896570 0.1120675 0.1206892 0.1379327 0.1637979 0.2758126 0.6896635
 0.8276264 0.9137683 0.9483156 0.9655139 0.9655139 0.9827875 0.9999857 0.9999857
   0.          .52E-4      .10E-3      .20E-3      .50E-3      .11E-2      .32E-2      .32E-2
   .20E-2      .80E-3      .40E-3      .20E-3      .10E-3      .40E-4      .26E-6      0.
   1   11    1            0.0D0       0.0D0       0.0D0
  13   11    1          500.0D0       0.0D0       0.0D0
  25   11    1         1000.0D0       0.0D0       0.0D0
  37   11    1         1500.0D0       0.0D0       0.0D0
  49   11    1         2000.0D0       0.0D0       0.0D0
  61   11    1         2500.0D0       0.0D0       0.0D0
  73   11    1         3000.0D0       0.0D0       0.0D0
  85   11    1         3500.0D0       0.0D0       0.0D0
  97   11    1         4000.0D0       0.0D0       0.0D0
 109   11    1         4500.0D0       0.0D0       0.0D0
 121   11    1         5000.0D0       0.0D0       0.0D0
 133   11    1         5500.0D0       0.0D0       0.0D0
 145   11    1         6000.0D0       0.0D0       0.0D0
 157    7    1         6500.0D0       0.0D0       0.0D0
 165    7    1         7000.0D0       0.0D0       0.0D0
 173    7    1         7500.0D0       0.0D0       0.0D0
 181    7    1         8000.0D0       0.0D0       0.0D0
 189    7    1         8500.0D0       0.0D0       0.0D0
 197   11    1         9000.0D0       0.0D0       0.0D0
 209   11    1         9500.0D0       0.0D0       0.0D0
 221   11    1        10000.0D0       0.0D0       0.0D0
 233   11    1        10500.0D0       0.0D0       0.0D0
 245   11    1        11000.0D0       0.0D0       0.0D0
 257   11    1        11500.0D0       0.0D0       0.0D0
 269   11    1        12000.0D0       0.0D0       0.0D0
 281   11    1        12500.0D0       0.0D0       0.0D0
 293   11    1        13000.0D0       0.0D0       0.0D0
 305   11    1        13250.0D0       0.0D0       0.0D0
 317   11    1        13500.0D0       0.0D0       0.0D0
```

Table 3 (Continue)

329	11	1	13750.0D0	0.0D0	0.0D0
341	11	1	14000.0D0	0.0D0	0.0D0
353	11	1	14250.0D0	0.0D0	0.0D0
365	11	1	14500.0D0	0.0D0	0.0D0
377	11	1	14750.0D0	0.0D0	0.0D0
389	11	1	15000.0D0	0.0D0	0.0D0
401	11	1	15250.0D0	0.0D0	0.0D0
413	11	1	15500.0D0	0.0D0	0.0D0
425	11	1	15750.0D0	0.0D0	0.0D0
437	11	1	16000.0D0	0.0D0	0.0D0
449	11	1	16250.0D0	0.0D0	0.0D0
461	11	1	16500.0D0	0.0D0	0.0D0
473	11	1	16600.0D0	0.0D0	0.0D0
485	11	1	16700.0D0	0.0D0	0.0D0
497	11	1	16800.0D0	0.0D0	0.0D0
509	11	1	16900.0D0	0.0D0	0.0D0
521	11	1	17000.0D0	0.0D0	0.0D0
533	11	1	17100.0D0	0.0D0	0.0D0
545	7	1	17250.0D0	0.0D0	0.0D0
553	3	1	17200.0D0	0.0D0	0.0D0
557	7	1	17400.0D0	0.0D0	0.0D0
565	0	1	17350.0D0	0.0D0	0.0D0
566	2	1	17300.0D0	0.0D0	0.0D0
569	6	1	17600.0D0	0.0D0	0.0D0
576	0	1	17550.0D0	0.0D0	0.0D0
577	0	1	17500.0D0	0.0D0	0.0D0
578	0	1	17450.0D0	0.0D0	0.0D0
579	1	1	17400.0D0	0.0D0	0.0D0
581	6	1	17800.0D0	0.0D0	0.0D0
588	0	1	17650.0D0	0.0D0	0.0D0
589	6	1	18000.0D0	0.0D0	0.0D0
1	12	12	0.0D0	0.0D0	0.0D0
2	12	12	500.0D0	0.0D0	0.0D0
3	12	12	1000.0D0	0.0D0	0.0D0
4	12	12	1500.0D0	0.0D0	0.0D0
5	12	12	2000.0D0	0.0D0	0.0D0
6	12	12	2200.0D0	0.0D0	0.0D0
7	12	12	2400.0D0	0.0D0	0.0D0
8	12	12	2600.0D0	0.0D0	0.0D0
9	12	12	2750.0D0	0.0D0	0.0D0
10	12	12	2900.0D0	0.0D0	0.0D0
11	12	12	2950.0D0	0.0D0	0.0D0
12	12	12	3000.0D0	0.0D0	0.0D0
157	5	8	0.0D0	0.0D0	0.0D0
158	5	8	500.0D0	0.0D0	0.0D0
159	5	8	1000.0D0	0.0D0	0.0D0
160	5	8	1500.0D0	0.0D0	0.0D0

Table 3(Continued)

161	5	8	2000.0D0	0.0D0	0.0D0
162	5	8	2200.0D0	0.0D0	0.0D0
163	5	8	2400.0D0	0.0D0	0.0D0
164	5	8	2600.0D0	0.0D0	0.0D0
205	1	1	2750.0D0	150.0D0	0.0D0
207	1	1	2950.0D0	50.0D0	0.0D0
209	4	1	0.0D0	500.0D0	0.0D0
214	2	1	2200.0D0	200.0D0	0.0D0
217	1	1	2750.0D0	150.0D0	0.0D0
219	1	1	2950.0D0	50.0D0	0.0D0
221	1	1	0.0D0	480.0D0	0.0D0
[illegible]	1	1	990.0D0	510.0D0	0.0D0
[illegible]	1	1	1980.0D0	200.0D0	0.0D0
227	1	1	2380.0D0	180.0D0	0.0D0
229	1	1	2740.0D0	150.0D0	0.0D0
231	1	1	2940.0D0	50.0D0	0.0D0
233	1	1	0.0	460.0	0.0
235	1	1	960.0	500.0	0.0
237	1	1	1920.0	210.0	0.0
239	1	1	2320.0	200.0	0.0
241	1	1	2700.0	150.0	0.0
243	1	1	2900.0	50.0	0.0
245	1	1	0.0	450.0	0.0
247	1	1	910.0	470.0	0.0
249	1	1	1810.0	210.0	0.0
251	1	1	2200.0	190.0	0.0
253	1	1	2570.0	180.0	0.0
255	1	1	2800.0	50.0	0.0
257	1	1	0.0	400.0	0.0
259	1	1	850.0	400.0	0.0
261	1	1	1660.0	200.0	0.0
263	1	1	2060.0	190.0	0.0
265	1	1	2440.0	150.0	0.0
267	1	1	2640.0	50.0	0.0
269	1	1	0.0	350.0	0.0
271	1	1	750.0	350.0	0.0
273	1	1	1500.0	200.0	0.0
275	1	1	1860.0	160.0	0.0
277	1	1	2200.0	170.0	0.0
279	1	1	2420.0	50.0	0.0
281	1	1	0.0	300.0	0.0
283	1	1	650.0	300.0	0.0
285	1	1	1330.0	160.0	0.0
287	1	1	1640.0	120.0	0.0
289	1	1	1880.0	120.0	0.0
291	1	1	2050.0	50.0	0.0
293	1	1	0.0	270.0	0.0
295	1	1	570.0	280.0	0.0

9 2 1 2 4 5 8 0 6 0 9

Table 3 (Continued)

297	1	1	1140.0	70.0	0.0
299	1	1	1300.0	100.0	0.0
301	1	1	1510.0	110.0	0.0
303	1	1	1670.0	50.0	0.0
305	1	1	0.0	250.0	0.0
307	1	1	530.0	240.0	0.0
309	1	1	1030.0	90.0	0.0
311	1	1	1200.0	100.0	0.0
313	1	1	1370.0	80.0	0.0
315	1	1	1500.0	50.0	0.0
317	1	1	0.0	220.0	0.0
319	1	1	460.0	240.0	0.0
321	1	1	920.0	90.0	0.0
323	1	1	1100.0	80.0	0.0
325	1	1	1250.0	50.0	0.0
327	1	1	1350.0	50.0	0.0
329	1	1	0.0	200.0	0.0
331	1	1	440.0	210.0	0.0
333	1	1	830.0	80.0	0.0
335	1	1	1000.0	80.0	0.0
337	1	1	1140.0	60.0	0.0
339	1	1	1250.0	50.0	0.0
341	1	1	0.0	190.0	0.0
343	1	1	400.0	190.0	0.0
345	1	1	750.0	80.0	0.0
347	1	1	900.0	80.0	0.0
349	1	1	1050.0	70.0	0.0
351	1	1	1170.0	50.0	0.0
353	1	1	0.0	170.0	0.0
355	1	1	350.0	190.0	0.0
357	1	1	670.0	80.0	0.0
359	1	1	820.0	80.0	0.0
361	1	1	960.0	90.0	0.0
363	1	1	1100.0	50.0	0.0
365	1	1	0.0	150.0	0.0
367	1	1	320.0	150.0	0.0
369	1	1	610.0	80.0	0.0
371	1	1	760.0	70.0	0.0
373	1	1	900.0	80.0	0.0
375	1	1	1030.0	50.0	0.0
377	1	1	0.0	140.0	0.0
379	1	1	270.0	160.0	0.0
381	1	1	560.0	70.0	0.0
383	1	1	700.0	60.0	0.0
385	1	1	840.0	70.0	0.0
387	1	1	960.0	50.0	0.0
389	1	1	0.0	120.0	0.0

Table 3 (Continued)

391	1	1	250.0	140.0	0.0
393	1	1	520.0	70.0	0.0
395	1	1	650.0	70.0	0.0
397	1	1	800.0	60.0	0.0
399	1	1	910.0	50.0	0.0
401	1	1	0.0	110.0	0.0
403	1	1	230.0	140.0	0.0
405	1	1	480.0	70.0	0.0
407	1	1	610.0	70.0	0.0
409	1	1	750.0	70.0	0.0
411	1	1	870.0	50.0	0.0
413	1	1	0.0	100.0	0.0
415	1	1	210.0	120.0	0.0
417	1	1	440.0	60.0	0.0
419	1	1	560.0	80.0	0.0
421	1	1	710.0	80.0	0.0
423	1	1	840.0	50.0	0.0
425	1	1	0.0	100.0	0.0
427	1	1	200.0	100.0	0.0
429	1	1	400.0	70.0	0.0
431	1	1	540.0	70.0	0.0
433	1	1	690.0	60.0	0.0
435	1	1	800.0	50.0	0.0
437	1	1	0.0	100.0	0.0
439	1	1	200.0	90.0	0.0
441	1	1	390.0	60.0	0.0
443	1	1	510.0	80.0	0.0
445	1	1	660.0	70.0	0.0
447	1	1	780.0	50.0	0.0
449	1	1	0.0	90.0	0.0
451	1	1	190.0	90.0	0.0
453	1	1	360.0	80.0	0.0
455	1	1	490.0	70.0	0.0
457	1	1	630.0	70.0	0.0
459	1	1	750.0	50.0	0.0
461	1	1	0.0	90.0	0.0
463	1	1	180.0	90.0	0.0
465	1	1	350.0	60.0	0.0
467	1	1	480.0	60.0	0.0
469	1	1	600.0	60.0	0.0
471	1	1	710.0	50.0	0.0
473	1	1	0.0	90.0	0.0
475	1	1	180.0	90.0	0.0
477	1	1	350.0	50.0	0.0
479	1	1	460.0	60.0	0.0
481	1	1	590.0	60.0	0.0
483	1	1	700.0	50.0	0.0

Table 3 (Continued)

485	1	1	0.0	90.0	0.0
487	1	1	180.0	80.0	0.0
489	1	1	340.0	60.0	0.0
491	1	1	460.0	50.0	0.0
493	1	1	570.0	70.0	0.0
495	1	1	690.0	50.0	0.0
497	1	1	0.0	90.0	0.0
499	1	1	180.0	80.0	0.0
501	1	1	340.0	60.0	0.0
503	1	1	450.0	50.0	0.0
505	1	1	560.0	60.0	0.0
507	1	1	670.0	50.0	0.0
509	1	1	0.0	90.0	0.0
511	1	1	180.0	80.0	0.0
513	1	1	340.0	50.0	0.0
515	1	1	450.0	50.0	0.0
517	1	1	550.0	60.0	0.0
519	1	1	660.0	50.0	0.0
521	1	1	0.0	90.0	0.0
523	1	1	180.0	80.0	0.0
525	1	1	340.0	50.0	0.0
527	1	1	450.0	50.0	0.0
529	1	1	550.0	50.0	0.0
531	1	1	650.0	50.0	0.0
533	1	1	0.0	90.0	0.0
535	1	1	180.0	80.0	0.0
537	1	1	340.0	50.0	0.0
539	1	1	450.0	50.0	0.0
541	1	1	550.0	50.0	0.0
543	1	1	650.0	50.0	0.0
545	1	1	0.0	90.0	0.0
547	1	1	180.0	80.0	0.0
549	1	1	340.0	50.0	0.0
551	1	1	450.0	50.0	0.0
553	1	1	550.0	50.0	0.0
555	1	1	650.0	50.0	0.0
557	1	1	0.0	90.0	0.0
559	1	1	180.0	80.0	0.0
561	1	1	340.0	50.0	0.0
563	1	1	450.0	50.0	0.0
565	1	1	550.0	50.0	0.0
567	1	1	650.0	50.0	0.0
569	1	1	0.0	90.0	0.0
571	1	1	180.0	80.0	0.0
573	1	1	340.0	50.0	0.0
575	1	1	450.0	50.0	0.0
577	1	1	550.0	50.0	0.0

Table 3 (Continued)

```
 579    1    1          650.0       50.0       0.0
 581    2    1          0.0D0      90.0D0     0.0D0
 584    1    1        260.0D0      80.0D0     0.0D0
 586    1    1        390.0D0      60.0D0     0.0D0
 588    0    1        490.0D0       0.0D0     0.0D0
 589    2    1          0.0D0      90.0D0     0.0D0
 592    1    1        260.0D0      80.0D0     0.0D0
 594    1    1        390.0D0      60.0D0     0.0D0

   1    1   13   14    2    1   11   12
 133  145  157  158  146    1
 139  151  163  164  152    1
 140  157  165  166  158    1    7    5
 175  197  209  210  198    1   11   31
 516  569  581  582  570    1
 522  575  587  588  576    1
 523  581  589  590  582    1
 528  586  594  595  587    1
   1  594    1      0.0        0.0

   0    1    2   29   30    1    2    7    1    2    7    6    1    2    0    0
   1    2    0    1    2
0.0        0.0            1.0D9 0.0
   1   29    1  232   12
                                                                END OF NPRS
   1   29    1    1    0
                                                                END OF IRFTYP
   1   29    1      0.0        0.0
                                                                END OF HCON
   1   29    1      -15000.0  0.0
                                                                END OF HMIN
   1   28  207    1    2    1   11    1    1
                                                                END OF RS ELEMENTS
0.0          650.000      1.0D9   650.000
 579  578  577  576  588  587  595
   1    6    1    1    0
                                                                END OF IDTYP
0.0          -4.0D-4      1.0D9   -4.0D-4
 152  164  172  180  188  196  204
   1    6    1    1    0
                                                                END OF ICTYP
   1    5    1    2    1    1    1
                                                                END OF ISC
                          END OF JOB
```

```
DIMENSION RFALL(1),RF(2,1),TRF(2,1)
DIMENSION NPRS(30),IRFTYP(30), IS(4,29)
DIMENSION DL(29),DCOSX(29),DCOSZ(29),NRSE(29),RSFLX(30),DCYFLX(30)
DIMENSION HCON(30),FLX(30),HMIN(30),NPCON(30),NPFLX(30),NPMIN(30)
DIMENSION HDB(1),HDBF(2,1),THDBF(2,1)
DIMENSION NPDB(7),IDTYP(7)
DIMENSION QCB(1),QCBF(2,1),TQCBF(2,1)
DIMENSION NPCB(7),ICTYP(7), ISC(5,6)
DIMENSION QNB(1),QNBF(1,1),TQNBF(1,1)
DIMENSION NPNB(1),INTYP(1), ISN(5,1)
DIMENSION DLN(1),DCOSXN(1),DCOSZN(1),MNES(1)
DIMENSION PROP(6,1)
DIMENSION THPROP(16,1),AKPROP(16,1),HPROP(16,1),CAPROP(16,1)
DIMENSION KPR(1),KDSK(1),TDTCH(1)
```

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

```
DATA MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNT1/528,595,132,132,14,1/
DATA JBAND/9/
DATA MXNDTC/1/
DATA MXSEL,MXSPR,MXSDP,MXWNP,MXWPR,MXWDP/1,1,1,1,1,1/
DATA MXRSEL,MXRSNP,MXRFPR,MXRPAR/29,30,1,2/
DATA MXDNP,MXDPR,MXDDP/7,1,2/
DATA MXCNP,MXCEL,MXCPR,MXCDP/7,6,1,2/
DATA MXNNP,MXNEL,MXNPR,MXNDP/1,1,1,1/
DATA MAXMAT,MXSPPM,MXMPPM/1,16,6/
```

Figures 17a-c show the Darcy velocity plot and the distribution of pressure head and total head, as simulated by the revised FEMWATER. The simulation is almost identical to that obtained by the original FEMWATER. This verify the revised algorithm of FEMWATER.

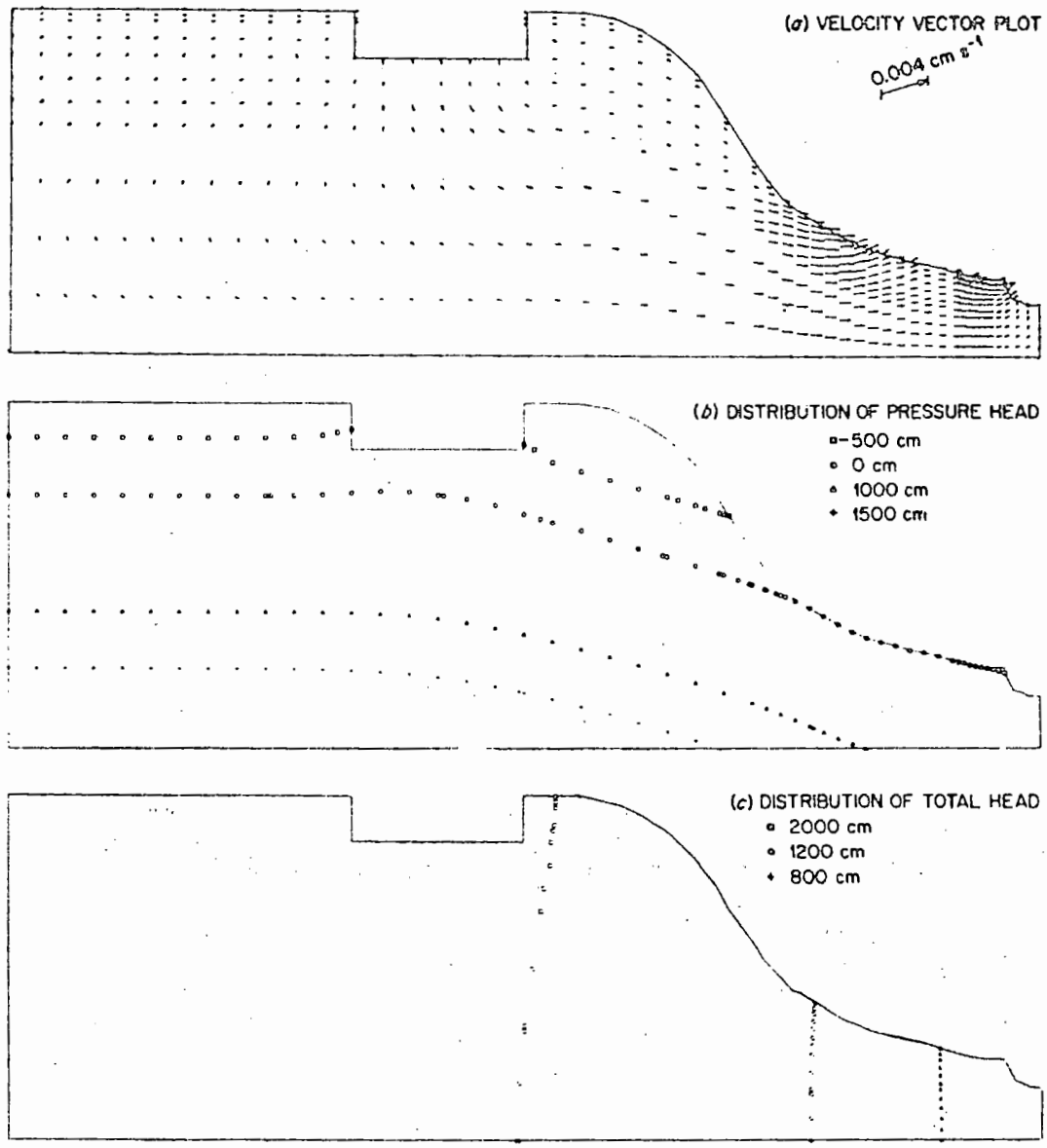


Fig. 17 Flow variables of seepage pond as simulated by the revised FEMWATER, (a) Darcy's velocity plot, (b) distribution of pressure head, and (c) distribution of total head.

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. Apply. Phys. 12:155-164.

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

Forsythe, G. E. and W. R. Wasow. 1960. Finite-Difference Methods for Partial Differential Equations. John Wiley & Sons, Inc., 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 Interior, Washington, D.C. 119 pp.

Huebner, K. H. 1975. The Finite Element Methods for Engineers, John Wiley & Sons, Inc., 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 Resources Res. Vol. 20, No. 8, 1099-1115.

Huyakorn, P. S. 1986. A Three-dimensional Finite Element model for simulating water flow in variably saturated porous media, Water Resources Res. Vol. 22, No. 13, 1790-1809.

Lapidus, L. and G. F. Pinder. 1982. Numerical Solution of Partial Differential Equations in Science and Engineering. John Wiley & Sons, Inc., New York. 677 pp.

Narasimhan, T. N. and P. A. Witherspoon. 1977. Numerical model for saturated-unsaturated flow in deformable porous media. Water Resources Res. Vol. 13, No. 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 Company, Scranton, Pennsylvania. 648 pp.

Polubarinova-Kochina, P. Ya. 1962. Theory of Groundwater Movement. Princeton University Press, Princeton, New Jersey. 613 pp.

Reeves, M. and J. O. Duguid. 1975. Water Movement through Saturated-Unsaturated-Unsaturated Porous Media: A Finite-Element Galerkin Model. ORNL-4927. Oak Ridge National Laboratory, Oak Ridge, Tennessee. 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, Washington.

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. IN Management of Gas and Leachate in Landfills. S. K. Banerji (ed.), EPA-600/9-77-026. Office of Research and Development, U.S. Environmental Protection Agency, Cincinnati, Ohio. 95-103.

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, New Jersey.

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. IN Flow through Porous Media, Chapter 8. R. Dewiest (ed.). McGraw-Hill Book Company, New York.

Wang, J. D. and J. J. Connor, 1975. Mathematical Modeling of Near-Coastal Circulation. Report No. MITSG 75-13. MIT, Cambridge, Massachusetts.

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, Tennessee. 153 pp.

Yeh, G. T. and R. J. Luxmoore. 1983. Modeling moisture and thermal transport in unsaturated porous media, J. Hydrology, 64:299-309.

9 2 1 2 4 5 8 0 6 1 9

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. Hydraulic Division, HY3, ASCE Vol. 111, 547-551.

Yeh, G. T. 1985. Comparison of successive iteration and direct methods to solve finite element equations of aquifer contaminant transport, Water Resources Res. Vol. 21, No. 3, 272-280.

Yeh, G. T. 1986. An orthogonal upstream finite element approach to modelling aquifer contaminant transport, Water Resources Res. Vol. 22, No. 6, 952-964.

Yeh, G. T. 1987a. 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, Tennessee.

Yeh, G. T., 1987b. 3DFEMWATER: A Three-dimensional Finite Element Model of WATER Flow through Saturated-Unsaturated Media. ORNL-6386. Oak Ridge National Laboratory, Oak Ridge, TN 37831.

9212458062 0

9 2 1 2 4 5 8 0 6 2 1

APPENDIX A: Listing of FORTRAN Source Code

```
C                                                                       COM  005
C       ************************************************************    COM  010
C       *                                                          *    COM  015
C       *                      FEMWATER                            *    COM  020
C       *                                                          *    COM  025
C       ************************************************************    COM  030
C                                                                       COM  035
C                                                                       COM  040
C  THIS COMPUTER CODE IS CONTAINED IN THE FOLLOWING REPORT:             COM  045
C  YEH, G. T., 1987.  "FEMWATER: A FINITE ELEMENT MODEL OF WATER FLOW   COM  050
C  THROUGH SATURATED-UNSATURATED POROUS MEDIA - FIRST REVISION,",       COM  055
C  ORNL-5567/R1, OAK RIDGE NATIONAL LABORATORY. OAK RIDGE, TN 37831     COM  060
C                                                                       COM  065
C  FOR ANY QUESTION, PLEASE CONTACT DR. G. T. YEH AT (615) 574-7285     COM  070
C                                                                       COM  075
C ****************************                                        * COM  080
C *                          *                                        * COM  085
C *  DEFINITION OF VARIABLES  *                                        * COM  090
C *                          *                                        * COM  095
C ****************************                                        * COM  100
C *                                                                    * COM  105
C * MAXEL  = MAXIMUM NO. OF ELEMENTS                                   * COM  110
C * MAXNP  = MAXIMUM NO. OF NODES                                      * COM  115
C * MAXBEL = MAXIMUM NO. OF BOUNDARY ELEMENTS                          * COM  120
C * MAXBNP = MAXIMUM NO. OF BOUNDARY NODAL POINTS                      * COM  125
C * MAXHBP = MAXIMUM NO. OF HALF BAND WIDTH                            * COM  130
C * MAXNTI = MAXIMUM NO. OF TIME STEPS                                 * COM  135
C * MXNDTC = MAXIMUM NO. OF DELT CHANGES                               * COM  140
C *                                                                    * COM  145
C *                                                                    * COM  150
C * NNP    = NO. OF NODAL POINTS                                       * COM  155
C * NEL    = NO. OF ELEMENTS                                           * COM  160
C * NBN    = NO. OF BOUNDARY NODES                                     * COM  165
C * NBEL   = NO. OF BOUNDARY ELEMENT-SIDES                             * COM  170
C * IBAND  = NO. OF BAND WIDTH                                         * COM  175
C * KGRAV  = GRAVITY TERM CONTROL, 0 = NOT INCLUDED, 1 = INCLUDED      * COM  180
C * NTI    = NO. OF TIME INCREMENTS                                    * COM  185
C * NDTCHG = NO. OF DELT CHANGES                                       * COM  190
C *                                                                    * COM  195
C * NCYL   = NO. OF CYCLES FOR ITERATING RAINFALL-SEEPAGE              * COM  200
C * NITER  = NO. OF ITERATIONS ALLOWED FOR SOLVING NONLINEAR EQATIONS  * COM  205
C * NSTRT  = NO. OF RESTART RECORD FILES                               * COM  210
C * KSTR   = STORE CONTROL INTEGER                                     * COM  215
C * KPRO   = LINE-PRINTING CONTROL FOR INITIAL TIME                    * COM  220
C * KDSKO  = DISK STORE CONTROL INTEGER                                * COM  225
C * KSS    = STEADY STATE SOLUTION CONTROL                             * COM  230
C *          0 = STEADY STATE SOLUTION IS WANTED EITHER AS FINAL       * COM  235
C *              SOLUTION OR AS INITIAL CONDIITION OF TRANSIENT STATE  * COM  240
C *              SOLUTION                                              * COM  245
C *          1 = ONLY TRANSIENT SOLUTION IS NEEDED                     * COM  250
```

9 2 1 2 4 5 8 0 6 2 2

```
C * KSP    = INDICATING IF SOIL CHARACTERISTICS ARE TO BE COMPUTED BY * COM  255
C *          INTERPOLATING THE TABLE OR BY ANALYTICAL FUNCTION,       * COM  260
C *          0 = COMPUTED ANALYTICALLY, 1 = INTERPOLATING THE TABLE   * COM  265
C * KCP    = SOIL PERMEABILITY INPUT CONTROL                          * COM  270
C *          0 = INPUT SATURATED HYDRAULIC CONDUCTIVITY TO ARRAY PROP * COM  275
C *          1 = INPUT SATURATED PERMEABILITY TO ARRAY PROP(I,3), ..  * COM  280
C *                                                                   * COM  285
C * DELT   = TIME SETP SIZE                                          * COM  290
C * CHNG   = PERCENTAGE CHANGE OF TIME STEP SIZE                     * COM  295
C * DELMAX = MAXIMUM TIME STEP SIZE ALLOWED                          * COM  300
C * TMAX   = MAXIMUM SIMULATION TIME ALLOWED                         * COM  305
C * DELTO  = INITIAL TIME STEP SIZE                                  * COM  310
C * TOLA   = TOTLERANCE FOR ITERATING STEADY STATE SOLUTION          * COM  315
C * TOLB   = TOLERANCE FOR ITERATING TRANSIENT SOLUTION OF PRESSURE  * COM  320
C * W      = TIME INTEGRATION PARAMETERS                             * COM  325
C *          0.5 = CRANK-NICOLSON CENTRAL DIFFERENCE                 * COM  330
C *          1.0 = BACKWARD DIFFERENCE (COMPLETE IMPLICIT SCHEME)    * COM  335
C * OME    = ITERATION PARAMETER                                     * COM  340
C *          0.0 TO 1.0 = UNDER RELAXATION                           * COM  345
C *          1.0 TO 2.0 = OVER-RELAXATION                            * COM  350
C * TIME   = TIME SINCE THE BEGINNING                                * COM  355
C *                                                                  * COM  360
C * CSFE   = COSINE OF THE ANGLE OF THE X-COORDINATE WITH HORIZONTAL * COM  365
C * SNFE   = SINE OF THE ANGLE OF THE X-COORDINATE WITH THE HOROZONTAL* COM  370
C *                                                                  * COM  375
C * MXSPR  = MAXIMUM NO. OF SOURCE PROFILES                          * COM  380
C * MXSDP  = MAXIMUM NO. OF DATA POINTS ON EACH SOURCE PROFILE       * COM  385
C * NSPR   = NO. OF SOURCE PROFILE                                   * COM  390
C * NSDP   = NO. OF DATA POINTS ON EACH OF THE SOURCE PROFILES       * COM  395
C * MXWNP  = MAXIMUM NO. OF WELL NODAL POINTS                        * COM  400
C * NWNP   = NO. OF WELL NODAL POINTS                                * COM  405
C * MXWPR  = MAXIMUM NO. OF WELL SOURCE PROFILES                     * COM  410
C * MXWDP  = MAXIMUM NO. OF DATA POINT IN EACH WELL SOURCE PROFILE   * COM  415
C * NWPR   = NO. OF WELL SOURCE/SINK PROFILES                        * COM  420
C * NWDP   = NO. OF DATA POINT IN EACH OF THE WELL SOURCE/SINK PROFILE* COM  425
C *                                                                  * COM  430
C * MXRSEL = MAXIMUM NO. OF RAINFALL-SEEPAGE ELEMENT SIDES           * COM  435
C * MXRSNP = MAXIMUM NO. OF RAINFALL SEEPAGE NODAL POINTS            * COM  440
C * MXRFPR = MAXIMUM NO. OF RAINFALL PROFILES                        * COM  445
C * MXRPAR = MAXIMUM NO. OF RAINFALL PARAMETERS ON EACH PROFILE      * COM  450
C * NRSEL  = NO. OF RAINFALL-SEEPAGE ELEMENT SIDES                   * COM  455
C * NRSN   = NO. OF RAINFALL SEEPAGE NODES                           * COM  460
C * NRFPR  = NO. OF RAINFALL PROFILE                                 * COM  465
C * NRFPAR = NO. OF RAINFALL PARAMETERS (DATA POINTS) ON EACH PROFILE * COM  470
C *                                                                  * COM  475
C * MXDNP  = MAXIMUM NO. OF DIRICHLET NODAL POINTS                   * COM  480
C * MXDPR  = MAXIMUM NO. OF DIRICHLET TOTAL HEAD PROFILES            * COM  485
C * MXDDP  = MAXIMUM NO. OF DATA POINTS ON EACH DIRICHLET PROFILE    * COM  490
C * NDNP   = NO. OF DIRICHLET NODAL POINTS                           * COM  495
C * NDPR   = NO. OF DIRICHLET TOTAL HEAD PROFILES                    * COM  500
```

```
C * NDDP   - NO. OF DIRICHLET DATA POINTS ON EACH DIRICHLET PROFILE   * COM  505
C *                                                                   * COM  510
C * MXCNP  - MAXIMUM NO. OF CAUCHY NODAL POINTS                       * COM  515
C * MXCEL  - MAXIMUM NO. OF CAUCHY ELEMENT SIDES                      * COM  520
C * MXCPR  - MAXIMUM NO. OF CAUCHY FLUX PROFILES                      * COM  525
C * MXCDP  - MAXIMUM NO. OF CAUCHY DATA POINTS ON EACH PROFILE        * COM  530
C * NCNP   - NO. OF CAUCHY NODAL POINTS                               * COM  535
C * NCEL   - NO. OF CAUCHY ELEMENT SIDES                              * COM  540
C * NCPR   - NO. OF CAUCY FLUX PROFILES                               * COM  545
C * NCDP   - NO. OF CAUCY FLUX DATA POINTS ON EACH PROFILE            * COM  550
C *                                                                   * COM  555
C * MXNNP  - MAXIMUM NO. OF NEUMANN NODAL POINTS                      * COM  560
C * MXNEL  - MAXIMUM NO. OF NEUMANN ELEMENT SIDES                     * COM  565
C * MXNPR  - MAXIMUM NO. OF NEUMANN-FLUX PROFILES                     * COM  570
C * MXNDP  - MAXIMUM NO. OF NEUMANN-FLUX DATA POINTS ON EACH PROFILE  * COM  575
C * NNNP   - NO. OF NEUMANN NODAL POINTS                              * COM  580
C * NNEL   - NO. OF NEUMANN ELEMENT SIDES                             * COM  585
C * NNPR   - NO. OF NEUMANN-FLUX PROFILES                             * COM  590
C * NNDP   - NO. OF NEUMANN-FLUX DATA POINTS ON EACH PROFILE          * COM  595
C *                                                                   * COM  600
C *                                                                   * COM  605
C * MAXMAT - MAXIMUM NO. OF MATERIAL TYPES (MEDIUM FORMATIONS)        * COM  610
C * MXSPPM - MAXIMUM NO. OF SOIL PARAMETERS PER MATERIAL, I. E. THE   * COM  615
C *          MAXIMUM NO. OF DATA POINTS ON SOIL CHARACTERISTIC CURVES * COM  620
C * MXMPPM - MAXIMUM NO. OF MATERIAL PROPERTIES PER MATERIAL          * COM  625
C *                                                                   * COM  630
C * NMAT   - NO. OF MATERIAL TYPES THAT FORM THE REGION OF INTEREST   * COM  635
C * NMPPM  - NO. OF MATERIAL PROPERTIES PER MATERIAL                  * COM  640
C * NSPPM  - NO. OF SOIL PARAMETERS PER MATERIAL                      * COM  645
C *                                                                   * COM  650
C * ILUMP  - LUMPING CONTROL, 0-NO LUMPING, 1-LUMPING                 * COM  655
C * IMID   - MID-DIFFERENCE CONTROL, 0-NOT MID-DIFFERENCING, 1-YES    * COM  660
C * IPNTS  - IS POINTWISE ITERATION SOLUTION METHOD USED? 0-NO, 1-YES * COM  665
C *                                                                   * COM  670
C *                                                                   * COM  675
C * FRATE(I) - FLOW RATE THROUGH I-TH TYPE BOUNDARY SEGMENT           * COM  680
C *            I - 1 - DIRICHLET BOUNDARY                             * COM  685
C *            I - 2 - NEUMANN BOUNDARY                               * COM  690
C *            I - 3 - CAUCHY BOUNDARY                                * COM  695
C *            I - 4 - SEEPAGE BOUNDARY                               * COM  700
C *            I - 5 - INFILTRATION BOUNDARY                          * COM  705
C *            I - 6 - NUMERICAL LOSS                                 * COM  710
C *            I - 7 - ENTIRE BOUNDARY                                * COM  715
C *            I - 8 - ARTIFICIAL SOURCE INTO THE REGION              * COM  720
C *            I - 9 - INCREASE IN THE WHOLE REGION OF INTEREST       * COM  725
C * FLOW(I) - AMOUNT OF FLOW THROUGH I-TH TYPE BOUNDARY IN THE SETP   * COM  730
C * TFLOW(I) - ACCUMULATED AMOUNT OF FLOW THROUGH I-TH TYPE BOUNDARY  * COM  735
C * NOTE: POSITIVE VALUES MEAN OUT FROM THE REGION, NEGATIVE VALUES   * COM  740
C * MEAN INTO THE REGION.                                             * COM  745
C *                                                                   * COM  750
```

```
C * X(N) - X-COORDINATE OF N-TH NODAL POINT, (L): X(MAXNP)                 * COM  755
C * Z(N) - Z-COORDINATE OF N-TH NODAL POINT, (L): X(MAXNP)                 * COM  760
C * IE(M,I) - GLOBAL NODE NUMBER OF THE I-TH NODE OF M-TH ELEMENT IF       * COM  765
C *          I IS BETWEEN 1 AND 4                                          * COM  770
C * IE(M,5) - INTEGER TO INDICATE THE MATERIAL TYPE OF M-TH ELEMENT        * COM  775
C * NOTE: IE IS DIMENSIONED IE(MAXEL,5)                                    * COM  780
C *                                                                        * COM  785
C * C(N,1) - AN ARRAY TO STORE THE ASSEMBLED GLOBAL MATRIX                 * COM  790
C *          SHOULD BE DIMENSIONED AS C(MAXNP,MAXHBP)                      * COM  795
C * LLNODE(I,N) - GLOBAL NODE NUMBER OF THE I-TH COMPRESSED LOWER-LEFT*    COM  800
C *               NODE TO THE N-TH GLOBAL NODE                             * COM  805
C * R(N) - PRESSURE HEAD LOAD OF N-TH NODE OR PRESSURE HEAD SOLUTION       * COM  810
C *        FROM SUBROUTINE BANSOL, (L): R(MAXNP)                           * COM  815
C * H(N) - PRESSURE HEAD AT N-TH NODE, (L): H(MAXNP)                       * COM  820
C * HP(N) - PRESSURE HEAD OF N-TH NODE AT PREVIOUS TIME STEP, (L)          * COM  825
C *         SHOULD BE DIMENSIONED AS HP(MAXNP)                             * COM  830
C * HW(N) - ITERATE PRESSURE HEAD OF N-TH NODE, (L): HW(MAXNP)             * COM  835
C * HT(N) - TOTAL HEAD OF N-TH NODE , (L): HT(MAXNP)                       * COM  840
C * VX(N) - X-COMPONENT VELOCITY AT N-TH NODE, (L/T): VX(MAXNP)           * COM  845
C * VZ(N) - Z-COMPONENT VELOCITY AT N-TH NODE, (L/T): VZ(MAXNP)           * COM  850
C * NPCNV(N) - NODAL POINT OF N-TH NONCONVERGENT NODE: NPCNV(MAXNP)       * COM  855
C *                                                                        * COM  860
C * TH(I,M) - MOISTURE CONTENT AT I-TH NODE OF M-TH ELEMENT                * COM  865
C *           SHOULD BE DIMENSIONED AS TH(4,MAXEL)                         * COM  870
C * DTH(I,M) - DTH/DH - WATER CAPACITY OF I-TH NODE OF M-TH ELEMENT        * COM  875
C *            (1/L), SHOULD BE DIMENSIONED AS DTH(4,MAXEL)                * COM  880
C * AKX(I,M) - XX-COMPONENT OF THE SATURATED HYDRAULIC CONDUCTIVITY AT*    COM  885
C *            I-TH NODE OF M-TH ELEMENT, (L/T): AKX(4,MAXEL)              * COM  890
C * AKZ(I,M) - ZZ-COMPONENT OF THE SATURATED HYDRAULIC CONDUCTIVITY AT*    COM  895
C *            I-TH NODE OF M-TH ELEMENT, (L/T): AKZ(4,MAXEL)              * COM  900
C * AKXZ(I,M) - XZ-COMPONENT OF THE SATURATED HYDRAULIC CONDUCTIVITY       * COM  905
C *             AT I-TH NODE OF M-TH ELEMENT, (L/T): AKXZ(4,MAXEL)         * COM  910
C *                                                                        * COM  915
C *                                                                        * COM  920
C * DLB(I) - LENGTH OF I-TH BOUNDARY ELEMENT SIDE, (L): DLB(MAXBEL)        * COM  925
C * DCOSXB(I) - X-DIRECTIONAL COSINE OF I-TH BOUNDARY ELEMENT SIDE         * COM  930
C *             SHOULD BE DIMENSIONED AS DCOSXB(MAXBEL)                    * COM  935
C * DCOSZB(I) - Z-DIRECTIONAL COSINE OF I-TH BOUNDARY ELEMENT SIDE         * COM  940
C *             SHOULD BE DIMENSIONED AS DCOSZB(MAXBEL)                    * COM  945
C * BFLX(I) - BOUNDARY FLUX AT I-TH BOUNDARY NODE, (L**3/T/L)              * COM  950
C *           SHOULD BE DIMENSIONED AS BFLX(MAXBNP)                        * COM  955
C * BFLXP(I) - BOUNDARY FLUX AT PREVIOUS TIME OF I-TH BOUNDARY NODE        * COM  960
C *            SOULD BE DIMENSIONED AS BFLXP(MAXBNP)                       * COM  965
C * NBE(I) - ELEMENT NUMBER OF I-TH BOUNDARY ELEMENT SIDE                  * COM  970
C *          SHOULD BE DIMENSIONED AS NBE(MAXBEL)                          * COM  975
C * ISB(1,I) - GLOBAL NODAL NUMBER OF THE FIRST NODE OF I-TH BOUNDARY *    COM  980
C *            SIDE                                                        * COM  985
C * ISB(2,I) - GLOBAL NODAL NUMBER OF THE SECOND NODE OF I-TH BOUNDARY*    COM  990
C *            SIDE                                                        * COM  995
```

```
C * ISB(3,I) - ELEMENT NBE(I)'S LOCAL NODAL NUMBER OF THE FIRST NODE  * COM 1000
C *            OF I-TH BOUNDARY SIDE                                  * COM 1005
C * ISB(4,I) - ELEMENT NBE(I)'S LOCAL NODAL NUMBER OF THE SECOND NODE * COM 1010
C *            OF I-TH BOUNDARY SIDE                                  * COM 1015
C * NPB(I) - GLOBAL NODE NUMBER OF I-TH BOUNDARY NODE                 * COM 1020
C *                                                                   * COM 1025
C *                                                                   * COM 1030
C * SOS(I) - SOURCE STRENGTH OF I-TH I-TH SOURCE PROFILE,             * COM 1035
C *          ((L**3/T)/L**3): SOS(MXSPR).                             * COM 1040
C * SOSF(I,J) - SOURCE STRENGTH OF I-TH DATA POINT IN J-TH SOURCE VS  * COM 1045
C *             TIME PROFILE, (L**3/T/L**3): SOSF(MXSDP,MXSPR).       * COM 1050
C * TSOSF(I,J) - TIME OF I-TH DATA POINT IN J-TH SOURCE VS TIME       * COM 1055
C *              PROFILE, (T): TSOSF(MXSDP,MXSPR)                     * COM 1060
C * LES(MP) - GLOBAL ELEMENT NUMBER OF MP-TH SOURCE ELEMENT           * COM 1065
C * ISTYP(MP) - TYPE OF SOURCE PROFILE ASSIGNED TO MP-TH ELEMENT      * COM 1070
C *                                                                   * COM 1075
C * NPW(NI) - GLOBAL NODE NUMBER OF NI-TH WELL NODE                   * COM 1080
C * IWTYP(NI) - TYPE OF WELL SOURCE/SINK PROFILE ASSIGNED TO -TH WELL * COM 1085
C * WSS(I) - VALUE OF I-TH WELL SOURCE/SINK AT PRESENT TIME, (L**2/T) * COM 1090
C * WSSF(J,I) - WELL SOURCE/SINK FUNCTION OF J-TH DATA POINT OF I-TH  * COM 1095
C *             PROFILE, (L**3/T/L).                                  * COM 1100
C * TWSSF(J,I) - TIME OF J-TH DATA POINT OF I-TH WELL SOURCE/SINK     * COM 1105
C *              PROFILE, (T)                                         * COM 1110
C *                                                                   * COM 1115
C *                                                                   * COM 1120
C * DL(MP) - LENGTH OF MP-TH RS BOUNDARY ELEMENT SIDE, (L)            * COM 1125
C *          SHOULD BE DIMENSIONED AS DL(MXRSEL)                      * COM 1130
C * DCOSX(MP) - DIRECTIONAL COSINE WITH X-AXIS OF MP-TH RS SIDE       * COM 1135
C *             SHOULD BE DIMENSIONED AS DCOSX(MXRSEL)                * COM 1140
C * DCOSZ(MP) - DIRECTIONAL COSINE WITH Z-AXIS OF MP-TH RS SIDE       * COM 1145
C *             SHOULD BE DIMENSIONED AS DCOSZ(MXRSES)                * COM 1150
C * DCYFLX(NP) - DARCY FLUX THROUGH NP-TH RS NODE                     * COM 1155
C *              (L**3/T/L-IN-Y-DIRECTION)                            * COM 1160
C * FLX(NP) - RAINFALL RATE THROUGH NP-TH RS NODE                     * COM 1165
C *           (L**3/T/L-IN-Y-DIRECTION)                               * COM 1170
C * RSFLX(NP) - DARCY VELOCITY AT NP-TH RS NODE                       * COM 1175
C *             (L/T), SHOULD BE DIMENSIONED AS RSFLX(MXRSNP)         * COM 1180
C * HCON(NP) - PONDING DEPTH ALLOWED FOR THE NP-TH NODE OF RAINFALL/  * COM 1185
C *            EVAPORATION SEEPAGE NODES, (L): HCON(MXRSNP)           * COM 1190
C * HMIN(NP) - MINIMUM PRESSURE HEAD ALLOWED FOR THE NP-TH RAINFALL/  * COM 1195
C *            EVAPORATION SEEPAGE NODES, (L): HMIN(MXRSNP)           * COM 1200
C * NRSE(I) - ELEMENT NUMBER TO WHICH I-TH RS BOUNDARY                * COM 1205
C *           SIDE BELONG, SHOULD BE DIMENSIONED AS NRSE(MXRSEL)      * COM 1210
C * IS(1,I) - COMPRESSED RAINFALL/EVAPORATION-SEEPAGE NODAL NUMBER OF * COM 1215
C *           THE FIRST NODE OF I-TH RS SIDE                          * COM 1220
C * IS(2,I) - COMPRESSED RAINFALL/EVAPORATION-SEEPAGE NODAL NUMBER OF * COM 1225
C *           THE SECOND NODE OF I-TH RS SIDE                         * COM 1230
C * IS(3,I) - ELEMENT NRSE(I)'S LOCAL NODAL NUMBER OF THE FIRST NODE  * COM 1235
C *           OF I-TH RAINFALL/EVAPORATION-SEEPAGE SIDE               * COM 1240
```

```
C * IS(4,I) - ELEMENT NRSE(I)'S LOCAL NODAL NUMBER OF THE SECOND NODE * COM 1245
C *           OF I-TH RAINFALL/EVAPORATION-SEEPAGE SIDE               * COM 1250
C * NOTE: "IS" SHOULD BE DIMENSIONED AS IS(4,MXRSEL)                  * COM 1255
C * NPRS(NP) - GLOBAL NODAL POINT NUMBER OF NP-TH RS-NODE             * COM 1260
C *            SHOULD BE DIMENSIONED AS NPRS(MXRSNP)                  * COM 1265
C * NPCON(NP) - PONDING DEPTH DIRICHLET BOUNDARY CONDITION INDICATOR  * COM 1270
C *             FOR THE NP-TH RS NODE, 0 - THIS NP-TH NODE IS NOT A   * COM 1275
C *             PONDING DEPTH DIRICHLET NODE, GLOBAL NODE NUMBER OF   * COM 1280
C *             NP-TH RS NODE - THIS NP-TH RS NODE IS A PONDING DEPTH * COM 1285
C *             DIRICHLET NODE                                        * COM 1290
C * NPMIN(NP) - MINIMUM PRESSURE HEAD DIRICHLET BOUNDARY CONDITION    * COM 1295
C *             INDICATOR FOR THE NP-TH RS NODE, 0 - NP-TH RS NODE IS * COM 1300
C *             NOT A MINIMUM PRESSURE HEAD DIRICHLET NODE, GLOBAL    * COM 1305
C *             NODE NUMBER OF NP-TH RS NODE - THIS NP-TH RS NODE IS  * COM 1310
C *             PRESCRIBE MINIMUM PRESSURE HEAD DIRICHLET NODE        * COM 1315
C * NPFLX(NP) - NEUMANN BOUNDARY CONDITION INDICATOR FOR THE NP-TH    * COM 1320
C *             NODE OF RAINFALL(+)/EVAPORATION(-)-SEEPAGE NODE, 0 -  * COM 1325
C *             THIS NP-TH RS NODE IS NOT A NEUMANN NODE FOR THE      * COM 1330
C *             TIME STEP, GLOBAL NODE NUMBER OF NP-TH RS NODE - THIS * COM 1335
C *             NP-TH RS NODE IS A NEUMANN NODE                       * COM 1340
C * IRSTYP(NP) - TYPE OF RAINFALL PROFILE ASSIGNED TO NP-TH NODE OF   * COM 1345
C *              THE RAINFALL-SEEPAGE NODES                           * COM 1350
C * TRF(I,J) - TIME OF I-TH DATA POINT ON J-TH RAINFALL-VS-TIME       * COM 1355
C *            PROFILE, (T): TRF(MXRPAR,MXRFPR)                       * COM 1360
C * RF(I,J) - RAINFALL OF I-TH DATA POINT ON J-TH RAINFALL-VS-TIME    * COM 1365
C *           PROFILE, (L/T): RF(MXRPAR,MXRFPR)                       * COM 1370
C * RFALL(I) - VALUE OF RAINFALL AT THE PRESENT TIME OF I-TH PROFILE  * COM 1375
C *                                                                   * COM 1380
C *                                                                   * COM 1385
C * HDB(I) - VALUE OF TOTAL HEAD AT THE PRESENT TIME OF I-TH TOTAL    * COM 1390
C *          HEAD VS TIME PROFILE FOR DIRICHELT BOUNDARY,             * COM 1395
C *          SHOULD BE DIMENSIONED AS HDB(MXDPR)                      * COM 1400
C * HDBF(I,J) - TOTAL HEAD OF I-TH DATA POINT IN J-TH DIRICHLET TOTAL * COM 1405
C *             HEAD VS TIME PROFILE, (L): HDBF(MXDDP,MXDPR)          * COM 1410
C * THDBF(I,J) - TIME OF I-TH DATA POINT IN J-TH DIRICHLET TOTAL HEAD * COM 1415
C *              VS TIME PROFILE, (T): THDBF(MXDDP,MXDPR)             * COM 1420
C *                                                                   * COM 1425
C * NPDB(NP) - GLOBAL NODAL POINT NUMBER OF NP-TH DIRICHLET BOUNDARY  * COM 1430
C *            NODE : NPDB(MXDNP)                                     * COM 1435
C * IDTYP(NP) - TYPE OF DIRICHLET TOTAL HEAD VS TIME PROFILE TO BE    * COM 1440
C *             ASSIGNED TO NP-TH DIRICHLET NODE: IDTYP(MXDNP)        * COM 1445
C *                                                                   * COM 1450
C *                                                                   * COM 1455
C * QCB(I) - VALUE OF FLUX AT THE PRESENT TIME OF I-TH CAUCHY FLUX VS * COM 1460
C *          TIME PROFILE FOR CAUCHY BOUNDARY, ((L**3/T)/L**2)        * COM 1465
C *          QCB(MXCPR)                                               * COM 1470
C * QCBF(I,J) - FLUX OF I-TH DATA POINT IN J-TH CAUCHY FLUX VS TIME   * COM 1475
C *             PROFILE  FOR CAUCHY BOUNDARY, ((L**3/T)/L**2)         * COM 1480
C *             QCBF(MXCDP,MXCPR)                                     * COM 1485
```

```
C * TQCBF(I,J) - TIME OF I-TH DATA POINT IN J-TH CAUCHY FLUX VS TIME  * COM 1490
C *              PROFILE FOR CAUCHY BOUNDARY, (T)                     * COM 1495
C *              TQCBF(MXCDP,MXCPR)                                   * COM 1500
C *                                                                   * COM 1505
C * NPCB(NP) - GLOBAL NODAL POINT NUMBER OF NP-TH CAUCHY BOUNDARY NODE* COM 1510
C *            NPCB(MXCNP)                                            * COM 1515
C * ICTYP(NP) - TYPE OF CAYCHY FLUX VS TIME PROFILE TO BE ASSIGNED TO * COM 1520
C *             NP-TH CAUCHY NODE: ICTYP(MXCNP)                       * COM 1525
C *                                                                   * COM 1530
C * ISC(1,I) - COMPRESSED CAUCHY NODAL NUMBER OF THE FIRST NODE OF    * COM 1535
C *            I-TH CAYCHY BOUNDARY SIDE                              * COM 1540
C * ISC(2,I) - COMPRESSED CAUCHY NODAL NUMBER OF THE SECOND NODE OF   * COM 1545
C *            I-TH CAYCHY BOUNDARY SIDE                              * COM 1550
C ******* NOTE: "ISC" SHOULD BE DIMENSIONED AS ISC(2,MXCEL)           * COM 1555
C *                                                                   * COM 1560
C *                                                                   * COM 1565
C * QNB(I) - VALUE OF FLUX AT THE PRESENT TIME OF I-TH NEUMANN FLUX VS* COM 1570
C *          TIME PROFILE FOR NEUMANN BOUNDARY, ((L**3/T)/L**2)       * COM 1575
C *          QNB(MXNPR)                                               * COM 1580
C * QNBF(I,J) - FLUX OF I-TH DATA POINT IN J-TH NEUMANN FLUX VS TIME  * COM 1585
C *             PROFILE  FOR NEUMANN BOUNDARY, ((L**3/T)/L**2)        * COM 1590
C *             QNBF(MXNDP,MXNPR)                                     * COM 1595
C * TQNBF(I,J) - TIME OF I-TH DATA POINT IN J-TH NEUMANN FLUX VS TIME * COM 1600
C *              PROFILE FOR NEUMANN BOUNDARY, (T)                    * COM 1605
C *              TQNBF(MXNDP,MXNPR)                                   * COM 1610
C *                                                                   * COM 1615
C * NPNB(NP) - GLOBAL NODAL POINT NUMBER OF NP-TH NEUMANN BOUNDARY    * COM 1620
C *            NODE: NPNB(MXNNP)                                      * COM 1625
C * INTYP(NP) - TYPE OF NEUMANN FLUX VS TIME PROFILE TO BE ASSIGNED TO* COM 1630
C *             NP-TH NEUMANN NODE: INTYP(MXNNP)                      * COM 1635
C *                                                                   * COM 1640
C * ISN(1,I) - COMPRESSED NEUMANN NODAL NUMBER OF THE FIRST NODE OF   * COM 1645
C *            I-TH NEUMANN BOUNDARY SIDE                             * COM 1650
C * ISN(2,I) - COMPRESSED NEUMANN NODAL NUMBER OF THE SECOND NODE OF  * COM 1655
C *            I-TH NEUMANN BOUNDARY SIDE                             * COM 1660
C * ISN(3,I) - ELEMENT MNES(I)'S LOCAL NODAL NUMBER OF THE FIRST NODE * COM 1665
C *            OF I-TH NEUMANN BOUNDARY SIDE                          * COM 1670
C * ISN(4,I) - ELEMENT MNES(I)'S LOCAL NODAL NUMBER OF THE SECOND NODE* COM 1675
C *            OF I-TH NEUMANN BOUNDARY SIDE                          * COM 1680
C ******* NOTE: "ISN" SHOULD BE DIMENSIONED AS ISN(4,MXNEL)           * COM 1685
C *                                                                   * COM 1690
C * DLN(MP) - LENGTH OF MP-TH NEUMANN BOUNDARY ELEMENT SIDE, (L)      * COM 1695
C *           SHOULD BE DIMENSIONED AS DLN(MXNEL)                     * COM 1700
C * DCOSXN(MP) - DIRECTIONAL COSINE WITH X-AXIS OF MP-TH NEUMANN      * COM 1705
C *              ELEMENT SIDE, SHOULD BE DIMENSIONED AS DCOSXN(MXNEL) * COM 1710
C * DCOSZN(MP) - DIRECTIONAL COSINE WITH Z-AXIS OF MP-TH NEUMANN      * COM 1715
C *              ELEMENT SIDE, SHOULD BE DIMENSIONED AS DCOSZN(MXNEL) * COM 1720
C * MNES(I) - ELEMENT NUMBER TO WHICH I-TH NEUMANN BOUNDARY ELEMENT   * COM 1725
C *           SIDE BELONG, SHOULD BE DIMENSIONED AS MNES(MXNEL)       * COM 1730
C *                                                                   * COM 1735
```

```
C *                                                                  * COM 1740
C * PROP(J,I) - J-TH MATERIAL PROPERTY OF I-TH MATERIALS             * COM 1745
C *             J - 1 - MODIFIED COMPRESSIBILITY OF THE MEDIUM, (1/L) * COM 1750
C *             J - 2 - MODIFIED COMPRESSIBILITY OF WATER, (1/L)     * COM 1755
C *             J - 3 - EFFECTIVE POROSITY, (IN DECIMALS)            * COM 1760
C *             J - 4 - SATURATED X-COMPONENT HYDRAULIC CONDUCTIVITY * COM 1765
C *             J - 5 - SATURATED Z-COMPONENT HYDRAULIC CONDUCTIVITY * COM 1770
C *             J - 6 - SATURATED XZ-COMPONENT HYDRAULUC CONDUCTIVITY * COM 1775
C ******* NOTE: "PROP" SHOULD BE DIMENSIONED AS PROP(MXMPPM,MAXMAT)  * COM 1780
C *                                                                  * COM 1785
C *                                                                  * COM 1790
C * THPROP(J,I) - MOISTURE CONTENT OF J-TH DATA POINT ON SOIL        * COM 1795
C *               CHARACTERISTIC CURVE FOR I-TH MATERIAL             * COM 1800
C * AKPROP(J,I) - RELATIVE HYDRAULIC CONDUCTIVITY OF J-TH DATA POINT * COM 1805
C *               SOIL CHARACTERISTIC CURVE FOR I-TH MATERIAL        * COM 1810
C * CAPROP(J,I) - WATER CAPACITY OF J-TH DATA POINT ON SOIL          * COM 1815
C *               CHARACTERISTIC CURVE FOR I-TH MATERIAL             * COM 1820
C ******* NOTE: " "THPROP, AKPROP, AND CAPROP" SHOULD BE DIMENSIONED * COM 1825
C *******          AS (MXSPPM,MAXMAT)                               * COM 1830
C *                                                                  * COM 1835
C *                                                                  * COM 1840
C * KPR(I) - LINE PRINT CONTROL FOR I-TH TIME STEP, SIMILAR TO KPRO  * COM 1845
C * KDSK(I) - DISK STORAGE CONTROL FOR I-TH TIME STEP SIMILAR TO KDSK0* COM 1850
C ******* NOTE: BOTH KPR AND KDSK" SHOULD BE DIMENSIONED AS (MAXNTI) * COM 1855
C * TDTCH(I) - TIME OF I-TH TIME TO RESET TIME STEP SIZE - DELTO, (T) * COM 1860
C *            SHOULD BE DIMENSIONED AS TDTCH(MXNDTC)                * COM 1865
C *                                                                  * COM 1870
C *                                                                  * COM 1875
```

```
C                                                                       MAIN 005
C ------- MAIN PROGRAM OF FEMWATER                                      MAIN 010
C                                                                       MAIN 015
      IMPLICIT REAL*8(A-H,O-Z)                                          MAIN 020
C                                                                       MAIN 025
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI,MXNDTC     MAIN 030
      COMMON /CPIS/ JBAND                                               MAIN 035
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,IBAND,KGRAV,NTI,NDTCHG            MAIN 040
C                                                                       MAIN 045
      COMMON /CINTE/ NCYL,NITER,NSTRT,KSTR,KPRO,KDSKO,KSS,KSP,NPITER    MAIN 050
      COMMON /CREAL/DELT,CHNG,DELMAX,TMAX,DELTO,TOLA,TOLB,W,OME,OMI,TIMEMAIN 055
      COMMON /COORD/ CSFE,SNFE,CYLIND                                   MAIN 060
C                                                                       MAIN 065
      COMMON /CSOS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP                    MAIN 070
      COMMON /CWSS/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP                    MAIN 075
C                                                                       MAIN 080
      COMMON /RSBC/ MXRSEL,MXRSNP,MXRFPR,MXRPAR,NRSEL,NRSN,NRFPR,NRFPAR MAIN 085
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP                    MAIN 090
      COMMON /CCBC/ MXCNP,MXCEL,MXCPR,MXCDP,NCNP,NCEL,NCPR,NCDP         MAIN 095
      COMMON /CNBC/ MXNNP,MXNEL,MXNPR,MXNDP,NNNP,NNEL,NNPR,NNDP         MAIN 100
C                                                                       MAIN 105
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM                                MAIN 110
      COMMON /CMTL/ NMAT,NMPPM,NSPPM                                    MAIN 115
C                                                                       MAIN 120
      COMMON /OPT/ ILUMP,IMID,IPNTS,IVML                                MAIN 125
C                                                                       MAIN 130
      COMMON /CFLOW/ FRATE(10),FLOW(10),TFLOW(10)                       MAIN 135
C                                                                       MAIN 140
      DIMENSION X(3111),Z(3111),IE(3000,5),LLNODE(9,3111)               MAIN 145
C                                                                       MAIN 150
      DIMENSION C(3111,53),R(3111),RL(3111),RI(3111)                    MAIN 155
      DIMENSION H(3111),HP(3111),HW(3111),HT(3111),VX(3111),VZ(3111)    MAIN 160
      DIMENSION NPCNV(3111), TH(4,3000),DTH(4,3000)                     MAIN 165
      DIMENSION AKX(4,3000),AKZ(4,3000),AKXZ(4,3000)                    MAIN 170
C                                                                       MAIN 175
      DIMENSION DLB(220),DCOSXB(220),DCOSZB(220),NBE(220),ISB(4,220)    MAIN 180
      DIMENSION BFLX(220),BFLXP(220),NPB(220)                           MAIN 185
C                                                                       MAIN 190
      DIMENSION SOS(3),SOSF(4,3),TSOSF(4,3), ISTYP(11),LES(11)          MAIN 195
      DIMENSION WSS(5),WSSF(8,5),TWSSF(8,5), IWTYP(10),NPW(10)          MAIN 200
C                                                                       MAIN 205
      DIMENSION RFALL(3),RF(24,3),TRF(24,3)                             MAIN 210
      DIMENSION NPRS(61),IRFTYP(61), IS(4,60)                           MAIN 215
      DIMENSION DL(60),DCOSX(60),DCOSZ(60),NRSE(60),RSFLX(61),DCYFLX(61)MAIN 220
      DIMENSION HCON(61),FLX(61),HMIN(61),NPCON(61),NPFLX(61),NPMIN(61) MAIN 225
C                                                                       MAIN 230
      DIMENSION HDB(1),HDBF(4,1),THDBF(4,1)                             MAIN 235
      DIMENSION NPDB(51),IDTYP(51)                                      MAIN 240
```

```
C                                                                       MAIN 245
      DIMENSION QCB(2),QCBF(4,2),TQCBF(4,2)                             MAIN 250
      DIMENSION NPCB(51),ICTYP(51), ISC(2,50)                           MAIN 255
C                                                                       MAIN 260
      DIMENSION QNB(1),QNBF(2,1),TQNBF(2,1)                             MAIN 265
      DIMENSION NPNB(61),INTYP(61), ISN(4,60)                           MAIN 270
      DIMENSION DLN(60),DCOSXN(60),DCOSZN(60),MNES(60)                  MAIN 275
C                                                                       MAIN 280
      DIMENSION PROP(6,3)                                               MAIN 285
      DIMENSION THPROP(52,3),AKPROP(52,3),HPROP(52,3),CAPROP(52,3)      MAIN 290
C                                                                       MAIN 295
      DIMENSION KPR(500),KDSK(500),TDTCH(20)                            MAIN 300
C                                                                       MAIN 305
C                                                                       MAIN 310
C ------- PASS THE PROGRAM TO GW2DXZ                                    MAIN 315
C                                                                       MAIN 320
      CALL GW2DXZ(X,Z,IE, C,R,H,HP,HW,HT,TH,DTH,VX,VZ,AKX,AKZ,          MAIN 325
     1 AKXZ,NPCNV, DLB,DCOSXB,DCOSZB,BFLX,BFLXP,NBE,ISB,NPB,            MAIN 330
     2 SOS,SOSF,TSOSF,LES,ISTYP, DL,DCOSX,DCOSZ,NRSE,DCYFLX,RSFLX,HCON, MAIN 335
     3 NPCON,NPFLX, RFALL,RF,TRF,FLX,NPRS,IRFTYP,IS, HDB,HDBF,THDBF,    MAIN 340
     4 NPDB,IDTYP, QCB,QCBF,TQCBF,NPCB,ICTYP,ISC, QNB,QNBF,TQNBF,       MAIN 345
     5 NPNB,INTYP,ISN,DLN,DCOSXN,DCOSZN,MNES,                           MAIN 350
     6 PROP,HPROP,AKPROP,THPROP,CAPROP, KPR,KDSK,TDTCH, HMIN,NPMIN,     MAIN 355
     7 NPW,IWTYP,WSS,TWSSF,WSSF, RL,RI,LLNODE)                          MAIN 360
C                                                                       MAIN 365
      STOP                                                              MAIN 370
      END                                                               MAIN 375
```

```
      BLOCK DATA                                                        BLOC 005
C                                                                       BLOC 010
      IMPLICIT REAL*8(A-H,O-Z)                                          BLOC 015
C                                                                       BLOC 020
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI,MXNDTC     BLOC 025
      COMMON /CPIS/ JBAND                                               BLOC 030
      COMMON /CSOS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP                    BLOC 035
      COMMON /CWSS/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP                    BLOC 040
      COMMON /RSBC/ MXRSEL,MXRSNP,MXRFPR,MXRPAR,NRSEL,NRSN,NRFPR,NRFPAR BLOC 045
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP                    BLOC 050
      COMMON /CCBC/ MXCNP,MXCEL,MXCPR,MXCDP,NCNP,NCEL,NCPR,NCDP         BLOC 055
      COMMON /CNBC/ MXNNP,MXNEL,MXNPR,MXNDP,NNNP,NNEL,NNPR,NNDP         BLOC 060
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM                                BLOC 065
C                                                                       BLOC 070
      DATA MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI/528,595,199,200,18,  BLOC 075
     > 500/                                                             BLOC 080
      DATA JBAND/9/                                                     BLOC 085
      DATA MXNDTC/20/                                                   BLOC 090
```

```
      DATA MXSEL,MXSPR,MXSDP,MXWNP,MXWPR,MXWDP/12,2,4,10,6,4/          BLOC 095
      DATA MXRSEL,MXRSNP,MXRFPR,MXRPAR/99,100,3,20/                    BLOC 100
      DATA MXDNP,MXDPR,MXDDP/40,2,4/                                   BLOC 105
      DATA MXCNP,MXCEL,MXCPR,MXCDP/30,29,2,4/                          BLOC 110
      DATA MXNNP,MXNEL,MXNPR,MXNDP/50,49,2,4/                          BLOC 115
      DATA MAXMAT,MXSPPM,MXMPPM/3,52,6/                                BLOC 120
C                                                                      BLOC 125
      END                                                              BLOC 130


      SUBROUTINE GW2DXZ(X,Z,IE, C,R,H,HP,HW,HT,TH,DTH,VX,VZ,AKX,AKZ,   GW2D 005
     1 AKXZ,NPCNV, DLB,DCOSXB,DCOSZB,BFLX,BFLXP,NBE,ISB,NPB,           GW2D 010
     2 SOS,SOSF,TSOSF,LES,ISTYP, DL,DCOSX,DCOSZ,NRSE,DCYFLX,RSFLX,HCON, GW2D 015
     3 NPCON,NPFLX, RFALL,RF,TRF,FLX,NPRS,IRFTYP,IS, HDB,HDBF,THDBF,   GW2D 020
     4 NPDB,IDTYP, QCB,QCBF,TQCBF,NPCB,ICTYP,ISC, QNB,QNBF,TQNBF,      GW2D 025
     5 NPNB,INTYP,ISN,DLN,DCOSXN,DCOSZN,MNES,                          GW2D 030
     6 PROP,HPROP,AKPROP,THPROP,CAPROP, KPR,KDSK,TDTCH, HMIN,NPMIN,    GW2D 035
     7 NPW,IWTYP,WSS,TWSSF,WSSF, RL,RI,LLNODE)                         GW2D 040
C                                                                      GW2D 045
      IMPLICIT REAL*8(A-H,O-Z)                                         GW2D 050
      REAL*4 SUBHD                                                     GW2D 055
C                                                                      GW2D 060
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI,MXNDTC    GW2D 065
      COMMON /CPIS/ JBAND                                              GW2D 070
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,IBAND,KGRAV,NTI,NDTCHG           GW2D 075
C                                                                      GW2D 080
      COMMON /CINTE/ NCYL,NITER,NSTRT,KSTR,KPRO,KDSKO,KSS,KSP,NPITER   GW2D 085
      COMMON /CREAL/DELT,CHNG,DELMAX,TMAX,DELTO,TOLA,TOLB,W,OME,OMI,TIMEGW2D 090
      COMMON /COORD/ CSFE,SNFE,CYLIND                                  GW2D 095
C                                                                      GW2D 100
      COMMON /CSOS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP                   GW2D 105
      COMMON /CWSS/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP                   GW2D 110
C                                                                      GW2D 115
      COMMON /RSBC/ MXRSEL,MXRSNP,MXRFPR,MXRPAR,NRSEL,NRSN,NRFPR,NRFPAR GW2D 120
C                                                                      GW2D 125
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP                   GW2D 130
      COMMON /CCBC/ MXCNP,MXCEL,MXCPR,MXCDP,NCNP,NCEL,NCPR,NCDP        GW2D 135
      COMMON /CNBC/ MXNNP,MXNEL,MXNPR,MXNDP,NNNP,NNEL,NNPR,NNDP        GW2D 140
C                                                                      GW2D 145
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM                               GW2D 150
C                                                                      GW2D 155
      COMMON /OPT/ ILUMP,IMID,IPNTS,IVML                               GW2D 160
C                                                                      GW2D 165
      COMMON /CFLOW/ FRATE(10),FLOW(10),TFLOW(10)                      GW2D 170
C                                                                      GW2D 175
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)                          GW2D 180
```

```
C
      DIMENSION C(MAXNP,MAXHBP),R(MAXNP),H(MAXNP),HP(MAXNP),HW(MAXNP),
     1 HT(MAXNP),TH(4,MAXEL),DTH(4,MAXEL),VX(MAXNP),VZ(MAXNP),
     2 AKX(4,MAXEL),AKZ(4,MAXEL),AKXZ(4,MAXEL), NPCNV(MAXNP)
      DIMENSION RL(MAXNP),RI(MAXNP),LLNODE(JBAND,MAXNP)
C
      DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),BFLX(MAXBNP),
     > BFLXP(MAXBNP),NBE(MAXBEL),ISB(4,MAXBEL),NPB(MAXBNP)
C
      DIMENSION SOS(MXSPR),SOSF(MXSDP,MXSPR),TSOSF(MXSDP,MXSPR)
      DIMENSION LES(MXSEL),ISTYP(MXSEL)
      DIMENSION NPW(MXWNP),IWTYP(MXWNP)
      DIMENSION WSS(MXWPR),WSSF(MXWDP,MXWPR),TWSSF(MXWDP,MXWPR)
C
      DIMENSION DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),NRSE(MXRSEL)
      DIMENSION DCYFLX(MXRSNP),RSFLX(MXRSNP),HCON(MXRSNP)
      DIMENSION NPCON(MXRSNP),NPFLX(MXRSNP)
      DIMENSION RFALL(MXRFPR),RF(MXRPAR,MXRFPR),TRF(MXRPAR,MXRFPR)
      DIMENSION FLX(MXRSNP),NPRS(MXRSNP),IRFTYP(MXRSNP),IS(4,MXRSEL)
      DIMENSION HMIN(MXRSNP),NPMIN(MXRSNP)
C
      DIMENSION HDB(MXDPR),HDBF(MXDDP,MXDPR),THDBF(MXDDP,MXDPR)
      DIMENSION NPDB(MXDNP),IDTYP(MXDNP)
C
      DIMENSION QCB(MXCPR),QCBF(MXCDP,MXCPR),TQCBF(MXCDP,MXCPR)
      DIMENSION NPCB(MXCNP),ICTYP(MXCNP), ISC(2,MXCEL)
C
      DIMENSION QNB(MXNPR),QNBF(MXNDP,MXNPR),TQNBF(MXNDP,MXNPR)
      DIMENSION NPNB(MXNNP),INTYP(MXNNP), ISN(4,MXNEL)
      DIMENSION DLN(MXNEL),DCOSXN(MXNEL),DCOSZN(MXNEL),MNES(MXNEL)
C
      DIMENSION PROP(MXMPPM,MAXMAT)
      DIMENSION THPROP(MXSPPM,MAXMAT),AKPROP(MXSPPM,MAXMAT),
     1 HPROP(MXSPPM,MAXMAT),CAPROP(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),IMOD,IBUG,ICHNG
C
      IF (NPROB.LE.0) GO TO 990
C
      PRINT 1000, NPROB,(TITLE(I),I=1,9),IMOD,IBUG,ICHNG
```

```
C                                                                       GW2D 440
C ------- READ AND PRINT INPUT DATA BY CALLING DATAIN                   GW2D 445
C                                                                       GW2D 450
      KOUT=0                                                            GW2D 455
C                                                                       GW2D 460
      CALL DATAIN(TITLE,NPROB,                                          GW2D 465
     1 KPR,KDSK,TDTCH, PROP, HPROP,THPROP,AKPROP,CAPROP, X,Z,IE,        GW2D 470
     2 H,HT,TH,VX,VZ, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,                    GW2D 475
     3 RF,TRF,NPRS,IRFTYP,IS,NRSE,DL,DCOSX,DCOSZ,HCON,NPCON,NPFLX,      GW2D 480
     4 HDBF,THDBF,NPDB,IDTYP, QCBF,TQCBF,NPCB,ICTYP,ISC,                GW2D 485
     5 QNBF,TQNBF,NPNB,INTYP,ISN,MNES,DLN,DCOSXN,DCOSZN, MAXDIF,ISTOP,  GW2D 490
     6 HMIN,NPMIN, NPW,IWTYP,WSSF,TWSSF, LES,ISTYP,SOSF,TSOSF)          GW2D 495
C                                                                       GW2D 500
      IF(IPNTS.EQ.1) CALL LNDGEN(LLNODE,IE,MAXNP,MAXEL,NNP,NEL)         GW2D 505
C                                                                       GW2D 510
      KDIG=NSTRT                                                        GW2D 515
      IF (ISTOP.GT.0) GO TO 990                                         GW2D 520
C                                                                       GW2D 525
C ------- COMPUTE BAND-WIDTH VARIABLES                                  GW2D 530
C                                                                       GW2D 535
      IHALFB=MAXDIF                                                     GW2D 540
      IBAND=2*IHALFB+1                                                  GW2D 545
      IHBP=IHALFB+1                                                     GW2D 550
      IF (IHBP.GT.MAXHBP .AND. IPNTS.EQ.0) GO TO 910                    GW2D 555
C                                                                       GW2D 560
C ------- PREPARE INITIAL OR PRE-INITIAL VARIABLES                      GW2D 565
C                                                                       GW2D 570
      IF(NDNP.NE.0) CALL INTERP(HDB,THDBF,HDBF,TIME,                    GW2D 575
     1              MXDPR,MXDDP,NDPR,NDDP)                              GW2D 580
      IF(NCNP.NE.0) CALL INTERP(QCB,TQCBF,QCBF,TIME,                    GW2D 585
     1              MXCPR,MXCDP,NCPR,NCDP)                              GW2D 590
      IF(NNNP.NE.0) CALL INTERP(QNB,TQNBF,QNBF,TIME,                    GW2D 595
     1              MXNPR,MXNDP,NNPR,NNDP)                              GW2D 600
C                                                                       GW2D 605
      IF(NSEL.NE.0) CALL INTERP(SOS,TSOSF,SOSF,TIME,MXSPR,MXSDP,        GW2D 610
     1 NSPR,NSDP)                                                       GW2D 615
      IF(NWNP.NE.0) CALL INTERP(WSS,TWSSF,WSSF,TIME,MXWPR,MXWDP,        GW2D 620
     1 NWPR,NWDP)                                                       GW2D 625
C                                                                       GW2D 630
C ------- PUT DIRICHLET BOUNDARY VALUES TO INITIAL CONDITIONS           GW2D 635
C                                                                       GW2D 640
      DO 130 I=1,NDNP                                                   GW2D 645
      NP=NPDB(I)                                                        GW2D 650
      ITYP=IDTYP(I)                                                     GW2D 655
      H(NP)=HDB(ITYP)-(Z(NP)*CSFE+X(NP)*SNFE)*DFLOAT(KGRAV)             GW2D 660
  130 CONTINUE                                                          GW2D 665
C                                                                       GW2D 670
      CALL SPROP(IE,H,TH,DTH,AKX,AKZ,AKXZ, PROP,THPROP,AKPROP,HPROP,    GW2D 675
     >           CAPROP, MAXEL,MAXNP, NEL,KSP)                          GW2D 680
C                                                                       GW2D 685
      CALL VELT(X,Z,IE, C,H,HT,VX,VZ,AKX,AKZ,AKXZ, MAXEL,MAXNP,MAXHBP)  GW2D 690
```

```
      KFLOW=-1                                                          GW2D 695
C                                                                       GW2D 700
      CALL SFLOW(X,Z,IE, H,HP,VX,VZ,TH,DTH,                             GW2D 705
     1 DLB,DCOSXB,DCOSZB,BFLX,BFLXP,ISB,NBE,NPB, LES,SOS,ISTYP,         GW2D 710
     2 NPRS,NPDB,NPCB,NPNB,  PROP, DELT, KFLOW, NPW,WSS,IWTYP)          GW2D 715
C                                                                       GW2D 720
      DO 140 I=1,9                                                      GW2D 725
      IF(I.EQ.9) GO TO 140                                              GW2D 730
      FLOW(I)=0.0                                                       GW2D 735
      TFLOW(I)=0.0                                                      GW2D 740
  140 CONTINUE                                                          GW2D 745
      FLOW(9)=0.0                                                       GW2D 750
C                                                                       GW2D 755
C ------- PRINT INITIAL OR PRE-INITIAL VARIABLES                        GW2D 760
C                                                                       GW2D 765
      KDIAG=0                                                           GW2D 770
C                                                                       GW2D 775
      CALL PRINTT(VX,VZ,H,HT,TH, NPB,BFLX, NPRS,RSFLX,NPCON,NPFLX,      GW2D 780
     1 SUBHD(1,I), TIME,DELT,KPRO,KOUT,KDIAG,-1, NPMIN)                 GW2D 785
C                                                                       GW2D 790
      CALL STORE(X,Z,IE,H,HT,TH,VX,VZ,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,    GW2D 795
     1 NPCON,NPFLX,TITLE,TIME,NPROB,NSTRT,NPMIN)                        GW2D 800
C                                                                       GW2D 805
      IF(IMOD.EQ.0) RETURN                                              GW2D 810
C                                                                       GW2D 815
      IF(KSTR.EQ.1 .AND. KSS.EQ.1 .AND. NSTRT.EQ.0 .AND. KDSKO.EQ.1)    GW2D 820
     > CALL STORE(X,Z,IE, H,HT,TH,VX,VZ,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,  GW2D 825
     1 NPCON,NPFLX, TITLE, TIME,     NPROB,NSTRT,NPMIN)                 GW2D 830
C                                                                       GW2D 835
      IF (KSS.NE.0) GO TO 500                                           GW2D 840
C                                                                       GW2D 845
C ------- PERFORM STEADY-STATE CALCULATION                              GW2D 850
C                                                                       GW2D 855
      IF (NRSN.EQ.0) GO TO 170                                          GW2D 860
C                                                                       GW2D 865
      DO 150 NPP=1,NRSN                                                 GW2D 870
      NPCON(NPP)=NPRS(NPP)                                              GW2D 875
      NPMIN(NPP)=0                                                      GW2D 880
  150 NPFLX(NPP)=0                                                      GW2D 885
C                                                                       GW2D 890
      NCHG=-1                                                           GW2D 895
      CALL BCPREP(IE, H,VX,VZ, DL,DCOSX,DCOSZ,DCYFLX,FLX,RSFLX,         GW2D 900
     > HCON,IS,NPRS,NPCON,NPFLX,IRFTYP,TRF,RF,RFALL. TIME,NCHG,         GW2D 905
     2 HMIN,NPMIN,X,Z)                                                  GW2D 910
C                                                                       GW2D 915
  170 DO 180 NP=1,NNP                                                   GW2D 920
  180 HP(NP)=H(NP)                                                      GW2D 925
C                                                                       GW2D 930
      KDIG=KDIG+1                                                       GW2D 935
      IF(IBUG.NE.0) PRINT 10400,KDIG,TIME,DELT                          GW2D 940
```

```
C                                                                       GW2D 945
C ------- ITERATION LOOP ON SEEPAGE-RAINFALL BOUNDARY CONDITIONS BEGINS GW2D 950
C                                                                       GW2D 955
      DO 390 ICY=1,NCYL                                                 GW2D 960
C                                                                       GW2D 965
      DO 210 NP=1,NNP                                                   GW2D 970
      HW(NP)=OME*H(NP)+(1.0D0-OME)*HP(NP)                               GW2D 975
      IF(IPNTS.EQ.1) RI(NP)=HW(NP)                                      GW2D 980
  210 CONTINUE                                                          GW2D 985
C                                                                       GW2D 990
C ------- ITERATION LOOP ON THE NON-LINEAR EQUATION BEGINS              GW2D 995
C                                                                       GW2D1000
      IF(IBUG.NE.0) PRINT 10401, ICY                                    GW2D1005
C                                                                       GW2D1010
C ------- PUT DIRICHLET BOUNDARY VALUES OF THE VARIABLE BOUNDARY        GW2D1015
C ------- INTO H, RI, AND HW                                            GW2D1020
C                                                                       GW2D1025
      IF(NRSEL.EQ.0) GO TO 240                                          GW2D1030
      DO 230 NPP=1,NRSN                                                 GW2D1035
      NI=NPMIN(NPP)                                                     GW2D1040
      IF(NI.EQ.0) GO TO 220                                             GW2D1045
      H(NI)=HMIN(NPP)                                                   GW2D1050
      RI(NI)=HMIN(NPP)                                                  GW2D1055
      HW(NI)=HMIN(NPP)                                                  GW2D1060
      GO TO 230                                                         GW2D1065
  220 NI=NPCON(NPP)                                                     GW2D1070
      IF(NI.EQ.0) GO TO 230                                             GW2D1075
      H(NI)=HCON(NPP)                                                   GW2D1080
      RI(NI)=HCON(NPP)                                                  GW2D1085
      HW(NI)=HCON(NPP)                                                  GW2D1090
  230 CONTINUE                                                          GW2D1095
  240 CONTINUE                                                          GW2D1100
C                                                                       GW2D1105
      DO 350 IT=1,NITER                                                 GW2D1110
C                                                                       GW2D1115
C ------- EVALUATE SOIL PROPERTIES FOR PREVIOUS ITERATE                 GW2D1120
C                                                                       GW2D1125
      CALL SPROP(IE,HW,TH,DTH,AKX,AKZ,AKXZ, PROP,THPROP,AKPROP,         GW2D1130
     >       HPROP,CAPROP, MAXEL,MAXNP, NEL,KSP)                        GW2D1135
C                                                                       GW2D1140
C ------- ASSEMBLE GLOBAL MATRIX                                        GW2D1145
C                                                                       GW2D1150
      CALL ASEMBL(X,Z,IE, C,R,HW,HP,TH,DTH,AKX,AKZ,AKXZ, PROP,          GW2D1155
     > MAXNP,MAXEL,MAXHBP, MAXMAT,MXMPPM, KSS,W,DELT, SOS,LES,ISTYP,    GW2D1160
     3        WSS,NPW,IWTYP,LLNODE)                                    GW2D1165
C                                                                       GW2D1170
C ------- APPLY STEADY-STATE BOUNDARY CONDITIONS                        GW2D1175
C                                                                       GW2D1180
      CALL BC(C,R, X,Z,IE,LLNODE, AKX,AKZ,AKXZ, FLX,HCON,NPCON,NPFLX,   GW2D1185
     1       HDB,IDTYP,NPDB, QCB,ICTYP,NPCB,ISC, QNB,INTYP,NPNB,ISN,    GW2D1190
     2       DLN,DCOSXN,DCOSZN,MNES, KSS, HMIN,NPMIN)                   GW2D1195
```

```
C
      IF(IPNTS.EQ.1) GO TO 250
C
      CALL BANSOL(1,C,R,NNP,IHBP,MAXNP,MAXHBP)
      CALL BANSOL(2,C,R,NNP,IHBP,MAXNP,MAXHBP)
      GO TO 300
C
  250 CALL PISS(R,RL,RI,C,OMI,TOLA,LLNODE,NNP,NPITER,MAXNP,MAXHBP,IBUG,
     1 1)
C
C ------- OBTAIN MAXIMUM RELATIVE DEVIATION FROM PREVIOUS ITERATE
C
  300 CONTINUE
      NPP=0
      RD=-1.0D0
      RES=-1.0D0
      DO 320 NP=1,NNP
      RESNP=DABS(R(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*R(NP)+(1.0D0-OME)*H(NP)
      IF(IPNTS.EQ.1) 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                                                       GW2D1455
      PRINT 10600, (NPCNV(NPP),NPP=1,NNCVN)                             GW2D1460
C                                                                       GW2D1465
  360 IF(ICHNG.EQ.0) GO TO 380                                          GW2D1470
      IF(NRSN.EQ.0) GO TO 380                                           GW2D1475
C                                                                       GW2D1480
C ------- PRINT RAINFALL-SEEPAGE B. C. CHANGE INFORMATION               GW2D1485
C                                                                       GW2D1490
      PRINT 10402, ICY                                                  GW2D1495
      DO 370 I=1,NRSN                                                   GW2D1500
      NP=NPRS(I)                                                        GW2D1505
      PRINT 10403, I,NP,NPCON(I),HCON(I),NPMIN(I),HMIN(I),NPFLX(I),     GW2D1510
     1   FLX(I),DCYFLX(I)                                               GW2D1515
  370 CONTINUE                                                          GW2D1520
C                                                                       GW2D1525
C ------- CALCULATE DARCY'S VELOCITY                                    GW2D1530
C                                                                       GW2D1535
  380 CALL SPROP(IE, H,TH,DTH,AKX,AKZ,AKXZ, PROP, THPROP,AKPROP,        GW2D1540
     >    HPROP,CAPROP, MAXEL,MAXNP, NEL,KSP)                           GW2D1545
C                                                                       GW2D1550
      CALL VELT(X,Z,IE, C,H,HT,VX,VZ,AKX,AKZ,AKXZ, MAXEL,MAXNP,MAXHBP)  GW2D1555
C                                                                       GW2D1560
      IF(NRSN .EQ. 0) GO TO 440                                         GW2D1565
C                                                                       GW2D1570
C ------- PREPARE BOUNDARY CONDITIONS ON THE VARIABLE-TYPE BOUNDARY FOR GW2D1575
C ------- NEXT CYCLE COMPUTATIONS.                                      GW2D1580
C                                                                       GW2D1585
      CALL BCPREP(IE, H,VX,VZ, DL,DCOSX,DCOSZ, DCYFLX,FLX,RSFLX,        GW2D1590
     > HCON,IS,NPRS,NPCON,NPFLX,IRFTYP,TRF,RF,RFALL, TIME,NCHG,         GW2D1595
     2 HMIN,NPMIN,X,Z)                                                  GW2D1600
C                                                                       GW2D1605
      IF(NCHG.EQ.0) GO TO 440                                           GW2D1610
C                                                                       GW2D1615
  390 CONTINUE                                                          GW2D1620
C                                                                       GW2D1625
C ------- END OF ITERATION LOOP ON SEEPACE-RAINFALL BOUNDARY CONDITIONS GW2D1630
C                                                                       GW2D1635
      PRINT 10610, ICY,IT,NCYL,NITER,RES,RD,NNCVN                       GW2D1640
  440 IF(NNCVN.EQ.0) GO TO 445                                          GW2D1645
      PRINT 10610, ICY,IT,NCYL,NITER,RES,RD,NNCVN                       GW2D1650
C                                                                       GW2D1655
C ------- COMPUTE FLUXES THROUGH ALL TYPES OF BOUNDARIES.               GW2D1660
C                                                                       GW2D1665
  445 KFLOW=0                                                           GW2D1670
      CALL SFLOW(X,Z,IE, H,HP,VX,VZ,TH,DTH,                             GW2D1675
     1 DLB,DCOSXB,DCOSZB,BFLX,BFLXP,ISB,NBE,NPB, LES,SOS,ISTYP,         GW2D1680
     2 NPRS,NPDB,NPCB,NPNB,  PROP, DELT, KFLOW, NPW,WSS,IWTYP)          GW2D1685
C                                                                       GW2D1690
      DO 450 I=1,9                                                      GW2D1695
      IF(I.EQ.9) GO TO 450                                              GW2D1700
```

```
      FLOW(I)=0.0                                                       GW2D1705
      TFLOW(I)=0.0                                                      GW2D1710
  450 CONTINUE                                                          GW2D1715
      FLOW(9)=0.0                                                       GW2D1720
C                                                                       GW2D1725
C ------- PRINT STEADY-STATE VARIABLES                                  GW2D1730
C                                                                       GW2D1735
      CALL PRINTT(VX,VZ,H,HT,TH, NPB,BFLX, NPRS,RSFLX,NPCON,NPFLX,      GW2D1740
     1 SUBHD(1,2), TIME,DELT, KPRO,KOUT,KDIAG,0, NPMIN)                 GW2D1745
C                                                                       GW2D1750
      IF(KSTR.EQ.1 .AND. KDSKO.EQ.1) CALL STORE(X,Z,IE,                 GW2D1755
     > H,HT,TH,VX,VZ,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,NPCON,NPFLX,         GW2D1760
     > TITLE, TIME,     NPROB,NSTRT, NPMIN)                             GW2D1765
C                                                                       GW2D1770
      IF (NTI.EQ.0) GO TO 100                                           GW2D1775
C                                                                       GW2D1780
      KSS=1                                                             GW2D1785
C                                                                       GW2D1790
C ------- PERFORM TRANSIENT-STATE CALCULATION                           GW2D1795
C                                                                       GW2D1800
  500 IF (NPSN.EQ.0) GO TO 550                                          GW2D1805
      IF (NSTRT.GT.0) GO TO 520                                         GW2D1810
C                                                                       GW2D1815
      DO 510 NPP=1,NPSN                                                 GW2D1820
      NPCON(NPP)=NPRS(NPP)                                              GW2D1825
      NPMIN(NPP)=0                                                      GW2D1830
  510 NPFLX(NPP)=0                                                      GW2D1835
C                                                                       GW2D1840
  520 NCHG=-1                                                           GW2D1845
C                                                                       GW2D1850
  550 TIME=TIME+DELT                                                    GW2D1855
      W1=W                                                              GW2D1860
      W2=1.-W                                                           GW2D1865
      KFLOW=1                                                           GW2D1870
      TFLOW(9)=0.0                                                      GW2D1875
C                                                                       GW2D1880
C ------- BEGIN THE TIME-MARCHING LOOP                                  GW2D1885
C                                                                       GW2D1890
      IDELT=0                                                           GW2D1895
      DO 890 ITM=1,NTI                                                  GW2D1900
      ITMITM=ITM                                                        GW2D1905
C                                                                       GW2D1910
      NCHG=-1                                                           GW2D1915
      CALL BCPREP(IE, H,VX,VZ, DL,DCOSX,DCOSZ,DCYFLX,FLX,RSFLX,         GW2D1920
     > HCON,IS,NPRS,NPCON,NPFLX,IRFTYP,TRF,RF,RFALL, TIME,NCHG,         GW2D1925
     2 HMIN,NPMIN,X,Z)                                                  GW2D1930
C                                                                       GW2D1935
C ------- PREPARE TRANSIENT BOUNDARY CONDITIONS AND SOURCE FOR THE STEP GW2D1940
C                                                                       GW2D1945
```

```
      IF(NDNP.NE.0) CALL INTERP(HDB,THDBF,HDBF,TIME,                    GW2D1950
     1                  MXDPR,MXDDP,NDPR,NDDP)                          GW2D1955
      IF(NCNP.NE.0) CALL INTERP(QCB,TQCBF,QCBF,TIME,                    GW2D1960
     1                  MXCPR,MXCDP,NCPR,NCDP)                          GW2D1965
      IF(NNNP.NE.0) CALL INTERP(QNB,TQNBF,QNBF,TIME,                    GW2D1970
     1                  MXNPR,MXNDP,NNPR,NNDP)                          GW2D1975
C                                                                       GW2D1980
      IF(NSEL.NE.0) CALL INTERP(SOS,TSOSF,SOSF,TIME,MXSPR,MXSDP,        GW2D1985
     1 NSPR,NSDP)                                                       GW2D1990
      IF(NWNP.NE.0) CALL INTERP(WSS,TWSSF,WSSF,TIME,MXWPR,MXWDP,        GW2D1995
     1 NWPR,NWDP)                                                       GW2D2000
C                                                                       GW2D2005
      DO 570 NP=1,NNP                                                   GW2D2010
      R(NP)=H(NP)                                                       GW2D2015
      HP(NP)=H(NP)                                                      GW2D2020
  570 CONTINUE                                                          GW2D2025
C                                                                       GW2D2030
      KDIG=KDIG+1                                                       GW2D2035
      IF(IBUG.NE.0 .AND. KPR(ITM).NE.0) PRINT 10400, KDIG,TIME,DELT     GW2D2040
C                                                                       GW2D2045
C ------- BEGIN ITERATION LOOP ON SEEPAGE-RAINFALL BOUNDARY CONDITIONS  GW2D2050
C                                                                       GW2D2055
      DO 690 ICY=1,NCYL                                                 GW2D2060
      IF(IBUG.NE.0 .AND. KPR(ITM).NE.0) PRINT 10401, ICY                GW2D2065
C                                                                       GW2D2070
      DO 580 NP=1,NNP                                                   GW2D2075
      H(NP)=OME*R(NP)+(1.0D0-OME)*H(NP)                                 GW2D2080
      IF(IPNTS.EQ.1) RI(NP)=H(NP)                                       GW2D2085
      HW(NP)=W1*(OME*H(NP)+(1.0D0-OME)*HP(NP))+W2*HP(NP)                GW2D2090
  580 CONTINUE                                                          GW2D2095
C                                                                       GW2D2100
C ------- BEGIN ITERATION LOOP ON THE NON-LINEAR EQUATION               GW2D2105
C                                                                       GW2D2110
C ------- PUT DIRICHLET BOUNDARY VALUES OF THE VARIABLE BOUNDARY        GW2D2115
C ------- INTO H, RI, AND HW                                            GW2D2120
C                                                                       GW2D2125
      IF(NRSEL.EQ.0) GO TO 595                                          GW2D2130
      DO 590 NPP=1,NRSN                                                 GW2D2135
      NI=NPMIN(NPP)                                                     GW2D2140
      IF(NI.EQ.0) GO TO 585                                             GW2D2145
      H(NI)=HMIN(NPP)                                                   GW2D2150
      RI(NI)=HMIN(NPP)                                                  GW2D2155
      HW(NI)=HMIN(NPP)                                                  GW2D2160
      GO TO 590                                                         GW2D2165
  585 NI=NPCON(NPP)                                                     GW2D2170
      IF(NI.EQ.0) GO TO 590                                             GW2D2175
      H(NI)=HCON(NPP)                                                   GW2D2180
      RI(NI)=HCON(NPP)                                                  GW2D2185
      HW(NI)=HCON(NPP)                                                  GW2D2190
  590 CONTINUE                                                          GW2D2195
```

```
  595 CONTINUE                                                          GW2D2200
C                                                                       GW2D2205
      DO 650 IT=1,NITER                                                 GW2D2210
C                                                                       GW2D2215
C ------- EVALUATE SOIL PROPERTIES FOR PREVIOUS ITERATE                GW2D2220
C                                                                       GW2D2225
      CALL SPROP(IE, HW,TH,DTH, AKX,AKZ,AKXZ, PROP, THPROP,             GW2D2230
     >           AKPROP,HPROP,CAPROP, MAXEL,MAXNP, NEL,KSP)             GW2D2235
C                                                                       GW2D2240
C -------- ASSEMBLE GLOBAL MATRIX.                                      GW2D2245
C                                                                       GW2D2250
      CALL ASEMBL(X,Z,IE, C,R,HW,HP,TH,DTH,AKX,AKZ,AKXZ, PROP,          GW2D2255
     > MAXNP,MAXEL,MAXHBP, MAXMAT,MXMPPM, KSS,W,DELT, SOS,LES,ISTYP,    GW2D2260
     3          WSS,NPW,IWTYP,LLNODE)                                   GW2D2265
C                                                                       GW2D2270
C ------- APPLY BOUNDARY CONDITIONS TO MODIFY THE GLOBAL MATRIX C AND   GW2D2275
C ------- THE LOAD VECTOR R.                                            GW2D2280
C                                                                       GW2D2285
      CALL BC(C,R, X,Z,IE,LLNODE, AKX,AKZ,AKXZ, FLX,HCON,NPCON,NPFLX,   GW2D2290
     1        HDB,IDTYP,NPDB, QCB,ICTYP,NPCB,ISC, QNB,INTYP,NPNB,ISN,   GW2D2295
     2        DIN,DCOSXN,DCOSZN,MNES, KSS, HMIN,NPMIN)                  GW2D2300
C                                                                       GW2D2305
      IF(IPNTS.EQ.1) GO TO 596                                          GW2D2310
C                                                                       GW2D2315
      CALL BANSOL(1,C,R,NNP,IHBP,MAXNP,MAXHBP)                          GW2D2320
      CALL BANSOL(2,C,R,NNP,IHBP,MAXNP,MAXHBP)                          GW2D2325
      GO TO 600                                                         GW2D2330
C                                                                       GW2D2335
  596 CALL PISS(R,RL,RI,C,OMI,TOLB,LLNODE,NNP,NPITER,MAXNP,MAXHBP,IBUG, GW2D2340
     1 KPR(ITM))                                                        GW2D2345
C                                                                       GW2D2350
C ------- OBTAIN MAXIMUM RELATIVE DEVIATION FROM PREVIOUS ITERATE       GW2D2355
C                                                                       GW2D2360
  600 CONTINUE                                                          GW2D2365
      NPP=0                                                             GW2D2370
      RD=-1.0D0                                                         GW2D2375
      RES=-1.0D0                                                        GW2D2380
      DO 620 NP=1,NNP                                                   GW2D2385
      RESNP=DABS(R(NP)-H(NP))                                           GW2D2390
      RES=DMAX1(RES,RESNP)                                              GW2D2395
      IF(H(NP) .NE. 0.0D0) RD=DMAX1(RD,DABS(RESNP/H(NP)))               GW2D2400
      IF(RESNP .LE. TOLB) GO TO 620                                     GW2D2405
      NPP=NPP+1                                                         GW2D2410
      NPCNV(NPP)=NP                                                     GW2D2415
  620 CONTINUE                                                          GW2D2420
C                                                                       GW2D2425
      NNCVN=NPP                                                         GW2D2430
C                                                                       GW2D2435
C ------- UPDATE PRESSURE WITH CURRENT ITERATE                          GW2D2440
C                                                                       GW2D2445
```

```
      DO 630 NP=1,NNP                                                   GW2D2450
      H(NP)=OME*R(NP)+(1.0D0-OME)*H(NP)                                 GW2D2455
      IF(IPNTS.EQ.1) RI(NP)=H(NP)                                       GW2D2460
      HW(NP)=W1*H(NP)+W2*HP(NP)                                         GW2D2465
  630 CONTINUE                                                          GW2D2470
C                                                                       GW2D2475
C ------- ESCAPE FROM ITERATION LOOP IF THE MAXIMUM RESIDUAL IS         GW2D2480
C ------- SUFFICIENTLY SMALL.                                           GW2D2485
C                                                                       GW2D2490
      IF(IBUG.NE.0 .AND. KPR(ITM).NE.0) PRINT 10200, IT,RES,RD,NNCVN    GW2D2495
      IF(IT.EQ.1 .AND. ITM.EQ.1) GO TO 650                              GW2D2500
      IF(RES.LT.TOLB) GO TO 660                                         GW2D2505
C                                                                       GW2D2510
  650 CONTINUE                                                          GW2D2515
C                                                                       GW2D2520
C ------- END THE ITERATION LOOP ON THE NON-LINEAR EQUATION             GW2D2525
C                                                                       GW2D2530
      PRINT 10710, ITM,ICY,IT,NITER,RES,RD,NNCVN                        GW2D2535
C                                                                       GW2D2540
      IF(IBUG.EQ.0 .OR. KPR(ITM).EQ.0) GO TO 660                        GW2D2545
C                                                                       GW2D2550
C ------- PRINT NONCONVERGING NODES                                     GW2D2555
C                                                                       GW2D2560
      PRINT 10500                                                       GW2D2565
      PRINT 10600, (NPCNV(NPP),NPP=1,NNCVN)                             GW2D2570
C                                                                       GW2D2575
  660 IF(ICHNG.EQ.0 .OR. KPR(ITM).EQ.0) GO TO 680                       GW2D2580
      IF(NRSN.EQ.0) GO TO 680                                           GW2D2585
C                                                                       GW2D2590
C ------- PRINT RAINFALL-SEEPAGE BOUNDARY CONDITION CHANGE INFORMATION  GW2D2595
C                                                                       GW2D2600
      PRINT 10402, ICY                                                  GW2D2605
      DO 670 I=1,NRSN                                                   GW2D2610
      NP=NPRS(I)                                                        GW2D2615
      PRINT 10403, I,NP,NPCON(I),HCON(I),NPMIN(I),HMIN(I),NPFLX(I),     GW2D2620
     1    FLX(I),DCYFLX(I)                                              GW2D2625
  670 CONTINUE                                                          GW2D2630
C                                                                       GW2D2635
C ------- CALCULATE DARCY'S VELOCITY                                    GW2D2640
C                                                                       GW2D2645
  680 CALL SPROP(IE, H,TH,DTH, AKX,AKZ,AKXZ, PROP, THPROP,AKPROP,       GW2D2650
     >       HPROP,CAPROP, MAXEL,MAXNP, NEL,KSP)                        GW2D2655
C                                                                       GW2D2660
      CALL VELT(X,Z,IE,C,H,HT,VX,VZ,AKX,AKZ,AKXZ,MAXEL,MAXNP,MAXHBP)    GW2D2665
C                                                                       GW2D2670
      IF(NRSN.EQ.0) GO TO 710                                           GW2D2675
C                                                                       GW2D2680
      CALL BCPREP(IE,H,VX,VZ, DL,DCOSX,DCOSZ, DCYFLX,FLX,RSFLX,         GW2D2685
     > HCON,IS,NPRS,NPCON,NPFLX,IRFTYP,TRF,RF,RFALL, TIME,NCHG,         GW2D2690
     2 HMIN,NPMIN,X,Z)                                                  GW2D2695
```

```
C                                                                       GW2D2700
      IF(NCHG.EQ.0) GO TO 710                                           GW2D2705
C                                                                       GW2D2710
  690 CONTINUE                                                          GW2D2715
C                                                                       GW2D2720
C ------- END ITERATION LOOP ON SEEPAGE-RAINFALL BOUNDARY CONDITIONS    GW2D2725
C                                                                       GW2D2730
      PRINT 10810, ITM,ICY,IT,NCYL,NITER,RES,RD,NNCVN                   GW2D2735
  710 IF(NNCVN.EQ.0) GO TO 715                                          GW2D2740
      PRINT 10810, ITM,ICY,IT,NCYL,NITER,RES,RD,NNCVN                   GW2D2745
C                                                                       GW2D2750
  715 IF(IMID.EQ.0) GO TO 740                                           GW2D2755
C                                                                       GW2D2760
      DO 720 I=1,NNP                                                    GW2D2765
  720 H(I)=2.0D0*H(I)-HP(I)                                             GW2D2770
C                                                                       GW2D2775
      DO 730 I=1,NDNP                                                   GW2D2780
      NP=NPDB(I)                                                        GW2D2785
      ITYP=IDTYP(I)                                                     GW2D2790
  730 H(NP)=HDB(ITYP)-(Z(NP)*CSFE+X(NP)*SNFE)*DFLOAT(KGRAV)             GW2D2795
C                                                                       GW2D2800
C ------- COMPUTE FLUXES THROUGH ALL TYPES OF BOUNDARIES                GW2D2805
C                                                                       GW2D2810
  740 CALL SFLOW(X,Z,IE, H,HP,VX,VZ,TH,DTH,                             GW2D2815
     1 DLB,DCOSXB,DCOSZB,BFLX,BFLXP,ISB,NBE,NPB, LES,SOS,ISTYP,         GW2D2820
     2 NPRS,NPDB,NPCB,NPNB,  PROP, DELT, KFLOW, NPW,WSS,IWTYP)          GW2D2825
C                                                                       GW2D2830
C ------- PRINT VARIABLES AT EACH TIME STEP                             GW2D2835
C                                                                       GW2D2840
      CALL PRINTT(VX,VZ,H,HT,TH, NPB,BFLX, NPRS,RSFLX,NPCON,NPFLX,      GW2D2845
     1 SUBHD(1,3), TIME,DELT, KPR(ITM),KOUT,KDIAG,ITMITM, NPMIN)        GW2D2850
C                                                                       GW2D2855
      IF(KSTR.EQ.1 .AND.KDSK(ITM).EQ.1) CALL STORE(X,Z,IE,H,HT,TH,VX,VZ,GW2D2860
     >    DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,NPCON,NPFLX, TITLE,             GW2D2865
     >     TIME,     NPROB,NSTRT, NPMIN)                                GW2D2870
C                                                                       GW2D2875
C ------- PREPARE FOR NEXT TIME STEP                                    GW2D2880
C                                                                       GW2D2885
      IF(TIME.GT.TMAX) GO TO 100                                        GW2D2890
      DELT=DELT*(1.0D0+CHNG)                                            GW2D2895
      DELT=DMIN1(DELT,DELMAX)                                           GW2D2900
      IF(IDELT.EQ.0) GO TO 880                                          GW2D2905
      IF(TIME.EQ.TDTCH(IDELT)) DELT=DELTO                               GW2D2910
  880 TIME=TIME+DELT                                                    GW2D2915
      IF(TIME.LT.TDTCH(IDELT+1)) GO TO 890                              GW2D2920
      IDELT=IDELT+1                                                     GW2D2925
      TIME=TIME-DELT                                                    GW2D2930
      DELT=TDTCH(IDELT)-TIME                                            GW2D2935
      IF(DELT.LE.0.0) DELT=DELTO                                        GW2D2940
      TIME=TIME+DELT                                                    GW2D2945
  890 CONTINUE                                                          GW2D2950
```

```
C                                                                       GW2D2955
C ------- END OF TIME-MARCHING LOOP                                     GW2D2960
C                                                                       GW2D2965
      GO TO 100                                                         GW2D2970
C                                                                       GW2D2975
  910 PRINT 10300,IHBP,MAXHBP                                           GW2D2980
C                                                                       GW2D2985
  990 RETURN                                                            GW2D2990
C                                                                       GW2D2995
   10 FORMAT(I5,9A8,I1,2I1)                                             GW2D3000
 1000 FORMAT(/8H1PROBLEM,I5,3H.. ,9A8,I1,2I1/)                          GW2D3005
10200 FORMAT(5X,I10,3X,E12.4,3X,E12.4,15X,I10)                          GW2D3010
10300 FORMAT(///26H HALF-BANDWIDTH-PLUS-ONE =,I4,                       GW2D3015
     > 25H EXCEEDS MAX. ALLOWABLE =,I4)                                 GW2D3020
10400 FORMAT(1H1,'*************************************************',GW2D3025
     1 '*************************************************************',GW2D3030
     2 '*****'///' DIAGNOSTIC TABLE',I4,'.. AT TIME =',1PD12.4,         GW2D3035
     3 ' ,(DELT =',1PD12.4,')')                                         GW2D3040
10401 FORMAT(//34H TABLE OF ITERATIVE PARAMETERS FOR,I3,9H-TH CYCLE//6X,GW2D3045
     > 9HITERATION,7X,8HRESIDUAL,6X,9HDEVIATION,6X,                     GW2D3050
     > 19HNO. NON-CONV.. NODES)                                         GW2D3055
10402 FORMAT(//53H TABLE OF RAINFALL/EVAPORATION-SEEPAGE B. C. USED FOR,GW2D3060
     1 I3,9H-TH CYCLE//6X,98H I  NPRS  NPCON     HCON     NPMIN     HMINGW2D3065
     1      NPFLX        FLX          DCYFLX FROM PREVIOUS CYCLE)      GW2D3070
10403 FORMAT(1H ,I8,I6,I7,D12.4,I7,D12.4,I7,2D12.4)                     GW2D3075
10500 FORMAT(//30H TABLE OF NON-CONVERGING NODES)                       GW2D3080
10600 FORMAT(/(5X,20I5))                                                GW2D3085
10210 FORMAT(1H0,'WARING: NON-CONVERGENCE OCCUR DURING STEADY STATE SOLUGW2D3090
     1TION AT',I3,' -TH CYCLE'/1H ,'IT = ',I3,'  .GT.  MAXIT = ',I3,    GW2D3095
     2 ',  RES =',D12.5,',  RD =',D12.4,',  NNCVN =',I4)                GW2D3100
10610 FORMAT(1H0,'ABSOLUTELY WARNING: STEADY STATE SOLUTION IS NG'/1H , GW2D3105
     > 'ICY = ',I3,'  IT = ',I3,'  MAXCY = ',I3,'  MAXIT = ',I3,        GW2D3110
     > ',  RES =',D12.4,',  RD =',D12.4,',  NNCVN =',I4)                GW2D3115
10710 FORMAT(1H0,'WARNING: NON-CONVERGENCE OCCUR AT',I5,' -TH TIME STEP'GW2D3120
     >,I3,' -TH CYCLE'/1H ,'IT = ',I3,'  .GT.  MAXIT = ',I3,2D12.4,I5)  GW2D3125
10810 FORMAT(1H0,'ABSOLUTELY WARNING: TRANSIENT SOLUTION IS NG AT ',I5, GW2D3130
     > ' -TH TIME STEP'/1H ,'ICY = ',I3,'  IT = ',I3,'  MAXCY = ',I3,   GW2D3135
     > '  MAXIT = ',I3,',  RES =',D12.4,',  RD =',D12.4,',  NNCVN =',I4)GW2D3140
      END                                                               GW2D3145
```

```
      SUBROUTINE DATAIN(TITLE,NPROB,                                    DATA 005
     1 KPR,KDSK,TDTCH, PROP, HPROP,THPROP,AKPROP,CAPROP, X,Z,IE,        DATA 010
     2 H,HT,TH,VX,VZ, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,                    DATA 015
     3 RF,TRF,NPRS,IRFTYP,IS,NRSE,DL,DCOSX,DCOSZ,HCON,NPCON,NPFLX,      DATA 020
     4 HDBF,THDBF,NPDB,IDTYP, QCBF,TQCBF,NPCB,ICTYP,ISC,                DATA 025
     5 QNBF,TQNBF,NPNB,INTYP,ISN,MNES,DLN,DCOSXN,DCOSZN, MAXDIF,ISTOP,  DATA 030
     6 HMIN,NPMIN, NPW,IWTYP,WSSF,TWSSF,LES,ISTYP,SOSF,TSOSF)           DATA 035
C                                                                       DATA 040
      IMPLICIT REAL*8(A-H,O-Z)                                          DATA 045
      REAL*4 PMAT,THPAR,AKPAR                                           DATA 050
C                                                                       DATA 055
C                                                                       DATA 060
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI,MXNDTC     DATA 065
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,IBAND,KGRAV,NTI,NDTCHG            DATA 070
C                                                                       DATA 075
      COMMON /CINTE/ NCYL,NITER,NSTRT,KSTR,KPRO,KDSKO,KSS,KSP,NPITER    DATA 080
      COMMON /CREAL/DELT,CHNG,DELMAX,TMAX,DELTO,TOLA,TOLB,W,OME,OMI,TIMEDATA 085
      COMMON /COORD/ CSFE,SNFE,CYLIND                                   DATA 090
C                                                                       DATA 095
      COMMON /CSOS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP                    DATA 100
      COMMON /CWSS/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP                    DATA 105
C                                                                       DATA 110
      COMMON /RSBC/ MXRSEL,MXRSNP,MXRFPR,MXRPAR,NRSEL,NRSN,NRFPR,NRFPAR DATA 115
C                                                                       DATA 120
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP                    DATA 125
      COMMON /CCBC/ MXCNP,MXCEL,MXCPR,MXCDP,NCNP,NCEL,NCPR,NCDP         DATA 130
      COMMON /CNBC/ MXNNP,MXNEL,MXNPR,MXNDP,NNNP,NNEL,NNPR,NNDP         DATA 135
C                                                                       DATA 140
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM                                DATA 145
      COMMON /CMTL/ NMAT,NMPPM,NSPPM                                    DATA 150
C                                                                       DATA 155
      COMMON /OPT/ ILUMP,IMID,IPNTS,IVML                                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
C                                                                       DATA 195
      DIMENSION HPROP(MXSPPM,MAXMAT),THPROP(MXSPPM,MAXMAT)              DATA 200
      DIMENSION AKPROP(MXSPPM,MAXMAT),CAPROP(MXSPPM,MAXMAT)             DATA 205
C                                                                       DATA 210
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)                           DATA 215
      DIMENSION H(MAXNP),HT(MAXNP),TH(4,MAXEL),VX(MAXNP),VZ(MAXNP)      DATA 220
C                                                                       DATA 225
      DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),NBE(MAXBEL),  DATA 230
     > ISB(4,MAXBEL),NPB(MAXBNP)                                        DATA 235
C                                                                       DATA 240
      DIMENSION LES(MXSEL),ISTYP(MXSEL)                                 DATA 245
      DIMENSION SOSF(MXSDP,MXSPR),TSOSF(MXSDP,MXSPR)                    DATA 250
      DIMENSION NPW(MXWNP),IWTYP(MXWNP)                                 DATA 255
```

```
      DIMENSION WSSF(MXWDP,MXWPR),TWSSF(MXWDP,MXWPR)                    DATA 260
C                                                                       DATA 265
      DIMENSION DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),HCON(MXRSNP),    DATA 270
     > NRSE(MXRSEL),IS(4,MXRSEL),NPRS(MXRSNP),NPCON(MXRSNP),            DATA 275
     > NPFLX(MXRSNP),IRFTYP(MXRSNP),TRF(MXRPAR,MXRFPR),                 DATA 280
     > RF(MXRPAR,MXRFPR)                                                DATA 285
      DIMENSION HMIN(MXRSNP),NPMIN(MXRSNP)                              DATA 290
C                                                                       DATA 295
      DIMENSION HDBF(MXDDP,MXDPR),THDBF(MXDDP,MXDPR)                    DATA 300
      DIMENSION NPDB(MXDNP),IDTYP(MXDNP)                                DATA 305
C                                                                       DATA 310
      DIMENSION QCBF(MXCDP,MXCPR),TQCBF(MXCDP,MXCPR)                    DATA 315
      DIMENSION NPCB(MXCNP),ICTYP(MXCNP), ISC(2,MXCEL)                  DATA 320
C                                                                       DATA 325
      DIMENSION QNBF(MXNDP,MXNPR),TQNBF(MXNDP,MXNPR)                    DATA 330
      DIMENSION NPNB(MXNNP),INTYP(MXNNP), ISN(4,MXNEL)                  DATA 335
      DIMENSION DLN(MXNEL),DCOSXN(MXNEL),DCOSZN(MXNEL),MNES(MXNEL)      DATA 340
C                                                                       DATA 345
      DIMENSION PMAT(3,6),AKPAR(3,8),THPAR(3,8)                         DATA 350
C                                                                       DATA 355
      DATA PMAT/4H    ,4H ALP,4H    ,4H   B,4HETAP,4H    ,4H    ,      DATA 360
     > 4H POR,4H    ,4H   K,4HSX/P,4HSX  ,4H   K,4HSZ/P,4HSZ  ,4H  KS,4HDATA 365
     >XZ/P,4HSXZ /                                                      DATA 370
C                                                                       DATA 375
      DATA THPAR/4H    ,4H TH1,4H    ,4H    ,4H TH2,4H    ,4H    ,     DATA 380
     > 4H  H0 ,4H    ,4H    ,4H  A1,4H    ,4H    ,4H  A2,4H    ,4H    , DATA 385
     > 4H  R1,4H    ,4H    ,4H  R2,4H    ,4H    ,4H   C,4H    /         DATA 390
C                                                                       DATA 395
      DATA AKPAR/4H    ,4H  B1,4H    ,4H    ,4H  B2,4H    ,18*4H    /  DATA 400
C                                                                       DATA 405
      ISTOP=0                                                           DATA 410
C                                                                       DATA 415
C ******* DATA SET 2: BASIC INTEGER PARAMETERS                          DATA 420
C                                                                       DATA 425
      READ 10, NNP,NEL,NMAT,NCM,NTI,KSS,KSP,NSPPM,NMPPM,KSTR,KCP,KGRAV, DATA 430
     1 NSTRT,NITER,NCYL,NDTCHG                                          DATA 435
      IF(NDTCHG.LE.0) NDTCHG=1                                          DATA 440
      READ 10, ILUMP,IMID,IPNTS,NPITER,IVML                             DATA 445
      PRINT 1000, NNP,NEL,NMAT,NCM,NTI,KSS,KSP,NSPPM,NMPPM,KSTR,KCP,    DATA 450
     1 KGRAV,NSTRT,NITER,NCYL,NDTCHG,NPITER                             DATA 455
      PRINT 1050, ILUMP,IMID,IPNTS,IVML                                 DATA 460
C                                                                       DATA 465
C ******* DATA SET 3: BASIC REAL PARAMETERS                             DATA 470
C                                                                       DATA 475
      READ 20, DELT,CHNG,DELMAX,TMAX,FE,TOLA,TOLB,RHO,GRAV,VISC,W,      DATA 480
     1 OME,OMI,CYLIND                                                   DATA 485
C                                                                       DATA 490
      DELTO=DELT                                                        DATA 495
      IF(TMAX.LE.0.0) TMAX=1.0D38                                       DATA 500
```

```
C
      PRINT 1100, DELT,CHNG,DELMAX,TMAX,FE,TOLA,TOLB,RHO,GRAV,VISC,W,
     1 OME,OMI,CYLIND
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
      PI=3.14159265
      FE=FE*PI/180.0D0
      CSFE=DCOS(FE)
      SNFE=DSIN(FE)
C
      IF(FE.EQ.0.0) CSFE=1.0D0
      IF(FE.EQ.0.0) SNFE=0.0
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
      IF (KSP.EQ.1) GO TO 120
C
C ------- SOIL PROPERTIES ARE TO BE REPRESENTED BY ANALYTIC FUNCTIONS
C
C ------- READ AND PRINT PARAMETERS REQUIRED TO COMPUTE MOISTURE
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)                                  DATA 760
      PRINT 1510, I,(AKPROP(J,I),J=1,NSPPM)                             DATA 765
  110 CONTINUE                                                          DATA 770
      GO TO 175                                                         DATA 775
  120 IF (NSPPM.EQ.0) GO TO 200                                         DATA 780
C                                                                       DATA 785
C ------- SOIL PROPERTIES ARE TO BE GIVEN IN TABULAR FORM               DATA 790
C                                                                       DATA 795
C ------- PRESSURE                                                      DATA 800
C                                                                       DATA 805
      DO 130 I=1,NMAT                                                   DATA 810
      READ 20, (HPROP(J,I),J=1,NSPPM)                                   DATA 815
  130 CONTINUE                                                          DATA 820
C                                                                       DATA 825
C ------- WATER CONTENT                                                 DATA 830
C                                                                       DATA 835
      DO 140 I=1,NMAT                                                   DATA 840
      READ 20, (THPROP(J,I),J=1,NSPPM)                                  DATA 845
  140 CONTINUE                                                          DATA 850
C                                                                       DATA 855
C ------- RELATIVE HYDRAULIC CONDUCTIVITY                               DATA 860
C                                                                       DATA 865
      DO 150 I=1,NMAT                                                   DATA 870
      READ 20, (AKPROP(J,I),J=1,NSPPM)                                  DATA 875
  150 CONTINUE                                                          DATA 880
C                                                                       DATA 885
C ------- WATER CAPACITY                                                DATA 890
C                                                                       DATA 895
      DO 160 I=1,NMAT                                                   DATA 900
      READ 20, (CAPROP(J,I),J=1,NSPPM)                                  DATA 905
  160 CONTINUE                                                          DATA 910
C                                                                       DATA 915
      DO 170 I=1,NMAT                                                   DATA 920
      PRINT 1700                                                        DATA 925
      PRINT 1710, I,(HPROP(J,I),THPROP(J,I),AKPROP(J,I),CAPROP(J,I),J=1,DATA 930
     1NSPPM)                                                            DATA 935
  170 CONTINUE                                                          DATA 940
C                                                                       DATA 945
  175 IF (KCP.EQ.0) GO TO 200                                           DATA 950
C                                                                       DATA 955
C ------- CONVERT FROM SATURATED PERMEABILITY TO SATURATED CONDUCTIVITY DATA 960
C                                                                       DATA 965
      DO 190 I=1,NMAT                                                   DATA 970
      PKCF=RHO*GRAV/VISC                                                DATA 975
      PROP(4,I)=PROP(4,I)*PKCF                                          DATA 980
      PROP(5,I)=PROP(5,I)*PKCF                                          DATA 985
      PROP(6,I)=PROP(6,I)*PKCF                                          DATA 990
  190 CONTINUE                                                          DATA 995
C                                                                       DATA1000
C ******* DATA SET 7: NODE COORDINATES                                  DATA1005
```

```
C                                                                       DATA1010
C ------- READ NODAL POINT COORDINATES                                  DATA1015
C                                                                       DATA1020
  200 CONTINUE                                                          DATA1025
      CALL READR(X,MAXNP,NNP)                                           DATA1030
      CALL READR(Z,MAXNP,NNP)                                           DATA1035
C                                                                       DATA1040
C ------- PRINT NODAL POINT COORDINATES                                 DATA1045
C                                                                       DATA1050
      LINE=0                                                            DATA1055
      DO 265 NP=1,NNP,4                                                 DATA1060
      NJMN=NP                                                           DATA1065
      NJMX=MIN0(NP+3,NNP)                                               DATA1070
      LINE=LINE+1                                                       DATA1075
      IF(MOD(LINE-1,50).EQ.0) PRINT 2100                                DATA1080
      PRINT 2110, (NJ,X(NJ),Z(NJ),NJ=NJMN,NJMX)                         DATA1085
  265 CONTINUE                                                          DATA1090
C                                                                       DATA1095
C ******* DATA SET 8: ELEMENT DATA                                      DATA1100
C                                                                       DATA1105
C ------- READ ELEMENT INDICES AND COMPUTE MAXIMUM NODAL DIFFERENCE FOR DATA1110
C ------- EACH ELEMENT                                                  DATA1115
C                                                                       DATA1120
      MAXDIF = 0                                                        DATA1125
      MJ = 0                                                            DATA1130
  270 READ 10, MI,(IE(MI,I),I=1,5),MODL,NLAY                            DATA1135
      MTYP=IE(MI,5)                                                     DATA1140
      MND = 0                                                           DATA1145
C                                                                       DATA1150
      NQ=4                                                              DATA1155
      NQ1=3                                                             DATA1160
      IEM4=IE(MI,4)                                                     DATA1165
      IF(IEM4.EQ.0) NQ=3                                                DATA1170
      IF(IEM4.EQ.0) NQ1=2                                               DATA1175
C                                                                       DATA1180
      DO 280 IQ=1,NQ1                                                   DATA1185
      IQ1=IQ+1                                                          DATA1190
      DO 280 JQ=IQ1,NQ                                                  DATA1195
      ND=IABS(IE(MI,IQ)-IE(MI,JQ))                                      DATA1200
      MND=MAX0(ND,MND)                                                  DATA1205
      MAXDIF=MAX0(ND,MAXDIF)                                            DATA1210
  280 CONTINUE                                                          DATA1215
C                                                                       DATA1220
  290 MJ = MJ + 1                                                       DATA1225
C                                                                       DATA1230
      IF (MI-MJ) 300,330,310                                            DATA1235
C                                                                       DATA1240
  300 PRINT 3000, MI                                                    DATA1245
      ISTOP = ISTOP + 1                                                 DATA1250
C                                                                       DATA1255
```

```
  310 DO 320 IQ=1,NQ                                                    DATA1260
  320 IE(MJ,IQ)=IE(MJ-1,IQ) + 1                                         DATA1265
      IE(MJ,5) = IE(MJ-1,5)                                             DATA1270
C                                                                       DATA1275
  330 IF (MJ.LT MI) GO TO 290                                           DATA1280
      IF (MJ.EQ.NEL) GO TO 370                                          DATA1285
      IF (MODL.LE.0) GO TO 270                                          DATA1290
C                                                                       DATA1295
      DO 360 I=1,NLAY                                                   DATA1300
      LL=2                                                              DATA1305
      DO 360 J=1,MODL                                                   DATA1310
      IF(MJ.EQ.MI) GO TO 350                                            DATA1315
C                                                                       DATA1320
      DO 340 KQ=1,NQ                                                    DATA1325
  340 IE(MJ,KQ)=IE(MJ-1,KQ) + LL                                        DATA1330
      IE(MJ,5)=IE(MJ-1,5)                                               DATA1335
C                                                                       DATA1340
  350 LL=1                                                              DATA1345
  360 MJ=MJ + 1                                                         DATA1350
      MJ = MJ - 1                                                       DATA1355
      IF (MJ.LT.NEL) GO TO 270                                          DATA1360
  370 CONTINUE                                                          DATA1365
C                                                                       DATA1370
      IHALFB=MAXDIF                                                     DATA1375
      IBAND=2*IHALFB+1                                                  DATA1380
      IHBP=IHALFB+1                                                     DATA1385
      PRINT 3150, IHALFB,IBAND,IHBP                                     DATA1390
C                                                                       DATA1395
C ******* DATA SET 9: MATERIAL CORRECTIONS                              DATA1400
C                                                                       DATA1405
      IF (NCM.LE.0) GO TO 410                                           DATA1410
      CALL READN(IE(1,5),MAXEL,NCM)                                     DATA1415
C                                                                       DATA1420
C ------- PRINT ELEMENT INCIDENCE AND MATERIAL TYPES FOR EACH ELEMENT   DATA1425
C                                                                       DATA1430
  410 LINE=0                                                            DATA1435
      DO 415 NI=1,NEL,3                                                 DATA1440
      NJMN=NI                                                           DATA1445
      NJMX=MIN0(NI+2,NEL)                                               DATA1450
      LINE=LINE+1                                                       DATA1455
      IF(MOD(LINE-1,50).EQ.0) PRINT 2700                                DATA1460
      PRINT 3100, (NJ,(IE(NJ,K),K=1,5),NJ=NJMN,NJMX)                    DATA1465
  415 CONTINUE                                                          DATA1470
C                                                                       DATA1475
C ------- CHECK IF MATERIAL TYPE FOR EACH ELEMENT IS CORRECT            DATA1480
C                                                                       DATA1485
      DO 420 M=1,NEL                                                    DATA1490
      MTYP=IE(M,5)                                                      DATA1495
      IF(MTYP.GT.0 .AND. MTYP.LE.NMAT) GO TO 420                        DATA1500
      PRINT 4200, M                                                     DATA1505
      ISTOP=ISTOP+1                                                     DATA1510
```

```
  420 CONTINUE                                                          DATA1515
C                                                                       DATA1520
      IF(ISTOP.EQ.0) GO TO 430                                          DATA1525
      PRINT 4300, ISTOP                                                 DATA1530
      STOP                                                              DATA1535
C                                                                       DATA1540
C ******* DATA SET 10: INITIAL CONDITIONS                               DATA1545
C                                                                       DATA1550
  430 TIME=0.0D0                                                        DATA1555
C                                                                       DATA1560
C -------- READ INITIAL CONDITIONS VIA LOGICAL UNIT 1 BECAUSE OF RESTART DATA1565
C                                                                       DATA1570
      IF (NSTRT.EQ.0) GO TO 450                                         DATA1575
C                                                                       DATA1580
C -------- READ INITIAL OR PRE-INITIAL CONDITIONS VIA LOGICAL UNIT 2    DATA1585
C -------- IF NSTRT .GT. 0                                              DATA1590
C                                                                       DATA1595
      REWIND 1                                                          DATA1600
      REWIND 2                                                          DATA1605
C                                                                       DATA1610
      READ(2) (DUM,I=1,9),IDUM,NPT,NET,NBN,NBEL,IDUM,NRSN               DATA1615
      IF (KSTR.EQ.1) WRITE(1) (TITLE(I),I=1,9),NPROB,NNP,NEL,NBN,NBEL,  DATA1620
     > NTI,NRSN                                                         DATA1625
C                                                                       DATA1630
      READ(2) (X(NP),NP=1,NPT),(Z(NP),NP=1,NPT),((IE(M,IQ),M=1,NET),IQ= DATA1635
     > 1,4),(DLB(M),M=1,NBEL),(DCOSXB(M),M=1,NBEL),                     DATA1640
     > (DCOSZB(M),M=1,NBEL),(NBE(M),M=1,NBEL),((ISB(IQ,M),IQ=1,4),M=1,  DATA1645
     > NBEL),(NPB(NP),NP=1,NBN)                                         DATA1650
C                                                                       DATA1655
      IF(KSTR.EQ.1) WRITE(1) (X(N),N=1,NNP),(Z(N),N=1,NNP),((IE(M,IQ),  DATA1660
     > M=1,NEL),IQ=1,4),(DLB(M),M=1,NBEL),(DCOSXB(M),M=1,NBEL),         DATA1665
     > (DCOSZB(M),M=1,NBEL),(NBE(M),M=1,NBEL),((ISB(IQ,M),IQ=1,4),M=1,  DATA1670
     > NBEL),(NPB(N),N=1,NBN)                                           DATA1675
C                                                                       DATA1680
      DO 440 ITM=1,NSTRT                                                DATA1685
      READ(2) TIME,(H(N),N=1,NPT),(HT(N),N=1,NPT),((TH(IQ,M),IQ=1,4),   DATA1690
     > M=1,NET),(VX(N),N=1,NPT),(VZ(N),N=1,NPT),(NPCON(N),N=1,NRSN),    DATA1695
     > (NPFLX(N),N=1,NRSN),(NPMIN(N),N=1,NRSN)                          DATA1700
C                                                                       DATA1705
      IF(KSTR.EQ.0) GO TO 440                                           DATA1710
         WRITE(1) TIME,(H(N),N=1,NNP),(HT(N),N=1,NNP),((TH(IQ,M),IQ=1,4)DATA1715
     >,M=1,NEL),(VX(N),N=1,NNP),(VZ(N),N=1,NNP),(NPCON(N),N=1,NNP),     DATA1720
     > (NPFLX(N),N=1,NRSN),(NPMIN(N),N=1,NRSN)                          DATA1725
  440 CONTINUE                                                          DATA1730
      GO TO 500                                                         DATA1735
C                                                                       DATA1740
C -------- READ INITIAL OR PRE-INITIAL CONDITIONS VIA CARDS IF NSTRT.EQ.0DATA1745
C                                                                       DATA1750
  450 CALL READR(H,MAXNP,NNP)                                           DATA1755
```

```
C                                                                       DATA1760
C ------- IDENTIFY BOUNDARY ELEMENTS AND COMPUTE DIRECTIONAL COSINES    DATA1765
C ------- OF ALL BOUNDARY ELEMENT SIDES                                 DATA1770
C                                                                       DATA1775
  500 CALL SURF(X,Z,IE, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,                  DATA1780
     > MAXNP,MAXEL, MAXBEL,MAXBNP)                                      DATA1785
C                                                                       DATA1790
C ******* DATA SET 11: TRANSIENT INTEGER PARAMETERS                     DATA1795
C                                                                       DATA1800
      READ 10, NSEL,NSPR,NSDP,NRSEL,NRSN,NRFPR,NRFPAR,NDNP,NDPR,NDDP,   DATA1805
     1 NCNP,NCEL,NCPR,NCDP, NNNP,NNEL,NNPR,NNDP, NWNP,NWPR,NWDP         DATA1810
      PRINT 5100, NSEL,NSPR,NSDP,NRSEL,NRSN,NRFPR,NRFPAR                DATA1815
      PRINT 5150, NDNP,NDPR,NDDP,NCNP,NCEL,NCPR,NCDP,NNNP,NNEL,NNPR,NNDPDATA1820
      PRINT 5155, NWNP,NWPR,NWDP                                        DATA1825
C                                                                       DATA1830
C ******* DATA SET 12: SOURCE DATA                                      DATA1835
C                                                                       DATA1840
      IF(NSEL.LE.0) GO TO 550                                           DATA1845
      PRINT 5300                                                        DATA1850
      DO 510 I=1,NSPR                                                   DATA1855
      READ 20, (TSOSF(J,I),SOSF(J,I),J=1,NSDP)                          DATA1860
      PRINT 5500, I                                                     DATA1865
      PRINT 5510, (TSOSF(J,I),SOSF(J,I),J=1,NSDP)                       DATA1870
  510 CONTINUE                                                          DATA1875
C                                                                       DATA1880
C ------- READ SOURCE ELEMENT AND TYPE ASSIGNED TO EACH SOURCE ELEMENT  DATA1885
C                                                                       DATA1890
      READ(5,10) (LES(I),I=1,NSEL)                                      DATA1895
      CALL READN(ISTYP,MXSEL,NSEL)                                      DATA1900
C                                                                       DATA1905
C ------- PRINT GLOBAL ELEMENT NUMBER AND PROFILE TYPE OF SOURCE ELEMENTDATA1910
C                                                                       DATA1915
      WRITE(6,5600)                                                     DATA1920
      DO 530 I=1,NSEL,5                                                 DATA1925
      NJMN=I                                                            DATA1930
      NJMX=MIN0(I+4,NSEL)                                               DATA1935
      WRITE(6,5650) (J,LES(J),ISTYP(J),J=NJMN,NJMX)                     DATA1940
  530 CONTINUE                                                          DATA1945
C                                                                       DATA1950
C -------  READ AND WRITE WELL SOURCE/SINK PROFILES                     DATA1955
C                                                                       DATA1960
  550 IF(NWNP.EQ.0) GO TO 600                                           DATA1965
      PRINT 5700                                                        DATA1970
      DO 570 I=1,NWPR                                                   DATA1975
      READ 20, (TWSSF(J,I),WSSF(J,I),J=1,NWDP)                          DATA1980
      PRINT 5710, I                                                     DATA1985
      PRINT 5510, (TWSSF(J,I),WSSF(J,I),J=1,NWDP)                       DATA1990
  570 CONTINUE                                                          DATA1995
C                                                                       DATA2000
C ------- READ WELL SOURCE/SINK NODES AND TYPE OF PROFILES ASSIGNED TO  DATA2005
C ------- EACH OF NWNP NODES.                                           DATA2010
```

```
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
      PRINT 5800
      DO 590 I=1,NWNP,5
      NJMN=I
      NJMX=MIN0(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(NRSEL.EQ.0) GO TO 700
C
      PRINT 6000
C
C ------- READ AND WRITE RAINFALL (+)/EVAPORATION (-) PROFILES
C
      PRINT 6100
      DO 610 I=1,NRFPR
         READ 20, (TRF(J,I),RF(J,I),J=1,NRFPAR)
      PRINT 6150, I
      PRINT 5510, (TRF(J,I),RF(J,I),J=1,NRFPAR)
  610 CONTINUE
C
C ------- READ AND WRITE GLOBAL NODAL NUMBER OF RS NODES
C
      CALL READN(NPRS,MXRSNP,NRSN)
      PRINT 6200, (NPRS(I),I=1,NRSN)
C
C ------- READ RAINFALL/EVAPORATION TYPE ASSIGNED TO EACH RS NODE
C
      CALL READN(IRFTYP,MXRSNP,NRSN)
C
C ------- READ PONDING DEPTH AND MINIMUM HEAD FOR EACH OF ALL RS NODES
C
      CALL READR(HCON,MXRSNP,NRSN)
      CALL READR(HMIN,MXRSNP,NRSN)
C
C ------- PRINT GLOBAL NODAL NUMBER, RAINFALL/EVAPORATION TYPE, PONDING
C ------- DEPTH, AND MINIMUM PRESSURE HEAD FOR ALL RS NODES
C
      LINE=0
      DO 620 I=1,NRSN,3
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 6220
      NJMN=I
      NJMX=MIN0(I+2,NRSN)
```

```
      PRINT 6250, (J,NPRS(J),IRFTYP(J),HCON(J),HMIN(J),J=NJMN,NJMX)
  620 CONTINUE
C
C ------- READ RAINFALL/EVAPORATION-SEEPAGE (RS) ELEMENT SIDES
C
      MPI=0
  630 READ 10, MI,NSEQ,M,IS1,IS2,MIAD,MAD,IS1AD,IS2AD
      IF(MI.EQ.0) GO TO 640
      MJ=MI+NSEQ
      DO 635 MP=MI,MJ
      I=MI+(MP-MI)*MIAD
      NRSE(I)=M+(MP-MI)*MAD
      IS(1,I)=IS1+(MP-MI)*IS1AD
      IS(2,I)=IS2+(MP-MI)*IS2AD
      MPI=MPI+1
  635 CONTINUE
      GO TO 630
  640 IF(MPI.EQ.NRSEL) GO TO 650
      PRINT 6500
      STOP
C
  650 CONTINUE
C
C ------- PRINT THE READ-IN RS ELEMENT SIDES
C
      LINE=0
      DO 660 MP=1,NRSEL,5
         LINE=LINE+1
         IF(MOD(LINE-1,50).EQ.0) PRINT 6600
      NJMN=MP
      NJMX=MIN0(MP+4,NRSEL)
      PRINT 6650, (J,NRSE(J),IS(1,J),IS(2,J),J=NJMN,NJMX)
  660 CONTINUE
C
C ------- COMPUTE LENGTH AND DIRECTIONAL COSINES OF ALL RAINFALL/
C ------- EVAPORATION-SEEPAGE ELEMENT SIDES
C
      DO 690 MPI=1,NRSEL
      MI=NRSE(MPI)
      DO 680 MPJ=1,NBEL
      MJ=NBE(MPJ)
      IF(MJ.NE.MI) GO TO 680
      I=IS(1,MPI)
      NI=NPRS(I)
      J=IS(2,MPI)
      NJ=NPRS(J)
      NIB=ISB(1,MPJ)
      NJB=ISB(2,MPJ)
      IF(NIB.EQ.NI .AND. NJB.EQ.NJ) GO TO 670
      IF(NIB.EQ.NJ .AND. NJB.EQ.NI) GO TO 665
```

```
      GO TO 680
C
  665 IS(1,MPI)=J
      IS(2,MPI)=I
  670 DO 675 JJ=3,4
  675 IS(JJ,MPI)=ISB(JJ,MPJ)
      DL(MPI)=DLB(MPJ)
      DCOSX(MPI)=DCOSXB(MPJ)
      DCOSZ(MPI)=DCOSZB(MPJ)
      GO TO 690
C
  680 CONTINUE
C
      PRINT 6800, MPI,I,J,NI,NJ,MI
C
  690 CONTINUE
C
C ------- PRINT COMPUTED RS ELEMENT SIDES
C
      LINE=0
      DO 695 MP=1,NRSEL,2
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 6900
      NJMN=MP
      NJMX=MIN0(MP+1,NRSEL)
      PRINT 6950, (J,NPRS(J),(IS(I,J),I=1,4),DL(J),DCOSX(J),DCOSZ(J),J=
     1 NJMN,NJMX)
  695 CONTINUE
C
C ******* DATA SET 14: 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)
      PRINT 7100, I
      PRINT 5510, (THDBF(J,I),HDBF(J,I),J=1,NDDP)
  710 CONTINUE
C
C ------- READ GLOBAL NODAL NUMBER OF ALL DIRICHLET NODES AND THE TYPE
C ------- OF TOTAL HEAD PROFILE ASSIGNED TO EACH OF THEM
C
      READ 10, (NPDB(I),I=1,NDNP)
      CALL READN(IDTYP,MXDNP,NDNP)
C
C ------- PRINT GLOBAL NODAL NUMBERS AND PROFILE TYPE OF DIRICHLET NODES
C
      LINE=0
```

```
      DO 720 I=1,NDNP,5                                                 DATA2770
      LINE=LINE+1                                                       DATA2775
      IF(MOD(LINE-1,50).EQ.0) PRINT 7200                                DATA2780
      NJMN=I                                                            DATA2785
      NJMX=MIN0(I+4,NDNP)                                               DATA2790
      PRINT 7250, (J,NPDB(J),IDTYP(J),J=NJMN,NJMX)                      DATA2795
  720 CONTINUE                                                          DATA2800
C                                                                       DATA2805
C ******* DATA SET 15: CAUCHY BOUNDARY CONDITIONS                       DATA2810
C                                                                       DATA2815
C ------- READ AND PRINT CAUCHY FLUX PROFILES                           DATA2820
C                                                                       DATA2825
  800 IF(NCNP.EQ.0) GO TO 900                                           DATA2830
      PRINT 8000                                                        DATA2835
      DO 810 I=1,NCPR                                                   DATA2840
      READ 20, (TQCBF(J,I),QCBF(J,I),J=1,NCDP)                          DATA2845
      PRINT 8100, I                                                     DATA2850
      PRINT 5510, (TQCBF(J,I),QCBF(J,I),J=1,NCDP)                       DATA2855
  810 CONTINUE                                                          DATA2860
C                                                                       DATA2865
C ------- READ GLOBAL NODAL NUMBER OF ALL CAUCHY NODES AND TYPE OF      DATA2870
C ------- CAUCHY FLUX PROFILE ASSIGNED TO EACH OF THEM                  DATA2875
C                                                                       DATA2880
      READ 10, (NPCB(I),I=1,NCNP)                                       DATA2885
C                                                                       DATA2890
      CALL READN(ICTYP,MXCNP,NCNP)                                      DATA2895
C                                                                       DATA2900
C ------- PRINT GLOBAL NODAL NUMBER AND PROFILE TYPE OF ALL CAUCHY NODESDATA2905
C                                                                       DATA2910
      LINE=0                                                            DATA2915
      DO 820 I=1,NCNP,5                                                 DATA2920
      LINE=LINE+1                                                       DATA2925
      IF(MOD(LINE-1,50).EQ.0) PRINT 8200                                DATA2930
      NJMN=I                                                            DATA2935
      NJMX=MIN0(I+4,NCNP)                                               DATA2940
      PRINT 7250, (J,NPCB(J),ICTYP(J),J=NJMN,NJMX)                      DATA2945
  820 CONTINUE                                                          DATA2950
C                                                                       DATA2955
C ------- READ CAUCHY BOUNDARY ELEMENT SIDES                            DATA2960
C                                                                       DATA2965
      MPI= 0                                                            DATA2970
  830 READ 10, MI,NSEQ,IS1,IS2,MIAD,IS1AD,IS2AD                         DATA2975
      IF(MI.EQ.0) GO TO 840                                             DATA2980
      MJ=MI+NSEQ                                                        DATA2985
      DO 835 MP=MI,MJ                                                   DATA2990
      I=MI+(MP-MI)*MIAD                                                 DATA2995
      ISC(1,I)=IS1+(MP-MI)*IS1AD                                        DATA3000
      ISC(2,I)=IS2+(MP-MI)*IS2AD                                        DATA3005
      MPI=MPI+1                                                         DATA3010
  835 CONTINUE                                                          DATA3015
```

```
      GO TO 830                                                         DATA3020
  840 IF(MPI.EQ.NCEL) GO TO 850                                         DATA3025
      PRINT 8300                                                        DATA3030
      STOP                                                              DATA3035
C                                                                       DATA3040
  830 CONTINUE                                                          DATA3045
C                                                                       DATA3050
C ------ PRINT THE READ-IN CAUCHY BOUNDARY ELEMENT SIDES                DATA3055
C                                                                       DATA3060
      LINE=0                                                            DATA3065
      DO 890 MP=1,NCEL,5                                                DATA3070
      LINE=LINE+1                                                       DATA3075
      IF(MOD(LINE-1,50).EQ.0) PRINT 8900                                DATA3080
      NJMN=MP                                                           DATA3085
      NJMX=MIN0(MP+4,NCEL)                                              DATA3090
      PRINT 8950, (J,ISC(1,J),ISC(2,J),J=NJMN,NJMX)                     DATA3095
  890 CONTINUE                                                          DATA3100
C                                                                       DATA3105
C ******* DATA SET 16:  NEUMANN BOUNDARY CONDITIONS                     DATA3110
C                                                                       DATA3115
C ------ READ AND PRINT NEUMANN FLUX PROFILES                           DATA3120
C                                                                       DATA3125
  900 IF(NNNP.EQ.0) GO TO 999                                           DATA3130
      PRINT 9000                                                        DATA3135
      DO 910 I=1,NNPR                                                   DATA3140
      READ 20, (TQNBF(J,I),QNBF(J,I),J=1,NNDP)                          DATA3145
      PRINT 9100, I                                                     DATA3150
      PRINT 5510, (TQNBF(J,I),QNBF(J,I),J=1,NNDP)                       DATA3155
  910 CONTINUE                                                          DATA3160
C                                                                       DATA3165
C ------- READ GLOBAL NODAL NUMBER OF ALL NEUMANN NODES AND TYPE OF     DATA3170
C ------- NEUMANN FLUX PROFILE ASSIGNED TO EACH OF THEM                 DATA3175
C                                                                       DATA3180
      READ 10, (NPNB(I),I=1,NNNP)                                       DATA3185
      CALL READN(INTYP,MXNNP,NNNP)                                      DATA3190
C                                                                       DATA3195
C ------- PRINT GLOBAL NODAL NUMBERS AND PROFILE TYPES OF NEUMANN NODES DATA3200
C                                                                       DATA3205
      LINE=0                                                            DATA3210
      DO 920 I=1,NNNP,5                                                 DATA3215
      LINE=LINE+1                                                       DATA3220
      IF(MOD(LINE-1,50).EQ.0) PRINT 9200                                DATA3225
      NJMN=I                                                            DATA3230
      NJMX=MIN0(I+4,NNNP)                                               DATA3235
      PRINT 7250, (J,NPNB(J),INTYP(J),J=NJMN,NJMX)                      DATA3240
  920 CONTINUE                                                          DATA3245
C                                                                       DATA3250
C ------- READ NEUMANN BOUNDARY ELEMENT SIDES                           DATA3255
C                                                                       DATA3260
      MPI=0                                                             DATA3265
```

```
  930 READ 10, MI,NSEQ,M,IS1,IS2,MIAD,MAD,IS1AD,IS2AD                DATA3270
      IF(MI.EQ.0) GO TO 940                                          DATA3275
      MJ=MI+NSEQ                                                     DATA3280
      DO 935 MP=MI,MJ                                                DATA3285
      I=MI+(MP-MI)*MIAD                                              DATA3290
      MNES(I)=M+(MP-MI)*MAD                                          DATA3295
      ISN(1,I)=IS1+(MP-MI)*IS1AD                                     DATA3300
      ISN(2,I)=IS2+(MP-MI)*IS2AD                                     DATA3305
      MPI=MPI+1                                                      DATA3310
  935 CONTINUE                                                       DATA3315
      GO TO 930                                                      DATA3320
  940 IF(MPI.EQ.NNEL) GO TO 950                                      DATA3325
      PRINT 9300                                                     DATA3330
      STOP                                                           DATA3335
C                                                                    DATA3340
  950 CONTINUE                                                       DATA3345
C                                                                    DATA3350
C ------- PRINT THE READ-IN NEUMANN BOUNDARY ELEMENT SIDES           DATA3355
C                                                                    DATA3360
      LINE=0                                                         DATA3365
      DO 960 MP=1,NNEL,5                                             DATA3370
      LINE=LINE+1                                                    DATA3375
      IF(MOD(LINE-1,50).EQ.0) PRINT 9600                             DATA3380
      NJMN=MP                                                        DATA3385
      NJMX=MIN0(MP+4,NNEL)                                           DATA3390
      PRINT 9650, (J,MNES(J),ISN(1,J),ISN(2,J),J=NJMN,NJMX)          DATA3395
  960 CONTINUE                                                       DATA3400
C                                                                    DATA3405
C ------- COMPUTE LENGTH AND DIRECTIONAL COSINES OF ALL NEUMANN      DATA3410
C ------- BOUNDARY ELEMENT SIDES                                     DATA3415
C                                                                    DATA3420
      DO 990 MPI=1,NNEL                                              DATA3425
      MI=MNES(MPI)                                                   DATA3430
      DO 980 MPJ=1,NBEL                                              DATA3435
      MJ=NBE(MPJ)                                                    DATA3440
      IF(MJ.NE.MI) GO TO 980                                         DATA3445
      I=ISN(1,MPI)                                                   DATA3450
      NI=NPNB(I)                                                     DATA3455
      J=ISN(2,MPI)                                                   DATA3460
      NJ=NPNB(J)                                                     DATA3465
      NIB=ISB(1,MPJ)                                                 DATA3470
      NJB=ISB(2,MPJ)                                                 DATA3475
      IF(NIB.EQ.NI .AND. NJB.EQ.NJ) GO TO 970                        DATA3480
      IF(NIB.EQ.NJ .AND. NJB.EQ.NI) GO TO 965                        DATA3485
      GO TO 980                                                      DATA3490
C                                                                    DATA3495
  965 ISN(1,MPI)=J                                                   DATA3500
      ISN(2,MPI)=I                                                   DATA3505
  970 DO 975 JJ=3,4                                                  DATA3510
  975 ISN(JJ,MPI)=ISB(JJ,MPJ)                                        DATA3515
```

```
      DLN(MPI)=DLB(MPJ)                                                 DATA3520
      DCOSXN(MPI)=DCOSXB(MPJ)                                           DATA3525
      DCOSZN(MPI)=DCOSZB(MPJ)                                           DATA3530
      GO TO 990                                                         DATA3535
C                                                                       DATA3540
  960 CONTINUE                                                          DATA3545
      PRINT 9800, MPI,I,J,NI,NJ,MI                                      DATA3550
  990 CONTINUE                                                          DATA3555
C                                                                       DATA3560
C -------- PRINT THE COMPUTED NEUMANN BOUNDARY ELEMENT SIDE INFORMATION DATA3565
C                                                                       DATA3570
      LINE=0                                                            DATA3575
      DO 995 MP=1,NNEL,2                                                DATA3580
      LINE=LINE+1                                                       DATA3585
      IF(MOD(LINE-1,50).EQ.0) PRINT 9900                                DATA3590
      NJMN=MP                                                           DATA3595
      NJMX=MIN0(MP+1,NNEL)                                              DATA3600
      PRINT 6950, (J,MNES(J),(ISN(I,J),I=1,4),DLN(J),DCOSXN(J),         DATA3605
     1 DCOSZN(J),J=NJMN,NJMX)                                           DATA3610
  995 CONTINUE                                                          DATA3615
C                                                                       DATA3620
  999 CONTINUE                                                          DATA3625
C                                                                       DATA3630
      RETURN                                                            DATA3635
C                                                                       DATA3640
   10 FORMAT(16I5)                                                      DATA3645
   20 FORMAT(8D10.3)                                                    DATA3650
   30 FORMAT(80I1)                                                      DATA3655
   40 FORMAT(3I5,5X,4D10.3)                                             DATA3660
C                                                                       DATA3665
 1000 FORMAT(40H0      **** BAISC INTEGER PARAMETERS ****//5X,         DATA3670
     1 40H NUMBER OF NODAL POINTS. . . . . . . . . . .,I5/ 5X,          DATA3675
     2 40H NUMBER OF ELEMENTS. . . . . . . . . . . . .,I5/ 5X,          DATA3680
     3 40H NUMBER OF DIFFERENT MATERIALS . . . . . . .,I5/ 5X,          DATA3685
     4 40H NUMBER OF CORRECTION MATERIALS. . . . . . .,I5/ 5X,          DATA3690
     5 40H NUMBER OF TIME INCREMENTS . . . . . . . . .,I5//5X,          DATA3695
     6 40H STEADY-STATE I.C. CONTROL . . . . . . . . .,I5/ 5X,          DATA3700
     7 40H SOIL-PROPERTY CONTROL . . . . . . . . . . .,I5/ 5X,          DATA3705
     8 40H NUMBER OF SOIL PARAMETERS . . . . . . . . .,I5/ 5X,          DATA3710
     9 40H NUMBER OF MATERIAL PROPERTIES . . . . . . .,I5//5X,          DATA3715
     A 40H AUXILIARY STORAGE CONTROL . . . . . . . . .,I5/ 5X,          DATA3720
     B 40H CONDUCTIVITY-PERMEABILITY CONTROL . . . . .,I5/ 5X,          DATA3725
     C 40H GRAVITY CONTROL . . . . . . . . . . . . . .,I5/ 5X,          DATA3730
     D 40H RESTART PARAMETER . . . . . . . . . . . . .,I5//5X,          DATA3735
     E 40H NO. OF ITERATIONS PER CYCLE . . . . . . . .,I5/ 5X,          DATA3740
     F 40H NO. OF CYCLES PER TIME STEP . . . . . . . .,I5/ 5X,          DATA3745
     G 40H NO. OF TIMES TO RESET TIME STEP SIZE . . .,I5/ 5X,           DATA3750
     H 40H NO. OF POINTWISE ITERATIONS ALLOWED . . . .,I5/)             DATA3755
```

```
1050 FORMAT(1H0,4X,                                                    DATA3760
    1 40H LUMPING INDICATOR, ILUMP. . . . . . . . .,I5/ 5X,            DATA3765
    2 40H TIME-DIFFERENCE INDICATOR, INID . . . . .,I5/ 5X,            DATA3770
    3 40H IS POINTWISE ITERATION SOLUTION USED?  ,I5/ 5X,              DATA3775
    4 40H IS VELOCITY-SOLVING MATRIX LUMPED? .   ,I5//)                DATA3780
1100 FORMAT(5X,40H TIME INCREMENT. . . . . . . . . . . . . . .,E15.6/ 5X,  DATA3785
    1 40H MULTIPLIER FOR INCREASING DELT. . . . . .,E15.6/ 5X,         DATA3790
    2 40H MAXIMUM VALUE OF DELT . . . . . . . . . .,E15.6/ 5X,         DATA3795
    3 40H MAXIMUM VALUE OF TIME . . . . . . . . . .,E15.6//5X,         DATA3800
    4 40H DEGREES OF PRIN-AXIS INCLINATION. . . . .,E15.6/ 5X,         DATA3805
    5 40H STEADY-STATE TOLERANCE. . . . . . . . . .,E15.6/ 5X,         DATA3810
    6 40H TRANSIENT-STATE TOLERANCE . . . . . . . .,E15.6//5X,         DATA3815
    7 40H DENSITY OF WATER. . . . . . . . . . . . .,E15.6/ 5X,         DATA3820
    8 40H ACCELERATION OF GRAVITY . . . . . . . . .,E15.6/ 5X,         DATA3825
    9 40H VISCOSITY OF WATER. . . . . . . . . . . .,E15.6//5X,         DATA3830
    A 40H TIME-INTEGRATION PARAMETER. . . . . . . .,E15.6/ 5X,         DATA3835
    B 40H ITERATION PARAMETER FOR NONLINEAR EQ.  ,E15.6/ 5X,           DATA3840
    C 40H RELAXATION PARAMETER FOR POINTWISE SOL.,E15.6/ 5X,           DATA3845
    D 40H CYLINDER COORDINATOR PARAMETER . . . . ,E15.6//)             DATA3850
1200 FORMAT(//6X,14HOUTPUT CONTROL)                                    DATA3855
1210 FORMAT(10X,50I2)                                                  DATA3860
1300 FORMAT(//6X,19HDISK OUTPUT CONTROL)                               DATA3865
1350 FORMAT(1H0,6X,'TIME OF CHANGING DELT'/(10X,8D12.4))               DATA3870
1400 FORMAT(36H1     **** MATERIAL PROPERTIES **** // 9H MAT. NO., 9(  DATA3875
    > 3A4))                                                            DATA3880
1410 FORMAT(I8,9D12.4)                                                 DATA3885
1500 FORMAT(44H1INPUT TABLE 3A. MOISTURE-CONTENT PARAMETERS//          DATA3890
    > 9H MAT. NO.,8(3A4))                                              DATA3895
1510 FORMAT(I8,9D12.4/(8X,9D12.4))                                     DATA3900
1600 FORMAT(40H1     **** CONDUCTIVITY PARAMETERS **** // 9H MAT. NO., DATA3905
    > 2(3A4))                                                          DATA3910
1700 FORMAT(53H1     **** SOIL PROPERTY INTERPOLATION VALUES ****   // DATA3915
    > 9H MAT. NO.,9X,8HPRESSURE,13X,16HMOISTURE CONTENT,4X,            DATA3920
    > 25H  RELATIVE CONDUCTIVITY ,6X,14HWATER CAPACITY)                DATA3925
1710 FORMAT(I8,D19.4,3D25.4/(2X,4D25.4))                               DATA3930
2100 FORMAT(38H1     **** NODAL COORDINATE DATA **** //                DATA3935
    > 4HNODE,10X,1HX,10X,1HZ,4X,4HNODE,10X,1HX,10X,1HZ,4X,4HNODE,      DATA3940
    > 10X,1HX,10X,1HZ,4X,4HNODE,10X,1HX,10X,1HZ/                       DATA3945
    > 27H***************************,3X,27H***************************,DATA3950
    >3X,27H***************************,3X,27H***************************DATA3955
    >*/)                                                               DATA3960
2110 FORMAT(1H ,I5,2D11.3,3X,I5,2D11.3,3X,I5,2D11.3,3X,I5,2D11.3)      DATA3965
2700 FORMAT(62H1     **** ELEMENT DATA: GLOBAL INDICES OF ELEMENT NODESDATA3970
    1 **** /3(5X,5H  ELM,5H NOD1,5H NOD2,5H NOD3,5H NOD4,5H MTYP)/     DATA3975
    2 3(5X,5H  ---,5H ----,5H ----,5H ----,5H ----,5H ----))           DATA3980
3000 FORMAT(////26H ERROR IN ELEMENT CARD NO.,I5///)                   DATA3985
3100 FORMAT(3(5X,6I5))                                                 DATA3990
3150 FORMAT(1H0/5X,31H *** MATRIX BAND SIZE: IHAFLB =,I3,10H   IBAND =,DATA3995
    1 I3,10H    IHBP =,I3,6H   ***)                                    DATA4000
4200 FORMAT(////40H ERROR IN MATERIAL TYPE CODE FOR ELEMENT,I5///)     DATA4005
```

```
4300 FORMAT(////28H EXECUTION HALTED BECAUSE OF,I5,13H FATAL ERRORS///)DATA4010
5100 FORMAT('1      **** TRANSIENT INTEGERS **** '// 5X,              DATA4015
    > 40H NO. OF SOURCE ELEMENT . . . . . . . .,I5/ 5X,               DATA4020
    1 40H NO. OF SOURCE PROFILES . . . . . . . . .,I5/ 5X,            DATA4025
    2 40H NO. OF DATA POINTS FOR EACH SOURCE PROF,I5/ 5X,             DATA4030
    3 40H NO. OF RAINFALL SEEPAGE ELEMENT S-DES .,I5/ 5X,             DATA4035
    4 40H NO. OF RAINFALL SEEPAGE NODES . . . . .,I5/ 5X,             DATA4040
    5 40H NO. OF RAINFALL PROFILES . . . . . . .,I5/ 5X,              DATA4045
    6 40H NO. OF DATA POINTS FOR EACH RAINFALL . .,I5/)               DATA4050
5150 FORMAT(1H /5X,                                                   DATA4055
    1 40H NO. OF DIRICHLET NODAL POINTS . . . . . .,I5/ 5X,           DATA4060
    2 40H NO. OF DIRICHLET HEAD PROFILES . . . . .,I5/ 5X,            DATA4065
    3 40H NO. OF DATA POINTS FOR DIRICHLET PROF..,I5/ 5X,             DATA4070
    4 40H NO. OF CAUCHY NODAL POINTS . . . . . . .,I5/ 5X,            DATA4075
    5 40H NO. OF CAUCHY ELEMENT SIDES . . . . . .,I5/ 5X,             DATA4080
    6 40H NO. OF CAUCHY FLUX PROFILES . . . . . .,I5/ 5X,             DATA4085
    7 40H NO. OF DATA POINT FOR CAUCHY PROFILES .,I5/ 5X,             DATA4090
    8 40H NO. OF NEUMANN NODAL POINTS . . . . . .,I5/ 5X,             DATA4095
    9 40H NO. OF NEUMANN ELEMENT SIDES . . . . . .,I5/ 5X,            DATA4100
    A 40H NO. OF NEUMANN FLUX PROFILES . . . . . .,I5/ 5X,            DATA4105
    B 40H NO. OF DATA POINTS FOR NEUMANN PROFILE ,I5/)                DATA4110
5155 FORMAT(1H /5X,                                                   DATA4115
    > 40H NO. OF WELL SOURCE/SINK NODES . . .... ,I5/ 5X,             DATA4120
    > 40H NO. OF WELL SOURCE/SINK PROFILES . . . ,I5/ 5X,             DATA4125
    > 40H NO. OF DATA POINTS IN WELL PROFILE . . ,I5/)                DATA4130
5300 FORMAT(1H1/5X,27H *** SOURCE INFORMATION ***)                    DATA4135
5500 FORMAT(1H0/5X,12H PROFILE NO.,I2,/ 5(4X,4HTIME,6X,6HSOURCE,2X)/  DATA4140
    > 5(4X,4H----,6X,6H------,2X))                                    DATA4145
5510 FORMAT(1H ,5(2D11.3))                                            DATA4150
5600 FORMAT(1H0//5X,' GLOBAL ELEMENT NUMBER AND PROFILE TYPE OF ELEMENTDATA4155
    1 SOURCE'//5X,5('   MP  LES STYP     '))                           DATA4160
5650 FORMAT(1H ,4X,5(3I5,5X))                                         DATA4165
5700 FORMAT(1H0///5X,37H *** WELL SOURCE/SINK INFORMATION ***)        DATA4170
5710 FORMAT(1H0/5X,12H PROFILE NO.,I2/ 4(4X,4HTIME,6X,6HSOURCE,2X)/   DATA4175
    1  4(4X,4H----,6X,6H -----,2X))                                   DATA4180
5800 FORMAT(1H0//10X,65H GLOBAL NODAL NUMBER AND PROFILE TYPE OF WELL SDATA4185
    1OUCE/SINK NODES   //5X,5(5H    I,5H  NPW,5H WTYP,5X))           DATA4190
5850 FORMAT(1H ,4X,5(3I5,5X))                                         DATA4195
6000 FORMAT(1H1/5X,46H **** RAINFALL-SEEPAGE BOUNDARY CONDITIONS ***) DATA4200
6100 FORMAT(1H0///10X,25H --- RAINFALL PROFILE ---)                   DATA4205
6150 FORMAT(1H0/5X,12H PROFILE NO.,I2,/ 5(4X,4HTIME,6X,6H RAINS,2X)/  DATA4210
    > 5(4X,4H----,6X,6H------,2X))                                    DATA4215
6200 FORMAT(1H0/10X,32H --- LIST OF RS-NODAL POINTS ---//(10X,20I5))  DATA4220
6220 FORMAT(1H0/10X,40H --- RAINFALL TYPE AND PONDING DEPTH ---//5X,  DATA4225
    1 3(5H    I,5H NPRS,5H RTYP,11H   HCON(I) ,11H   HMIN(I) .2X)/)   DATA4230
6250 FORMAT(1H ,4X,3(3I5,D11.3,D11.3,2X))                             DATA4235
6500 FORMAT(1H0,10X,61H *** ERROR IN READING RAINFALL-SEEPAGE ELEMENT DATA4240
    1SIDE STOP ***)                                                   DATA4245
6600 FORMAT(1H0//10X,50H --- RAINFALL-SEEPAGE ELEMENT SIDE DATA ---   DATA4250
    1   //5X,5(5H   MP,5H   M,5H CNP1,5H CNP2,2X))                    DATA4255
```

```
6650 FORMAT(1H ,4X,5(4I5,2X))                                          DATA4260
6800 FORMAT(1H0,5X,I5,61H-TH RS SIDE IS NOT ONE OF THE BOUNDARY SIDES; DATA4265
    1 I,J,NI,NJ,MI =,5I5)                                              DATA4270
6900 FORMAT(1H1//10X,59H *** COMPUTED RAINFALL SEEPAGE ELEMENT SIDE INFDATA4275
    1ORMATION ***//5X,2(5H   MP,5H    M,5H  NP1,5H  NP2,3H L1,3H L2,11HDATA4280
    2   DL(MP)  ,11H DCOSX(MP) ,11H DCOSZ(MP) ,1X))                    DATA4285
6950 FORMAT(1H ,4X,2(4I5,2I3,3D11.3,1X))                               DATA4290
7000 FORMAT(1H1/5X,40H **** DIRICHLET BOUNDARY CONDITIONS ****)        DATA4295
7100 FORMAT(1H0/5X,12H PROFILE NO.,I2,/ 4(4X,4HTIME,6X,6H HEAD ,2X)/   DATA4300
    > 4(4X,4H----,6X,6H ---- ,2X))                                     DATA4305
7200 FORMAT(1H0//10X,65H GLOBAL NODAL NUMBER AND PROFILE TYPE OF DIRICHDATA4310
    1LET BOUNDARY NODES//5X,5(5H    I,5H NPDB,5H DTYP,5X))            DATA4315
7250 FORMAT(1H ,4X,5(3I5,5X))                                          DATA4320
8000 FORMAT(1H1/5X,38H **** CAUCHY  BOUNDARY CONDITIONS ****)          DATA4325
8100 FORMAT(1H0/5X,12H PROFILE NO.,I2/ 4(4X,4HTIME,6X,6H FLUX ,2X)/    DATA4330
    > 4(4X,4H----,6X,6H ---- ,2X))                                     DATA4335
8200 FORMAT(1H0//10X,62H GLOBAL NODAL NUMBER AND PROFILE TYPE OF CAUCHYDATA4340
    1 BOUNDARY NODES//5X,5(5H    I,5H NPCB,5H CTYP,5X)/)               DATA4345
8300 FORMAT(/// 38H ERROR IN READING CAUCHY  ELEMENT SIDE//)           DATA4350
8900 FORMAT(1H0//10X,26H CAUCHY ELEMENT SIDE DATA //5X,                DATA4355
    1 5(5H   MP,5H CNP1,5H CNP2,5X))                                   DATA4360
8950 FORMAT(1H ,4X,5(3I5,5X))                                          DATA4365
9000 FORMAT(1H1/5X,38H **** NEUMANN BOUNDARY CONDITIONS ****)          DATA4370
9100 FORMAT(1H0/5X,12H PROFILE NO.,I2/ 4(4X,4HTIME,6X,6H FLUX ,2X)/    DATA4375
    > 4(4X,4H----,6X,6H ---- ,2X))                                     DATA4380
9200 FORMAT(1H0//10X,63H GLOBAL NODAL NUMBER AND PROFILE TYPE OF NEUMANDATA4385
    1N BOUNDARY NODES//5X,5(5H    I,5H NPNB,5H NTYP,5X))               DATA4390
9300 FORMAT(/// 38H ERROR IN READING NEUMANN ELEMENT SIDE//)           DATA4395
9600 FORMAT(1H0//10X,27H NEUMANN ELEMENT SIDE DATA //5X,               DATA4400
    1 5(5H   MP,5H    M,5H CNP1,5H CNP2,2X))                           DATA4405
9650 FORMAT(1H ,4X,5(4I5,2X))                                          DATA4410
9800 FORMAT(1H0,5X,I5,65H-TH NEUMANN SIDE IS NOT ONE OF THE BOUNDARY SIDATA4415
    1DES; I,J,NI,NJ,MI =,5I5)                                          DATA4420
9900 FORMAT(1H1//10X,50H *** COMPUTED NEUMANN ELEMENT SIDE INFORMATION DATA4425
    1***//5X,2(5H   MP,5H    M,5H CNP1,5H CNP2,3H L1,3H L2,11H   DL(MP)DATA4430
    2   ,11H DCOSXN(MP),11H DCOSZM(MP),1X))                            DATA4435
      END                                                              DATA4440


      SUBROUTINE SURF(X,Z,IE, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,           SURF 005
     > MAXNP,MAXEL, MAXBEL,MAXBNP)                                     SURF 010
C                                                                      SURF 015
      IMPLICIT REAL*8(A-H,O-Z)                                         SURF 020
C                                                                      SURF 025
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,IBAND,KGRAV,NTI                  SURF 030
C                                                                      SURF 035
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)                          SURF 040
```

```
      DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),NBE(MAXBEL),  SURF 045
     > ISB(4,MAXBEL),NPB(MAXBNP)                                        SURF 050
C                                                                       SURF 055
C ------- FIND SURFACE SIDES BY LOCATING NONDUPLICATED SIDES            SURF 060
C                                                                       SURF 065
      NBEL=0                                                            SURF 070
      NBN=0                                                             SURF 075
      DO 390 MI=1,NEL                                                   SURF 080
      NQ=4                                                              SURF 085
      IEM4=IE(MI,4)                                                     SURF 090
      IF(IEM4.EQ.0) NQ=3                                                SURF 095
      DO 380 IQ=1,NQ                                                    SURF 100
      IQ1=IQ+1                                                          SURF 105
      IF(IQ.EQ.NQ) IQ1=1                                                SURF 110
C                                                                       SURF 115
      DO 190 MJ=1,NEL                                                   SURF 120
      IF(MJ.EQ.MI) GO TO 190                                            SURF 125
      MQ=4                                                              SURF 130
      IEM4=IE(MJ,4)                                                     SURF 135
      IF(IEM4.EQ.0) MQ=3                                                SURF 140
      DO 180 JQ=1,MQ                                                    SURF 145
      JQ1=JQ+1                                                          SURF 150
      IF(JQ.EQ.MQ) JQ1=1                                                SURF 155
      IF(IE(MI,IQ).EQ.IE(MJ,JQ).AND.IE(MI,IQ1).EQ.IE(MJ,JQ1)) GO TO 380 SURF 160
      IF(IE(MI,IQ).EQ.IE(MJ,JQ1).AND.IE(MI,IQ1).EQ.IE(MJ,JQ)) GO TO 380 SURF 165
  180 CONTINUE                                                          SURF 170
  190 CONTINUE                                                          SURF 175
C                                                                       SURF 180
C ******* A BOUNDARY SIDE HAS BEEN FOUND.                               SURF 185
C                                                                       SURF 190
      NI=IE(MI,IQ)                                                      SURF 195
      NJ=IE(MI,IQ1)                                                     SURF 200
      NBEL=NBEL+1                                                       SURF 205
      NBE(NBEL)=MI                                                      SURF 210
      ISB(1,NBEL)=NI                                                    SURF 215
      ISB(2,NBEL)=NJ                                                    SURF 220
      ISB(3,NBEL)=IQ                                                    SURF 225
      ISB(4,NBEL)=IQ1                                                   SURF 230
C                                                                       SURF 235
      IF(NBEL.GT.1) GO TO 200                                           SURF 240
C                                                                       SURF 245
      NBN=NBN+1                                                         SURF 250
      NPB(NBN)=NI                                                       SURF 255
      NBN=NBN+1                                                         SURF 260
      NPB(NBN)=NJ                                                       SURF 265
C                                                                       SURF 270
  200 DO 210 I=1,NBN                                                    SURF 275
      IJ=NPB(I)                                                         SURF 280
      IF(IJ.EQ.NI) GO TO 250                                            SURF 285
  210 CONTINUE                                                          SURF 290
```

```
      NBN=NBN+1                                                         SURF 295
      NPB(NBN)=NI                                                       SURF 300
C                                                                       SURF 305
  250 DO 260 J=1,NBN                                                    SURF 310
      IJ=NPB(J)                                                         SURF 315
      IF(IJ.EQ.NJ) GO TO 290                                            SURF 320
  260 CONTINUE                                                          SURF 325
      NBN=NBN+1                                                         SURF 330
      NPB(NBN)=NJ                                                       SURF 335
C                                                                       SURF 340
  290 CONTINUE                                                          SURF 345
C                                                                       SURF 350
  380 CONTINUE                                                          SURF 355
  390 CONTINUE                                                          SURF 360
C                                                                       SURF 365
C ------- CALCULATE SIDE LENGTH AND DIRECTION COSINES                   SURF 370
C                                                                       SURF 375
      DO 490 MP=1,NBEL                                                  SURF 380
      NI=ISB(1,MP)                                                      SURF 385
      NJ=ISB(2,MP)                                                      SURF 390
      DX=X(NJ)-X(NI)                                                    SURF 395
      DZ=Z(NJ)-Z(NI)                                                    SURF 400
      DLB(MP)=DSQRT(DX*DX+DZ*DZ)                                        SURF 405
C                                                                       SURF 410
      BETA=DATAN2(DZ,DX)                                                SURF 415
      DCOSXB(MP)=DSIN(BETA)                                             SURF 420
      DCOSZB(MP)=-DCOS(BETA)                                            SURF 425
  490 CONTINUE                                                          SURF 430
C                                                                       SURF 435
C ------- PRINT BOUNDARY SIDE INFORMATION                               SURF 440
C                                                                       SURF 445
      LINE=0                                                            SURF 450
      DO 695 MP=1,NBEL,2                                                SURF 455
      LINE=LINE+1                                                       SURF 460
      IF(MOD(LINE-1,50).EQ.0) PRINT 6900                                SURF 465
      NJMN=MP                                                           SURF 470
      NJMX=MIN0(MP+1,NBEL)                                              SURF 475
      PRINT 6950, (J, NBE(J),(ISB(I,J),I=1,4),DLB(J),DCOSXB(J),DCOSZB(J)SURF 480
     1,J=NJMN,NJMX)                                                     SURF 485
  695 CONTINUE                                                          SURF 490
C                                                                       SURF 495
 6900 FORMAT(1H1//10X,51H *** COMPUTED BOUNDARY ELEMENT SIDE INFORMATIONSURF 500
     1 ***//5X,2(5H   MP,5H    M,5H  NP1,5H  NP2,3H L1,3H L2,11H   DLB(MSURF 505
     2P) ,11H DCOSXB(MP),11H DCOSZB(MP),1X))                            SURF 510
 6950 FORMAT(1H ,4X,2(4I5,2I3,3D11.3,1X))                               SURF 515
C                                                                       SURF 520
      RETURN                                                            SURF 525
      END                                                               SURF 530
```

```
      SUBROUTINE LNDGEN(LLNODE,IE,MAXNP,MAXEL,NNP,NEL)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /CPIS/ JBAND
C
      DIMENSION LLNODE(JBAND,MAXNP),IE(MAXEL,5)
C
      DO 490 NP=1,NNP
C
      DO 110 I=1,JBAND
  110 LLNODE(I,NP)=0
C
      NLNOD=0
      KOUNT=0
C
      DO 390 M=1,NEL
      IEM1=IE(M,1)
      IEM2=IE(M,2)
      IEM3=IE(M,3)
      IEM4=IE(M,4)
      NQ=4
      IF(IEM4.EQ.0) NQ=3
C
      IF(IEM4.EQ.0) GO TO 210
C
C ------- FOR THE CASE OF QUADRILATERAL ELEMENT
C
      IF(IEM1.NE.NP .AND. IEM2.NE.NP .AND. IEM3.NE.NP .AND. IEM4.NE.NP)
     1 GO TO 390
      GO TO 220
C
C ------- FOR THE CASE OF TRIANGULAR ELEMENT
C
  210 IF(IEM1.NE.NP .AND. IEM2.NE.NP .AND. IEM3.NE.NP) GO TO 390
C
C ------- MERGE THE QUADRILATERAL AND TRIANGULAR ELEMENT BRANCH
C
  220 DO 290 IQ=1,NQ
      NI=IE(M,IQ)
C
C ******* COMPRESS THE NODE 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
      LLNODE(KOUNT,NP)=NI
      GO TO 290
```

```
C                                                                       LNDG 260
C ------- IF NLNOD IS GREATER THAN 1, WE HAVE TO CHECK IF NI IS THE     LNDG 265
C ------- NODE ALREADY COMPRESSED? IF YES, SKIP. IF NOT INCREASE THE    LNDG 270
C ------- KOUNT.                                                        LNDG 275
C                                                                       LNDG 280
  230 DO 240 J=1,KOUNT                                                  LNDG 285
      NJ=LLNODE(J,NP)                                                   LNDG 290
      IF(NI.EQ.NJ) GO TO 290                                            LNDG 295
  240 CONTINUE                                                          LNDG 300
C                                                                       LNDG 305
C ------- THE NODE NI HAS NOT BEEN COMPRESSED YET.  HENCE COMPRESS IT.  LNDG 310
C                                                                       LNDG 315
      KOUNT=KOUNT+1                                                     LNDG 320
      LLNODE(KOUNT,NP)=NI                                               LNDG 325
C                                                                       LNDG 330
  290 CONTINUE                                                          LNDG 335
C                                                                       LNDG 340
  390 CONTINUE                                                          LNDG 345
C                                                                       LNDG 350
      IF(KOUNT.LE.JBAND) GO TO 490                                      LNDG 355
      KONT=KOUNT-1                                                      LNDG 360
      JBND=JBAND-1                                                      LNDG 365
      WRITE(6,1000) NP,KONT,JBND                                        LNDG 370
      STOP                                                              LNDG 375
C                                                                       LNDG 380
  490 CONTINUE                                                          LNDG 385
C                                                                       LNDG 390
      LINE=0                                                            LNDG 395
      DO 590 NP=1,NNP,2                                                 LNDG 400
      NJMN=NP                                                           LNDG 405
      NJMX=MIN0(NP+1,NNP)                                               LNDG 410
      LINE=LINE+1                                                       LNDG 415
      IF(MOD(LINE-1,50).EQ.0) PRINT 5000                                LNDG 420
      PRINT 5100, (NJ,(LLNODE(I,NJ),I=1,JBAND),NJ=NJMN,NJMX)            LNDG 425
  590 CONTINUE                                                          LNDG 430
C                                                                       LNDG 435
 1000 FORMAT(1H0//5X,' ****',I5,'-TH NODE HAS',I3,                      LNDG 440
     1 'NODES SURROUNDING IT, WHICH IS MORE THAN JBAND - 1 =',I3,       LNDG 445
     2 '  STOP  ***')                                                   LNDG 450
 5000 FORMAT(1H1/5X,47H ** GENERATED SURROUNDING NODES OF ALL NODES **,/LNDG 455
     1 2(2X,5H   NP,5H    1,5H    2,5H    3,5H    4,5H    5,5H    6,5H LNDG 460
     2  7,5H    8,5H    9,5H   10,5H   11)/ 2(2X,5H   --,5H    -,5H   -LNDG 465
     3,5H    -,5H    -,5H    -,5H    -,5H    -,5H    -,5H    -,5H   --, LNDG 470
     4 5H   --)//)                                                      LNDG 475
 5100 FORMAT(1H ,2(2X,12I5))                                            LNDG 480
C                                                                       LNDG 485
      RETURN                                                            LNDG 490
      END                                                               LNDG 495
```

```
      SUBROUTINE VELT(X,Z,IE, C,H,HT,VX,VZ,AKX,AKZ,AKXZ,
     > MAXEL,MAXNP,MAXHBP)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,IBAND,KGRAV,NTI
      COMMON /COORD/ CSFE,SNFE,CYLIND
      COMMON /OPT/ ILUMP,IMID,IPNTS,IVML
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)
      DIMENSION C(MAXNP,MAXHBP),H(MAXNP),HT(MAXNP),VX(MAXNP),VZ(MAXNP),
     > AKX(4,MAXEL),AKZ(4,MAXEL),AKXZ(4,MAXEL)
C
      DIMENSION QB(4,4),QRX(4),QRZ(4),XQ(4),ZQ(4),HTQ(4),
     > AKXG(4),AKZG(4),AKXZG(4)
C
      IHALFB=(IBAND-1)/2
      IHBP=IHALFB+1
      AGRAV=DFLOAT(KGRAV)
C
C ------- INITIATE THE DARCY VELOCITY VX(NP) AND VZ(NP)
C
      DO 100 NP=1,NNP
      VX(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*(X(NP)*SNFE+Z(NP)*CSFE)
C
C ------- INITIATE MATRIX C(NP,IB)
C
      IF(IPNTS.EQ.1 .OR. IVML.EQ.1) GO TO 150
      DO 110 NP=1,NNP
      DO 110 IB=1,IHBP
  110 C(NP,IB)=0.0
      GO TO 190
  150 DO 155 NP=1,NNP
  155 C(NP,1)=0.0
  190 CONTINUE
C
C ------- ASSEMBLE GLOBAL MATRIX AND LOAD VECTOR.
C
      DO 290 M=1,NEL
      NQ=4
      IEM4=IE(M,4)
      IF(IEM4.EQ.0) NQ=3
C
      DO 210 IQ=1,NQ
      NP=IE(M,IQ)
```

```
      XQ(IQ)=X(NP)                                                      VELT 260
      ZQ(IQ)=Z(NP)                                                      VELT 265
  210 HTQ(IQ)=HT(NP)                                                    VELT 270
C                                                                       VELT 275
      DO 215 KG=1,NQ                                                    VELT 280
      AKXG(KG)=AKX(KG,M)                                                VELT 285
      AKXZG(KG)=AKXZ(KG,M)                                              VELT 290
  215 AKZG(KG)=AKZ(KG,M)                                                VELT 295
C                                                                       VELT 300
      IF(NQ.EQ.3) CALL Q3D(QB,QRX,QRZ,XQ,ZQ,AKXG,AKZG,AKXZG,HTQ)       VELT 305
      IF(NQ.EQ.4) CALL Q4D(QB,QRX,QRZ,XQ,ZQ,AKXG,AKZG,AKXZG,HTQ)       VELT 310
C                                                                       VELT 315
      DO 280 IQ=1,NQ                                                    VELT 320
      NI=IE(M,IQ)                                                       VELT 325
      DO 240 JQ=1,NQ                                                    VELT 330
      NJ=IE(M,JQ)                                                       VELT 335
C                                                                       VELT 340
      IF(IPNTS.EQ.1 .OR. IVML.EQ.1) GO TO 230                           VELT 345
C                                                                       VELT 350
      IF(NJ.LT.NI) GO TO 240                                            VELT 355
      IB=NJ-NI+1                                                        VELT 360
      C(NI,IB)=C(NI,IB)+QB(IQ,JQ)                                       VELT 365
      GO TO 240                                                         VELT 370
C                                                                       VELT 375
  230 C(NI,1)=C(NI,1)+QB(IQ,JQ)                                         VELT 380
  240 CONTINUE                                                          VELT 385
C                                                                       VELT 390
      VX(NI)=VX(NI)+QRX(IQ)                                             VELT 395
      VZ(NI)=VZ(NI)+QRZ(IQ)                                             VELT 400
  280 CONTINUE                                                          VELT 405
C                                                                       VELT 410
  290 CONTINUE                                                          VELT 415
C                                                                       VELT 420
C ------- SOLVE THE MATRIX EQUATION CX=B                                VELT 425
C                                                                       VELT 430
      IF(IPNTS.EQ.1 .OR. IVML.EQ.1) GO TO 360                           VELT 435
C                                                                       VELT 440
      CALL BANSOL(1,C,VX,NNP,IHBP,MAXNP,MAXHBP)                         VELT 445
      CALL BANSOL(2,C,VX,NNP,IHBP,MAXNP,MAXHBP)                         VELT 450
      CALL BANSOL(2,C,VZ,NNP,IHBP,MAXNP,MAXHBP)                         VELT 455
      GO TO 390                                                         VELT 460
C                                                                       VELT 465
  360 DO 370 NP=1,NNP                                                   VELT 470
      VX(NP)=VX(NP)/C(NP,1)                                             VELT 475
  370 VZ(NP)=VZ(NP)/C(NP,1)                                             VELT 480
C                                                                       VELT 485
  390 CONTINUE                                                          VELT 490
C                                                                       VELT 495
      RETURN                                                            VELT 500
      END                                                               VELT 505
```

```
      SUBROUTINE Q4D(QB,QRX,QRZ,XQ,ZQ,AKXG,AKZG,AKXZG,HTQ)
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 N(4)
C
      COMMON /COORD/ CSFE,SNFE,CYLIND
C
      DIMENSION QB(4,4),QRX(4),QRZ(4),XQ(4),ZQ(4),HTQ(4),
     * AKXG(4),AKZG(4),AKXZG(4)
      DIMENSION S(4),T(4),DNX(4),DNZ(4)
      DIMENSION PJAB(2,2),DNSS(4),DNTT(4)
C
      DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-
     * 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /
C
C ------- INITIATE MATRICES QB(IQ,JQ) AND QRX(IQ) & QRZ(IQ)
C
      DO 100 IQ=1,4
      QRX(IQ)=0.0
      QRZ(IQ)=0.0
      DO 100 JQ=1,4
  100 QB(IQ,JQ)=0.0
C
C ------- SUMMATION OF THE INTEGRAND OVER THE GAUSSIAN POINTS
C
      DO 490 KG=1,4
C
C ------- DETERMINE LOACAL COORDINATE (SS,TT) OF GAUSSIAN POINT KG
C
      SS=P*S(KG)
      TT=P*T(KG)
C
C ------- CALCULATE VALUES OF BASIS FUNCTIONS N(IQ) AND THEIR
C ------- DERIVATIVES DNX(IQ) AND DNZ(IQ) W.R.T. X AND Z, RESPECTIVELY
C ------- AT THE GAUSSIAN POINT KG.
C
         CALL BASE(N,DNSS,DNTT,SS,TT)
C
      DO 210 I=1,2
      DO 210 J=1,2
  210 PJAB(I,J)=0.0
C
      DO 220 I=1,4
      PJAB(1,1)=PJAB(1,1)+ZQ(I)*DNTT(I)
      PJAB(1,2)=PJAB(1,2)-ZQ(I)*DNSS(I)
      PJAB(2,1)=PJAB(2,1)-XQ(I)*DNTT(I)
      PJAB(2,2)=PJAB(2,2)+XQ(I)*DNSS(I)
  220 CONTINUE
C
      DJAC=PJAB(2,2)*PJAB(1,1)-PJAB(1,2)*PJAB(2,1)
```

```
      DJACI=1.0D0/DJAC                                                  Q4D  255
C                                                                       Q4D  260
      DO 230 I=1,2                                                      Q4D  265
      DO 230 J=1,2                                                      Q4D  270
      PJAB(I,J)=PJAB(I,J)*DJACI                                         Q4D  275
  230 CONTINUE                                                          Q4D  280
C                                                                       Q4D  285
      DO 260 I=1,4                                                      Q4D  290
      DNX(I)=DNSS(I)*PJAB(1,1)+DNTT(I)*PJAB(1,2)                        Q4D  295
      DNZ(I)=DNSS(I)*PJAB(2,1)+DNTT(I)*PJAB(2,2)                        Q4D  300
  260 CONTINUE                                                          Q4D  305
C                                                                       Q4D  310
      R=0.0                                                             Q4D  315
      DO 270 I=1,4                                                      Q4D  320
      R=R+XQ(I)*N(I)                                                    Q4D  325
  270 CONTINUE                                                          Q4D  330
C                                                                       Q4D  335
      AKXK=AKXG(KG)                                                     Q4D  340
      AKZK=AKZG(KG)                                                     Q4D  345
      AKXZK=AKXZG(KG)                                                   Q4D  350
      RR=1.0D0                                                          Q4D  355
      IF(CYLIND.NE.0.0) RR=R                                            Q4D  360
C                                                                       Q4D  365
C ------- ACCUMULATE THE SUMS TO OBTAIN THE MATRIX INTEGRALS QB(IQ,JQ)  Q4D  370
C ------- AND QRX(IQ) & QRZ(IQ)                                        Q4D  375
C                                                                       Q4D  380
      DO 390 IQ=1,4                                                     Q4D  385
      DO 390 JQ=1,4                                                     Q4D  390
      QB(IQ,JQ)=QB(IQ,JQ)+ N(IQ)*N(JQ)*DJAC*RR                          Q4D  395
      QRX(IQ)=QRX(IQ)-N(IQ)*HTQ(JQ)*(AKXK*DNX(JQ)+AKXZK*DNZ(JQ))*DJAC*RRQ4D  400
      QRZ(IQ)=QRZ(IQ)-N(IQ)*HTQ(JQ)*(AKXZK*DNX(JQ)+AKZK*DNZ(JQ))*DJAC*RRQ4D  405
  390 CONTINUE                                                          Q4D  410
C                                                                       Q4D  415
  490 CONTINUE                                                          Q4D  420
C                                                                       Q4D  425
      RETURN                                                            Q4D  430
      END                                                               Q4D  435
```

```
      SUBROUTINE Q3D(QB,QRX,QRZ,XQ,ZQ,AKXG,AKZG,AKXZG,HQ)               Q3D  005
C                                                                       Q3D  010
      IMPLICIT REAL*8(A-H,O-Z)                                          Q3D  015
C                                                                       Q3D  020
      COMMON /COORD/ CSFE,SNFE,CYLIND                                   Q3D  025
C                                                                       Q3D  030
```

```
      DIMENSION QB(4,4),QRX(4),QRZ(4),XQ(4),ZQ(4),HQ(4),
     > AKXG(4),AKZG(4),AKXZG(4)
      DIMENSION A(3),B(3),C(3)
      DATA OSIX/0.1666666666666667D0/
      DATA OTWLVE/0.8333333333333333D-1/
C
      DO 110 IQ=1,4
      QRX(IQ)=0.0
      QRZ(IQ)=0.0
      DO 110 JQ=1,4
      QB(IQ,JQ)=0.0
  110 CONTINUE
C
      X1=XQ(1)
      X2=XQ(2)
      X3=XQ(3)
      Z1=ZQ(1)
      Z2=ZQ(2)
      Z3=ZQ(3)
C
      A(1)=X2*Z3 - X3*Z2
      A(2)=X3*Z1 - X1*Z3
      A(3)=X1*Z2 - X2*Z1
C
      R=(X1+X2+X3)/3.0D0
      RR=1.0D0
      IF(CYLIND.NE.0.0) RR=R
C
      AREA2=DABS(A(1)+A(2)+A(3))
      AREA=AREA2/2.0D0*RR
C
      B(1)=(Z2-Z3)/AREA2
      B(2)=(Z3-Z1)/AREA2
      B(3)=(Z1-Z2)/AREA2
      C(1)=(X3-X2)/AREA2
      C(2)=(X1-X3)/AREA2
      C(3)=(X2-X1)/AREA2
C
      AOSIX=AREA*OSIX
      AOTWLV=AREA*OTWLVE
      QB(1,1)=AOSIX
      QB(1,2)=AOTWLV
      QB(1,3)=AOTWLV
      QB(2,1)=AOTWLV
      QB(2,2)=AOSIX
      QB(2,3)=AOTWLV
      QB(3,1)=AOTWLV
      QB(3,2)=AOTWLV
      QB(3,3)=AOSIX
C
```

```
      DHDX=0.0                                                          Q3D  285
      DHDZ=0.0                                                          Q3D  290
      DO 200 IQ=1,3                                                     Q3D  295
      DHDX=DHDX+HQ(IQ)*B(IQ)                                            Q3D  300
  200 DHDZ=DHDZ+HQ(IQ)*C(IQ)                                            Q3D  305
C                                                                       Q3D  310
      QRX(1)=-AREA*DHDX*(AKXG(1)*OSIX+(AKXG(2)+AKXG(3))*OTWLVE) -       Q3D  315
     1 AREA*DHDZ*(AKXZG(1)*OSIX+(AKXZG(2)+AKXZG(3))*OTWLVE)             Q3D  320
      QRX(2)=-AREA*DHDX*((AKXG(1)+AKXG(3))*OTWLVE+AKXG(2)*OSIX) -       Q3D  325
     1 AREA*DHDZ*((AKXZG(1)+AKXZG(3))*OTWLVE+AKXZG(2)*OSIX)             Q3D  330
      QRX(3)=-AREA*DHDX*((AKXG(1)+AKXG(2))*OTWLVE+AKXG(3)*OSIX) -       Q3D  335
     1 AREA*DHDZ*((AKXZG(1)+AKXZG(2))*OTWLVE+AKXZG(3)*OSIX)             Q3D  340
C                                                                       Q3D  345
      QRZ(1)=-AREA*DHDZ*(AKZG(1)*OSIX+(AKZG(2)+AKZG(3))*OTWLVE) -       Q3D  350
     1 AREA*DHDX*(AKXZG(1)*OSIX+(AKXZG(2)+AKXZG(3))*OTWLVE)             Q3D  355
      QRZ(2)=-AREA*DHDZ*((AKZG(1)+AKZG(3))*OTWLVE+AKZG(2)*OSIX) -       Q3D  360
     1 AREA*DHDX*((AKXZG(1)+AKXZG(3))*OTWLVE+AKXZG(2)*OSIX)             Q3D  365
      QRZ(3)=-AREA*DHDZ*((AKZG(1)+AKZG(2))*OTWLVE+AKZG(3)*OSIX) -       Q3D  370
     1 AREA*DHDX*((AKXZG(1)+AKXZG(2))*OTWLVE+AKXZG(3)*OSIX)             Q3D  375
C                                                                       Q3D  380
      RETURN                                                            Q3D  385
      END                                                               Q3D  390


      SUBROUTINE SPROP(IE,H,TH,DTH,AKX,AKZ,AKXZ, PROP,THPROP,AKPROP,    SPRO 005
     > HPROP,CAPROP, MAXEL,MAXNP, NEL,KSP)                              SPRO 010
C                                                                       SPRO 015
      IMPLICIT REAL*8(A-H,O-Z)                                          SPRO 020
      REAL*8 N(4)                                                       SPRO 025
C                                                                       SPRO 030
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM                                SPRO 035
      COMMON /CMTL/ NMAT,NMPPM,NSPPM                                    SPRO 040
C                                                                       SPRO 045
      DIMENSION IE(MAXEL,5),H(MAXNP),TH(4,MAXEL),DTH(4,MAXEL),          SPRO 050
     > AKX(4,MAXEL),AKZ(4,MAXEL),AKXZ(4,MAXEL)                          SPRO 055
C                                                                       SPRO 060
      DIMENSION PROP(MXMPPM,MAXMAT)                                     SPRO 065
      DIMENSION THPROP(MXSPPM,MAXMAT),AKPROP(MXSPPM,MAXMAT),            SPRO 070
     1 HPROP(MXSPPM,MAXMAT),CAPROP(MXSPPM,MAXMAT)                       SPRO 075
C                                                                       SPRO 080
      DIMENSION HQ(4), S(4),T(4), HKG(4)                                SPRO 085
C                                                                       SPRO 090
      DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-    SPRO 095
     > 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /             SPRO 100
```

```
C                                                                       SPRO 105
C -------- TH, DTH/DH, AKX, AND AKZ ARE OBTAINED BY TABLE               SPRO 110
C                                                                       SPRO 115
      DO 900 M=1,NEL                                                    SPRO 120
      NQ=4                                                              SPRO 125
      IEM4=IE(M,4)                                                      SPRO 130
      IF(IEM4.EQ.0) NQ=3                                                SPRO 135
C                                                                       SPRO 140
      DO 110 IQ=1,NQ                                                    SPRO 145
      NP=IE(M,IQ)                                                       SPRO 150
  110 HQ(IQ)=H(NP)                                                      SPRO 155
C                                                                       SPRO 160
      IF(NQ.EQ.3) GO TO 160                                             SPRO 165
C                                                                       SPRO 170
C ------- EVALUATE PRESSURE AT FOUR GAUSSIAN POINTS FOR QUADRILATERAL   SPRO 175
C ------- ELEMENT.                                                      SPRO 180
C                                                                       SPRO 185
      DO 150 KG=1,NQ                                                    SPRO 190
      SS=P*S(KG)                                                        SPRO 195
      TT=P*T(KG)                                                        SPRO 200
      SM=1.0D0-SS                                                       SPRO 205
      SP=1.0D0+SS                                                       SPRO 210
      TM=1.0D0-TT                                                       SPRO 215
      TP=1.0D0+TT                                                       SPRO 220
C                                                                       SPRO 225
      N(1)=0.25D0*SM*TM                                                 SPRO 230
      N(2)=0.25D0*SP*TM                                                 SPRO 235
      N(3)=0.25D0*SP*TP                                                 SPRO 240
      N(4)=0.25D0*SM*TP                                                 SPRO 245
C                                                                       SPRO 250
      HKG(KG)=0.0                                                       SPRO 255
      DO 120 IQ=1,NQ                                                    SPRO 260
  120 HKG(KG)=HKG(KG)+HQ(IQ)*N(IQ)                                      SPRO 265
  150 CONTINUE                                                          SPRO 270
      GO TO 180                                                         SPRO 275
C                                                                       SPRO 280
C ------- EVALUATE PRESSURE AT NODES FOR TRIANGULAR ELEMENT.            SPRO 285
C                                                                       SPRO 290
  160 DO 170 IQ=1,NQ                                                    SPRO 295
  170 HKG(IQ)=HQ(IQ)                                                    SPRO 300
C                                                                       SPRO 305
  180 MTYP=IE(M,5)                                                      SPRO 310
      SATKX=PROP(4,MTYP)                                                SPRO 315
      SATKZ=PROP(5,MTYP)                                                SPRO 320
      SATKXZ=PROP(6,MTYP)                                               SPRO 325
C                                                                       SPRO 330
      IF(KSP.EQ.0) GO TO 600                                            SPRO 335
C                                                                       SPRO 340
C                                                                       SPRO 345
C ------- TH, DTH/DH, AKX, AKZ, AND AKXZ ARE OBTAINED BY TABLE          SPRO 350
C                                                                       SPRO 355
```

```
      DO 500 KG=1,NQ                                                    SPRO 360
      HNP=HKG(KG)                                                       SPRO 365
      IF(HNP.GT.HPROP(1,MTYP)) GO TO 200                                SPRO 370
C                                                                       SPRO 375
C ------- FOR CASE  WHEN PRESSURE IS LESS THAN MINIMUM TABULET VALUE.   SPRO 380
C                                                                       SPRO 385
      JL=1                                                              SPRO 390
      JU=2                                                              SPRO 395
      A=0.0                                                             SPRO 400
      GO TO 400                                                         SPRO 405
C                                                                       SPRO 410
  200 IF(HNP.LT.HPROP(NSPPM,MTYP)) GO TO 250                            SPRO 415
C                                                                       SPRO 420
C ------- FOR CASE  WHEN PRESSURE IS  GT THAN MAXIMUM TABULET VALUE.    SPRO 425
C                                                                       SPRO 430
      JL=NSPPM                                                          SPRO 435
      JU=1                                                              SPRO 440
      A=0.0                                                             SPRO 445
      GO TO 400                                                         SPRO 450
C                                                                       SPRO 455
C ------- FOR CASE  WHEN PRESSURE LIES BETWEEN TABULET VALUES.          SPRO 460
C                                                                       SPRO 465
  250 DO 300 J=2,NSPPM                                                  SPRO 470
      JU=J                                                              SPRO 475
      IF(HPROP(J,MTYP).GT.HNP) GO TO 350                                SPRO 480
C                                                                       SPRO 485
  300 CONTINUE                                                          SPRO 490
  350 JL=JU-1                                                           SPRO 495
      A=(HNP-HPROP(JL,MTYP))/(HPROP(JU,MTYP)-HPROP(JL,MTYP))            SPRO 500
C                                                                       SPRO 505
C ------- THREE CASES MERGE HERE.                                       SPRO 510
C                                                                       SPRO 515
  400 TH(KG,M)=THPROP(JL,MTYP)+A*(THPROP(JU,MTYP)-THPROP(JL,MTYP))      SPRO 520
      DTH(KG,M)=CAPROP(JL,MTYP)+A*(CAPROP(JU,MTYP)-CAPROP(JL,MTYP))     SPRO 525
      USKFCT=AKPROP(JL,MTYP)+A*(AKPROP(JU,MTYP)-AKPROP(JL,MTYP))        SPRO 530
      AKX(KG,M)=SATKX*USKFCT                                            SPRO 535
      AKZ(KG,M)=SATKZ*USKFCT                                            SPRO 540
      AKXZ(KG,M)=SATKXZ*USKFCT                                          SPRO 545
C                                                                       SPRO 550
  500 CONTINUE                                                          SPRO 555
C                                                                       SPRO 560
      GO TO 900                                                         SPRO 565
C                                                                       SPRO 570
  600 CONTINUE                                                          SPRO 575
C                                                                       SPRO 580
C ------- TH, DTH/DH, AKX, AKZ, AND AKXZ ARE OBTAINED ANALYTICALLY.     SPRO 585
C ------- THE READER MUST SUPPLY THE FUNCTIONAL FORM OF FKX, FKZ, AND   SPRO 590
C ------- FTH BELOW                                                     SPRO 595
C                                                                       SPRO 600
```

```
C ------- THE FOLLOWING IS JUST AN EXAMPLE.                                SPRO 605
C ------- WCR= THPROP(MTYP,1)=0.065, 0.050 FOR TWO SAMPLE MATERUALS       SPRO 610
C ------- WCS=THPROP(MTYP,2)=0.364, 0.341 FOR TWO SAMPLE MATERIALS        SPRO 615
C ------- RN=THPROP(MTYP,3)=1.092217, 1.546937 FOR TWO SAMPLE MATERIALS SPRO 620
C ------- ALPH=THPROP(MTYP,4)=0.109, 0.002166 FOR TWO SAMPLE MATERIALS   SPRO 625
C                                                                         SPRO 630
        WCR=THPROP(1,MTYP)                                                SPRO 635
        WCS=THPROP(2,MTYP)                                                SPRO 640
        RN=THPROP(3,MTYP)                                                 SPRO 645
        ALPH=THPROP(4,MTYP)                                               SPRO 650
        RM=1.0D0-1.0D0/RN                                                 SPRO 655
        DO 800 KG=1,NQ                                                    SPRO 660
           HNP=HKG(KG)                                                    SPRO 665
           HNP=-HNP                                                       SPRO 670
C                                                                         SPRO 675
C ------- SATURATED CONDITION                                             SPRO 680
C                                                                         SPRO 685
           IF(HNP.GT.0.0) GO TO 700                                       SPRO 690
           TH(KG,M)=WCS                                                   SPRO 695
           DTH(KG,M)=0.0D0                                                SPRO 700
           AKX(KG,M)=SATKX                                                SPRO 705
           AKZ(KG,M)=SATKZ                                                SPRO 710
           AKXZ(KG,M)=SATKXZ                                              SPRO 715
           GO TO 800                                                      SPRO 720
C                                                                         SPRO 725
C ------- UNSATURATED CASE                                                SPRO 730
C                                                                         SPRO 735
  700      THMKG=WCR+(WCS-WCR)/(1.0D0+(ALPH*HNP)**RN)**RM                 SPRO 740
           TH(KG,M)=THMKG                                                 SPRO 745
           RWC=(THMKG-WCR)/(WCS-WCR)                                      SPRO 750
           TERM=(1.0D0-RWC**(1.0D0/RM))**RM                               SPRO 755
           RK=DSQRT(RWC)*(1.0D0-TERM)*(1.0D0-TERM)                        SPRO 760
           AKX(KG,M)=SATKX*RK                                             SPRO 765
           AKZ(KG,M)=SATKZ*RK                                             SPRO 770
           AKXZ(KG,M)=SATKXZ*RK                                           SPRO 775
           DTH(KG,M)=ALPH*(RN-1.0D0)*TERM*RWC**(1.0D0/RM)*(WCS-WCR)       SPRO 780
C                                                                         SPRO 785
  800    CONTINUE                                                         SPRO 790
  900 CONTINUE                                                            SPRO 795
      RETURN                                                              SPRO 800
      END                                                                 SPRO 805
```

```
      SUBROUTINE BCPREP(IE, H,VX,VZ, DL,DCOSX,DCOSZ,DCYFLX,FLX,RSFLX,
     > HCON,IS,NPRS,NPCON,NPFLX,IRFTYP,TRF,RF,RFALL, TIME,NCHG,
     2 HMIN,NPMIN,X,Z)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,IBAND,KGRAV,NTI
      COMMON /RSBC/ MXRSEL,MXRSNP,MXRFPR,MXRPAR,NRSEL,NRSN,NRFPR,NRFPAR
      COMMON /COORD/ CSFE,SNFE,CYLIND
C
      DIMENSION IE(MAXEL,5),H(MAXNP),VX(MAXNP),VZ(MAXNP)
C
      DIMENSION DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),DCYFLX(MXRSNP),
     > FLX(MXRSNP),RSFLX(MXRSNP),HCON(MXRSNP),IS(4,MXRSEL),NPRS(MXRSNP),
     > NPCON(MXRSNP),NPFLX(MXRSNP),IRFTYP(MXRSNP),TRF(MXRPAR,MXRFPR),
     > RF(MXRPAR,MXRFPR),RFALL(MXRFPR)
      DIMENSION HMIN(MXRSNP),NPMIN(MXRSNP)
      DIMENSION X(MAXNP),Z(MAXNP)
C
C ------- CALCULATE THE RAINFALL RFALL(I) FROM EACH PROFILE
C
      IF (NRFPR.EQ.0) GO TO 200
      DO 160 I=1,NRFPR
         DO 140 J=2,NRFPAR
            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)
            TRFJ=TRF(J,I)
            ABC=RFJ-RFJM1
            ABCD=TRFJ-TRFJM1
            RFALL(I)=RFJM1+(TIME-TRFJM1)*ABC/ABCD
            GO TO 160
  140    CONTINUE
         RFALL(I)=0.0
  160 CONTINUE
C
C ------- DETERMINE NORMAL RAINFALLS FLX(I) AND DARCY FLUXES DCYFLX(I)
C ------- FOR EACH RAINFALL-SEEPAGE NODAL POINT.
C
  200 DO 210 NP=1,NRSN
         FLX(NP)=0.
  210    DCYFLX(NP)=0.
C
      DO 290 MP=1,NRSEL
         I=IS(1,MP)
         J=IS(2,MP)
         NI=NPRS(I)
```

```
         NJ=NPRS(J)                                                     BCPR 255
C                                                                       BCPR 260
      R=(X(NI)+X(NJ))/2.0D0                                             BCPR 265
      RR=1.0D0                                                          BCPR 270
      IF(CYLIND.NE.0.0) RR=R                                            BCPR 275
C                                                                       BCPR 280
         ITYP=IRFTYP(I)                                                 BCPR 285
         JTYP=IRFTYP(J)                                                 BCPR 290
         RFNI=0.                                                        BCPR 295
         RFNJ=0.                                                        BCPR 300
         IF(ITYP.GT.0) RFNI=RFALL(ITYP)                                 BCPR 305
         IF(JTYP.GT.0) RFNJ=RFALL(JTYP)                                 BCPR 310
C                                                                       BCPR 315
C ------- OBTAIN RAINFALL RATES RFNI AND RFNJ AT POINTS NI AND NJ NORMALBCPR 320
C ------- TO THE SIDE SUBTENDED BY THESE POINTS                         BCPR 325
C                                                                       BCPR 330
         PROJ= DCOSX(MP)*SNFE+DCOSZ(MP)*CSFE                            BCPR 335
         RFNI=-RFNI*PROJ*DL(MP)*RR                                      BCPR 340
         RFNJ=-RFNJ*PROJ*DL(MP)*RR                                      BCPR 345
C                                                                       BCPR 350
C ------- CALCULATE RAINFALL FLUX PASSING THROUGH SIDE (NI,NJ) AND      BCPR 355
C ------- DIVIDE IT INTO TWO PARTS FLX(I) AND FLX(J).  PERFORM A SIMILARBCPR 360
C ------- OPERATION TO OBTAIN DARCY FLUXES DCYFLX(I) AND DCYFLX(J).     BCPR 365
C                                                                       BCPR 370
         FLX(I)=FLX(I)+RFNI/3.0D0+RFNJ/6.0D0                            BCPR 375
         FLX(J)=FLX(J)+RFNI/6.0D0+RFNJ/3.0D0                            BCPR 380
C                                                                       BCPR 385
C ------- COMPUTE THE FLUXES THROUGH POINTS NI AND NJ USING THE         BCPR 390
C ------- WHOLE BOUNDARY LENGTH.                                        BCPR 395
C                                                                       BCPR 400
         FNI=(VX(NI)*DCOSX(MP)+VZ(NI)*DCOSZ(MP))*DL(MP)*RR              BCPR 405
         FNJ=(VX(NJ)*DCOSX(MP)+VZ(NJ)*DCOSZ(MP))*DL(MP)*RR              BCPR 410
C                                                                       BCPR 415
C ------- DISTRIBUTE THE ABOVE FLUXES TO TWO END POINTS OF THE SIDE     BCPR 420
C                                                                       BCPR 425
         DCYFLX(I)=DCYFLX(I)+FNI/3.0D0+FNJ/6.0D0                        BCPR 430
         DCYFLX(J)=DCYFLX(J)+FNJ/3.0D0+FNI/6.0D0                        BCPR 435
C                                                                       BCPR 440
  290    CONTINUE                                                       BCPR 445
C                                                                       BCPR 450
C ------- CHANGE TO FLUX OR HEAD CONDITIONS, AS NECESSARY, AND SO       BCPR 455
C ------- INDICATE IN THE ARRAYS NPFLX(NPP) AND NPCON(NPP).             BCPR 460
C                                                                       BCPR 465
      IF (NCHG.NE.(-1)) GO TO 300                                       BCPR 470
      NCHG=0                                                            BCPR 475
      RETURN                                                            BCPR 480
C                                                                       BCPR 485
  300 NCHG=0                                                            BCPR 490
C                                                                       BCPR 495
      DO 390 NPP=1,NRSN                                                 BCPR 500
```

```
C
      DCYNNP=DCYFLX(NPP)
      FLXNNP=FLX(NPP)
      HCONNP=HCON(NPP)
      HMINNP=HMIN(NPP)
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
  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                                             BCPR 755
      IF(HMINNP.LE.H(NP)) GO TO 390                                     BCPR 760
      NPMIN(NPP)=NPFLX(NPP)                                             BCPR 765
      NPFLX(NPP)=0                                                      BCPR 770
      NCHG=NCHG+1                                                       BCPR 775
      GO TO 390                                                         BCPR 780
C                                                                       BCPR 785
C ------- CHECK IF THE CHANGE FROM PONDING CONDITION TO EVAPORATION-FLUXBCPR 790
C ------- CONDITION IS NECESSARY?                                       BCPR 795
C                                                                       BCPR 800
  360 NP=NPCON(NPP)                                                     BCPR 805
      IF(NP.EQ.0) GO TO 370                                             BCPR 810
      IF(DCYNNP.GE.0.0) GO TO 390                                       BCPR 815
      NPFLX(NPP)=NPCON(NPP)                                             BCPR 820
      NPCON(NPP)=0                                                      BCPR 825
      NCHG=NCHG+1                                                       BCPR 830
      GO TO 390                                                         BCPR 835
C                                                                       BCPR 840
C ------- CHECK IF THE CHANGE FROM MINIMUM PRESSURE HEAD CONDITION TO   BCPR 845
C ------- EVAPORATION-FLUX CONDITION IS NECESSARY?                      BCPR 850
C                                                                       BCPR 855
  370 NP=NPMIN(NPP)                                                     BCPR 860
      IF(NP.EQ.0) GO TO 390                                             BCPR 865
C     IF(DCYNNP.LT.0.0) GO TO 380                                       BCPR 870
      IF(DCYNNP.LT.FLXNNP) GO TO 390                                    BCPR 875
      NPFLX(NPP)=NPMIN(NPP)                                             BCPR 880
      NPMIN(NPP)=0                                                      BCPR 885
      NCHG=NCHG+1                                                       BCPR 890
C     GO TO 390                                                         BCPR 895
C                                                                       BCPR 900
C 380 PRINT 1000, NPP                                                   BCPR 905
C     STOP                                                              BCPR 910
C                                                                       BCPR 915
  390    CONTINUE                                                       BCPR 920
C                                                                       BCPR 925
C1000 FORMAT('1 **** WARNING DURING NON-RAINFALL PERIOD',I3,            BCPR 930
C    1 '-TH RS NODE USING MINIMUM HEAD CONDITION RESULTS IN DARCY ',    BCPR 935
C    2 'FLUX INTO THE REGION'/1X,'THIS SHOULD NOT HAPPEN.  HENCE ',     BCPR 940
C    3 'EITHER MINIMUM HEAD IS NOT SMALL ENOUGH OR THE INTERIOR ',      BCPR 945
C    4 'NODE PRESSURE IS TO SMALL: STOP')                               BCPR 950
C                                                                       BCPR 955
      RETURN                                                            BCPR 960
      END                                                               BCPR 965
```

```
      SUBROUTINE ASEMBL(X,Z,IE, C,R,HW,HP,TH,DTH,AKX,AKZ,AKXZ,PROP,
     > MAXNP,MAXEL,MAXHBP, MAXMAT,MXMPPM, KSS,W,DELT, SOS,LES,ISTYP,
     3 WSS,NPW,IWTYP,LLNODE)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /CPIS/ JBAND
      COMMON /CSOS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP
      COMMON /CWSS/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,IBAND,KGRAV,NTI
      COMMON /COORD/ CSFE,SNFE,CYLIND
      COMMON /OPT/ ILUMP,IMID,IPNTS,IVML
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)
      DIMENSION C(MAXNP,MAXHBP),R(MAXNP), HP(MAXNP),HW(MAXNP),
     > TH(4,MAXEL),DTH(4,MAXEL),AKX(4,MAXEL),AKZ(4,MAXEL),AKXZ(4,MAXEL)
      DIMENSION LLNODE(JBAND,MAXNP)
      DIMENSION PROP(MXMPPM,MAXMAT)
      DIMENSION SOS(MXSPR), LES(MXSEL),ISTYP(MXSEL)
      DIMENSION WSS(MXWPR), NPW(MXWNP),IWTYP(MXWNP)
C
      DIMENSION QA(4,4),QB(4,4),RQ(4),THG(4),DTHG(4),AKXG(4),AKZG(4),
     > AKXZG(4),XQ(4),ZQ(4),IEM(4)
C
      DATA TWOPI/6.283185307D0/
C
      AGRAV=DFLOAT(KGRAV)
      IHALFB=(IBAND-1)/2
      IHBP=IHALFB+1
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 IF(IPNTS.EQ.1) GO TO 140
      DO 110 NP=1,NNP
      R(NP)=0.0
      DO 110 IB=1,IHBP
      C(NP,IB)=0.0
  110 CONTINUE
      GO TO 190
  140 DO 145 NP=1,NNP
      R(NP)=0.0
      DO 145 I=1,JBAND
```

```
      C(NP,I)=0.0                                                       ASEM 255
  145 CONTINUE                                                          ASEM 260
  190 CONTINUE                                                          ASEM 265
C                                                                       ASEM 270
C ------- START TO ASSEMBLE OVER ALL ELEMENTS                           ASEM 275
C                                                                       ASEM 280
      DO 690 M=1,NEL                                                    ASEM 285
C                                                                       ASEM 290
      NQ=4                                                              ASEM 295
      IEM4=IE(M,4)                                                      ASEM 300
      IF(IEM4.EQ.0) NQ=3                                                ASEM 305
C                                                                       ASEM 310
C ------- COMPUTE MATRICES QA(IQ,JQ), QB(IQ,JQ), AND RQ(IQ) FOR EACH M  ASEM 315
C                                                                       ASEM 320
      MTYP=IE(M,5)                                                      ASEM 325
      ALP=PROP(1,MTYP)                                                  ASEM 330
      BETAP=PROP(2,MTYP)                                                ASEM 335
      POR=PROP(3,MTYP)                                                  ASEM 340
C                                                                       ASEM 345
      DO 210 IQ=1,NQ                                                    ASEM 350
      NP=IE(M,IQ)                                                       ASEM 355
      IEM(IQ)=NP                                                        ASEM 360
      XQ(IQ)=X(NP)                                                      ASEM 365
      ZQ(IQ)=Z(NP)                                                      ASEM 370
  210 CONTINUE                                                          ASEM 375
C                                                                       ASEM 380
      DO 220 KG=1,NQ                                                    ASEM 385
      THG(KG)=TH(KG,M)                                                  ASEM 390
      DTHG(KG)=DTH(KG,M)                                                ASEM 395
      AKXG(KG)=AKX(KG,M)                                                ASEM 400
      AKZG(KG)=AKZ(KG,M)                                                ASEM 405
      AKXZG(KG)=AKXZ(KG,M)                                              ASEM 410
  220 CONTINUE                                                          ASEM 415
C                                                                       ASEM 420
      WSOSM=0.0                                                         ASEM 425
      IF(NSEL.LE.0) GO TO 250                                           ASEM 430
      DO 230 I=1,NSEL                                                   ASEM 435
      MP=LES(I)                                                         ASEM 440
      IF(MP.EQ.M) GO TO 240                                             ASEM 445
  230 CONTINUE                                                          ASEM 450
      GO TO 250                                                         ASEM 455
  240 ITYP=ISTYP(I)                                                     ASEM 460
      WSOSM=SOS(ITYP)                                                   ASEM 465
  250 CONTINUE                                                          ASEM 470
C                                                                       ASEM 475
      IF(NQ.EQ.3) CALL Q3(QA,QB,RQ,THG,DTHG,AKXG,AKZG,AKXZG,XQ,ZQ,WSOSM,ASEM 480
     >            ALP,BETAP,POR,AGRAV)                                  ASEM 485
      IF(NQ.EQ.4) CALL Q4(QA,QB,RQ,THG,DTHG,AKXG,AKZG,AKXZG,XQ,ZQ,WSOSM,ASEM 490
     >            ALP,BETAP,POR,AGRAV)                                  ASEM 495
```

```
C                                                                       ASEM 500
C ------- ASSEMBLE QA(IQ,JQ) AND QB(IQ,JQ) INTO THE GLOBAL MATRIX       ASEM 505
C ------- C(NP,IB) = B + A/DELT AND FORM THE GLOBAL LOAD VECTOR R(NP).  ASEM 510
C ------- SINCE C IS SYMMETRIC, ONLY THE UPPER HALF BAND IS STORED.     ASEM 515
C                                                                       ASEM 520
      IF(IMID.EQ.1) GO TO 400                                           ASEM 525
C                                                                       ASEM 530
C ------- FOR THE CASES OF NON MID-DIFFERENCES                          ASEM 535
C                                                                       ASEM 540
      DO 390 IQ=1,NQ                                                    ASEM 545
      NI=IEM(IQ)                                                        ASEM 550
      R(NI)=R(NI)-RQ(IQ)                                                ASEM 555
      DO 340 JQ=1,NQ                                                    ASEM 560
      NJ=IEM(JQ)                                                        ASEM 565
      QA(IQ,JQ)=QA(IQ,JQ)*DELTI                                         ASEM 570
      R(NI)=R(NI)+(QA(IQ,JQ)-W2*QB(IQ,JQ))*HP(NJ)                       ASEM 575
C                                                                       ASEM 580
      IF(IPNTS.EQ.1) GO TO 320                                          ASEM 585
C                                                                       ASEM 590
      IF(NJ.LT.NI) GO TO 340                                            ASEM 595
      IB=NJ-NI+1                                                        ASEM 600
      C(NI,IB)=C(NI,IB)+QA(IQ,JQ)+W1*QB(IQ,JQ)                          ASEM 605
      GO TO 340                                                         ASEM 610
C                                                                       ASEM 615
  320 DO 325 I=1,JBAND                                                  ASEM 620
      LNODE=LLNODE(I,NI)                                                ASEM 625
      IF(NJ.EQ.LNODE) GO TO 330                                         ASEM 630
  325 CONTINUE                                                          ASEM 635
C                                                                       ASEM 640
      WRITE(6,1000) NI,M,JQ                                             ASEM 645
      STOP                                                              ASEM 650
C                                                                       ASEM 655
  330 C(NI,I)=C(NI,I)+QA(IQ,JQ)+W1*QB(IQ,JQ)                            ASEM 660
C                                                                       ASEM 665
  340 CONTINUE                                                          ASEM 670
  390 CONTINUE                                                          ASEM 675
      GO TO 690                                                         ASEM 680
C                                                                       ASEM 685
C ------- FOR THE CASES OF MID-DIFFERENCES                              ASEM 690
C                                                                       ASEM 695
  400 DO 490 IQ=1,NQ                                                    ASEM 700
      NI=IEM(IQ)                                                        ASEM 705
      R(NI)=R(NI)-RQ(IQ)                                                ASEM 710
      DO 440 JQ=1,NQ                                                    ASEM 715
      NJ=IEM(JQ)                                                        ASEM 720
      QA(IQ,JQ)=2.0D0*QA(IQ,JQ)*DELTI                                   ASEM 725
      R(NI)=R(NI)+QA(IQ,JQ)*HP(NJ)                                      ASEM 730
C                                                                       ASEM 735
      IF(IPNTS.EQ.1) GO TO 420                                          ASEM 740
C                                                                       ASEM 745
```

```
      IF(NJ.LT.NI) GO TO 440                                            ASEM 750
      IB=NJ-NI+1                                                        ASEM 755
      C(NI,IB)=C(NI,IB)+QA(IQ,JQ)+QB(IQ,JQ)                             ASEM 760
      GO TO 440                                                         ASEM 765
C                                                                       ASEM 770
  420 DO 425 I=1,JBAND                                                  ASEM 775
      LNODE=LLNODE(I,NI)                                                ASEM 780
      IF(NJ.EQ.LNODE) GO TO 430                                         ASEM 785
  425 CONTINUE                                                          ASEM 790
C                                                                       ASEM 795
      WRITE(6,1000) NI,M,JQ                                             ASEM 800
      STOP                                                              ASEM 805
C                                                                       ASEM 810
  430 C(NI,I)=C(NI,I)+QA(IQ,JQ)+QB(IQ,JQ)                               ASEM 815
C                                                                       ASEM 820
  440 CONTINUE                                                          ASEM 825
  490 CONTINUE                                                          ASEM 830
C                                                                       ASEM 835
  690 CONTINUE                                                          ASEM 840
C                                                                       ASEM 845
C ------- INCORPORATE WELL SOURCE/SINK                                  ASEM 850
C                                                                       ASEM 855
      IF(NWNP.EQ.0) GO TO 910                                           ASEM 860
      DO 790 I=1,NWNP                                                   ASEM 865
      NI=NPW(I)                                                         ASEM 870
      ITYP=IWTYP(I)                                                     ASEM 875
      R(NI)=R(NI)+WSS(ITYP)/TWOPI**CYLIND                               ASEM 880
  790 CONTINUE                                                          ASEM 885
C                                                                       ASEM 890
  910 CONTINUE                                                          ASEM 895
C                                                                       ASEM 900
 1000 FORMAT(1H1/5X,'*** WARNING: NONE OF THE LOWER-LEFT NODE IN EQUATIOASEM 905
     1N',I3,/5X,'***  IS CORRESPONDING TO ',I5,'-TH ELEMENT-S',I2,     ASEM 910
     2'-TH NODE; STOP  ****')                                           ASEM 915
C                                                                       ASEM 920
      RETURN                                                            ASEM 925
      END                                                               ASEM 930


      SUBROUTINE Q4(QA,QB,RQ,THG,DTHG,AKXG,AKZG,AKXZG,XQ,ZQ,WSOSM,      Q4   005
     > ALP,BETAP,POR,AGRAV)                                             Q4   010
C                                                                       Q4   015
      IMPLICIT REAL*8 (A-H,O-Z)                                         Q4   020
      REAL*8 N(4)                                                       Q4   025
C                                                                       Q4   030
      COMMON /COORD/ CSFE,SNFE,CYLIND                                   Q4   035
      COMMON /OPT/ ILUMP,IMID,IPNTS,IVML                                Q4   040
```

```
C
      DIMENSION QA(4,4),QB(4,4),RQ(4),THG(4),DTHG(4),AKXG(4),AKZG(4),
     > AKXZG(4),XQ(4),ZQ(4)
      DIMENSION S(4),T(4),DNX(4),DNZ(4)
      DIMENSION PJAB(2,2),DNSS(4),DNTT(4)
C
      DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-
     > 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /
C
C ------- INITIATE MATRICES QA, QB, AND RQ
C
      DO 110 IQ=1,4
      RQ(IQ)=0.0
      DO 110 JQ=1,4
      QA(IQ,JQ)=0.0
      QB(IQ,JQ)=0.0
  110 CONTINUE
C
      DO 490 KG=1,4
C
C ------- DETERMINE LOACAL COORDINATE (SS,TT) OF GAUSSIAN POINT KG
C
      SS=P*S(KG)
      TT=P*T(KG)
C
C ------- CALCULATE VALUES OF BASIS FUNCTIONS N(IQ) AND THEIR
C ------- DERIVATIVES DNX(IQ) AND DNZ(IQ) W.R.T. X AND Z, RESPECTIVELY
C ------- AT THE GAUSSIAN POINT KG.
C
         CALL BASE(N,DNSS,DNTT,SS,TT)
C
      DO 210 I=1,2
      DO 210 J=1,2
  210 PJAB(I,J)=0.0
C
      DO 220 I=1,4
      PJAB(1,1)=PJAB(1,1)+ZQ(I)*DNTT(I)
      PJAB(1,2)=PJAB(1,2)-ZQ(I)*DNSS(I)
      PJAB(2,1)=PJAB(2,1)-XQ(I)*DNTT(I)
      PJAB(2,2)=PJAB(2,2)+XQ(I)*DNSS(I)
  220 CONTINUE
C
      DJAC=PJAB(2,2)*PJAB(1,1)-PJAB(1,2)*PJAB(2,1)
      DJACI=1.0D0/DJAC
C
      DO 230 I=1,2
      DO 230 J=1,2
      PJAB(I,J)=PJAB(I,J)*DJACI
  230 CONTINUE
C
```

```
      DO 260 I=1,4                                                      Q4  295
      DNX(I)=DNSS(I)*PJAB(1,1)+DNTT(I)*PJAB(1,2)                        Q4  300
      DNZ(I)=DNSS(I)*PJAB(2,1)+DNTT(I)*PJAB(2,2)                        Q4  305
  260 CONTINUE                                                          Q4  310
C                                                                       Q4  315
      R=0.0                                                             Q4  320
      DO 270 I=1,4                                                      Q4  325
      R=R+XQ(I)*N(I)                                                    Q4  330
  270 CONTINUE                                                          Q4  335
      RR=1.0D0                                                          Q4  340
      IF(CYLIND.NE.0.0) RR=R                                            Q4  345
C                                                                       Q4  350
      AKXOP=AKXG(KG)                                                    Q4  355
      AKXZQP=AKXZG(KG)                                                  Q4  360
      AKZQP=AKZG(KG)                                                    Q4  365
      THQP=THG(KG)                                                      Q4  370
      DTHQP=DTHG(KG)                                                    Q4  375
C                                                                       Q4  380
C ------- ACCUMULATE THE SUMS TO OBTAIN THE MATRIX INTEGRALS QQ(IQ,JQ)  Q4  385
C ------- AND RQ(IQ)                                                    Q4  390
C                                                                       Q4  395
      FHP=ALP*THQP/POR+BETAP*THQP+DTHQP                                 Q4  400
      FHP=FHP*DJAC*RR                                                   Q4  405
      AKXQP=AKXQP*DJAC*RR                                               Q4  410
      AKXZQP=AKXZQP*DJAC*RR                                             Q4  415
      AKZQP=AKZQP*DJAC*RR                                               Q4  420
      WSOSK=WSOSM*DJAC*RR                                               Q4  425
C                                                                       Q4  430
      DO 390 IQ=1,4                                                     Q4  435
      RQ(IQ)=RQ(IQ)+AGRAV*(DNX(IQ)*(AKXQP*SNFE+AKXZQP*CSFE) +           Q4  440
     1        DNZ(IQ)*(AKXZQP*SNFE+AKZQP*CSFE)) - WSOSK*N(IQ)           Q4  445
      DO 390 JQ=1,4                                                     Q4  450
      QA(IQ,JQ)=QA(IQ,JQ) + FHP*N(IQ)*N(JQ)                             Q4  455
      QB(IQ,JQ)=QB(IQ,JQ) + DNX(IQ)*(AKXQP*DNX(JQ)+AKXZQP*DNZ(JQ)) +   Q4  460
     1          DNZ(IQ)*(AKXZQP*DNX(JQ)+AKZQP*DNZ(JQ))                  Q4  465
  390 CONTINUE                                                          Q4  470
C                                                                       Q4  475
  490 CONTINUE                                                          Q4  480
C                                                                       Q4  485
      IF(ILUMP.NE.0) GO TO 600                                          Q4  490
C                                                                       Q4  495
C ------- IF MASS MATRIX IS NOT TO BE LUMPED , RETURN                   Q4  500
C                                                                       Q4  505
      RETURN                                                            Q4  510
C                                                                       Q4  515
C ------- FOR THE CASES WHEN THE MASS MATRIX IS TO BE LUMPED            Q4  520
C                                                                       Q4  525
  600 CONTINUE                                                          Q4  530
      DO 690 I=1,4                                                      Q4  535
      SUM=0.0                                                           Q4  540
```

```
      DO 640 J=1,4                                                      Q4   545
      SUM=SUM+QA(I,J)                                                   Q4   550
  640 QA(I,J)=0.0                                                       Q4   555
C                                                                       Q4   560
      QA(I,I)=SUM                                                       Q4   565
  690 CONTINUE                                                          Q4   570
C                                                                       Q4   575
      RETURN                                                            Q4   580
      END                                                               Q4   585
```

```
      SUBROUTINE BASE(N,DNSS,DNTT,SS,TT)                                BASE 005
C                                                                       BASE 010
      IMPLICIT REAL*8 (A-H,O-Z)                                         BASE 015
      REAL*8 N(4)                                                       BASE 020
C                                                                       BASE 025
      DIMENSION DNSS(4),DNTT(4)                                         BASE 030
C                                                                       BASE 035
      SM=1.0D0-SS                                                       BASE 040
      SP=1.0D0+SS                                                       BASE 045
      TM=1.0D0-TT                                                       BASE 050
      TP=1.0D0+TT                                                       BASE 055
      N(1)=0.25D0*SM*TM                                                 BASE 060
      N(2)=0.25D0*SP*TM                                                 BASE 065
      N(3)=0.25D0*SP*TP                                                 BASE 070
      N(4)=0.25D0*SM*TP                                                 BASE 075
      DNSS(1)=-0.25D0*TM                                                BASE 080
      DNSS(2)=0.25D0*TM                                                 BASE 085
      DNSS(3)=0.25D0*TP                                                 BASE 090
      DNSS(4)=-0.25D0*TP                                                BASE 095
      DNTT(1)=-0.25D0*SM                                                BASE 100
      DNTT(2)=-0.25D0*SP                                                BASE 105
      DNTT(3)=0.25D0*SP                                                 BASE 110
      DNTT(4)=0.25D0*SM                                                 BASE 115
      RETURN                                                            BASE 120
      END                                                               BASE 125
```

```
      SUBROUTINE Q3(QA,QB,RQ,THG,DTHG,AKXG,AKZG,AKXZG,XQ,ZQ,WSOSM,
     1 ALP,BETAP,POR,AGRAV)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /COORD/ CSFE,SNFE,CYLIND
      COMMON /OPT/ ILUMP,IMID,IPNTS,IVML
C
      DIMENSION QA(4,4),QB(4,4),RQ(4),AKXG(4),AKZG(4),AKXZG(4),
     > XQ(4),ZQ(4),THG(4),DTHG(4)
      DIMENSION A(3),B(3),C(3)
C
C     DATA OSIX/0.1666666666666667D0/
C     DATA OTWLVE/0.8333333333333333D-1/
      DATA OTHRTY/0.3333333333333333D-1/
      DATA OSIXTY/0.1666666666666667D-1/
C
      DO 110 IQ=1,4
      RQ(IQ)=0.0
      DO 110 JQ=1,4
      QA(IQ,JQ)=0.0
      QB(IQ,JQ)=0.0
  110 CONTINUE
C
      X1=XQ(1)
      X2=XQ(2)
      X3=XQ(3)
      Z1=ZQ(1)
      Z2=ZQ(2)
      Z3=ZQ(3)
C
      A(1)=X2*Z3 - X3*Z2
      A(2)=X3*Z1 - X1*Z3
      A(3)=X1*Z2 - X2*Z1
C
      R=(X1+X2+X3)/3.0D0
      RR=1.0D0
      IF(CYLIND.NE.0.0) RR=R
C
      AREA2=DABS(A(1)+A(2)+A(3))
      AREA=AREA2/2.0D0*RR
C
      B(1)=(Z2-Z3)/AREA2
      B(2)=(Z3-Z1)/AREA2
      B(3)=(Z1-Z2)/AREA2
      C(1)=(X3-X2)/AREA2
      C(2)=(X1-X3)/AREA2
      C(3)=(X2-X1)/AREA2
C
      F1= ALP*THG(1)/POR+BETAP*THG(1) + DTHG(1)
```

```
      F2= ALP*THG(2)/POR+BETAP*THG(2) + DTHG(2)
      F3= ALP*THG(3)/POR+BETAP*THG(3) + DTHG(3)
C
      QA(1,1)=AREA*(F1*0.1D0 + F2*OTHRTY + F3*OTHRTY)
      QA(1,2)=AREA*(F1*OTHRTY + F2*OTHRTY + F3*OSIXTY)
      QA(1,3)=AREA*(F1*OTHRTY + F2*OSIXTY + F3*OTHRTY)
      QA(2,1)=AREA*(F1*OTHRTY + F2*OTHRTY + F3*OSIXTY)
      QA(2,2)=AREA*(F1*OTHRTY + F2*0.1D0 + F3*OTHRTY)
      QA(2,3)=AREA*(F1*OSIXTY + F2*OTHRTY + F3*OTHRTY)
      QA(3,1)=AREA*(F1*OTHRTY + F2*OSIXTY + F3*OTHRTY)
      QA(3,2)=AREA*(F1*OSIXTY + F2*OTHRTY + F3*OTHRTY)
      QA(3,3)=AREA*(F1*OTHRTY + F2*OTHRTY + F3*0.1D0)
C
      AREAD=AREA/3.0D0
      AKXQS=AKXG(1)+AKXG(2)+AKXG(3)
      AKZQS=AKZG(1)+AKZG(2)+AKZG(3)
      AKXZQS=AKXZG(1)+AKXZG(2)+AKXZG(3)
C
      DO 210 IQ=1,3
      DO 210 JQ=1,3
      QB(IQ,JQ)=AREAD*(B(IQ)*B(JQ)*AKXQS+C(IQ)*C(JQ)*AKZQS +
     1          (B(IQ)*C(JQ)+B(JQ)*C(IQ))*AKXZQS)
  210 CONTINUE
C
      BICOEF=((AKXG(1)+AKXG(2)+AKXG(3))*SNFE +
     1 (AKXZG(1)+AKXZG(2)+AKXZG(3))*CSFE)*AGRAV
      CICOEF=((AKXZG(1)+AKXZG(2)+AKXZG(3))*SNFE +
     1 (AKZG(1)+AKZG(2)+AKZG(3))*CSFE)*AGRAV
      DO 310 IQ=1,3
      RQ(IQ)=AREAD*(B(IQ)*BICOEF+C(IQ)*CICOEF-WSOSM)
  310 CONTINUE
C
      IF(ILUMP.EQ.0) RETURN
C
C ------- FOR THE CASES WHEN THE MASS MATRIX IS TO BE LUMPED
C
      DO 690 I=1,3
      SUM=0.0
      DO 640 J=1,3
      SUM=SUM+QA(I,J)
  640 QA(I,J)=0.0
      QA(I,I)=SUM
  690 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE BC(C,R, X,Z,IE,LRN,AKX,AKZ,AKXZ, FLX,HCON,NPCON,NPFLX,
     1       HDB,IDTYP,NPDB, QCB,ICTYP,NPCB,ISC, QNB,INTYP,NPNB,ISN,
     2       DLN,DCOSXN,DCOSZN,MNES, KSS, HMIN,NPMIN)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI
      COMMON /CPIS/ JBAND
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,IBAND,KGRAV,NTI
      COMMON /RSBC/ MXRSEL,MXRSNP,MXRFPR,MXRPAR,NRSEL,NRSN,NRFPR,NRFPAR
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP
      COMMON /CCBC/ MXCNP,MXCEL,MXCPR,MXCDP,NCNP,NCEL,NCPR,NCDP
      COMMON /CNBC/ MXNNP,MXNEL,MXNPR,MXNDP,NNNP,NNEL,NNPR,NNDP
      COMMON /COORD/ CSFE,SNFE,CYLIND
      COMMON /OPT/ ILUMP,IMID,IPNTS,IVML
CC
      DIMENSION C(MAXNP,MAXHBP),R(MAXNP)
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),LRN(JBAND,MAXNP)
      DIMENSION AKX(4,MAXEL),AKZ(4,MAXEL),AKXZ(4,MAXEL)
      DIMENSION FLX(MXRSNP),HCON(MXRSNP),NPCON(MXRSNP),NPFLX(MXRSNP)
      DIMENSION HMIN(MXRSNP),NPMIN(MXRSNP)
C
      DIMENSION HDB(MXDPR),IDTYP(MXDNP),NPDB(MXDNP)
      DIMENSION QCB(MXCPR),ICTYP(MXCNP),NPCB(MXCNP),ISC(2,MXCEL)
C
      DIMENSION QNB(MXNPR),INTYP(MXNNP),NPNB(MXNNP),ISN(4,MXNEL)
      DIMENSION DLN(MXNEL),DCOSXN(MXNEL),DCOSZN(MXNEL),MNES(MXNEL)
C
      DIMENSION RQ(2),XQ(4),ZQ(4),KQ(2)
      DIMENSION AKXG(4),AKZG(4),AKXZG(4)
C
      AGRAV=DFLOAT(KGRAV)
C
      IHALFB=(IBAND-1)/2
      IHBP=IHALFB+1
C
C ******* APPLY CAUCHY BOUNDARY CONDITIONS
C
  250 IF(NCNP.LE.0) GO TO 300
      DO 260 MP=1,NCEL
         I=ISC(1,MP)
         NI=NPCB(I)
         J=ISC(2,MP)
         NJ=NPCB(J)
C
      RR=((X(NI)+X(NJ))/2.0D0)
      IF(CYLIND.EQ.0.0) RR=1.0D0
C
         DX=X(NI)-X(NJ)
         DZ=Z(NI)-Z(NJ)
```

```
      EL=RR*DSQRT(DX*DX+DZ*DZ)                                          BC  255
      ITYPI=ICTYP(I)                                                    BC  260
      ITYPJ=ICTYP(J)                                                    BC  265
      R(NI)=R(NI)-EL*(QCB(ITYPI)/3.0D0+QCB(ITYPJ)/6.0D0)                BC  270
      R(NJ)=R(NJ)-EL*(QCB(ITYPI)/6.0D0+QCB(ITYPJ)/3.0D0)                BC  275
  260 CONTINUE                                                          BC  280
C                                                                       BC  285
C ******* APPLY VARIABLE (RAINFALL-SEEPAGE) BOUNDARY CONDITIONS         BC  290
C                                                                       BC  295
  300 IF(NRSN.EQ.0) GO TO 500                                           BC  300
C                                                                       BC  305
C -------- DIRICHLET PART OF VARIABLE BOUNDARY CONDITIONS               BC  310
C                                                                       BC  315
      IF(IPNTS.EQ.1) GO TO 400                                          BC  320
C                                                                       BC  325
      DO 390 NPP=1,NRSN                                                 BC  330
C                                                                       BC  335
C             MODIFY LOAD VECTOR FOR NON-ZERO HCON                      BC  340
C                                                                       BC  345
      NI=NPCON(NPP)                                                     BC  350
      IF(NI.NE.0) GO TO 305                                             BC  355
      NI=NPMIN(NPP)                                                     BC  360
      IF(NI.NE.0) GO TO 310                                             BC  365
      GO TO 390                                                         BC  370
  305 BB=HCON(NPP)                                                      BC  375
      GO TO 315                                                         BC  380
  310 BB=HMIN(NPP)                                                      BC  385
C                                                                       BC  390
  315 IF(BB.EQ.0.0) GO TO 340                                           BC  395
      DO 320 IB=1,IHALFB                                                BC  400
      NJ=NI-IB                                                          BC  405
      IF(NJ.LT.1) GO TO 325                                             BC  410
      JB=IB+1                                                           BC  415
  320 R(NJ)=R(NJ)-BB*C(NJ,JB)                                           BC  420
  325 DO 330 IB=1,IHALFB                                                BC  425
      NJ=NI+IB                                                          BC  430
      IF(NJ.GT.NNP) GO TO 340                                           BC  435
      JB=IB+1                                                           BC  440
  330 R(NJ)=R(NJ)-BB*C(NI,JB)                                           BC  445
  340 R(NI)=BB                                                          BC  450
C                                                                       BC  455
C             ZERO COLUMN NPCON(NPP)                                    BC  460
C                                                                       BC  465
      DO 350 IB=1,IHALFB                                                BC  470
      NJ=NI-IB                                                          BC  475
      IF(NJ.LT.1) GO TO 360                                             BC  480
      JB=IB+1                                                           BC  485
  350 C(NJ,JB)=0.0                                                      BC  490
C                                                                       BC  495
C             MODIFY ROW NPCON(NPP)                                     BC  500
```

```
C
  360 DO 370 KB=1,IHBP
  370 C(NI,KB)=0.0
      C(NI,1)=1.0D0
  390 CONTINUE
      GO TO 450
C
  400 DO 440 NPP=1,NRSN
      NI=NPCON(NPP)
      IF(NI.NE.0) GO TO 405
      NI=NPMIN(NPP)
      IF(NI.NE.0) GO TO 410
      GO TO 440
  405 BB=HCON(NPP)
      GO TO 415
  410 BB=HMIN(NPP)
C
  415 R(NI)=BB
      DO 430 I=1,JBAND
      C(NI,I)=0.0
      LNODE=LRN(I,NI)
      IF(LNODE.EQ.NI) C(NI,I)=1.0D0
  430 CONTINUE
  440 CONTINUE
C
C -------- CAUCHY PART OF VARIABLE BOUNDARY CONDITIONS
C
  450 DO 490 NPP=1,NRSN
         NP=NPFLX(NPP)
         IF (NP.EQ.0) GO TO 490
         R(NP)=R(NP)-FLX(NPP)
  490 CONTINUE
C
C ******* APPLY NEUMANN BOUNDARY CONDITIONS
C
  500 IF(NNNP.EQ.0) GO TO 600
      DO 530 MP=1,NNEL
         I=ISN(1,MP)
         NI=NPNB(I)
         J=ISN(2,MP)
         NJ=NPNB(J)
C
C -------- MODIFY LOAD VECTOR DUE TO NEUMANN FLUX
C
      RR=((X(NI)+X(NJ))/2.0D0)
      IF(CYLIND.EQ.0.0) RR=1.0D0
C
         EL=RR*DLN(MP)
         ITYPI=INTYP(I)
         ITYPJ=INTYP(J)
```

```
         R(NI)=R(NI)-EL*(QNB(ITYPI)/3.0D0+QNB(ITYPJ)/6.0D0)
         R(NJ)=R(NJ)-EL*(QNB(ITYPI)/6.0D0+QNB(ITYPJ)/3.0D0)
C
C -------- MODIFY GRAVITY FLUX DUE TO GRAVITY TERM
C
         M=MNES(MP)
      NQ=4
      IEM4=IE(M,4)
      IF(IEM4.EQ.0) NQ=3
C
         KQ(1)=ISN(3,MP)
         KQ(2)=ISN(4,MP)
C
         DO 520 IQ=1,NQ
            NP=IE(M,IQ)
            XQ(IQ)=X(NP)
            ZQ(IQ)=Z(NP)
  520    CONTINUE
C
      DO 525 KG=1,NQ
      AKXG(KG)=AKX(KG,M)
      AKZG(KG)=AKZ(KG,M)
      AKXZG(KG)=AKXZ(KG,M)
  525 CONTINUE
C
      CALL Q2GF(RQ,KQ,EL,DCOSXN(MP),DCOSZN(MP),XQ,ZQ,
     1 AKXG,AKZG,AKXZG,CSFE,SNFE,AGRAV)
C
         R(NI)=R(NI)+RQ(1)
         R(NJ)=R(NJ)+RQ(2)
  530 CONTINUE
  600 IF(NDNP.LE.0) GO TO 900
C
C ******* APPLY DIRICHLET BOUNDARY CONDITIONS
C
      IF(IPNTS.EQ.1) GO TO 200
      DO 190 NPP=1,NDNP
C
C -------- MODIFY LOAC VECTOR FOR NON-ZERO DIRICHLET BOUNDARY VALUES
C
      NI=NPDB(NPP)
      ITYP=IDTYP(NPP)
      BB=HDB(ITYP)-(Z(NI)*CSFE+X(NI)*SNFE)*AGRAV
C
      IF(BB.EQ.0.0) GO TO 140
      DO 110 IB=1,IHALFB
      NJ=NI-IB
      IF(NJ.LT.1) GO TO 120
      JB=IB+1
      R(NJ)=R(NJ) - BB*C(NJ,JB)
```

```
  110 CONTINUE                                                          BC  1005
  120 DO 130 IB=1,IHALFB                                                BC  1010
      NJ=NI+IB                                                          BC  1015
      IF(NJ.GT.NNP) GO TO 140                                           BC  1020
      JB=IB+1                                                           BC  1025
      R(NJ)=R(NJ)-BB*C(NI,JB)                                           BC  1030
  130 CONTINUE                                                          BC  1035
C                                                                       BC  1040
  140 R(NI)=BB                                                          BC  1045
C                                                                       BC  1050
C -------- ZERO COLUMN NPDB(NPP)                                        BC  1055
C                                                                       BC  1060
      DO 150 IB=1,IHALFB                                                BC  1065
      NJ=NI-IB                                                          BC  1070
      IF(NJ.LT.1) GO TO 160                                             BC  1075
      JB=IB+1                                                           BC  1080
      C(NJ,JB)=0.0                                                      BC  1085
  150 CONTINUE                                                          BC  1090
C                                                                       BC  1095
C -------- MODIFY ROW NPDB(NPP)                                         BC  1100
C                                                                       BC  1105
  160 DO 170 KB=1,IHBP                                                  BC  1110
      C(NI,KB)=0.0                                                      BC  1115
  170 CONTINUE                                                          BC  1120
      C(NI,1)=1.0D0                                                     BC  1125
  190 CONTINUE                                                          BC  1130
      GO TO 900                                                         BC  1135
C                                                                       BC  1140
  200 DO 220 NPP=1,NDNP                                                 BC  1145
      NI=NPDB(NPP)                                                      BC  1150
      ITYP=IDTYP(NPP)                                                   BC  1155
      BB=HDB(ITYP)-(Z(NI)*CSFE+X(NI)*SNFE)*AGRAV                        BC  1160
      R(NI)=BB                                                          BC  1165
      DO 210 I=1,JBAND                                                  BC  1170
      C(NI,I)=0.0                                                       BC  1175
      LNODE=LPN(I,NI)                                                   BC  1180
      IF(LNODE.EQ.NI) C(NI,I)=1.0D0                                     BC  1185
  210 CONTINUE                                                          BC  1190
  220 CONTINUE                                                          BC  1195
C                                                                       BC  1200
  900 CONTINUE                                                          BC  1205
C                                                                       BC  1210
      RETURN                                                            BC  1215
      END                                                               BC  1220
```

```
      SUBROUTINE Q2GF(RQ,KQ,DL,DCOSX,DCOSZ,XQ,ZQ,AKXG,AKZG,AKXZG,
     1 CSFE,SNFE,AGRAV)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION RQ(2),XQ(4),ZQ(4),KQ(2)
      DIMENSION AKXG(4),AKZG(4),AKXZG(4)
C
      DATA OTHRD/0.3333333333333333D0/
      DATA OSIX/0.1666666666666667D0/
C
C ------- INITIALIZE NODAL COMPONENTS OF LINE INTEGRAL.
C
      DO 110 JQ=1,2
  110 RQ(JQ)=0.0
C
      I=KQ(1)
      J=KQ(2)
C
      VN1=(DCOSX*(CSFE*AKXZG(I) + SNFE*AKXG(I)) + DCOSZ*(CSFE*AKZG(I)+
     1 SNFE*AKXZG(I)))*AGRAV
      VN2=(DCOSX*(CSFE*AKXZG(J) + SNFE*AKXG(J)) + DCOSZ*(CSFE*AKZG(J)+
     1 SNFE*AKXZG(J)))*AGRAV
C
C --------- CONTRIBUTION FROM VN1
C
      RQ(1)=RQ(1) + DL*VN1*OTHRD
      RQ(2)=RQ(2) + DL*VN1*OSIX
C
C -------- CONTRIBUTION FROM VN2
C
      RQ(1)=RQ(1) + DL*VN2*OSIX
      RQ(2)=RQ(2) + DL*VN2*OTHRD
C
      RETURN
      END
```

```
      SUBROUTINE BANSOL(KKK,C,R,NNP,IHBP,MAXNP,MAXHBP)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION C(MAXNP,MAXHBP),R(MAXNP)
      IHALFB=IHBP-1
      NNP1=NNP-1
C
C  IF KKK = 1, THEN TRIANGULARIZE THE BAND MATRIX C(NP,IB), BUT
```

```
C  IF KKK = 2, THEN SIMPLY SOLVE WITH THE NEW RIGHT-HAND SIDE R(NP)      BANS 045
C                                                                        BANS 050
      IF (KKK.EQ.2) GO TO 50                                             BANS 055
C                                                                        BANS 060
C  TRIANGULARIZE MATRIX C                                                BANS 065
C                                                                        BANS 070
      NU=NNP-IHALFB                                                      BANS 075
      DO 20 NI=1,NU                                                      BANS 080
         NJ=NI-1                                                         BANS 085
         PIVOTI=1.0D0/C(NI,1)                                            BANS 090
         DO 20 LB=2,IHBP                                                 BANS 095
            A=C(NI,LB)*PIVOTI                                            BANS 100
            NK=NJ+LB                                                     BANS 105
            JB=0                                                         BANS 110
            DO 10 KB=LB,IHBP                                             BANS 115
               JB=JB+1                                                   BANS 120
   10          C(NK,JB)=C(NK,JB)-A*C(NI,KB)                              BANS 125
   20       C(NI,LB)=A                                                   BANS 130
      NL=NU+1                                                            BANS 135
      DO 40 NI=NL,NNP1                                                   BANS 140
         NJ=NI-1                                                         BANS 145
         MB=NNP-NJ                                                       BANS 150
         PIVOTI=1.0D0/C(NI,1)                                            BANS 155
         DO 40 LB=2,MB                                                   BANS 160
            A=C(NI,LB)*PIVOTI                                            BANS 165
            NK=NJ+LB                                                     BANS 170
            JB=0                                                         BANS 175
            DO 30 KB=LB,MB                                               BANS 180
               JB=JB+1                                                   BANS 185
   30          C(NK,JB)=C(NK,JB)-A*C(NI,KB)                              BANS 190
   40       C(NI,LB)=A                                                   BANS 195
      RETURN                                                             BANS 200
C                                                                        BANS 205
C  MODIFY LOAD VECTOR R                                                  BANS 210
C                                                                        BANS 215
   50 NU=NNP-IHALFB                                                      BANS 220
      DO 60 NI=1,NU                                                      BANS 225
         NJ=NI-1                                                         BANS 230
         A=R(NI)                                                         BANS 235
         R(NI)=A/C(NI,1)                                                 BANS 240
         DO 60 LB=2,IHBP                                                 BANS 245
            NK=NJ+LB                                                     BANS 250
   60       R(NK)=R(NK)-C(NI,LB)*A                                       BANS 255
      NL=NU+1                                                            BANS 260
      DO 70 NI=NL,NNP1                                                   BANS 265
         NJ=NI-1                                                         BANS 270
         MB=NNP-NJ                                                       BANS 275
         A=R(NI)                                                         BANS 280
         R(NI)=A/C(NI,1)                                                 BANS 285
         DO 70 LB=2,MB                                                   BANS 290
```

92/1245806695

```
            NK=NJ+LB                                                    BANS 295
   70       R(NK)=R(NK)-C(NI,LB)*A                                      BANS 300
C                                                                       BANS 305
C  BACK-SOLVE                                                           BANS 310
C                                                                       BANS 315
      R(NNP)=R(NNP)/C(NNP,1)                                            BANS 320
      DO 80 IB=1,IHALFB                                                 BANS 325
         NI=NNP-IB                                                      BANS 330
         NJ=NI-1                                                        BANS 335
         MB=IB+1                                                        BANS 340
         DO 80 KB=2,MB                                                  BANS 345
            NK=NJ+KB                                                    BANS 350
   80       R(NI)=R(NI)-C(NI,KB)*R(NK)                                  BANS 355
      DO 90 IB=IHBP,NNP1                                                BANS 360
         NI=NNP-IB                                                      BANS 365
         NJ=NI-1                                                        BANS 370
         DO 90 KB=2,IHBP                                                BANS 375
            NK=NJ+KB                                                    BANS 380
   90       R(NI)=R(NI)-C(NI,KB)*R(NK)                                  BANS 385
      RETURN                                                            BANS 390
      END                                                               BANS 395
```

```
      SUBROUTINE PISS(R,RL,RI,C,OMI,EPS,LRN,NNP,NITER,MAXNP,MAXBW,IBUG, PISS 005
     1 KTIM)                                                            PISS 010
C                                                                       PISS 015
      IMPLICIT REAL*8(A-H,O-Z)                                          PISS 020
C                                                                       PISS 025
      COMMON /CPIS/ JBAND                                               PISS 030
C                                                                       PISS 035
      DIMENSION R(MAXNP),C(MAXNP,MAXBW)                                 PISS 040
      DIMENSION RL(MAXNP),RI(MAXNP),LRN(JBAND,MAXNP)                    PISS 045
C                                                                       PISS 050
C ------- STORE THE LOAD VECTOR IN TEMPORARY STORAGE RL(MAXNP).         PISS 055
C                                                                       PISS 060
      DO 190 NP=1,NNP                                                   PISS 065
  190 RL(NP)=R(NP)                                                      PISS 070
C                                                                       PISS 075
C ******* NOW R(MAXNP) IS FREE AND CAN BE USED TO STORE THESOLUTION.    PISS 080
C                                                                       PISS 085
C ------- PRINT ITERATION INFORMATION IF DESIRED.                       PISS 090
C                                                                       PISS 095
      IF(IBUG.NE.0 .AND. KTIM.NE.0) PRINT 1000                          PISS 100
C                                                                       PISS 105
      DO 290 IT=1,NITER                                                 PISS 110
C                                                                       PISS 115
C ------- FOR EACH ITERATION, PUT THE LOAD VECTOR INTO R(MAXNP).        PISS 120
```

```
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
```

```
  270 DO 280 NP=1,NNP                                                   PISS 375
  280 RI(NP)=R(NP)                                                      PISS 380
C                                                                       PISS 385
C -------  PRINT ITERATION INFORMATION IF DESIRED.                     PISS 390
C                                                                       PISS 395
      IF(IBUG.NE.0 .AND. KTIM.NE.0) PRINT 1100, IT,ABSERR,RELERR,NNCVN  PISS 400
C                                                                       PISS 405
      IF(IT.EQ.1) GO TO 290                                             PISS 410
      IF(ABSERR .LE. EPS) GO TO 990                                     PISS 415
C                                                                       PISS 420
  290 CONTINUE                                                          PISS 425
C                                                                       PISS 430
      PRINT 2000, IT,NITER,ABSERR,RELERR,NNCVN                          PISS 435
C                                                                       PISS 440
  990 CONTINUE                                                          PISS 445
C                                                                       PISS 450
 1000 FORMAT(1H0/75X,5H   IT,12H   ABS ERR  ,12H   REL ERR  ,           PISS 455
     1 13H NO. OF NODES/75X,5H   --,12H   --- ---  ,12H   --- ---  ,    PISS 460
     2 13H --- -- -----)                                                PISS 465
 1100 FORMAT(75X,I5,2D12.4,5X,I5)                                       PISS 470
 2000 FORMAT(1H0/70X,43H ::: WARNING: NO CONVERGENCE IN PISS AFTER ,    PISS 475
     1 I4,11H ITERATIONS/75X,9H NPITER =,I4,11H  ABSERR =,D12.4/75X,    PISS 480
     2 11H   RELERR =,D12.4,11H    NNCVN =,I4)                          PISS 485
C                                                                       PISS 490
      RETURN                                                            PISS 495
      END                                                               PISS 500
```

```
      SUBROUTINE SFLOW(X,Z,IE,H,HP,VX,VZ,TH,DTH,                        SFLO 005
     1 DLB,DCOSXB,DCOSZB,BFLX,BFLXP,ISB,NBE,NPB, LES,SOS,ISTYP,         SFLO 010
     2 NPRS,NPDB,NPCB,NPNB,  PROP, DELT, KFLOW, NPW,WSS,IWTYP)          SFLO 015
C                                                                       SFLO 020
      IMPLICIT REAL*8(A-H,O-Z)                                          SFLO 025
C                                                                       SFLO 030
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI            SFLO 035
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,IBAND,KGRAV,NTI                   SFLO 040
C                                                                       SFLO 045
      COMMON /CSOS/ MXSEL,MXSPR,MXSDP,NSEL,NSPR,NSDP                    SFLO 050
      COMMON /CWSS/ MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP                    SFLO 055
C                                                                       SFLO 060
      COMMON /RSBC/ MXRSEL,MXRSNP,MXRFPR,MXRPAR,NRSEL,NRSN,NRFPR,NRFPAR SFLO 065
      COMMON /CDBC/ MXDNP,MXDPR,MXDDP,NDNP,NDPR,NDDP                    SFLO 070
      COMMON /CCBC/ MXCNP,MXCEL,MXCPR,MXCDP,NCNP,NCEL,NCPR,NCDP         SFLO 075
      COMMON /CNBC/ MXNNP,MXNEL,MXNPR,MXNDP,NNNP,NNEL,NNPR,NNDP         SFLO 080
C                                                                       SFLO 085
      COMMON /COORD/ CSFE,SNFE,CYLIND                                   SFLO 090
```

9 2 1 2 4 5 8 0 6 9 8

```
      COMMON /SMTL/ MAXMAT,MXSPPM,MXMPPM                                SFLO 095
C                                                                       SFLO 100
      COMMON /CFLOW/ FRATE(10),FLOW(10),TFLOW(10)                       SFLO 105
C                                                                       SFLO 110
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)                           SFLO 115
C                                                                       SFLO 120
      DIMENSION H(MAXNP),HP(MAXNP),VX(MAXNP),VZ(MAXNP)                  SFLO 125
      DIMENSION TH(4,MAXEL),DTH(4,MAXEL)                                SFLO 130
C                                                                       SFLO 135
      DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),BFLX(MAXBNP), SFLO 140
     > BFLXP(MAXBNP),NBE(MAXBEL),ISB(4,MAXBEL),NPB(MAXBNP)              SFLO 145
C                                                                       SFLO 150
      DIMENSION SGS(MXSPR),LES(MXSEL),ISTYP(MXSEL)                      SFLO 155
      DIMENSION WSS(MXWPR),NPW(MXWNP),IWTYP(MXWNP)                      SFLO 160
C                                                                       SFLO 165
      DIMENSION NPRS(MXRSNP),NPDB(MXDNP),NPCB(MXCNP),NPNB(MXNNP)        SFLO 170
C                                                                       SFLO 175
      DIMENSION PROP(MXMPPM,MAXMAT)                                     SFLO 180
C                                                                       SFLO 185
      DIMENSION XQ(4),ZQ(4),DHQ(4),THG(4)                               SFLO 190
C                                                                       SFLO 195
      DATA TWOPI/6.283185307D0/                                         SFLO 200
      DATA QSOS/0.0D0/                                                  SFLO 205
C                                                                       SFLO 210
      DO 110 NP=1,NBN                                                   SFLO 215
      BFLXP(NP)=BFLX(NP)                                                SFLO 220
  110 BFLX(NP)=0.0                                                      SFLO 225
C                                                                       SFLO 230
C ******* CALCULATE VOLUMETRIC FLOW RATE PER UNIT Y-LENGTH FOR ALL      SFLO 235
C ******* BOUNDARY NODES                                                SFLO 240
C                                                                       SFLO 245
      DO 170 MP=1,NBEL                                                  SFLO 250
      M=NBE(MP)                                                         SFLO 255
      NI=ISB(1,MP)                                                      SFLO 260
      NJ=ISB(2,MP)                                                      SFLO 265
      RR=((X(NI)+X(NJ))/2.0D0)                                          SFLO 270
      IF(CYLIND.EQ.0.0) RR=1.0D0                                        SFLO 275
      DO 120 I=1,NBN                                                    SFLO 280
      IJ=NPB(I)                                                         SFLO 285
      IF(IJ.NE.NI) GO TO 120                                            SFLO 290
      NII=I                                                             SFLO 295
      GO TO 130                                                         SFLO 300
  120 CONTINUE                                                          SFLO 305
  130 DO 140 J=1,NBN                                                    SFLO 310
      IJ=NPB(J)                                                         SFLO 315
      IF(IJ.NE.NJ) GO TO 140                                            SFLO 320
      NJJ=J                                                             SFLO 325
      GO TO 150                                                         SFLO 330
  140 CONTINUE                                                          SFLO 335
  150 CONTINUE                                                          SFLO 340
```

```
C                                                                       SFLO 345
C ------- COMPUTE THE CONTRIBUTION OF SIDE (NI,NJ) TO THE TOTAL FLUX    SFLO 350
C ------- THROUGH POINTS NI AND NJ, RESPECTIVELY.                       SFLO 355
C                                                                       SFLO 360
      FNI=(VX(NI)*DCOSXB(MP)+VZ(NI)*DCOSZB(MP))*DLB(MP)*RR              SFLO 365
      FNJ=(VX(NJ)*DCOSXB(MP)+VZ(NJ)*DCOSZB(MP))*DLB(MP)*RR              SFLO 370
C                                                                       SFLO 375
C ------- DISTRIBUTE THE ABOVE FLUXES TO TOW END POINTS OF THE SIDE     SFLO 380
C                                                                       SFLO 385
      BFLX(NII)=BFLX(NII)+FNI/3.0D0+FNJ/6.0D0                           SFLO 390
      BFLX(NJJ)=BFLX(NJJ)+FNJ/3.0D0+FNI/6.0D0                           SFLO 395
C                                                                       SFLO 400
  170 CONTINUE                                                          SFLO 405
C                                                                       SFLO 410
      IF (KFLOW.GT.0) GO TO 200                                         SFLO 415
      DO 180 NP=1,NBN                                                   SFLO 420
  180 BFLXP(NP)=BFLX(NP)                                                SFLO 425
C                                                                       SFLO 430
      DO 190 I=1,9                                                      SFLO 435
  190 TFLOW(I)=0.0                                                      SFLO 440
C                                                                       SFLO 445
C ******* DETERMINE TOTAL FLOWS AND TOTAL FLOW RATES THROUGH VARIOUS    SFLO 450
C ******* TYPES OF BOUNDARIES, STARTING WITH THE NET FLOWS THROUGH THE  SFLO 455
C ******* ENTIRE BOUNDARY.                                              SFLO 460
C                                                                       SFLO 465
  200 SUM=0.                                                            SFLO 470
      SUMP=0.                                                           SFLO 475
      DO 210 NP=1,NBN                                                   SFLO 480
      SUM=SUM+BFLX(NP)                                                  SFLO 485
  210 SUMP=SUMP+BFLXP(NP)                                               SFLO 490
C                                                                       SFLO 495
      FRATE(7)=SUM                                                      SFLO 500
      FLOW(7)=0.5D0*(SUM+SUMP)*DELT                                     SFLO 505
C                                                                       SFLO 510
C ******* THE DIRICHLET BOUNDARY                                        SFLO 515
C                                                                       SFLO 520
      FRATE(1)=0.                                                       SFLO 525
      FLOW(1)=0.                                                        SFLO 530
      IF(NDNP.LE.0) GO TO 400                                           SFLO 535
      SUM=0.                                                            SFLO 540
      SUMP=0.                                                           SFLO 545
      DO 330 NPP=1,NDNP                                                 SFLO 550
      NP=NPDB(NPP)                                                      SFLO 555
         DO 310 I=1,NBN                                                 SFLO 560
      IJ=NPB(I)                                                         SFLO 565
      IF(IJ.NE.NP) GO TO 310                                            SFLO 570
      NII=I                                                             SFLO 575
      GO TO 320                                                         SFLO 580
  310 CONTINUE                                                          SFLO 585
  320 CONTINUE                                                          SFLO 590
```

```
      SUM=SUM+BFLX(NII)                                                 SFLO 595
      SUMP=SUMP+BFLXP(NII)                                              SFLO 600
  330 CONTINUE                                                          SFLO 605
      FRATE(1)=SUM                                                      SFLO 610
      FLOW(1)=0.5D0*(SUM+SUMP)*DELT                                     SFLO 615
C                                                                       SFLO 620
C ******* THE CAUCHY BOUNDARY                                           SFLO 625
C                                                                       SFLO 630
  400 FRATE(2)=0.0                                                      SFLO 635
      FLOW(2)=0.0                                                       SFLO 640
      IF(NCNP.LE.0) GO TO 500                                           SFLO 645
      SUM=0.0                                                           SFLO 650
      SUMP=0.0                                                          SFLO 655
      DO 430 NPP=1,NCNP                                                 SFLO 660
         NP=NPCB(NPP)                                                   SFLO 665
         DO 410 I=1,NBN                                                 SFLO 670
            IJ=NPB(I)                                                   SFLO 675
            IF(IJ.NE.NP) GO TO 410                                      SFLO 680
            NII=I                                                       SFLO 685
            GO TO 420                                                   SFLO 690
  410    CONTINUE                                                       SFLO 695
  420    CONTINUE                                                       SFLO 700
         SUM=SUM+BFLX(NII)                                              SFLO 705
         SUMP=SUMP+BFLXP(NII)                                           SFLO 710
  430 CONTINUE                                                          SFLO 715
      FRATE(2)=SUM                                                      SFLO 720
      FLOW(2)=0.5D0*(SUM+SUMP)*DELT                                     SFLO 725
C                                                                       SFLO 730
C ******* THE NEUMANN BOUNDARY                                          SFLO 735
C                                                                       SFLO 740
  500 FRATE(3)=0.                                                       SFLO 745
      FLOW(3)=0.                                                        SFLO 750
      IF(NNNP.LE.0) GO TO 600                                           SFLO 755
      SUM=0.                                                            SFLO 760
      SUMP=0.                                                           SFLO 765
      DO 530 NPP=1,NNNP                                                 SFLO 770
         NP=NPNB(NPP)                                                   SFLO 775
      DO 510 I=1,NBN                                                    SFLO 780
      IJ=NPB(I)                                                         SFLO 785
      IF(IJ.NE.NP) GO TO 510                                            SFLO 790
      NII=I                                                             SFLO 795
      GO TO 520                                                         SFLO 800
  510 CONTINUE                                                          SFLO 805
  520 CONTINUE                                                          SFLO 810
      SUM=SUM+BFLX(NII)                                                 SFLO 815
      SUMP=SUMP+BFLXP(NII)                                              SFLO 820
  530 CONTINUE                                                          SFLO 825
      FRATE(3)=SUM                                                      SFLO 830
      FLOW(3)=0.5D0*(SUM+SUMP)*DELT                                     SFLO 835
C                                                                       SFLO 840
```

```
C ******* THE RAINFALL-SEEPAGE BOUNDARY                                  SFLO 845
C                                                                        SFLO 850
  600 FRATE(4)=0.                                                        SFLO 855
      FLOW(4)=0.                                                         SFLO 860
      FRATE(5)=0.                                                        SFLO 865
      FLOW(5)=0.                                                         SFLO 870
      IF(NRSN.LE.0) GO TO 700                                            SFLO 875
      SUMS=0.                                                            SFLO 880
      SUMSP=0.                                                           SFLO 885
      SUMR=0.                                                            SFLO 890
      SUMRP=0.                                                           SFLO 895
      DO 640 NPP=1,NRSN                                                  SFLO 900
         NP=NPRS(NPP)                                                    SFLO 905
         DO 610 I=1,NBN                                                  SFLO 910
            IJ=NPB(I)                                                    SFLO 915
            IF(IJ.NE.NP) GO TO 610                                       SFLO 920
            NII=I                                                        SFLO 925
            GO TO 620                                                    SFLO 930
  610       CONTINUE                                                     SFLO 935
  620    CONTINUE                                                        SFLO 940
         BFLXA=BFLX(NII)                                                 SFLO 945
         IF (BFLXA.LT.0.D0) GO TO 630                                    SFLO 950
         SUMS=SUMS+BFLX(NII)                                             SFLO 955
         SUMSP=SUMSP+BFLXP(NII)                                          SFLO 960
         GO TO 640                                                       SFLO 965
  630    SUMR=SUMR+BFLX(NII)                                             SFLO 970
         SUMRP=SUMRP+BFLXP(NII)                                          SFLO 975
  640    CONTINUE                                                        SFLO 980
      FRATE(4)=SUMS                                                      SFLO 985
      FLOW(4)=0.5D0*(SUMS+SUMSP)*DELT                                    SFLO 990
      FRATE(5)=SUMR                                                      SFLO 995
      FLOW(5)=0.5D0*(SUMR+SUMRP)*DELT                                    SFLO1000
C                                                                        SFLO1005
C ******* THE UNSPECIFIED BOUNDARY, I. E. BOUNDARY WITH ZERO TOTAL FLUX SFLO1010
C                                                                        SFLO1015
  700 SUM=0.                                                             SFLO1020
      SUMP=0.                                                            SFLO1025
      DO 710 I=1,5                                                       SFLO1030
      SUM=SUM+FRATE(I)                                                   SFLO1035
  710 SUMP=SUMP+FLOW(I)                                                  SFLO1040
      FRATE(6)=FRATE(7)-SUM                                              SFLO1045
      FLOW(6)=FLOW(7)-SUMP                                               SFLO1050
C                                                                        SFLO1055
C ******* CALCULATE THE INCREASE IN THE WATER CONTENT AND THE SOURCE     SFLO1060
C                                                                        SFLO1065
      QSOSP=QSOS                                                         SFLO1070
      QSOS=0.0                                                           SFLO1075
      QTH=0.                                                             SFLO1080
      DO 830 M=1,NEL                                                     SFLO1085
      NQ=4                                                               SFLO1090
```

9 2 1 2 4 5 8 0 7 0 2

```
      IEM4=IE(M,4)                                                      SFL01095
      IF(IEM4.EQ.0) NQ=3                                                SFL01100
      MTYP=IE(M,5)                                                      SFL01105
         ALP=PROP(1,MTYP)                                               SFL01110
         BETAP=PROP(2,MTYP)                                             SFL01115
         POR=PROP(3,MTYP)                                               SFL01120
C                                                                       SFL01125
      SOURCE=0.0                                                        SFL01130
      IF(NSEL.LE.0.0) GO TO 805                                         SFL01135
      DO 802 I=1,NSEL                                                   SFL01140
      MP=LES(I)                                                         SFL01145
      IF(MP.EQ.M) GO TO 804                                             SFL01150
  802 CONTINUE                                                          SFL01155
      GO TO 805                                                         SFL01160
  804 ITYP=ISTYP(I)                                                     SFL01165
      SOURCE=SOS(ITYP)                                                  SFL01170
  805 CONTINUE                                                          SFL01175
C                                                                       SFL01180
      DO 810 IQ=1,NQ                                                    SFL01185
            NP=IE(M,IQ)                                                 SFL01190
            XQ(IQ)=X(NP)                                                SFL01195
            ZQ(IQ)=Z(NP)                                                SFL01200
            DHQ(IQ)=H(NP)-HP(NP)                                        SFL01205
      IF(KFLOW.LE.0) DHQ(1Q)=1.0D0                                      SFL01210
  810 CONTINUE                                                          SFL01215
C                                                                       SFL01220
      DO 820 KG=1,NQ                                                    SFL01225
      THG(KG)=TH(KG,M)                                                  SFL01230
      IF(KFLOW.GT.0) THG(KG)=DTH(KG,M)+ALP*THG(KG)/POR+BETAP*THG(KG)    SFL01235
  820 CONTINUE                                                          SFL01240
C                                                                       SFL01245
      IF(NQ.EQ.3) CALL Q3TH(QTHM,DHQ,XQ,ZQ,THG,QSOSM,SOURCE)            SFL01250
      IF(NQ.EQ.4) CALL Q4TH(QTHM,DHQ,XQ,ZQ,THG,QSOSM,SOURCE)            SFL01255
C                                                                       SFL01260
         QSOS=QSOS-QSOSM                                                SFL01265
         QTH=QTH+QTHM                                                   SFL01270
  830 CONTINUE                                                          SFL01275
C                                                                       SFL01280
      IF(NWNP.EQ.0) GO TO 850                                           SFL01285
      DO 840 I=1,NWNP                                                   SFL01290
      ITYP=IWTYP(I)                                                     SFL01295
  840 QSOS=QSOS-WSS(ITYP)/TWOPI**CYLIND                                 SFL01300
  850 CONTINUE                                                          SFL01305
C                                                                       SFL01310
      IF(KFLOW.GT.0) GO TO 880                                          SFL01315
      QSOSP=QSOS                                                        SFL01320
CC                                                                      SFL01325
  880 FRATE(8)=QSOS                                                     SFL01330
      FLOW(8)=0.5D0*(QSOS+QSOSP)*DELT                                   SFL01335
      FLOW(9)=QTH                                                       SFL01340
```

```
      FRATE(9)=FLOW(9)/DELT                                             SFLO1345
      IF(KFLOW.LE.0) FRATE(9)=-(FRATE(7)+FRATE(8))                      SFLO1350
C                                                                       SFLO1355
      DO 910 I=1,9                                                      SFLO1360

  910    TFLOW(I)=TFLOW(I)+FLOW(I)                                      SFLO1365
C                                                                       SFLO1370
      RETURN                                                            SFLO1375
      END                                                               SFLO1380


      SUBROUTINE Q4TH(QTHM,DHQ,XQ,ZQ,THG,QSOSM,SOURCE)                  Q4TH 005
C                                                                       Q4TH 010
      IMPLICIT REAL*8 (A-H,O-Z)                                         Q4TH 015
      REAL*8 N(4)                                                       Q4TH 020
C                                                                       Q4TH 025
      COMMON /COORD/ CSFE,SNFE,CYLIND                                   Q4TH 030
C                                                                       Q4TH 035
      DIMENSION THG(4),S(4),T(4),XQ(4),ZQ(4),DHQ(4)                     Q4TH 040
      DIMENSION PJAB(2,2),DNSS(4),DNTT(4)                               Q4TH 045
C                                                                       Q4TH 050
      DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-    Q4TH 055
     > 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /             Q4TH 060
C                                                                       Q4TH 065
      QSOSM=0.0                                                         Q4TH 070
      QTHM=0.                                                           Q4TH 075
      DO 490 KG=1,4                                                     Q4TH 080
C                                                                       Q4TH 085
C ------- DETERMINE LOACAL COORDINATE (SS,TT) OF GAUSSIAN POINT KG      Q4TH 090
C                                                                       Q4TH 095
      SS=P*S(KG)                                                        Q4TH 100
      TT=P*T(KG)                                                        Q4TH 105
C                                                                       Q4TH 110
C ------- CALCULATE VALUES OF BASIS FUNCTIONS N(IQ)                     Q4TH 115
C                                                                       Q4TH 120
         CALL BASE(N,DNSS,DNTT,SS,TT)                                   Q4TH 125
C                                                                       Q4TH 130
      DO 210 I=1,2                                                      Q4TH 135
      DO 210 J=1,2                                                      Q4TH 140
  210 PJAB(I,J)=0.0                                                     Q4TH 145
C                                                                       Q4TH 150
      DO 220 I=1,4                                                      Q4TH 155
      PJAB(1,1)=PJAB(1,1)+ZQ(I)*DNTT(I)                                 Q4TH 160
      PJAB(1,2)=PJAB(1,2)-ZQ(I)*DNSS(I)                                 Q4TH 165
      PJAB(2,1)=PJAB(2,1)-XQ(I)*DNTT(I)                                 Q4TH 170
```

9 2 1 2 4 5 8 0 7 0 4

```
      PJAB(2,2)=PJAB(2,2)+XQ(I)*DNSS(I)                                 Q4TH 175
  220 CONTINUE                                                          Q4TH 180
C                                                                       Q4TH 185
      DJAC=PJAB(2,2)*PJAB(1,1)-PJAB(1,2)*PJAB(2,1)                      Q4TH 190
C                                                                       Q4TH 195
C ------- INTERPOLATE TO OBTAIN WATER CONTENT AT THE GAUSSIAN POINT KG  Q4TH 200
C                                                                       Q4TH 205
      DHQP=0.0                                                          Q4TH 210
      R=0.0                                                             Q4TH 215
      DO 390 IQ=1,4                                                     Q4TH 220
      DHQP=DHQP+DHQ(IQ)*N(IQ)                                           Q4TH 225
      R=R+XQ(IQ)*N(IQ)                                                  Q4TH 230
  390 CONTINUE                                                          Q4TH 235
C                                                                       Q4TH 240
      THQP=DHQP*THG(KG)                                                 Q4TH 245
      RR=1.0D0                                                          Q4TH 250
      IF(CYLIND.NE.0.0) RR=R                                            Q4TH 255
C                                                                       Q4TH 260
C ------- ACCUMULATE THE SUM TO EVALUATE THE INTEGRAL                   Q4TH 265
C                                                                       Q4TH 270
         QSOSM=QSOSM+SOURCE*DJAC*RR                                     Q4TH 275
         QTHM=QTHM+THQP*DJAC*RR                                         Q4TH 280
  490 CONTINUE                                                          Q4TH 285
C                                                                       Q4TH 290
      RETURN                                                            Q4TH 295
      END                                                               Q4TH 300


      SUBROUTINE Q3TH(QTHM,DHQ,XQ,ZQ,THG,QSOSM,SOURCE)                  Q3TH 005
C                                                                       Q3TH 010
      IMPLICIT REAL*8 (A-H,O-Z)                                         Q3TH 015
C                                                                       Q3TH 020
      COMMON /COORD/ CSFE,SNFE,CYLIND                                   Q3TH 025
C                                                                       Q3TH 030
      DIMENSION XQ(4),ZQ(4),DHQ(4),THG(4)                               Q3TH 035
      DIMENSION A(3),STRGE(3)                                           Q3TH 040
C                                                                       Q3TH 045
      DATA OSIX/0.1666666666666667D0/                                   Q3TH 050
      DATA OTWLVE/0.8333333333333333D-1/                                Q3TH 055
C                                                                       Q3TH 060
      X1=XQ(1)                                                          Q3TH 065
      X2=XQ(2)                                                          Q3TH 070
      X3=XQ(3)                                                          Q3TH 075
      Z1=ZQ(1)                                                          Q3TH 080
      Z2=ZQ(2)                                                          Q3TH 085
      Z3=ZQ(3)                                                          Q3TH 090
```

```
      A(1)=X2*Z3 - X3*Z2                                                Q3TH 095
      A(2)=X3*Z1 - X1*Z3                                                Q3TH 100
      A(3)=X1*Z2 - X2*Z1                                                Q3TH 105
C                                                                       Q3TH 110
      R=(X1+X2+X3)/3.0D0                                                Q3TH 115
      RR=1.0D0                                                          Q3TH 120
      IF(CYLIND.NE.0.0) RR=R                                            Q3TH 125
C                                                                       Q3TH 130
      AREA2=DABS(A(1)+A(2)+A(3))                                        Q3TH 135
      AREA=AREA2/2.0D0*RR                                               Q3TH 140
C                                                                       Q3TH 145
      DO 320 IQ=1,3                                                     Q3TH 150
      STRGE(IQ)=THG(IQ)                                                 Q3TH 155
  320 CONTINUE                                                          Q3TH 160
C                                                                       Q3TH 165
      QTHM=STRGE(1)*(DHQ(1)*OSIX+(DHQ(2)+DHQ(3))*OTWLVE)*AREA +         Q3TH 170
     1     STRGE(2)*(DHQ(2)*OSIX+(DHQ(3)+DHQ(1))*OTWLVE)*AREA +         Q3TH 175
     2     STRGE(3)*(DHQ(3)*OSIX+(DHQ(1)+DHQ(2))*OTWLVE)*AREA           Q3TH 180
C                                                                       Q3TH 185
      QSOSM=-SOURCE*AREA                                                Q3TH 190
C                                                                       Q3TH 195
      RETURN                                                            Q3TH 200
      END                                                               Q3TH 205
```

```
      SUBROUTINE PRINTT(VX,VZ,H,HT,TH, NPB,BFLX,NPRS,RSFLX,NPCON,NPFLX, PRIN 005
     1  SUBHD, TIME,DELT, KPR,KOUT,KDIAG,ITIM, NPMIN)                   PRIN 010
C                                                                       PRIN 015
      IMPLICIT REAL*8(A-H,O-Z)                                          PRIN 020
      REAL*4 SUBHD                                                      PRIN 025
C                                                                       PRIN 030
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI            PRIN 035
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,IBAND,KGRAV,NTI                   PRIN 040
      COMMON /RSBC/ MXRSEL,MXRSNP,MXRFPR,MXRPAR,NRSEL,NRSN,NRFPR,NRFPAR PRIN 045
C                                                                       PRIN 050
      COMMON /CFLOW/ FRATE(10),FLOW(10),TFLOW(10)                       PRIN 055
C                                                                       PRIN 060
      DIMENSION VX(MAXNP),VZ(MAXNP),H(MAXNP),HT(MAXNP),TH(4,MAXEL)      PRIN 065
      DIMENSION NPB(MAXBNP),BFLX(MAXBNP),NPRS(MXRSNP),RSFLX(MXRSNP),    PRIN 070
     > NPCON(MXRSNP),NPFLX(MXRSNP), NPMIN(MXRSNP)                       PRIN 075
C                                                                       PRIN 080
      DIMENSION SUBHD(8)                                                PRIN 085
C                                                                       PRIN 090
      IF(KPR.EQ.0) RETURN                                               PRIN 095
C                                                                       PRIN 100
C ------- PRINT DIAGNOSTIC FLOW INFORMATION                             PRIN 105
```

```
C
  110 KDIAG=KDIAG+1
      KDIA=KDIAG-1
      PRINT 1000, KDIA,TIME,DELT,ITIM,(FRATE(I),FLOW(I),TFLOW(I),I=1,9)
      IF (NRSN.EQ.0) GO TO 130
      DO 120 NPP=1,NRSN
         NP=NPRS(NPP)
         DO 115 I=1,NBN
            IJ=NPB(I)
            IF(IJ.NE.NP) GO TO 115
            NKK=I
            GO TO 120
  115       CONTINUE
  120    RSFLX(NPP)=BFLX(NKK)
      PRINT 1100
      PRINT 1110, (RSFLX(NPP),NPP=1,NRSN)
      PRINT 1120, (NPCON(NPP),NPP=1,NRSN)
      PRINT 1125, (NPMIN(NPP),NPP=1,NRSN)
      PRINT 1130,  (NPFLX(NPP),NPP=1,NRSN)
  130 IF (KPR.EQ.1) RETURN
C
C ------- PRINT PRESSURE HEADS
C
      KOUT=KOUT+1
      LINE=0
      DO 210 NI=1,NNP,8
         NJMN=NI
         NJMX=MIN0(NI+7,NNP)
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 2000, KOUT,TIME,
     >     DELT,IBAND,ITIM,(SUBHD(I),I=1,8)
  210    PRINT 2100, NI,(H(NJ),NJ=NJMN,NJMX)
      IF (KPR.EQ.2) RETURN
C
C ------- PRINT TOTAL HEADS
C
      KOUT=KOUT+1
      LINE=0
      DO 310 NI=1,NNP,8
         NJMN=NI
         NJMX=MIN0(NI+7,NNP)
      LINE=LINE+1
      IF(MOD(LINE-1,50).EQ.0) PRINT 3000, KOUT,TIME,
     >     DELT,IBAND,ITIM,(SUBHD(I),I=1,8)
  310    PRINT 3100, NI,(HT(NJ),NJ=NJMN,NJMX)
      IF(KPR.EQ.3) RETURN
C
C ------- PRINT WATER CONTENTS
C
      KOUT=KOUT+1
```

```
      LINE=0                                                            PRIN 360
      DO 410 M=1,NEL,2                                                  PRIN 365
      LINE=LINE+1                                                       PRIN 370
      IF(MOD(LINE-1,50).EQ.0) PRINT 4000, KOUT,TIME,                    PRIN 375
     >     DELT,IBAND,ITIM,(SUBHD(I),I=1,8)                             PRIN 380
         NJMN=M                                                         PRIN 385
         NJMX=MIN0(M+1,NEL)                                             PRIN 390
  410    PRINT 4100,  (MJ,(TH(IQ,MJ),IQ=1,4),MJ=NJMN,NJMX)              PRIN 395
      IF (KPR.EQ.4) RETURN                                              PRIN 400
C                                                                       PRIN 405
C ------- PRINT DARCY VELOCITIES                                        PRIN 410
C                                                                       PRIN 415
      KOUT=KOUT+1                                                       PRIN 420
      LINE=0                                                            PRIN 425
      DO 510 NP=1,NNP,4                                                 PRIN 430
      LINE=LINE+1                                                       PRIN 435
      IF(MOD(LINE-1,50).EQ.0) PRINT 5000, KOUT,TIME,                    PRIN 440
     >     DELT,IBAND,ITIM,(SUBHD(I),I=1,8)                             PRIN 445
         NJMN=NP                                                        PRIN 450
         NJMX=MIN0(NP+3,NNP)                                            PRIN 455
  510    PRINT 5100,  (NJ,VX(NJ),VZ(NJ),NJ=NJMN,NJMX)                   PRIN 460
C                                                                       PRIN 465
      RETURN                                                            PRIN 470
C                                                                       PRIN 475
 1000 FORMAT(1H1,' TABLE OF SYSTEM-FLOW PARAMETERS',2X,'TABLE: ',I4,    PRIN 480
     > '.. AT TIME =',1PD12.4,' ,(DELT =',1PD12.4,')',' ITIM=',I4//5X,  PRIN 485
     > ' TYPE OF FLOW',35X,'RATE(L**3/T/L)',3X,'INC. FLOW(L**3/L)',4X,  PRIN 490
     > 'TOTAL FLOW(L**3/L)'/5X,                                         PRIN 495
     1 40H  1. FLOW THROUGH DIRICHLET NODES .. . .,3(E12.4,10X)/5X,     PRIN 500
     2 40H  2. FLOW THROUGH CAUCHY NODES . . . . .,3(E12.4,10X)/5X,     PRIN 505
     3 40H  3. FLOW THROUGH NEUMANN NODES .  . . .,3(E12.4,10X)/5X,     PRIN 510
     4 40H  4. FLOW THROUGH SEEPAGE NODES . . . . .,3(E12.4,10X)/5X,    PRIN 515
     5 40H  5. FLOW THROUGH INFILTRATION NODES . .,3(E12.4,10X)/5X,     PRIN 520
     6 40H  6. FLOW THROUGH UNSPECIFIED NODES  . .,3(E12.4,10X)/5X,     PRIN 525
     7 40H  7. NET FLOW THROUGH ENTIRE BOUNDARY  .,3(E12.4,10X)/5X,     PRIN 530
     8 40H  8. ARTIFICIAL SOURCES/SINKS . . . .   ,3(E12.4,10X)/5X,     PRIN 535
     9 40H  9. INCREASE IN WATER CONTENT . . . . .,3(E12.4,10X)/5X,     PRIN 540
     A ' *** NOTE: (+) = OUT FROM, (-) = INTO THE REGION. '//)          PRIN 545
 1100 FORMAT(/50H RAINFALL-SEEPAGE NODAL FLOWS (((L**3/T)/L**2).L).)    PRIN 550
 1110 FORMAT(8D15.4)                                                    PRIN 555
 1120 FORMAT(1H0,16H VALUES OF NPCON/(8I15))                            PRIN 560
 1125 FORMAT(1H0,16H VALUES OF NPMIN/(8I15))                            PRIN 565
 1130 FORMAT(1H0,16H VALUES OF NPFLX/(8I15))                            PRIN 570
 2000 FORMAT(13H1OUTPUT TABLE,I4,30H.. PRESSURE HEADS(L) AT TIME =,     PRIN 575
     > 1PD12.4,9H ,(DELT =,1PD12.4,15H),(BAND WIDTH =,I4,1H),6H  IT =,  PRIN 580
     > I5//1X,8A4/1X,7H NODE I,5X,40HPRESSURE HEAD (L) OF NODES I,I+1,..PRIN 585
     >,I+7 /)                                                           PRIN 590
 2100 FORMAT(I7,8(1PD15.4))                                             PRIN 595
```

```
 3000 FORMAT(13H1OUTPUT TABLE,I4,26H. TOTAL HEADS(L) AT TIME =,1PD12.4,  PRIN 600
     > 9H ,(DELT =,1PD12.4,15H),(BAND WIDTH =,I4,1H),6H  IT =,I5//1X,8A4PRIN 605
     > /1X,7H NODE I,5X,37HTOTAL HEAD (L) OF NODES I,I+1,...,I+7/)      PRIN 610
 4000 FORMAT(13H1OUTPUT TABLE,I4,36H. WATER CONTENT(L**3/L**3) AT TIME =PRIN 615
     >,1PD12.4,9H ,(DELT =,1PD12.4,15H),(BAND WIDTH =,I4,1H),6H  IT =,  PRIN 620
     2 I5//1X,8A4/30X,16H GAUSSIAN POINTS,40X,16H GAUSSIAN POINTS/      PRIN 625
     3 2(17X,1H1,11X,1H2,11X,1H3,11X,1H4,6X)/2(3X,7HELEMENT,2X,         PRIN 630
     > 46H**********************************************,2X)/)         PRIN 635
 4100 FORMAT(1H ,2(1X,I7,2X,1PD12.4,1PD12.4,1PD12.4,1PD12.4,2X))        PRIN 640
 5000 FORMAT(13H1OUTPUT TABLE,I4,35H.. DARCY VELOCITIES (L/T) AT TIME =,PRIN 645
     > 1PD12.4,9H ,(DELT =,1PD12.4,15H),(BAND WIDTH =,I4,1H),6H  IT =,  PRIN 650
     > I5//1X,8A4/2X,4HNODE,9X,2HVX,9X,2HVZ,4X,4HNODE,9X,2HVX,9X,2HVZ,  PRIN 655
     > 4X,4HNODE,9X,2HVX,9X,2HVZ,4X,4HNODE,9X,2HVX,9X,2HVZ/             PRIN 660
     > 27H***************************,3X,27H***************************,PRIN 665
     >3X,27H***************************,3X,27H***************************PRIN 670
     >*/)                                                               PRIN 675
 5100 FORMAT(1H ,I5,2D11.3,3X,I5,2D11.3,3X,I5,2D11.3,3X,I5,2D11.3)      PRIN 680
      END                                                               PRIN 685
```

```
      SUBROUTINE STORE(X,Z,IE,H,HT,TH,VX,VZ,DLB,DCOSXB,DCOSZB,NBE,ISB,  STOR 005
     1 NPB,NPCON,NPFLX, TITLE, TIME,     NPROB,NSTRT, NPMIN)            STOR 010
C                                                                       STOR 015
      IMPLICIT REAL*8(A-H,O-Z)                                          STOR 020
C                                                                       STOR 025
      COMMON /SGEOM/ MAXEL,MAXNP,MAXBEL,MAXBNP,MAXHBP,MAXNTI            STOR 030
      COMMON /CGEOM/ NNP,NEL,NBN,NBEL,IBAND,KGRAV,NTI                   STOR 035
      COMMON /RSBC/ MXRSEL,MXRSNP,MXRFPR,MXRPAR,NRSEL,NRSN,NRFPR,NRFPAR STOR 040
C                                                                       STOR 045
      DIMENSION TITLE(9)                                                STOR 050
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)                           STOR 055
      DIMENSION H(MAXNP),HT(MAXNP),VX(MAXNP),VZ(MAXNP),TH(4,MAXEL)      STOR 060
      DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),NBE(MAXBEL),  STOR 065
     > ISB(4,MAXBEL),NPB(MAXBNP)                                        STOR 070
      DIMENSION NPCON(MXRSNP),NPFLX(MXRSNP),NPMIN(MXRSNP)               STOR 075
C                                                                       STOR 080
      DATA NPPROB/-1/                                                   STOR 085
C                                                                       STOR 090
      IF (NSTRT.GT.0) GO TO 110                                         STOR 095
      IF (NPPROB.EQ.(-1)) REWIND 1                                      STOR 100
      IF (NPPROB.EQ.NPROB) GO TO 110                                    STOR 105
      WRITE(1) (TITLE(I),I=1,9),NPROB,NNP,NEL,NBN,NBEL,NTI,NRSN         STOR 110
      WRITE(1) (X(N),N=1,NNP),(Z(N),N=1,NNP),((IE(M,I),M=1,NEL),I=1,4), STOR 115
     1 (DLB(M),M=1,NBEL),(DCOSXB(M),M=1,NBEL),(DCOSZB(M),M=1,NBEL),     STOR 120
     2 (NBE(M),M=1,NBEL),((ISB(I,M),I=1,4),M=1,NBEL),(NPB(N),N=1,NBN)   STOR 125
      NPPROB=NPROB                                                      STOR 130
```

```
C                                                                       STOR 135
      RETURN                                                            STOR 140
C                                                                       STOR 145
  110 WRITE(1) TIME,(H(N),N=1,NNP),(HT(N),N=1,NNP),((TH(I,M),I=1,4),M=1,STOR 150
     1 NEL),(VX(N),N=1,NNP),(VZ(N),N=1,NNP),(NPCON(N),N=1,NRSN),        STOR 155
     2 (NPFLX(N),N=1,NRSN), (NPMIN(N),N=1,NRSN)                         STOR 160
      RETURN                                                            STOR 165
      END                                                               STOR 170
```

```
      SUBROUTINE INTERP(FALL,TRF,RF,TIME,MXPR,MXDP,NPR,NDP)             INTE 005
C                                                                       INTE 010
      IMPLICIT REAL*8(A-H,O-Z)                                          INTE 015
C                                                                       INTE 020
      DIMENSION FALL(MXPR),TRF(MXDP,MXPR),RF(MXDP,MXPR)                 INTE 025
C                                                                       INTE 030
      DO 160 I=1,NPR                                                    INTE 035
      DO 140 J=2,NDP                                                    INTE 040
      IF(TRF(J-1,I).LE.TIME .AND. TIME.LE.TRF(J,I)) GO TO 120           INTE 045
      GO TO 140                                                         INTE 050
  120 RFJM1=RF(J-1,I)                                                   INTE 055
      TRFJM1=TRF(J-1,I)                                                 INTE 060
      RFJ=RF(J,I)                                                       INTE 065
        TRFJ=TRF(J,I)                                                   INTE 070
      ABC=RFJ-RFJM1                                                     INTE 075
      ABCD=TRFJ-TRFJM1                                                  INTE 080
       FALL(I)=RFJM1+(TIME-TRFJM1)*ABC/ABCD                             INTE 085
      GO TO 160                                                         INTE 090
  140 CONTINUE                                                          INTE 095
      FALL(I)=0.0                                                       INTE 100
  160 CONTINUE                                                          INTE 105
C                                                                       INTE 110
      RETURN                                                            INTE 115
      END                                                               INTE 120
```

```
      SUBROUTINE READR(F,MAXNOD,NNP)                                    REDR 005
C                                                                       REDR 010
      IMPLICIT REAL*8(A-H,O-Z)                                          REDR 015
C                                                                       REDR 020
      DIMENSION F(MAXNOD)                                               REDR 025
C                                                                       REDR 030
      NODES=0                                                           REDR 035
  150 READ 40, NI,NSEQ,NAD,FNI,FAD,FRD                                  REDR 040
```

```
      IF(NI.EQ.0) GO TO 170                                             REDR 045
      NJ=NI+NSEQ                                                        REDR 050
      DO 160 N=NI,NJ                                                    REDR 055
         NODES=NODES+1                                                  REDR 060
         I=NI+(N-NI)*NAD                                                REDR 065
      IF(FRD.NE.0.0) GO TO 155                                          REDR 070
         F(I)=FNI+FAD*DFLOAT(N-NI)                                      REDR 075
      GO TO 160                                                         REDR 080
  155 I1=I-NAD                                                          REDR 085
      IF(N.EQ.NI) F(I)=FNI                                              REDR 090
      IF(N.EQ.NI) DINC=1.0D0                                            REDR 095
      IF(N.GT.NI) DINC=DINC*(1.0D0+FRD)                                 REDR 100
      IF(N.GT.NI) F(I)=F(I1)+FAD*DINC                                   REDR 105
  160 CONTINUE                                                          REDR 110
      GO TO 150                                                         REDR 115
  170 IF(NODES.EQ.NNP) GO TO 180                                        REDR 120
      PRINT 1100                                                        REDR 125
      STOP                                                              REDR 130
  180 IF(NNP.LE.MAXNOD) GO TO 190                                       REDR 135
      PRINT 1200                                                        REDR 140
      STOP                                                              REDR 145
  190 CONTINUE                                                          REDR 150
C                                                                       REDR 155
   40 FORMAT(3I5,5X,4D10.3)                                             REDR 160
 1100 FORMAT(1H1/1H ,64H *** ERROR IN EXECUTING READR SINCE NODES  .NE. REDR 165
     > NNP:   STOP ***)                                                 REDR 170
 1200 FORMAT(1H1/1H ,46H *** NNP .GT. MAXNOD  IN EXECUTING READR: STOP) REDR 175
C                                                                       REDR 180
      RETURN                                                            REDR 185
      END                                                               REDR 190


      SUBROUTINE READN(INDTYP,MXTYP,NTYPE)                              REDN 005
C                                                                       REDN 010
      IMPLICIT REAL*8(A-H,O-Z)                                          REDN 015
C                                                                       REDN 020
      DIMENSION INDTYP(MXTYP)                                           REDN 025
C                                                                       REDN 030
      NTYPES=0                                                          REDN 035
  110 READ 30,NI,NSEQ,NAD,NITYP,NTYPAD                                  REDN 040
      IF(NI.EQ.0) GO TO 130                                             REDN 045
      NJ=NI+NSEQ                                                        REDN 050
      DO 120 N=NI,NJ                                                    REDN 055
      I=NI+(N-NI)*NAD                                                   REDN 060
      INDTYP(I)=NITYP + (N-NI)*NTYPAD                                   REDN 065
      NTYPES=NTYPES+1                                                   REDN 070
  120 CONTINUE                                                          REDN 075
      GO TO 110                                                         REDN 080
```

```
  130 IF(NTYPES.EQ.NTYPE) GO TO 140                                    REDN 085
      PRINT 1100                                                       REDN 090
      STOP                                                             REDN 095
  140 IF(NTYPE.LE.MXTYP) GO TO 150                                     REDN 100
      PRINT 1200                                                       REDN 105
      STOP                                                             REDN 110
  150 CONTINUE                                                         REDN 115
C                                                                      REDN 120
   30 FORMAT(5I5)                                                      REDN 125
 1100 FORMAT(1H1/1H ,64H *** ERROR IN EXECUTING READN SINCE NTYPES .NE. REDN 130
     > NTYPE: STOP ***)                                                REDN 135
 1200 FORMAT(1H1/1H ,46H *** NTYPE .GT. MXTYP IN EXECUTING READN: STOP) REDN 140
C                                                                      REDN 145
      RETURN                                                           REDN 150
      END                                                              REDN 155
```

9 2 1 2 4 5 8 0 7 1 2

INTERNAL DISTRIBUTION

1. S. I. Auerbach
2. W. J. Boegly, Jr.
3. N. H. Cutshall
4. E. C. Davis
5. C. S. Haase
6. D. D. Huff
7. S. Y. Lee
8-17. T. E. Myrick
18. W. W. Pitt
19-20. D. E. Reichle
21. T. H. Row
22-23. F. E. Sharples
24. S. H. Stow
25-34. L. E. Stratton
35. J. Switek
36-45. G. T. Yeh
46. Central Research Library
47-61. ESD Library
62-63. Laboratory Records Dept.
64. Laboratory Records, ORNL-RC
65. ORNL Patent Section
66. ORNL Y-12 Technical Library

EXTERNAL DISTRIBUTION

67. V. Dean Adams, Tennessee Technological University, Cookeville, TN 38501
68. M. U. Ahmad, Department of Geology, Ohio University, Athens, OH 45701
69. L. R. Ahuja, Water Quality and Watershed Research Laboratory, P.O. Box 1430, Agricultural Research Services, Southern Plains Area, USDA, Durant, OK 74702-1430
70. F. R. Baddour, Eastern Research Group, Inc., 6 Whittemore Street, Arlington, MA 02174
71. Andrew Barry, Virginia Polytechnic Institute and State University, Department of Agronomy, Blacksburg, VA 24061
72. Paul Beck, Ministry of the Environment, 135 St. Clair Ave. West, Toronto Ontario CANADA M4V 1P5
73. D. Bedcock, Gannett Fleming Engineers, P.O. Box 1963, Harrisburg, PA 17105
74. Beniwal Bowser and Morner, P.O. Box 51, Dayton, Ohio 45401
75. T. F. Bergen, Computer Application Engineers, Inc., P.O. Box 408, Exton, PA 19341
76. M. P. Bergeron, USGS, Water Resources Division, 521 West Seneca Street, Ithaca, NY 14850
77. R. P. Berube, Office of Environmental Guidance and Compliance, EH-23, U.S. Department of Energy, Washington, DC 20585
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. C. M. Borgstrom, Director, Office of NEPA Project Assistance, EH-25, U.S. Department of Energy, Washington, DC 20585

81. Efthimios Bouloutas, MIT, Dept. of Civil Engineering, Cambridge, MA 02139
82. Luis Bravo, Radian Corporation, 3200 E. Chapel Hill Rd., Research Triangle Park, NC 27709
83. Eshel Bresler, Soil Physicist, Agricultural Research Organization, The Volcanic Center, Institute of Soil and Water, P.O. Box 6, Bet Dagan, 50-250, Israel
84. John Broadway, Eastern Environmental Radiation Facility, P.O. Box 3009, Montgomery, AL 36193
85. S. R. Brown, Department of Hydrology, University of Arizona, Tucson, AZ 85721
86. T. W. Broyd, Atkins Research and Development, Woodcote Grove Ashley Rd., Epsom Surrey KT18 5BW England
87. W. Brutsaert, School of Civil Engineering, Cornell University, Ithaca, NY 14850
88. A. Burges, Golder Associates, 4104 148th Avenue, N.E., Redmond, WA 98052
89. Christopher S. Burke, Harza Engineering Company, 150 South Wacker Drive, Chicago, IL 60606
90. J. Thomas Callahan, Associate Director, Ecosystem Studies Program, Room 336, 1800 G Street, NW, National Science Foundation, Washington, DC 20550
91. G. N. Camp, Jr., 810 Smiley Ave., Cincinnati, OH 45240
92. D. B. Chambers, Executive Vice President, SENES Consultants Limited, 499 McNicoll Ave., Willowdale, Ontario, CANADA M2H 2C9
93. Jui-Lin Chang, College of Mineral and Energy Resources, West Virginia University, Morgantown, WV 26506-6070
94. Shoou-Yuh Chang, Dept. of Civil Engineering, University of Missouri-Rolla, Rolla, MO 65401
95. S. T. Chau, Alberta Environmental Protection Service, 14th Floor, Standard Life Center, 10405 Jasper Ave., Edmonton, Alberta, Canada
96. T. C. Chee, R&D and Byproducts Division, DP-123 (GTN), U.S. Department of Energy, Washington, DC 20545
97. Yee-Ping P. Chia, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
98. James Chieh, Ecology and Environment, Inc., 195 Sugg Road, P.O. Box D, Buffalo, NY 14225
99. S. W. Childs, Department of Soil Science, Oregon State University, Corvallis, OR 97331
100. 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
101. Royce Cline, North Dakota State Water Commission, 900 East Blvd., Bismarck, ND 58505
102. C. C. Clyde, Utah Water Resources Laboratory, Utah State University, Logan, UT 84322
103. C. R. Cole, Pacific Northwest Laboratory, P.O. Box 999, Richland, WA 99352
104. M. A. Collins, Illinois State Water Survey, 2204 Griffith Drive, Champaign, IL 61820

9 2 1 2 4 5 8 0 7 1 4

105. E. F. Conti, Office of Nuclear Regulatory Research, Nuclear Regulatory Commission, MS-1130-SS, Washington, DC 20555
106. D. Cooper, School of Public Health, Harvard University, 665 Huntington Ave., Boston, MA 02115
107. Ashok S. Damle, Research Triangle Institute, P.O. Box 12194, Research Triangle Park, NC 27709
108. S. R. Davidson, SCS Engineers, 11260 Roger Bacon Drive, Reston, VA 22090
109. D. G. deMarsily, Ecole Nationale, Superieure des Mines, Centre d'Informatique Geologigue, 35 Rue Saint-Honore, 77305 Fontainebleau, France
110. A. K. Deb, Roy F. Weston, Inc., Weston Way, West Center, PA 19380
111. D. B. Dunbar, Woodward-Clyde Consultants, Three Embarcadero Center, Suite 700, San Francisco, CA 94111
112. Amos Eddy, Director, OCS, Oklahoma Climatological Survey, 710 Asp, Suite 8, Norman, OK 73019
113. B. E. Ehleringer, Geosciences, Inc., San Diego, CA 92123
114. Aly El-Kadi, International Groundwater Modeling Center, Holcomb Research Institute, Butler University, 4600 Sunset Ave., Indianapolis, IN 46208
115. G. L. Evavold, REREM, Inc. 408 Board of Trade Bldg., 301 West First Street, Duluth, MN 55802
116. Greg Faber, DPRA Development Planning and Research Associates, Inc., 200 Research Drive, P.O. Box 727, Manhattan, KS 66502
117. Pat Fennessy, Golder Associates, 4104 148th Avenue, N.E., Redmond, WA 98052
118. G. J. Foley, Office of Environmental Process and Effects Research, U.S. Environmental Protection Agency, 401 M Street, SW, RD-682, Washington, DC 20460
119. W. A. Frankhouser, Director, Operations and Traffic Division, DP-122 (GTN), U.S. Department of Energy, Washington, DC 20545
120. R. A. Freeze, Dept. of Geological Sciences, University of British Columbia, Vancouver, B.C. Canada
121. Jesus Lopez Garcia, Division of Recursos Hidraulicos, INTECSA, Francisco Gervas, 6, 28020 Madrid, Spain
122. W. J. Gburek, U.S. Department of Agriculture, Northwest Watershed Research Center, 110 Research Building A, University Park, PA 16802
123. Ferruc.io Gera, ISMES SPA, sede de Roma, via dei Crociferi 44-00187 ROMA
124. C. G. Glyde, Utah Water Resources Laboratory, Utah State University, Logan, UT 84222
125. C. R. Goldman, Professor of Limnology, Director of Tahoe Research Group, Division of Environmental Studies, University of California, Davis, CA 94616
126. Dan Goode, Environmental Engineering Dept., GCA Corporation, 213 Burlington Road, Bedford, MA 01730
127. Stephen Gorelick, USGS, Water Resources Division, MS-21, 345 Middlefield Road, Menlo Park, CA 94205
128. Jennifer B. Gorrell, Water and Earth Science Associates, LTD, Box 430, CARP, Ontario KOA 1LO

129. A. Berge Cureghian, Project Management Division, Performance Analysis Department, Battelle Columbus, 505 King Avenue, Columbus, OH 43201
130. D. A. Hamilton, Waste Management Division, Dept. of Natural Resources, Stevens T. Mason Building, P.O. Box 30028, Lansing, MI 48909
131. J. M. Hamlett, Dept. of Agriculture Engineering, Iowa State University, Ames, IA 50011
132. Keith Hansen, Law Environmental Services, 2749 Delk Road, Marietta, GA 30067
133. J. E. Hawley, Chen and Associates, 96 South Zuni Street, Denver, CO 80223
134. M. E. Herman, USGS, Water Resources Division, Rm. 227, Federal Bldg., 705 North Plaza Street, Carson City, NV 89701
135. L. A. Hermann, Dept. of Geosciences, Louisiana Tech University, Ruston, LA 71272
136. Beverly Herzog, Illinois State Geological Survey, Natural Resources Bldg., 615 E. Peabody Drive, Champaign, IL 61820
137. Francois E. Heuze, Lawrence Livermore National Laboratory, Lawrence Livermore, CA 94550
138. Richard Heystee, Ontario Hydro, 700 University Avenue, Toronto, Ontario M5G 1X6 Canada
139. Jimmy Ho, NUS Corporation, Park West Two, Cliff Mine Road, Pittsburgh, PA 15275
140. J. A. Hoopes, 1261B Engineering Bldg., Department of Civil and Environmental Engineering, University of Wisconsin-Madison, Madison, WI 53706
141. J. W. Hopmans, Dept of Hydraulics and Catchment Hydrology, Agricultural University, Wagenigen, The Netherlands
142. J. Horyna, Department of Radiation Protection, Nuclear Research Institute 250-68 Rez., Prague, Czechoslovakia
143. Sharon M. Hotz, Harding Lawson Associates, 7655 Redwood Blvd., Novato, CA 94948
144. Tai-Dan Hso, WRC Engineering, Inc., 1660 South Albion Street, Suite 500, Denver, CO 80222
145. J. W. Huckabee, Manager, Ecological Studies Program, Electric Power Research Institute, 3412 Hillview Avenue, P.O. Box 10412, Palo Alto, CA 94303
146. Paul Hwang, ES Engineering Science, Two Flint Hill, 10521 Rosehaven Street, Fairfax, VA 22030-2899
147. Judith Jeffcoate, Scicon Computer Services Limited, Brick Close Kiln Farm, Milton Keynes, Great Britain MK1 3EJ
148. Cady Johnson, Mifflin & Associates, Inc., 2700 Sunset, Suite B-13, Las Vegas, NV 89120
149. R. Johnson, Department of Environmental Sciences, The Oregon Graduate Center, 19600 NW Walker Road, Beaverton, OR 97006
150. A. Jonasson, Department of Geology, Chalmers University of Technology, S-412, 96 Gothenburg, Sweden
151. David B. Jones, O'brien & Gere, Box 4873, 1304 Buckley Road, Syracuse, NY 13231

152. George Y. Jordy, Director, Office of Program Analysis, Office of Energy Research, ER-30, G-226, U.S. Department of Energy, Washington, DC 20545
153. Kyla S. Juvonen, Woodward-Clyde Consultants, Stanford Place 3, Suite 100, 4582 South Ulster Street Parkway, Denver, CO 80237
154. J. L. Kauschinger, Dept. of Civil Engineering, Tufts University, Medford, MA 02155
155. D. Kent, Whitehurst Hall, Oklahoma University, Stillwater, OK 74078
156. Raz Khaleel, New Mexico Tech, Dept. of Geosciences, Socorro, NM 87801
157. Anne Klock, Earth Technology Corporation, 3777 Long Beach Blvd., Long Beach, CA 90807
158. Nic Korte, P.O. Box 2567, Grand Junction, CO 81502
159. W. J. Kutz, HARZA Engineering Company, 150 South Wacker Drive, Chicago, IL 60606
160. David Langseth, Metcalf & Eddy, Inc., 50 Stanford Street, Boston, MA 02114
161. T. H. Larson, State Geological Survey Division, Illinois Department of Energy and Natural Resources, Natural Resources Bldg., 615 East Peabody Drive, Champaign, IL 61820
162. R. T. Lawson, Environmental Sciences Division, Australian Atomic Energy Commission, New Illanarra Rd., Lucas Heights, N.S.W., Australia
163. Denis R. LeBlanc, USGS, Water Resources Division, 150 Causeway Street, Suite 1001, Boston, MA 02114
164. D. B. Leclaire, Director, Office of Defense Waste and Transportation Management, DP-12 (GTN), U.S. Department of Energy, Washington, DC 20545
165. T. C. Lederman, Radiological Sciences Institute, State of New York, Dept. of Health, The Governor Nelson A. Rockefeller Empire State Plaza, Albany, NY 12201
166. Tien-chang Lee, University of California-Riverside, Dept. of Earth Sciences, Riverside, CA 92521
167. Joan Lefler, Environmental Review Division, Office of Testing and Evaluation Office of Toxic Substances, Environmental Protection Agency, 401 M Street, SW, Washington, DC 20460
168. J. F. Lentz, Applied Physics Laboratory, The Johns Hopkins University, Johns Hopkins Rd., Laurel, MD 20810
169. C. C. Leser, Dept. of Civil Engineering, University of Notre Dame, Notre Dame, IN 46556
170. J. T. Lin, United Industries Corp, Executive Plaza, Suite 211, 12835 Bell-Red Road, Bellevue, WA 98005
171. Stanley E. Logan, S. E. Logan and Associates, Inc., Santa Fe, NM 87501
172. G. V. Loganathan, Virginia Polytechnic Institute and State University, Dept. of Civil Engineering, Blacksburg, VA 24061
173. Y. C. Lu, Dept. of Civil Engineering, National Taiwan University, Taipei, Taiwan
174. M. A. Luckett, GURR & Associates, Inc., 2000 East Edgewood Drive, Suite 1, Lakeland, FL 33803

9 2 1 2 4 5 8 0 7 1 7

175. P. R. Manam, Sargent & Lundy Engineers, 55 East Monroe Street, Chicago, IL 60603
176. C. J. Mankin, Director, Oklahoma Geological Survey, The University of Oklahoma, 830 Van Vleet Oval, Room 163, Norman, OK 73019
177. A. Marples, Safety and Medical Services, British Nuclear Fuels Limited, Sellafield Seascale, Cumbria CA20 1PG, England
178. M. L. Maslia, USGS, 6481 Peachtree Industrial Blvd., Doraville, CA 30360
179. 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
180. Marry McGuire, University of Pittsburgh, Pittsburgh, PA 15005
181. D. C. McKinney, Department of Environmental Resources Engineering, Humboldt State University, Arcata, CA 95521
182. C. F. McLane III, Dept. of Environmental Sciences, University of Virginia, Charlottesville, VA 22903
183. G. L. Meyer, Environmental Protection Agency, 401 M Street SW, MS-ANR459, Washington, DC 20460
184. W. B. Miller, Sandia National Laboratory, P.O. Box 5800, Albuquerque, NM 87185
185. M. T. Mills, CorStar, 7315 Wisconsin, North Tower #702, Bethesda, MD 20814
186. Jan Mockler, Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439
187. David E. Mohr, Jacobs Engineering Group, Inc., 5301 Central Avenue, Suite 1700, Albuquerque, NM 87108
188. Fred H. Molz, Director, School of Engineering and Engineering Experiment Station, Auburn University, Auburn, AL 36849
189. H. A. Mooney, Department of Biological Sciences, Stanford University, Stanford, CA 94305
190. G. E. Moore, System Applications Inc., 101 Lucas Valley Road, San Rafael, CA 94903
191. Sharon M. Motz, Harding Lawson Associates, 7655 Redwood Blvd., Novato, CA 94948
192. Fabio Musmeci, ENEA CASACCIA, PAS/ISP/INGSITI, Via Anguillarese 301, 006 Roma Italy
193. J. D. Nasson, NEBG Installation Representative, Mail Code 311, General Electric Company, 175 Curtner Ave, San Jose, CA 95125
194. Wan Mokhtar Nawang, Universiti Tecknologi Malaysia, Kuala Lumpur, Malaysia
195. R. Obrist, Eidg. Techn. Hochschule, Institut fur Hydromechanik and Wasserwirtschaft, CH-8093 Zurich-Honggerberg, Switzerland
196. Edward O'Donnell, Division of Radiation Programs and Earth Sciences, U.S. Nuclear Regulatory Commission, Mail Stop 1130 SS, Washington, DC 20555
197. P. M. Ornstein, MS-697, U.S. Nuclear Regulatory Commission, 7915 Eastern Ave., Silver Spring, MD 20910

198. C. A. Oster, Energy Systems Department, Pacific Northwest Laboratories, Richland, WA 99352
199. Frank Parker, Dept. of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN 37235
200. J. C. Parker, Department of Agronomy, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061
201. D. V. Patel, Southern Company Services, Birmingham, AL 35202
202. Jack Paul, PEDCO, Inc., 1006 N. Bowen Road, Arlington, TX 76012
203. G. F. Pinder, Dept. of Civil and Environmental Engineering, Princeton University, Princeton, NJ 08544
204. James Pittman, State of Maryland, Environmental Health Administration, 201 West Preston Street, Baltimore, MD 21201
205. Wilfred Pon, Environsphere Company, Two World Trade Center, New York, NY 10048
206. Susan Power, ES Engineering-Science, 290 Elwood Davis Road, Liverpool, NY 13088
207. David Prudic, USGS, Water Resources Division, Room 229, Federal Building, Carson City, NV 89701
208. Ellen J. Quinn, Northeast Utilities, P.O. Box 270, Hartford, CT 06141
209. P.A.C. Raats, Institute for Soil Fertility, P.O. Box 30003, 9750 Ra Haren (Gr), Netherlands
210. M. S. Rahman, North Carolina State University, Dept. of Civil Engineering, Raleigh, NC 27695
211. Vishna Ranganathan, Dept. of Geology, Louisiana State University, Baton Rouge, LA 70803
212. Gregory Reed, Department of Civil Engineering, The University of Tennessee, Knoxville, TN 37916
213. Ed Repa, JRB Associates, 8400 West Park Drive, McLean, VA 22102
214. Antonio Marozzi Righetto, Universidade de Sao Paulo, Depto. de Hidraulica e Saneamento, Escola de Engenharia de S. Carlos, 13.560 - Sao Carlos, SP, Brazil
215. James W. Roberts, 1824 Arcadia Drive, Lafayette, IN 47905
216. John C. Rodgers, Group LS-6, Environmental Sciences Group, Los Alamos National Laboratory, Los Alamos, NM 87545
217. William L. Rohrer, Pope Reid Associates, Inc., 24 E. 6th Street, Suite 813, St. Paul, MN 55101
218. E. M. Romney, University of California, Los Angeles, 900 Veteran Avenue, Los Angeles, CA 90024
219. Wyn Ross, Desert Research Institute, University of Nevada, Reno, NV 89506
220. Philip Rosten, Jr., ERM-Southeast, Inc., P.O. Box 881, Brentwood, TN 37027
221. Dauglas Rumbaugh, Woodward-Clyde Consultants, 5120 Butler Pike, Plymouth Meeting, PA 19462
222. Lisa A. Sabol, IT Corporation, 10 Duff Road, Pittsburgh, PA 15235
223. Z. Saleem, Ebasco Services, Inc., 2211 West Meadowview Road, Greensboro, NC 27407
224. Karen A. Savinda, D'Appolonia, 10 Duff Road, Pittsburgh, PA 15235
225. J. L. Schnoor, Division of Energy Engineering, The University of Iowa, Iowa City, IA 52242

226. John E. Sevee, Sevee & Maher Engineers, P.A., 668 Main Street, Westbrook, ME 04092
227. Thomas R. Schultz, Wahler Associates, 12477 W. Cedar Drive, Suite 206, Lakewood, CO 80228
228. Joanna I. Scott, TASC-The Analytical Sciences Corporation, One Jacob Way, Reading, MA 01867
229. Maurice Sell, 1980 Philips Drive, Northglenn, CO 80233
230. Jacob Shapiro, Radiation Protection Officer, University Health Services, Harvard University, Cambridge, MA 02138
231. Derek Snowball, SCICON, Scicon Limited, Tick Close, Kiln Farm, Milton Keynes MK11 3EJ England
232. Marios Sophocleous, Kansas Geological Survey, 1930 Avenue A, Campus West, The University of Kansas, Lawrence, KS 66044
233. Eveerett P. Spring, NW Watershed Research Center, USDA, Suite 116, 1175 South Orchard, Boise, ID 83705
234. Jack Sosebee, Environmental Sciences and Engineering, Inc., 7332 S. Alton Way, Suite H-I, Englewood, CO 80110
235. R. H. Stacks, Department of Pollution Control and Ecology, State of Arkansas 8001 National Drive, P.O. Box 9583, Little Rock, AR 72209
236. R. J. Starmer, HLW Technical Development Branch, Office of Nuclear Material Safety and Safeguards, Nuclear Regulatory Commission, Room 427-SS, Washington, DC 20555
237. F. Stäuffer, Institute fur Hydromechanik and Wasserwirtschaft, ETH-Honggerberg, CH-8093 Zurich, Switzerland
238. J. G. Steger, Environmental Sciences Group, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545
239. D. B. Stephens, New Mexico Tech, Department of Geosciences, Socorro, NM 87801
240. R. J. Stern, Director, Office of Environmental Compliance, MS PE-25, FORRESTAL, U.S. Department of Energy, 1000 Independence Avenue, SW, Washington, DC 20585
241. M. T. Stewart, University of South Florida, Tampa, FL 33620
242. C. John Suen, Brookhaven National Laboratory, Nuclear Waste and Materials Technology Division, Upton, Long Island, NY 11973
243. Angelo Tagliacozzo, Gibbs & Hill, Inc., 11 Penn Plaza, New York, NY 10001
244. Robert J. Telford, ACRES Consulting Limited, 5259 Dorchester Road, P.O. Box 1001, Niagara Falls, Ontarion L2E 6W1 CANADA
245. Khanh T. Tran, AMI Applied Modeling Inc., Warner Financial Plaza, 6355 Topanga Canyon Blvd., Suite 433, Woodland Hills, CA 91367
246. Howard Trussell, 1261 Engineering Bldg., University of Wisconsin Madison, Madison, WI 53706
247. Steve Y. Tsai, Division of Environmental Impact Studies, Argonne National Laboratory, 9700 South Case Avenue, Argonne, IL 60439
248. S. B. Upchurch, University of South Florida, Tampa, FL 33620
249. M. van Genuchten, U S. Salinity Laboratory, 4500 Glenwood Drive, Riverside, CA 92501
250. John J. Vasconcelos, James M. Montgomery Engineers, 250 North Madison Avenue, Pasadena, CA 91101

251. Monique Villars, EWA, Environment and Water Resources Management, 12800 Industrial Park Blvd., Minneapolis, MN 55441
252. Jan Wagner, School of Chemical Engineering, Oklahoma State University, Stillwater, OK 74078
253. Ken Walker, Department of Geology, The University of Tennessee, Knoxville, TN 37916
254. Terry Warner, Rio Algom Mining Corp., La Sal Route, Moab, UT 84532
255. Leonard H. Weinstein, Program Director of Environmental Biology, Cornell University, Boyce Thompson Institute for Plant Research, Ithaca, NY 14853
256. T. C. Wells, D'Appolonia Consulting Engineers, Inc., 10 Duff Road, Pittsburgh, PA 15235
257. D. E. Wessol, Reactor Physics Branch, EG&G Idaho, Inc., P.O. Box 1625, Idaho Falls, ID 83415
258. T. E. White, School of Civil Engineering, Purdue University, West Lafayette, IN 47907
259. S. L. Whiteside, Geotechnical Engineers, Inc., 1017 Main Street, Winchester, MA 01890
260. D. Widmayer, U.S. Miclear Regulatory Commission, 7915 Eastern Ave., Silver Spring, MD 20920
261. P. J. Wierenga, Department of Agronomy, New Mexico State University, P.O. Box 3Q, Las Cruces, NM 88003
262. Raymond G. Wilhour, Chief, Air Pollution Effects Branch, Corvallis Environmental Research Laboratory, U.S. Environmental Protection Agency, 200 SW 35th Street, Corvallis, OR 97330
263. Patrick J. Withkowski, Dept. of Civil Engineering, Princeton University, Princeton, NJ 08544
264. Frank J. Wobber, Ecological Research Division, Office of Health and Environmental Research, Office of Energy Research, MS-E201, Department of Energy, Washington, DC 20545
265. M. Gordon Wolman, The Johns Hopkins University, Department of Geography and Environmental Engineering, Baltimore, MD 21218
266. J. R. Yearsley, U.S. Environmental Protection Agency, REGION 10, 1200 Sixth Avenue, Seattle, WA 98101
267. Jim Yeh, Dept. of Environmental Sciences, University of Virginia, Charlottesville, VA 22903
268. Young S. Yoon, Boyle Engineering Corporation, Suite 200, 165 South Union, Lakewood, CO 80228
269. H. H. Zehner, U.S. Geological Survey-Water Resources Division, 1013 N. Broadway, Knoxville, TN 37917
270. H. D. Yeh, Department of Civil Engineering, National Chiao-Tung University, Hsinchu, Taiwan
271. Paolo Zannetti, AeroVironment Inc., 825 Myrtle Ave., Monrovia, CA 91016-3424
272. Feng-quing Zhuang, Institute of Mechanics, Chinese Academy of Sciences, Beijing, Peoples' Republic of China

273. P. B. Zielinski, Water Resources Institute, Clemson University, Clemson, SC 29631
274. 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
275-304. Technical Information Center, Oak Ridge, TN 37831

☆ U.S. GOVERNMENT PRINTING OFFICE: 1987-748-168/60079

END

DATE FILMED

11 / 10 / 87