ORNL--5976

DE84 002980

Contract No. W-7405-eng-26

FEMA: A Finite Element Model of Water Flow Through Aquifers

G. T. Yeh and D. D. Huff

**ENVIRONMENTAL SCIENCES DIVISION
Publication No. 2240**

**NUCLEAR WASTE PROGRAMS
(Activity No. AR 05 15 15 0; ONL-WL09)**

Date Published: November 1983

**OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY**

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.

ACKNOWLEDGMENT

The financial support of this study came from the Office of Defense Waste and By-Product Management, U.S. Department of Energy, under contract W-7405-eng-26 with Union Carbide Corporation.

ABSTRACT

YEH, G. T., and D. D. HUFF. 1983. FEWA: A Finite Element Model of Water Flow Through Aquifers. ORNL-5976. Oak Ridge National Laboratory, Oak Ridge, Tennessee. 222 pp.

This report documents the implementation and demonstration of a Finite Element model of Water flow through Aquifers (FEWA). The particular features of FEWA are its versatility and flexibility to deal with as many real-world problems as possible. Point as well as distributed sources/sinks are included to represent recharges/pumpings and rainfall infiltrations. All sources/sinks can be transient or steady state. Prescribed hydraulic head on the Dirichlet boundaries (usually bordering streams and/or lakes) and fluxes on Neumann or Cauchy boundaries (usually bordering adjacent aquifers) can be time-dependent or constant. Source/sink strength over each element and node, hydraulic head at each Dirichlet boundary node, and flux at each boundary segment can vary independently of each other. Either completely confined or completely unconfined aquifers, or partially confined and partially unconfined aquifers can be dealt with effectively. Discretization of a compound region with very irregular curved boundaries is made easy by including both quadrilateral and triangular elements in the formulation. Large-field problems can be solved efficiently by including a pointwise iterative solution strategy as an optional alternative to the direct elimination solution method for the matrix equation approximating the partial differential equation of groundwater flow. FEWA also includes transient flow through confining leaky aquifers lying above and/or below the aquifer of interest. The model is verified against three simple cases to which analytical solutions are available. It is then demonstrated by two examples of how the model can be applied to heterogeneous and anisotropic aquifers with transient boundary conditions, time-dependent sources/sinks, and confining aquitards for a confined aquifer of variable thickness and for a free surface problem in an unconfined aquifer, respectively.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	v
LIST OF TABLES	ix
LIST OF FIGURES	xi
I. INTRODUCTION	1
II. MATHEMATICAL STATEMENTS	2
III. FINITE ELEMENT APPROXIMATION	16
IV. COMPUTER PROGRAM STRUCTURE	33
V. VERIFICATION AND DEMONSTRATION APPLICATIONS	43
VI. REFERENCES	105
VII. GLOSSARY OF NOTATIONS	107
APPENDIX A: DATA INPUT GUIDE	113
APPENDIX B: OUTPUT	141
APPENDIX C: LISTING OF FORTRAN IV SOURCE PROGRAM	143

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Listing of alternative numerical schemes	29
2	Input data of verification example 1	47
3	Input data of verification example 2	51
4	Input data of verification example 3	55
5	Input data of verification example 3 with triangular element discretization	58
6	Typical material properties for sandstone, silt loam, and clay	65
7	Input data set for demonstration example 1	68
8	Input data set for demonstration example 2	88

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1	Sketch of an arbitrary material volume and its enclosing surface	3
2	Sketch of a region of interest and its three types of boundaries	9
3	Example discretization by 80 nodes and 69 elements (with 59 quadrilateral and 10 triangular elements)	17
4	Relationships between globally defined and element-wise defined basis functions	18
5	Global coordinate vs local coordinate for a typical quadrilateral element	20
6	(a) Three-node linear triangular element e in global coordinates (x,y) , and (b) area and global coordinates of a linear triangular element e	22
7	Program structure of FEWA	35
8	(a) Problem sketch of verification example 1 for flow in a confined aquifer and (b) finite element discretization of verification example 1	44
9	Hydraulic head distribution as predicted by Eq. (104) and FEWA at time equal to 30 minutes (o and —), (b) 5 h (Δ and —), and 10 h ($\square$ and —)	48
10	(a) Problem sketch of verification example 2 for flow in an unconfined aquifer, and (b) finite element discretization of verification example 2	50
11	Hydraulic head distribution as predicted by Eq. (108) (o) and FEWA (—)	52
12	(a) Problem sketch of verification example 3 for flow in an unconfined aquifer, and (b) finite element discretization of verification example 3	53

<u>Figure</u>		<u>Page</u>
13	Hydraulic head distribution as predicted by Eq. (112) (o) and FEWA (—)	56
14	Discretization of verification example 3 with triangular elements	57
15	Problem sketch of demonstration example 1 for flow in a confined aquifer: (a) three-dimensional view and (b) plane view	64
16	Transient water level at the lake and stream	66
17	Normal fluxes imposed on the Neumann boundary	66
18	Operation of the pump at point P	67
19	Finite element discretization of demonstrative example 1	67
20	Distribution of hydraulic head as simulated by FEWA for demonstration example 1 at various times	75
21	Plot of vertically integrated Darcy fluxes as simulated by FEWA for demonstration example 1 at various times	80
22	Problem sketch of demonstration example 2 for flow in an unconfined aquifer: (a) three-dimensional view and (b) plane view	86
23	Profile of net rainfall and evapotranspiration	87
24	Distribution of hydraulic head as simulated by FEWA for demonstration example 2 at various times	95
25	Plot of vertically integrated Darcy fluxes as simulated by FEWA for demonstration example 2 at various times	100

I. INTRODUCTION

To study the transport of dissolved constituents in aquifers, the velocity field as well as the piezometric head distribution therein must be determined first. Numerous models utilizing finite difference and finite element methods have been reported in the literature for simulating the flow field in aquifers (Pinder 1974, Trescott et al. 1976, Bredehoeft and Pinder 1970, Lappala 1980). However, few large computational codes have been designed to handle transient sources/sinks, Dirichlet boundary values, Neumann or Cauchy fluxes, and leaky confining beds (aquitards) simultaneously. In real-world problems, recharge and pumping are likely to be distributed over the region of interest and be time-dependent; the given boundary piezometric heads are, with rare exception, varying with time and along the Dirichlet nodes; the prescribed boundary fluxes are both spatially and temporally distributed over Neumann or Cauchy surfaces; and the head above and below leaky confining beds are definitely changing with time and varying over the region. Some aquifers may be partially confined and partially unconfined. None of the existing codes can effectively deal with the mixed confined-unconfined aquifers. For prototype applications, a computational model must be capable of dealing with these aspects simultaneously. Few existing computer codes, if any, address these situations fully. This report is intended to document the implementation and demonstration of a finite element model designed to treat these complex transient real-world problems. In addition, both quadrilateral and triangular elements, or a combination of the two, are used to facilitate the discretization of a compound region with highly irregular curved boundaries. Also, a pointwise iteration solution strategy for solving the matrix equation approximating the partial differential equation is included as an alternative to the direct solution method. This inclusion makes the model efficient and applicable to large-field problems when the bandwidth of the matrix becomes large.

II. MATHEMATICAL STATEMENTS

A formal mathematical derivation for the transient, vertically integrated flow in the horizontal plane will be presented based on the general three-dimensional saturated-flow equation. We attempt to include all important steps in the development, to account for the assumptions and their bases, and to incorporate boundary conditions as much as possible. Where numerical parameters are needed in the constituent equations resulting from vertical integration, numbers or relationships based on experience are indicated. The model is thus intended to be truly predictive provided boundary conditions and recharging/pumping can be prescribed.

II.1 Three-Dimensional Flows

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). Based on (1) the continuity of fluid, (2) the continuity of solid, (3) the motion of fluid (Darcy's law), (4) the consolidation of media, and (5) the compressibility of water, one can derive a combined governing equation for the distribution of pressure head in the aquifer. The derivation of this combined equation will be described below.

The continuity of fluid is expressed in integral form as:

$$\frac{D}{Dt} \int_V n_e \rho_f dv = - \int_{\Gamma} \vec{n} \cdot (\rho_f \vec{V}_{fs}) d\Gamma + \int_V \rho_f q dv \quad (1)$$

where t is the time; $D()/Dt$ is the material derivative of $()$ with respect to time; n_e is the effective porosity; ρ_f is the fluid density; $\vec{n}$ is a unit vector normal to the surface Γ (Fig. 1); $\vec{V}_{fs}$ is the fluid velocity relative to the solid matrix; v is the volume of material containing constant amount of solids; q is internal source of water; and Γ is the surface enclosing the volume v . It should be noted that the volume v will change with time because of consolidation of the aquifer matrix.

ORNL-DWG 82-18425

Fig. 1. Sketch of an arbitrary material volume and its enclosing surface.

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

$$\int_V \frac{\partial \rho_f n_e}{\partial t} dv + \int_{\Gamma} \vec{n} \cdot (\rho_f n_e \vec{V}_S) d\Gamma + \int_{\Gamma} \vec{n} \cdot (\rho_f \vec{V}_{fs}) d\Gamma = \int_V \rho_f q dv \quad (2)$$

in which $\vec{V}_S$ is the velocity of the deformable surface Γ due to consolidation. Applying the Gaussian divergence theorem to Eq. (2) and using the fact that v is arbitrary, one can obtain the following continuity equation of fluid in differential form:

$$\frac{\partial \rho_f n_e}{\partial t} + \nabla \cdot (\rho_f n_e \vec{V}_S) + \nabla \cdot (\rho_f \vec{V}_{fs}) = \rho_f q \quad (3)$$

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:

$$\frac{D}{Dt} \int_V \rho_s (1-n_e) dv = 0 \quad (4)$$

where ρ_s is the density of solid grains. Again applying the Reynolds' transport theorem to Eq. (4), we obtain:

$$\int_V \frac{\partial(1-n_e)\rho_s}{\partial t} dv + \int_V \vec{n} \cdot [(1-n_e)\rho_s \vec{V}_s] d\Gamma = 0 \quad (5)$$

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

$$\frac{\partial(1-n_e)\rho_s}{\partial t} + \nabla \cdot [(1-n_e)\rho_s \vec{V}_s] = 0 \quad (6)$$

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

$$\frac{\partial \rho_s}{\partial t} + \vec{V}_s \cdot \nabla \rho_s = 0 \quad (7)$$

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

$$\frac{\partial(1-n_e)}{\partial t} + \nabla \cdot [(1-n_e)\vec{V}_s] = 0 \quad (8)$$

Equations (3) and (8) are derived based on the continuity law of fluid and solid grains, with the assumption that the grains are incompressible. These two equations involve four state variables, ρ_f , n_e , $\vec{V}_{fs}$, and $\vec{V}_s$. 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 two constitutive relationships must be empirically established among ρ_f , n_e , $\vec{V}_{fs}$, and $\vec{V}_s$. Alternatively, three of these four state variables may be related empirically or theoretically to a fifth common state variable. The alternative approach is generally adopted since Darcy has empirically proposed a law (Darcy's law) stating that the relative fluid velocity $\vec{V}_{fs}$ is proportional to the gradient of hydraulic head. The proportionality constant is termed the hydraulic conductivity. This empirical law, in fact, can be derived theoretically from the continuity of momentum with the assumptions of neglecting the inertial forces and linearizing the frictional force (Polubarinova-Kochina 1962).

The extension of Darcy's law to an anisotropic media, for $\rho_f = \text{const}$ or $\rho_f = \rho_f(p)$, may be written in the form of:

$$\vec{V}_{fs} = - \bar{K} \cdot \nabla \phi \quad (9)$$

where $\bar{K}$ is the hydraulic conductivity tensor and ϕ is the hydraulic head. The hydraulic conductivity tensor is a function of fluid and medium properties and is defined as

$$\bar{K} = \frac{\rho_f g}{\mu_f} \bar{P} \quad (10)$$

where g is the acceleration of gravity, μ_f is the fluid viscosity, and $\bar{P}$ is the intrinsic permeability tensor of the medium. The hydraulic head ϕ may be split into two parts:

$$\phi = \psi + z \quad (11)$$

in which ψ is the pressure head and z is the potential head. The pressure head is defined as

$$\psi = \frac{p - p_0}{\rho_f g} = \frac{\sigma}{\rho_f g} \quad (12)$$

where p is the pressure; p_0 is a datum pressure; and $\sigma = p - p_0$ is the incremental pressure of the fluid. The pressure head Ψ is the fifth common state variable mentioned earlier.

The consolidation law of the medium is used to relate the solid velocity $\vec{V}_s$ to the pressure head Ψ in this paragraph. The three-dimensional consolidation equation developed by Biot (1940) is

$$(\lambda_s + 2\mu_s)\nabla^2 e = \nabla^2 \sigma \quad (13)$$

where λ_s and μ_s are Lamé constants for the medium and e is the dilation of the medium. The dilatation e and the solid velocity V_s are defined by:

$$e = \nabla \cdot \vec{U} \quad \text{and} \quad \vec{V}_s = \frac{\partial \vec{U}}{\partial t} \quad (14)$$

where $\vec{U}$ is the displacement vector of the medium under consolidation. Taking the divergence of $\vec{V}_s$, and from Eq. (14), we have:

$$\nabla \cdot \vec{V}_s = \frac{\partial}{\partial t}(\nabla \cdot \vec{U}) = \frac{\partial e}{\partial t} \quad (15)$$

Equations (12), (13), and (15) have implicitly established the constitutive relationship between $\nabla \cdot \vec{V}_s$ and the pressure head Ψ . To explicitly express this relationship, we integrate Eq. (13) to yield

$$(\lambda_s + 2\mu_s)e = \sigma + f \quad (16)$$

where f is the integration function of both space and time and must satisfy the Laplace's equation $\nabla^2 f = 0$ for all time. To simplify the matter further, we will only consider vertical consolidation. Under this condition, it has been shown (Verrijt 1969) that the integration function f is equal to 0. It then follows from Eq. (16) that

$$\frac{\partial e}{\partial t} = \alpha \frac{\partial \sigma}{\partial t} \quad (17)$$

where

$$\alpha = \frac{1}{(\lambda_s + 2\mu_s)} \quad (18)$$

in which α is the coefficient of consolidation of the medium. Substituting Eqs. (12) and (17) into (15), we obtain:

$$\nabla \cdot \vec{V}_s = \frac{\partial e}{\partial t} = \alpha \frac{\partial \sigma}{\partial t} = \alpha \rho_f g \frac{\partial \Psi}{\partial t} = \alpha' \frac{\partial \Psi}{\partial t} \quad (19)$$

The compressibility of the fluid, β , is defined by:

$$\beta = \frac{1}{\rho_f} \frac{d\rho_f}{dp} \quad (20)$$

Rewriting Eq. (20) in the following form,

$$d\rho_f = \rho_f \beta dp \quad (21)$$

we will have:

$$\frac{\partial \rho_f}{\partial t} = \rho_f \beta \frac{\partial p}{\partial t} = \rho_f \beta' \frac{\partial \Psi}{\partial t} \quad (22)$$

where

$$\beta' = \rho_f g \beta \quad (23)$$

is the modified compressibility of water.

Equations (3), (8), (9), (19), and (22) [representing the continuity of fluid, the continuity of solids, Darcy's law, the vertical consolidation of the solid matrix, and the compressibility of water, respectively] contain five variables ρ_f , n_e , $\vec{V}_{fs}$, $\vec{V}_s$, and Ψ . 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 five equations into a single one under certain assumptions to simplify the problem. The simplification is demonstrated below.

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

$$n_e \left(\frac{\partial \rho_f}{\partial t} + \vec{V}_s \cdot \nabla \rho_f \right) + \rho_f \left[\frac{\partial n_e}{\partial t} + \nabla \cdot (n_e \vec{V}_s) \right] + \nabla \cdot (\rho_f \vec{V}_{fs}) = \rho_f q \quad (24)$$

and

$$\frac{\partial n_e}{\partial t} + \nabla \cdot (n_e \vec{V}_s) = \nabla \cdot \vec{V}_s \quad (25)$$

respectively. Neglecting the second-order term, $\vec{V}_s \cdot \nabla \rho_f$, and substituting Eq. (25) into (24), we obtain:

$$n_e \frac{\partial \rho_f}{\partial t} + \rho_f \nabla \cdot \vec{V}_s + \nabla \cdot (\rho_f \vec{V}_{fs}) = \rho_f q \quad (26)$$

Substituting Eqs. (9), (19), and (22) into (26), we have:

$$\rho_f n_e \beta' \frac{\partial \Psi}{\partial t} + \rho_f \alpha' \frac{\partial \Psi}{\partial t} = \nabla \cdot (\rho_f \vec{K} \cdot \nabla \Phi) + \rho_f q \quad (27)$$

Expanding the third term of Eq. (27) and neglecting the second-order term $[(\vec{K} \cdot \nabla \Phi) \cdot (\nabla \rho_f)]$, we finally have the following governing equation:

$$S_s \frac{\partial \Psi}{\partial t} = \nabla \cdot (\vec{K} \cdot \nabla \Phi) + q \quad (28)$$

where

$$S_s = \beta' n_e + \alpha' \quad (29)$$

is the specific storage. Typical values of α' and β' are on the order of $3.0 \times 10^{-8} \text{ cm}^{-1}$ and $1.0 \times 10^{-8} \text{ cm}^{-1}$, respectively.

To completely define the problem Eq. (28) must be constrained by initial and boundary conditions. This can be done using physical considerations. Three types of boundary conditions can be prescribed for Eq. (28). The first type is the Dirichlet (specified head) boundary on which the pressure head Ψ is given as:

$$\Psi = \Psi_D \quad \text{on} \quad B_D(x,y,z,t) = 0 \quad (30)$$

where Ψ_D is the Dirichlet head, B_D is part of the boundary $B(x,y,z,t) = 0$ (Fig. 2), and x, y, z are the spatial Cartesian coordinates. The second type is the Neumann (specified gradient flux) boundary on which the gradient flux is given as:

$$-\vec{n} \cdot (\vec{K} \cdot \nabla \Psi) = q_N \quad \text{on} \quad B_N(x,y,z,t) = 0 \quad (31)$$

where q_N is the prescribed Neumann flux and B_N is part of the boundary B (Fig. 2). The third type is the Cauchy (specified flow) boundary on which the total flux is given as:

$$-\vec{n} \cdot (\vec{K} \cdot \nabla \Phi) = q_C \quad \text{on} \quad B_C(x,y,z,t) = 0 \quad (32)$$

where q_C is the Cauchy flux and B_C is part of the boundary B (Fig. 2).

ORNL-DWG 82-13488

Fig. 2. Sketch of a region of interest and its three types of boundaries.

It is assumed that an initial distribution of pressure head can be described to complete the mathematical statement of the problem:

$$\Psi = \Psi_I \quad \text{in} \quad R \quad (33)$$

where R is the region of interest (Fig. 2). The initial pressure head Ψ_I can be obtained from the field measurements or from the simulation of the steady state version of Eq. (28).

If the boundary B is independent of time, the specification of boundary conditions outlined above is relatively easy. However, for many problems, a portion of the boundary is moving with time and a governing equation is required to specify such a kinematic condition as follows:

$$\frac{\partial F}{\partial t} + \vec{U}_F \cdot \nabla F = I \quad (34)$$

where $\vec{U}_F$ is the velocity of the moving surface $F(x,y,z,t) = 0$ and I is the net flux through the moving surface F .

Equations (28) through (34) describe general transient three-dimensional problems in saturated aquifers. Unfortunately, because of the insurmountable difficulties associated with computational time and computer storage requirements, no computer model exists to solve Eqs. (28) through (34) in their general forms. Although a number of three-dimensional or quasi-three-dimensional saturated flow models (Gupta et al. 1975; Trescott 1975, Trescott et al. 1976) have been developed, their applicability to real-world problems under transient conditions is generally limited to confined aquifers because the moving boundary condition, Eq. (34), has not been included.

II.2 Vertically Averaged Flows

In general, flow through a porous medium is transient and three-dimensional and is described by Eqs. (28) through (34). However, since the geometry of most aquifers is such that they are thin in the vertical direction relative to their horizontal dimensions, a simple

approach can be employed. According to this approach, we assume that the flow in the aquifer is essentially horizontal everywhere and the vertical component of the flow may be neglected. The error introduced by this assumption is small in most cases of practical interest. The assumption will fail in the region where the flow has a large vertical component, for example, in the vicinity of partially penetrating wells or outlets in the form of springs. However, even in these cases, at some distance (as a rule of thumb, a distance larger than 1.5 times the thickness of the aquifer at the vicinity) from the source/sink, the assumption of essentially horizontal flow is still valid. The assumption is also applicable to leaky aquifers from the law of refraction of streamlines (Bear 1972). The assumption of essentially horizontal flow is in fact the basis of Dupuit assumption. The vertically integrated equations and variables may then adequately describe the system of the essentially horizontal flows in aquifers.

Let the x - and y -coordinates be on a horizontal plane and the z -coordinate be in the vertical direction. The governing equations are integrated vertically to eliminate the z dependence. Beginning with Eq. (28), we may formally write:

$$\int_{z_0}^{h_s} \left[S_s \frac{\partial \psi}{\partial t} - \nabla \cdot (\bar{K} \cdot \nabla \phi) \right] dz = \int_{z_0}^{h_s} q dz \quad (35)$$

where $z = z_0$ and $z = h_s$ are the lower and upper boundaries, respectively, of the aquifer. Both z_0 and h_s are functions of x and y . In the case of confined aquifers, the boundary will be fixed and thus independent of time. On the other hand, in the case of unconfined aquifers, the upper phreatic boundary is allowed to move freely due to infiltration and pumping or recharges. Under this circumstance, h_s is then a function of time as well.

Using the Leibnitz rule and Dupuit assumption, the integration of Eq. (35) can be carried out for confined and unconfined aquifers, respectively. For confined aquifers, h_s is independent of time, hence we will obtain the following equation after integration:

$$bS_s \frac{\partial h}{\partial t} = \nabla_H \cdot (b\bar{K} \cdot \nabla_H h) - Q_U - Q_L + W \quad (36)$$

where $b = h_s - z_0$ is the aquifer thickness; ∇_H is the Del Operator on the horizontal coordinates, x and y ; Q_U and Q_L are water fluxes through the upper and lower semi-impervious confining layers, respectively; W is the vertically integrated internal source/sink, representing recharges or withdrawals; and h is the hydraulic head given by

$$h = \bar{h}(x, y, t) = \frac{1}{b} \int_{z_0}^{h_s} \phi(x, y, z, t) dz \quad (37)$$

In Eq. (35), W , Q_U , and Q_L are given by:

$$W = \int_{z_0}^{h_s} q dz \quad (38)$$

$$Q_U = \bar{k} \cdot (\bar{K} \cdot \nabla \phi) \Big|_{z=h_s} \quad (39)$$

and

$$Q_L = - \bar{k} \cdot (\bar{K} \cdot \nabla \phi) \Big|_{z=z_0} \quad (40)$$

where $\bar{k}$ is the vertical unit vector. In this report, Q_U and Q_L are approximated by

$$Q_U = K_U (h - h_U) \quad (41)$$

and

$$Q_L = K_L (h - h_L) \quad (42)$$

where K_U and K_L are the leakage coefficients for the upper and lower confining layers; and h_U and h_L are the hydraulic heads above and below upper and lower confining layers, respectively. In fact, K_U and K_L are approximately equal to the hydraulic conductivities of the upper and lower confining layers divided by their corresponding thickness.

For the case of unconfined aquifers, h_s is also a function of time. Thus, the kinematic condition, Eq. (34), can be written as

$$S_y \frac{\partial h_s}{\partial t} = (\bar{K} \cdot \nabla \phi) \Big|_{z=h_s} - \nabla h_s \cdot (\bar{K} \cdot \nabla \phi) \Big|_{z=z_0} + I \quad (43)$$

where S_y is the specific yield which is approximately equal to the effective porosity, n_e . Integrating Eq. (35) and using the kinematic condition, Eq. (43), we will obtain:

$$(bS_s + S_y) \frac{\partial h}{\partial t} = \nabla_H \cdot (b\bar{K} \cdot \nabla_H h) - Q_L + W + I \quad (44)$$

It should be noted that in deriving Eq. (44), we have assumed that $h = h_s$. Equations (36) and (44) can be rewritten as

$$S \frac{\partial h}{\partial t} = \nabla_H \cdot (\bar{T} \cdot \nabla_H h) - Q_U - Q_L + S_w + \sum_{p=1}^{N_p} \delta(x-x_p) \delta(y-y_p) S_p \quad (45)$$

where N_p is the number of point source/sink; S_p is the strength of point source/sink located at (x_p, y_p) ; δ is the delta function;

$$\bar{T} = b\bar{K} \quad \text{and} \quad S_w = W + I \quad (46a)$$

are the transmissivity tensor and the strength of distributed source/sink respectively; and

$$S = bS_s + S_y \quad (46b)$$

is the generalized storage coefficient. In the case of confined aquifers, T is independent of h , while in the case of unconfined aquifers T is a function of h resulting in a nonlinear equation. In Eq. (45), Q_H is zero and S is given by Eq. (46b) for unconfined aquifers, whereas $S_W = W$ and $S = bS_s$ for confined aquifers.

II.3 Initial and Boundary Conditions

The initial and boundary conditions for Eq. (45) can be obtained by integrating Eqs. (33) and (30) through (32), respectively. Starting with Eq. (33), we first add a gravitational head z to both sides of the equation and integrate it vertically to yield:

$$h = h_I(x,y) \quad \text{in} \quad R \quad (47)$$

where h_I is the prescribed initial condition, which can be defined or obtained by solving the steady state version of Eq. (45). Similarly, the integration of Eq. (30) will yield the Dirichlet boundary condition as follows:

$$h = h_D(x,y,t) \quad \text{on} \quad B_D(x,y) = 0 \quad (48)$$

where h_D is the given Dirichlet hydraulic head. However, because of the elimination of z dependence, the integration of Eqs. (31) and (32) will render the identical condition:

$$-\vec{n} \cdot (\vec{T} \cdot \nabla_H h) = q_F \quad \text{on} \quad B_F(x,y) = 0 \quad (49)$$

where $q_F(x,y,t)$ is the prescribed flux resulting from the vertical integration of either q_N or q_C and B_F is the union of B_N and B_C .

Equations (45) and (47) through (49) constitute a general mathematical statement of the physical problem of flow in aquifers. 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 and complex geometries and its simplicity in handling boundary conditions, whereas the FDM has to use regular rectangular system and to interpolate the flux boundary conditions.

III. FINITE ELEMENT APPROXIMATIONS

III.1 Derivation of a Set of Ordinary Differential Equations with FEM

Equations (45) through (49) permit a very large number of numerical approximations. Finite element methods are employed in this report. Because applications of finite element methods to groundwater flow have been fully addressed (Pinder and Gray 1977), the theoretical basis of the method will not be 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 of basis functions, (4) define weighting functions, (5) define a residual, (6) set the residual to zero, (7) derive the coefficient matrix, (8) incorporate boundary conditions into the coefficient matrix, and (9) solve the matrix equation. These procedures are addressed below.

Step 1--The first step is to subdivide the region into an assemblage of M smaller subdomains called elements interconnected by N vertices called nodes. An element can either be quadrilateral or triangular. 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 domain is approximated by interpolating the values at N nodes. Thus, the second step is to define the interpolating functions (basis functions). Let $N_j(x,y)$ be the basis function of nodal point j in the two-dimensional space. The basis function, $N_j(x,y)$, 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, l , and m (Fig. 4). Furthermore, $N_j(x,y)$ 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 82-20016

2 NODE NUMBER
 ⑤ TRIANGULAR ELEMENT
 2 QUADRILATERAL ELEMENT

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

ORNL-DWG 82-10307A

Fig. 4. Relationships between globally defined and element-wise defined basis functions.

In 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:

$$N_3^a = N_j \quad \text{over element a} \quad (50a)$$

$$N_2^b = N_j \quad \text{over element b} \quad (50b)$$

$$N_4^c = N_j \quad \text{over element c} \quad (50c)$$

$$N_1^d = N_j \quad \text{over element d} \quad (50d)$$

where N_3^a , N_2^b , N_4^c , and N_1^d are the basis functions for node 3 of element a, node 2 of element b, node 4 of element c, node 1 of element d, respectively. Similarly, we may obtain

$$N_1^a = N_f, \quad N_2^a = N_i, \quad \text{and} \quad N_4^a = N_g \quad \text{over element a} \quad (50e)$$

$$N_1^b = N_g, \quad N_3^b = N_k, \quad \text{and} \quad N_4^b = N_h \quad \text{over element b} \quad (50f)$$

$$N_1^c = N_i, \quad N_2^c = N_l, \quad \text{and} \quad N_3^c = N_m \quad \text{over element c} \quad (50g)$$

$$N_2^d = N_m \quad \text{and} \quad N_3^d = N_k \quad \text{over element d} \quad (50h)$$

where N_f , N_g , N_h , N_i , N_k , N_l , and N_m are basis functions of nodes f, g, h, i, k, l, and m, respectively; N_1^a , N_2^a , and N_4^a are basis functions for nodes 1, 2, and 4, respectively, of element a; N_1^b , N_3^b , and N_4^b are basis functions for nodes 1, 3, and 4, respectively, of element b; N_1^c , N_2^c , and N_3^c are basis functions for nodes 1, 2, and 3, respectively,

of element c; and N_2^d and N_3^d are basis functions for nodes 2 and 3, respectively, of element d. 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 functions are employed:

$$\begin{aligned} N_1^e &= (1-\xi)(1-\eta)/4, & N_2^e &= (1+\xi)(1-\eta)/4 \\ N_3^e &= (1+\xi)(1+\eta)/4, & N_4^e &= (1-\xi)(1+\eta)/4 \end{aligned} \quad (51)$$

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

$$x = \sum_{j=1}^4 x_j N_j^e(\xi, \eta) \quad \text{and} \quad y = \sum_{j=1}^4 y_j N_j^e(\xi, \eta) \quad (52)$$

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

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

regardless of the shape of the quadrilateral element in the global coordinate system. Since the basis functions are used for coordinate transformation, this element formulation is termed isoparametric.

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

$$N_1^e = L_1, \quad N_2^e = L_2, \quad \text{and} \quad N_3^e = L_3 \quad (53)$$

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

$$L_1 = (a_1 + b_1x + c_1y)/(2A) \quad (54a)$$

$$L_2 = (a_2 + b_2x + c_2y)/(2A) \quad (54b)$$

$$L_3 = (a_3 + b_3x + c_3y)/(2A) \quad (54c)$$

where

$$a_1 = x_2y_3 - x_3y_2, \quad b_1 = y_2 - y_3, \quad c_1 = x_3 - x_2 \quad (54d)$$

$$a_2 = x_3y_1 - x_1y_3, \quad b_2 = y_3 - y_1, \quad c_2 = x_1 - x_3 \quad (54e)$$

$$a_3 = x_1y_2 - x_2y_1, \quad b_3 = y_1 - y_2, \quad c_3 = x_2 - x_1 \quad (54f)$$

and A is the area of the triangle. It is seen from Eqs. (54a) through (54f) that

$$L_1 + L_2 + L_3 = 0 \quad (55)$$

Thus, it is clear that only two of the area coordinates can be independent, just as in the original coordinate system, where there are

ORNL-DWG 32-20017

Fig. 6. (a) Three-node linear triangular element e in global coordinate (x, y) , and (b) area and global coordinate of a linear triangular element e .

only two independent coordinates, x and y . 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. 6b). 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

$$h \approx \hat{h} = \sum_{j=1}^N h_j(t) N_j(x, y) \quad (56)$$

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 (FEM) and finite difference methods (FDM) is that the former is based directly upon approximation of the function as in Eq. (56), whereas the latter is based on approximation of derivatives (Hilderbrand 1968).

Step 4--The fourth step of 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.:

$$W_i(x, y) = N_i(x, y), \quad i = 1, 2, 3, \dots, N \quad (57)$$

where W_i is the weighting function for nodal point i .

Step 5--Since $\hat{h}$ is only an approximate solution of Eq. (45), it will not satisfy the equation. Thus, the fifth step of FEM is to define a residual R_r by:

$$R_r = S \frac{\partial \hat{h}}{\partial t} - \nabla_H \cdot (\mathbf{T} \cdot \nabla_H \hat{h}) + \hat{Q}_U + \hat{Q}_L - S_w - \sum_{p=1}^N S_p \delta(x-x_p) \delta(y-y_p) \quad (58)$$

where

$$\hat{Q}_U = K_U(\hat{h} - h_U) \quad \text{and} \quad \hat{Q}_L = K_L(\hat{h} - h_L) \quad (59)$$

Step 6--According to the principle of FEM, we choose the coefficients h_j 's in Eq. (56) such that the residual in Eq. (58) weighted by the set of weighting functions is zero. In other words, the sixth step of FEM applied to this problem is

$$\begin{aligned} \int_R W_i R_r dR &= \int_R W_i \left\{ \frac{\partial \hat{h}}{\partial t} S - \nabla_H \cdot (\bar{T} \cdot \nabla_H \hat{h}) + \hat{Q}_U + \hat{Q}_L - S_w \right\} dR \\ &- \int_R \sum_{p=1}^P S_p \delta(x-x_p) \delta(y-y_p) dR = 0 \end{aligned}$$

$$\text{for } i = 1, 2, 3, \dots, N \quad (60)$$

Step 7--The seventh step of FEM is to derive a matrix equation based on Eq. (60). This can be done rather easily. Substituting Eqs. (56) and (57) into (60) and interchanging the order of integration and summation, we have:

$$[A] \left\{ \frac{dh}{dt} \right\} + [B] \{h\} = \{D\} + \{Q\} \quad (61)$$

where $\{h\}$ is the column vector representing the values of h at nodes, and the matrix $[A]$ is given by:

$$\begin{aligned} A_{ij} &= \int_R N_i S N_j dR = \sum_{e \in M} \int_{R_e} N_i S N_j dR \\ &= \sum_{e \in M_e} \int_{R_e} N_\alpha^e S N_\beta^e dR \end{aligned} \quad (62)$$

in which R_e is the region of element e , M_e is the set of elements that have a local side α - β coinciding with the global side i - j , and N_α^e and N_β^e are the α -th and β -th local basis functions of element e .

Similarly, the matrix $[B]$ and column vectors $\{D\}$ and $\{Q\}$ are given by

$$\begin{aligned}
 B_{ij} &= \int_R [(\nabla_H N_i) \cdot \bar{T} \cdot (\nabla_H N_j) + N_i (K_U + K_L) N_j] dR \\
 &= \sum_{e \in M} \int_{R_e} [(\nabla_H N_i) \cdot \bar{T} \cdot (\nabla_H N_j) + N_i (K_U + K_L) N_j] dR \\
 &= \sum_{e \in M_e} \int_{R_e} [(\nabla_H N_\alpha^e) \cdot \bar{T} \cdot (\nabla_H N_\beta^e) + N_\alpha^e (K_U + K_L) N_\beta^e] dR \quad (63)
 \end{aligned}$$

$$\begin{aligned}
 D_i &= \int_R N_i [K_U h_U + K_L h_L + S_w + \sum_{p=1} S_p \delta(x-x_p) \delta(y-y_p)] dR \\
 &= \sum_{e \in M} \int_{R_e} N_i [K_U h_U + K_L h_L + S_w] dR + \sum_{p=1} S_p N_i(x_p, y_p) \\
 &= \sum_{e \in M_e} \int_{R_e} N_\alpha^e (K_U h_U + K_L h_L + S_w) dR + S_i \quad (64)
 \end{aligned}$$

and

$$\begin{aligned}
 Q_i &= \int_B N_i \vec{n} \cdot (\bar{T} \cdot \nabla_H h) dB = \sum_{e \in N_s} \int_{B_e} N_i \vec{n} \cdot (\bar{T} \cdot \nabla_H h) dB \\
 &= \sum_{e \in N_{se}} \int_{B_e} N_\alpha^e \vec{n} \cdot (\bar{T} \cdot \nabla_H h) dB \quad (65)
 \end{aligned}$$

where N_s is the set of all global boundary segments; N_{se} is the set of boundary segments that have a local node α coinciding with the global node i . It should be noted that S_i in Eq. (64) is obtained by assuming all S_p 's are located on the nodes.

The reduction of the partial differential equation [Eq. (45)] to the matrix equation [Eq. (61)] boils down to the evaluation of integrals on the right-hand side of the last equal sign in each of the equations Eqs. (62) through (64) for every element e to yield the element matrices $[A^e]$ and $[B^e]$ and the element column vector $\{D^e\}$ as:

$$A_{\alpha\beta}^e = \int_{R_e} N_{\alpha}^e S N_{\beta}^e dR \quad (66a)$$

$$B_{\alpha\beta}^e = \int_{R_e} [(\nabla_H N_{\alpha}^e) \cdot \bar{T} \cdot (\nabla_H N_{\beta}^e) + N_{\alpha}^e (K_U + K_L) N_{\beta}^e] dR \quad (66b)$$

and

$$D_{\alpha}^e = \int_{R_e} N_{\alpha}^e (K_U h_U + K_L h_L + S_w) dR \quad (66c)$$

where the superscript e denotes the element and $\alpha, \beta = 1, 2, 3$, or 4 for linear quadrilateral elements or $\alpha, \beta = 1, 2$, or 3 for linear triangular elements. For a quadrilateral element, Eqs. (66a) through (66c) are computed by Gaussian quadrature (Conte 1965) because it is not easy to solve Equation 52 for ξ and η in terms of x and y . On the other hand, for a triangular element, the transformation from the global coordinate (x, y) to the local coordinate (L_1, L_2, L_3) is already given explicitly by Eqs. (54a) through (54c). This fact and the following integration identity,

$$\int_{R_e} L_1^n L_2^m L_3^k dR = 2A \frac{m! n! k!}{(m+n+k+2)!} \quad (67)$$

make the evaluation of Eqs. (66a) through (66c) for linear triangular element obtainable analytically. With the element matrices $[A^e]$ and $[B^e]$ and the element volume vector $\{D^e\}$ computed, the global matrices

[A] and [B] and the global column vector {D} are then assembled element by element.

Step 8--The matrix equation, Eq. (61), is singular. To make the problem uniquely defined, boundary conditions [Eqs. (48) and (49)], must be implemented. Hence, the eighth step of FEM is the incorporation of boundary conditions into the matrix equation [Eq. (61)]. For the Dirichlet-type boundary conditions, an identity equation is generated for each node and the evaluation of Eq. (65) is not needed. The detailed method of applying this type of boundary condition can be found elsewhere (Yeh and Ward 1980). For the Neumann-type boundary conditions, we simply substitute Eq. (49) into Eq. (65) to yield:

$$Q_i = - \sum_{e \in N_{se}} \int_{B_e} N_{\alpha}^e q_F dB \quad (68a)$$

which is independent of the hydraulic head h , thus becoming the load factor of i -th algebraic equation. It is seen that the global column vector {Q} is obtained by assembling over all flux boundary-sides, the element column vector {Q^e} whose α -th entry is

$$Q_{\alpha}^e = \int_{R_e} N_{\alpha}^e q_F dB \quad (68b)$$

After incorporation of the boundary conditions, Eq. (61) becomes:

$$[M]\left\{\frac{dh}{dt}\right\} + [S]\{h\} = \{Y\} \quad (69)$$

where [M], [S], and {Y} are the modified [A], [B], and {D} + {Q}, respectively.

Step 9--The final step of the FEM is to solve Eq. (69) with the initial conditions [Eq. (47)] as will be described in the following subsection.

III.2. Temporal Integration

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, the 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:

$$[M]({h}_{t+\Delta t} - {h}_t)/\Delta t + \frac{1}{2} [S]({h}_{t+\Delta t} + {h}_t) = \{Y\} \quad (70)$$

where $[M]$, $[S]$, and $\{Y\}$ are evaluated at $(t + \Delta t/2)$. In the second method the backward difference formulation may be written as:

$$[M]({h}_{t+\Delta t} - {h}_t)/\Delta t + [S]{h}_{t+\Delta t} = \{Y\} \quad (71)$$

where $[M]$, $[S]$, and $\{Y\}$ are evaluated at $t + \Delta t$. In the third optional method, the values of the 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:

$$\left(\frac{2}{\Delta t} [M] + [S]\right){h}_{t+\Delta t/2} - \frac{2}{\Delta t} [M]{h}_t = \{Y\} \quad (72a)$$

and

$${h}_{t+\Delta t} = 2{h}_{t+\Delta t/2} - {h}_t \quad (72b)$$

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

Three other optional numerical schemes are presented in this report. These are obtained by lumping the mass matrix $[A]$ in Eq. (62). Referring to $[A]$, one may recall that this is a unit matrix if the finite difference formulation is adopted 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 preferable to use the consistent mass matrix (mass matrix without lumping). Therefore, options are provided for the lumping of matrix [A]. More explicitly, [A] will be lumped according to:

$$A_{ij} = \sum_{e \in M_e} \left(\sum_{\beta=1}^4 \int_{R_e} N_{\alpha}^e S N_{\beta}^e dR \right) \quad \text{if } j = i \quad (73a)$$

and

$$A_{ij} = 0 \quad \text{if } j \neq i \quad (73b)$$

with the mass matrix [A] lumped according to Eqs. (73a) and (73b). The three time-marching schemes (i.e., central difference, backward difference, and mid-difference) are then applied to reduce the set of ordinary differential equations [Eq. (69)] to a set of algebraic equations.

Thus, depending on the method of time marching and the treatment of mass matrix, six alternative numerical schemes are made available in this report (Table 1). For example, scheme 1 uses the central difference time marching with no lumping on the mass matrix.

Table 1. Listing of alternative numerical schemes

	Time marching			Mass matrix	
	Central	Backward	Mid-difference	No lumping	Lumping
1	X			X	
2		X		X	
3	X				X
4		X			X
5			X	X	
6			X		X

Any of the numerical schemes in Table 1 will result in a matrix equation of the following type after initial and boundary conditions are incorporated:

$$[C]\{h\} = \{R\} \quad (74)$$

where $[C]$ is the coefficient matrix and $\{R\}$ is the known load vector. The matrix equation, Eq. (74), is normally solved by direct elimination methods. For large-field problems, the band width of the coefficient matrix is often very large. This makes the direct elimination method inefficient and requires large amounts of computer storage. This report provides the pointwise iterative method as an alternative option to overcome the problem. However, it must be noted that when the aquifer is anisotropic, the finite element method often results in an ill-conditioned matrix (a matrix is ill-conditioned when it is far from diagonally dominant) which, as our experience has indicated, usually makes certain pointwise iterative method e.g., the Jacobian simultaneous displacement, nonconvergent. Hence, appropriate iteration methods, such as the successive over-relaxation used in this report, should be chosen and not be used indiscriminately. A new finite element scheme has been developed that yields a well-conditioned matrix and enables the application of all three most commonly used pointwise iteration methods (Jacobian simultaneous displacement, Gauss-Seidel successive displacement, and successive over-relaxation) to such a matrix even in an anisotropic medium (Yeh 1983).

III.3. Mass Balance Computation

One of the most important aspects in numerical modeling of groundwater 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 $B(x,y) = 0$ can be obtained by integrating Eq. (45):

$$F_V = \int_R \left(S \frac{\partial h}{\partial t} + Q_U + Q_L - S_w \right) dR - \sum_{p=1}^{N_p} S_p \quad (75a)$$

and

$$F_B = - \int_B F_n dB \quad (75b)$$

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 into the region, and F_n is the outward normal flux. In fact, F_n can be defined as:

$$F_n = - \vec{n} \cdot \vec{T} \cdot \nabla_H h \quad (76)$$

Having obtained the hydraulic-head field (h) by solving Eq. (45), one could integrate Eqs. (75a) and (75b) independently. If the solution for h is free of error, one would expect the equality of two integrals. In the present report, the integral in Eq. (75a) is broken into four parts:

$$F_P = \int_R S \frac{\partial h}{\partial t} dR \quad (77a)$$

$$F_{UA} = \int_R Q_U dR \quad (77b)$$

$$F_{LA} = \int_R Q_L dR \quad (77c)$$

and

$$F_S = \int_R S_w dR + \sum_{i=1}^{N_p} S_i \quad (77d)$$

where F_D , F_{UA} , F_{LA} and F_S represent the volumetric rates due to head change, upper leaky bed, lower leaky bed, and artificial sources, respectively. Similarly, the integral in Eq. (75b) is broken into three parts:

$$F_D = - \int_{B_D} F_n \, dB \quad (78a)$$

$$F_F = - \int_{B_F} F_n \, dB \quad (78b)$$

and

$$F_L = - \int_{B-B_D-B_F} F_n \, dB \quad (78c)$$

in which F_D , F_F , and F_L represent fluxes into the region through a Dirichlet boundary, B_D , a flux boundary, B_F , and an unspecified boundary, $B-B_D-B_F$.

For an exact solution, the net influx across the whole boundary, F_B , should equal the total volumetric increase, F_V . In addition, F_L should theoretically be equal to zero. However, in any practical simulation, F_B will not be equal to F_V , 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.

IV. COMPUTER PROGRAM STRUCTURE

IV.1. Purpose of the FEWA Program

The source program of FEWA is designed to solve for the following initial and boundary value problem:

Governing Equation:

$$S \frac{\partial h}{\partial t} = \nabla_H \cdot (\bar{T} \cdot \nabla_H h) - K_U(h-h_U) - K_L(h-h_L) \\ + S_w + \sum_{i=1} S_i \delta(x-x_i) \delta(y-y_i) \quad (79)$$

where S_w is the source/sink of water representing recharge-discharge rate per unit horizontal area ($L^3/T/L^2$).

Initial Condition:

$$h(x,y,0) = h_I(x,y) \quad \text{in } R \quad (80)$$

where (x,z) is the prescribed initial value function, which may also be obtained by solving the steady-state version of Eq. (79) under time-invariant boundary conditions.

Boundary Conditions:

$$h = h_D(x_b, y_b, t) \quad \text{on } B_D \quad (81)$$

and

$$-\vec{n} \cdot (\bar{T} \cdot \nabla_H h) = q_F(x_b, y_b, t) \quad \text{on } B_F \quad (82)$$

where $h_D(x_b, y_b, t)$ is a prescribed hydraulic head as a function of time t and spatial coordinates (x_b, y_b) on the Dirichlet boundary B_D , and $q_F(x_b, y_b, t)$ is the given water flux through the flow-through boundary B_F , also as function of time t and spatial coordinates (x_b, y_b) on B_F . It should be noted that $q_F(x_b, y_b, t)$ is positive if it is directed out from the region of interest and negative if directed into the region.

IV.2 Structure and Function of FEWA Program

To numerically solve the mathematical model defined by Eqs. (79) through (82), the FEWA code consists of a MAIN program and 18 subroutines. These subroutines are named according to their functions. The MAIN program is utilized to specify the dimension of arrays and to assign integer parameters of the array size. It also initializes the arrays. The control and coordinate activity are performed by the subroutine GW2D. Figure 7 shows the structure of the program. The functions of these subroutines are described below.

Subroutine GW2D

The subroutine GW2D 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 the array dimensions throughout all subroutines.

The subroutine will perform either the steady-state computation alone ($KSS = 0$ and $NTI = 0$), or a transient computation using the steady state as the initial condition ($KSS = 0$ and $NTI > 0$), or the transient computation using user-supplied initial conditions ($KSS = 1$ and $NTI > 0$). GW2D calls subroutine DATAIN to read data in; subroutine INTERP to compute sources/sinks and boundary conditions at current time step by interpolating the tabular sources/sinks, Dirichlet head, and fluxes; subroutine SPRDP to obtain the transmissivity tensor based on hydraulic head and hydraulic conductivity in the case of

ORNL-DWG 82-18427

Fig. 7. Program structure of FEWA.

unconfined aquifer; subroutine VELT to compute vertically integrated flux; subroutine ASEMBL to assemble the element matrix over all elements; subroutine BC to implement Dirichlet and flux boundary conditions; subroutine BANSOL to solve the resulting matrix equation; subroutine SFLOW to calculate flux through all types of boundaries and water accumulated in the media; subroutine PRINTT to print out the results; and subroutine STORE to store the flow variables for input to a material transport model or for plotting. These calling sequences are shown in Fig. 7.

Subroutine DATAIN

Subroutine DATAIN reads all data input described in Appendix A except the card group 1. It also prints all the input information and calls subroutine SURF to identify the surface element-sides and boundary nodes.

Subroutine SURF

Subroutine SURF identifies the boundary sides, sequences the boundary nodes, and computes the length and directional cosines of each side. The element numbers associated with the boundary sides are stored in NBE; the boundary nodes are stored in NPB. The length and directional cosines for each side are stored in DLB and DCOSXB and DCOSZB, respectively. The local and global nodal numbers of two nodes of each side are in ISB. The information is returned to the subroutine DATAIN for other users.

Subroutine READR

This subroutine is called by the subroutine DATAIN to generate real numbers for Data Set 6 and Data Set 10 of Appendix A. 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 Set 9, Data Set 12(b), Data Set 13(c), Data Set 14(c),

Data Set 15(c), and Data Set 16(c). As with subroutine READR, an automatic generation mechanism for regularly patterned data is built in this subroutine.

Subroutine INTERP

This subroutine is called by the controlling subroutine GW2D to compute the hydraulic heads of the upper and lower leaking aquifers, respectively, and the sources/sinks, Dirichlet hydraulic head, and prescribed water fluxes if required. The computation is based on the linear interpolation of the input values of these quantities.

Subroutine SPROP

This subroutine is called if and only if the aquifer is unconfined. Under such circumstances, subroutine SPROP is called by GW2D to compute the transmissivity tensor components, AKX(M,KG), AKXZ(M,KG), and AKZ(M,KG), at the Gaussian point KG of the quadrilateral element M, where KG = 1, 2, ... 4 and M = 1, 2, ... NEL (NEL is the number of elements). Or in the case of triangular elements, it computes these components at the three vertices of the element.

Subroutine VELT

This subroutine calls Q4D or Q3D to evaluate the element matrix. It then sums over all element matrices to form a matrix equation governing the vertically integrated flux components at all nodal points. Subsequently, it calls subroutine BANSOL to yield the solution. The computed field of the vertically integrated flux is then returned to GW2D through the argument.

Subroutines Q4D and Q3D

Subroutines Q4D and Q3D are called by the Subroutine VELT to compute the element matrix given by:

$$QQ(I,J) = \int_{R_e} N_i^e N_j^e dR \quad (83a)$$

where N_i^e and N_j^e are the base functions for nodal points i and j , respectively. They also evaluate the element load vector given by:

$$RQ(I) = - \int_{R_e} N_i^e \vec{e}_x \cdot (\vec{T} \cdot \nabla_H N_j^e) h_j dR \quad (83b)$$

or

$$RQ(I) = - \int_{R_e} N_i^e \vec{e}_y \cdot (\vec{T} \cdot \nabla_H N_j^e) h_j dR \quad (83c)$$

where h_j is the hydraulic head at nodal point j , and $\vec{e}_x$ and $\vec{e}_y$ are the unit vectors along the x - and y - coordinates, respectively. The computation of these quantities is straightforward. Q4D calculates the integrations in Eqs. (83a) and (83b) or (83c) for quadrilateral elements and Q3D evaluates those integrations for triangular elements.

Subroutine BASE

This subroutine is called by Q4D, Q4TH, and Q4 to evaluate the value of the base functions and their derivatives with respect to the local coordinates at a Gaussian point.

Subroutine BANSOL

This subroutine is called by VELT and GW2D to solve for the matrix equation of the type

$$[C]\{x\} = \{y\} \quad (84)$$

where $[C]$ is a 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 $\{x\}$ and stores it in $\{y\}$. The computation is a standard Gaussian elimination procedure.

Subroutine ASEMBL

This subroutine calls Q4 or Q3 to evaluate the element matrix. It then sums over all elements to form a matrix equation governing the hydraulic head at all nodal points.

Subroutines Q4 and Q3

Subroutines Q4 and Q3 are called by ASEMBL to compute the element matrix given by:

$$QA(I,J) = \int_{R_e} N_i^e S N_j^e dR \quad (85a)$$

and

$$QB(I,J) = \int_{R_e} [(\nabla_H N_i^e) \cdot \bar{T} \cdot (\nabla_H N_j^e) + N_i^e (K_U + K_L) N_j^e] dR \quad (85b)$$

Q4 is to compute Eqs. (85a and (85b) for quadrilateral elements, while Q3 is to compute those for triangular elements. They also calculate the load vector given by:

$$RQ(I) = \int_{R_e} N_i^e (S_w + \gamma_U h_U + K_L h_L) N_j^e dR \quad (86)$$

Subroutine BC

This subroutine evaluates both Dirichlet and flux 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 the flux boundary, the integration of the boundary flux is added to the load vector.

Subroutine PRINTT

This subroutine is used to line-print the flow variables. These include the fluxes through variable boundary segments, the hydraulic head, and vertically integrated flux 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, FEMA. The information stored includes TITLE, NPROB, NNP, NEL, NBN, NBEL, NTI, NHUPR, NHLPR, X, Z, B, IE, DLB, DCOSXB, DCOSZB, NBE, ISB, NPB, RKU, IHUTYP, RKL, IHLTYP, and TIME, H, VX, VZ, HU, HL for each desired time step.

Subroutine SFLOW

This subroutine is used to compute the flow rates through various types of boundaries and the increasing rate of water in the region of interest. FRATE(6) stores the flow rate through the whole boundary enclosing the region of interest. It is given by:

$$\text{FRATE}(6) = \int_B (V_x n_x + V_z n_z) dB \quad (87)$$

in which B is the global boundary of the region of interest, V_x and V_z are the vertically integrated flux components on the boundary in the x- and z-direction, respectively, and n_x and n_z are the directional cosines of the outward unit vector normal to the boundary B. FRATE(1) and FRATE(2) store the flow rates through Dirichlet boundary B_D and flux boundary B_F , respectively, given by:

$$\text{FRATE}(1) = \int_{B_D} (V_x n_x + V_z n_z) dB \quad (88)$$

and

$$\text{FRATE}(2) = \int_{B_F} (V_x n_x + V_z n_z) dB \quad (89)$$

FRATE(3) and FRATE(4) store the flow rates going out of the region through the upper and lower confining leaky beds, respectively, given by:

$$\text{FRATE}(3) = - \int_R K_U (h_U - h) dR \quad (90)$$

and

$$\text{FRATE}(4) = - \int_R K_L (h_L - h) dR \quad (91)$$

FRATE(5), which is related to the numerical loss, is given by:

$$\text{FRATE}(5) = \text{FRATE}(6) - [\text{FRATE}(1) + \text{FRATE}(2)] \quad (92)$$

FRATE(7) and is used to store an increased rate of water within media and FRATE(8) to store source rate, respectively:

$$\text{FRATE}(7) = \int_R S \frac{\partial h}{\partial t} dR \quad (93)$$

and

$$\text{FRATE}(8) = - \int_R S_w dR - \sum_{i=1} S_i \quad (94)$$

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

$$\text{FRATE}(7) = - [\text{FRATE}(6) + \text{FRATE}(3) + \text{FRATE}(4) + \text{FRATE}(8)] \quad (95)$$

and FRATE(5) should be equal to zero.

Equation (95) simply states that the negative rate of water going out from the region through the entire boundary, due to a source/sink and because of leakage, is equal to the rate of water accumulated in the region. The rate of water going out from the region through the

entire boundary is FRATE(6), the rate due to the artificial source/sink is FRATE(8), leakage through the upper leaky bed is FRATE(3), and leakage through the lower leaky bed is FRATE(4). The rate of water accumulated in the region is represented by FRATE(7).

Subroutines Q4TH and Q3TH

These two subroutines are used to compute the contributions to FRATE(3), FRATE(4), FRATE(7), and FRATE(8) by quadrilateral and triangular elements, respectively:

$$UPLEKM = \int_{R_e} K_U (h_U - h) dR \quad (96)$$

$$DNLEKM = \int_{R_e} K_L (h_L - h) dR \quad (97)$$

$$QTHM = \int_{R_e} S(h - h_p) dR \quad (98)$$

and

$$SOUSM = \int_{R_e} S_w dR \quad (99)$$

where h_p is the hydraulic head at the previous time step and h is the hydraulic head at the present time step. The computation of the above integrations is straightforward.

V. VERIFICATIONS AND DEMONSTRATION APPLICATIONS

Five examples are presented in this section, three of which are used to verify the accuracy of the computer code FEWA. The other two are used to demonstrate how FEWA can be applied to heterogeneous and anisotropic media under transient Dirichlet and Neumann boundary conditions and with time-dependent sources/sinks for both confined and unconfined aquifers. Three verification examples are: (1) transient simulation in a confined aquifer without sources/sinks or leakage, (2) steady-state simulation in an unconfined aquifer with sources/sinks but without leakage, and (3) steady-state computation in an unconfined aquifer with both sources/sinks and leakage.

V.1 Example 1

Consider a confined aquifer with a thickness of $b = 10$ m (Fig. 8a). Assume the aquifer is filled with a porous medium which has a compressibility of skeleton of $\alpha' = 10^{-5} \text{ m}^{-1}$ and a compressibility of water of $\beta' = 4 \times 10^{-6} \text{ m}^{-1}$. The hydraulic conductivity of the medium is $K = 4.4 \cdot 10^{-7} \text{ m/s}$, and the effective porosity is $n_e = 0.25$. This confined aquifer would then have a generalized storage coefficient of $S = 1.1 \cdot 10^{-4}$ and a transmissivity of $T = 4.4 \times 10^{-6} \text{ m}^2/\text{s}$. Let us further assume that no leaking to or from the underlying or overlying aquifer would take place. This assumption can be satisfied by specifying both K_U and K_L equal to zero. We will also assume that the internal source/sink is zero. Initially the aquifer is observed to have a piezometric head $h = H = 25$ m (Fig. 8a). If one starts to pump $2Q = 4.4 \times 10^{-6} \text{ m}^3/\text{s}$ of water per unit width perpendicular to x - z plane at $x = 0$, what will the piezometric head distribution be along x -axis at any time?

ORNL-DWG 82-18421

Fig. 8. (a) Problem sketch of verification example 1 for flow in a confined aquifer, and (b) finite element discretization of verification example 1.

The above problem can be described by the following set of equations:

Governing Equation:

$$S \frac{\partial h}{\partial t} = T \frac{\partial^2 h}{\partial x^2} \quad (100)$$

Initial Conditions:

$$h = H \quad \text{at} \quad t = 0 \quad \text{and} \quad 0 < x < \infty \quad (101)$$

Boundary Conditions:

$$h = H \quad \text{at} \quad t > 0 \quad \text{and} \quad x = L \quad (102)$$

and

$$T \frac{\partial h}{\partial t} = Q \quad \text{at} \quad t > 0 \quad \text{and} \quad x = 0 \quad (103)$$

Equations (100) through (103) are in fact the special cases of Eqs. (45) and (47) through (49), respectively. It can be verified that the solution of Eq. (100) subject to initial and boundary conditions, Eqs. (101) through (103), is given by (Carslaw and Jaeger 1959) as:

$$h = H - \frac{2Q}{T} \left\{ \left(\frac{Tt}{S\pi} \right)^{1/2} \cdot \exp\left[-\frac{Sx^2}{4\pi Tt}\right] - \frac{x}{2} \operatorname{erfc}\left(\frac{\sqrt{Sx}}{2\sqrt{Tt}}\right) \right\} \quad (104)$$

To compare the FEWA calculation with that obtained by Eq. (104), a total of approximately 10.64 h real-time simulation is made. With this real time, a length of approximately $L = 298.2$ m (Fig. 8a) can be considered infinite. To start the simulation, the region is divided into 98 elements and 198 nodes (Fig. 8b). The smallest element (Element No. 1 in Fig. 8b) has a length equal to 1.0 m, and each of

the 97 subsequent elements is increased by a factor of 1.02 over its previous element. Thus, the longest element (Element No. 98 in Fig. 8b) has a length equal to $1.0 \cdot (1.0 + 0.02)^{97} = 6.83$ m. To minimize the band width of the coefficient matrix, the nodes are numbered to proceed most rapidly along the direction that contains the least number of nodes. A variable time step size is used. The initial time step size is 60 s and each subsequent step size is increased by 10% with a constraint that the maximum step size is 600 s. It can be computed that a total number of time steps required for 10.64 h (to be exact 3.8301×10^4 s) is 79 (method of computing the number of time steps is given in the footnote of Data Set 3 in Appendix A).

With the above problem description, the input data for this case can be coded according to Appendix A. The complete input data set is given in Table 2. The output is given in Appendix B. Figure 9 shows the head distribution as simulated by FEWA and Eq. (99) at time equal to 30 min, 5 h, and 10 h. They show almost perfect agreement between FEWA prediction and analytical solution. In fact, computer printout indicated that the maximum error in the drawdown (= initial head - hydraulic head) as predicted by FEWA is less than two tenths of a percent.

Table 2. Input data of verification example 1

```

1 VERIFICATION, COMPINED, NO SOURCE, NO LEAKY, TRANSIENT
----- DATA SET 2: INTEGER PARANTERS -----
198 98 1 0 79 1 1 0 0 40 0 0 1 1
----- DATA SET 3: BASIC REAL PARAMETERS -----
60.0D0 0.10D0 600.0D0 9999999.0 0.000001 0.000001 1000.0D0 9.8D00
1.3D-3 1.0D0 1.0D0
----- DATA SET 4: PRINTER, DISK STORE AND TIME RESETTING CONTROL -----
333 1 1 1 1 1 3 1 13 1 1 1 3 1 1 1 3 1 1 1 3 1 1 1 1 3 3
111 111 111
1.0D20
----- DATA SET 5: MATERIAL PROPERTIES -----
1.0D-5 4.0D-6 0.25D0 4.4D-7 0.0D00 0.0D00 0.0D00
----- DATA SET 6: AQUIFER THICKNESS -----
1 197 1 10.0D0 0.0
----- DATA SET 7: COORDINATE -----
1 0 2 0.0 1.0D0 0.0D0
3 97 2 1.0D00 1.00D0 0.02D0
2 0 2 0.0 1.0D0 0.0D0
4 97 2 1.0D00 1.00D0 0.02D0

1 98 2 0.0 0.0 0.0
2 98 2 1.0D0 0.0 0.0
----- DATA SET 8: ELEMENT INCIDENCES -----
1 1 3 4 2 1 1 98
----- DATA SET 10: INITIAL CONDITIONS -----
1 197 1 25.00D0 0.000D0 0.0
----- DATA SET 11: TRANSIENT INTEGER PARAMETERS -----
1 2 1 2 2 1 2 1 2 1 2 1 2
----- DATA SET 12: ELEMENT WIDE SOURCE/SINK -----
0.0 0.0 1.0D20 0.0
1 97 1 1 0
----- DATA SET 13: DIRICHLET BOUNDARY CONDITIONS -----
0.0 25.0D0 1.0D20 25.0D0
197 198
1 1 1 1 0
----- DATA SET 14: NEUMANN BOUNDARY CONDITIONS -----
0.0 2.2D-6 1.0D20 2.2D-6
1 2
1 1 1 1 0
1 0 1 2 0 0
----- DATA SET 15: UPPER LEAKY AQUIFER -----
0.0 0.0 1.0D20 0.0
0.0
1 97 1 1 0
----- DATA SET 16: LOWER LEAKY AQUIFER -----
0.0 0.0 1.0D20 0.0
0.0
1 97 1 1 0

```

Fig. 9. Hydraulic head distribution as predicted by Eq. (104) and FEWA at time equal to 30 min (o and —), 5 h (Δ and —), and 10 h ($\square$ and —), respectively.

V.2 Example 2

As a second verification example, consider a strip of parallel rivers (Fig. 10a). Although the rivers do not penetrate fully the aquifer, we will assume that at $x = 0$ and $x = L$, we have vertical equipotential, $h_0 = 5$ m and $h_L = 10$ m, respectively. The hydraulic conductivity of the aquifer is assumed to be $K = 10^{-5}$ m/s, and a uniform source $S_w = 10^{-8}$ m/s is infiltrating through the ground surface to the aquifer. We will further assume that the bottom of the aquifer is impervious, i.e., $K_L = 0$. The question being asked is: what is the distribution of piezometric head along the x -axis under steady-state conditions?

The problem can be posed as a special case of Eqs. (45) and (47) through (49) as follows:

Governing Equation:

$$K \frac{\partial}{\partial x} \left(h \frac{\partial h}{\partial x} \right) + S_w = 0 \quad (105)$$

Boundary Conditions:

$$h = h_0 \quad \text{at} \quad x = 0 \quad (106)$$

and

$$h = h_L \quad \text{at} \quad x = L \quad (107)$$

Eqs. (105) can be verified to have the following solution (Bear 1977):

$$h = \{ h_0^2 + \frac{x}{L} (h_L^2 - h_0^2) + \frac{S_w}{K} x(L - x) \}^{1/2} \quad (108)$$

For FEWA simulation, the region is discretized by 98 elements and 198 nodes (Fig. 10b). Each element has a length of 4 m. With this discretization, the input data can now be prepared according to Appendix A. The complete input data set is given in Table 3. The output is given in Appendix B. Since only the steady-state simulation is desired, both KSS and NTI in Data Set 2 should be set to 0.

ORNL-DWG 82-18429

Fig. 10. (a) Problem sketch of verification example 2 for flow in an unconfined aquifer, and (b) finite element discretization of verification example 2.

Table 3. Input data of verification example 2

```

2 VERIFICATION, UNCONFINED AQUIFER, WITH SOURCES BUT WITHOUT LEAKING
----- DATA SET 2: INTEGER PARAMETERS -----
198 98 1 0 0 0 1 0 0 40 0 0 0 1
----- DATA SET 3: BASIC REAL PARAMETERS -----
60.0 0.1 600.0 95.999.0 0.000001 0.000001 1000.0 9.8
1.3D-3 1.0 1.0
----- DATA SET 4: PRINTER, DISK STORE AND TIME RESETING CONTROL -----
33
11
1.0D20
----- DATA SET 5: MATERIAL PROPERTIES -----
0.0 0.0 0.25D0 1.0D-5 0.0 0.0 0.25D0
----- DATA SET 6: BOTTOM ELEVATION -----
1 197 1 0.0 0.0

----- DATA SET 7: COORDINATE -----
1 98 2 0.0 4.0D0 0.00D0
2 98 2 0.0 4.0D0 0.00D0

1 98 2 0.0 0.0 0.0
2 98 2 1.0D0 0.0 0.0

----- DATA SET 8: ELEMENT INCIDENCES -----
1 1 3 4 2 -1 1 98

----- DATA SET 10: INITIAL CONDITIONS -----
1 47 2 5.0D00 0.2D00 0.0
97 50 2 14.4D00 -0.088D0 0.0
2 47 2 5.0D00 0.2D00 0.0
98 50 2 14.4D00 -0.088D0 0.0

----- DATA SET 11: TRANSIENT INTEGER PARAMETERS -----
0 0 0 0 4 2 2 1 2 1 2 1 2
----- DATA SET 12: ELEMENT WIDE SOURCE/SINK -----
0.0 1.0D-8 1.0D20 1.0D-8
1 97 1 1 0

----- DATA SET 13: DIRICHLET BOUNDARY CONDITIONS -----
0.0 5.0D0 1.0D20 5.0D0
0.0 10.0D0 1.0D20 10.0D0
1 2 197 198
1 1 1 1 0
3 1 1 2 0

----- DATA SET 15: UPPER LEAKY AQUIFER -----
0.0 0.0 1.0D20 0.0
0.0
1 97 1 1 0

----- DATA SET 16: LOWER LEAKY AQUIFER -----
0.0 0.0 1.0D20 0.0
0.0
1 97 1 1 0

```

Furthermore, DELT, CHNG, DELMAX, TMAX, and TOLB in Data Set 3 can be anything because they are not used due to the steady-state calculation. Figure 11 shows the comparison of head distribution as computed by Eq. (108) and FEWA. Almost perfect agreement is obtained. Computer printout revealed that the FEWA simulation deviated from the analytical solution after the fourth digit.

Fig. 11. Hydraulic head distribution as predicted by Eq. (108) (o) and FEWA (—), respectively.

V.3 Example 3

As a third verification example, consider an aquifer which overlies a thin layer of aquitard (Fig. 12a) which is impervious to the left of $x = 0$ and is semipervious to the right of $x = 0$ with a leakage coefficient $(K_L) = 10^{-7} \text{ s}^{-1}$. On the ground surface a constant recharge rate of $(S_w) = 2.5 \times 10^{-7} \text{ m/s}$ is added to the aquifer. Assume it is observed at $x = 0$, that the piezometric head is constant at $h = h_0 = 20 \text{ m}$ in reference to the bottom of the aquifer (Fig. 12a). We will further assume that below the semipervious layer, a suction head of $h = h_L = -10 \text{ m}$ is maintained. The question is: what is the distribution of hydraulic head along the x -axis under steady-state conditions?

Fig. 12. (a) Problem sketch of verification example 3 for flow in an unconfined aquifer, and (b) finite element discretization of verification example 3.

The problem can be stated mathematically as follows:

Governing Equation:

$$\frac{\partial}{\partial x} \left(Kh \frac{\partial h}{\partial x} \right) - K_L (h - h_L) + S_w = 0 \quad (109)$$

Boundary Conditions:

$$h = h_0 \quad \text{at} \quad x = 0 \quad (110)$$

and

$$- Kh \frac{\partial h}{\partial x} = 0 \quad \text{at} \quad x = L \quad (111)$$

with the additional requirement that $h = 0$ at $x = L$ (Bear 1977). Equation (109) is obtained from Eq. (45) by deleting the transient term on its left-hand side and by specifying $K_y = 0$. Equations (110) and (111) are special cases of Eqs. (48) and (49), respectively. The analytical solution of Eqs. (109) through (111) is given by Bear (1977):

$$h = h_0 + \frac{Ax^2}{6} - \left(\frac{2}{3} Ah_0 + B \right)^{1/2} x \quad (112)$$

where

$$A = K_L/K \quad \text{and} \quad B = -(K_L h_L + S_w)/K \quad (113)$$

It can be shown that, with the additional requirement that $h = 0$ at $x = L$, L is given by the following formula:

$$L = 3 \left[\left(\frac{2}{3} Ah_0 + B \right)^{1/2} - B^{1/2} \right] \quad (114)$$

With the hydraulic properties and the source strength given in the example, L can be computed to be approximately equal to 54.772 m.

Table 4. Input data of verification example 3

```

3A. VERIFICATION, UNCONFINED AQUIFER, WITH SOURCE & LEAKY, QUADRILATERAL
----- DATA SET 2: INTEGER PARAMETERS -----
112 55 1 0 0 0 1 0 0 40 0 0 0 1
----- DATA SET 3: BASIC REAL PARAMETERS -----
60.0 0.1 600.0 9999999.0 0.000001 0.000001 1000.0 9.8
1.3D-3 1.0 1.0
----- DATA SET 4: PRINTER, DISK STORE AND TIME RESETING CONTROL -----
33
11
1.0D20
----- DATA SET 5: MATERIAL PROPERTIES -----
0.0 0.0 0.25D0 1.0D-5 0.0 0.0 0.25D0
----- DATA SET 6: BOTTOM ELEVATION -----
1 111 1 0.0 0.0

----- DATA SET 7: COORDINATE -----
1 54 2 0.0 1.0D0 0.0
112 0 1 54.772D00 0.0D0 0.0
2 54 2 0.0 1.0D0 0.0
111 0 1 54.772D00 0.0D0 0.0

1 55 2 0.0 0.0 0.0
2 55 2 1.0D0 0.0 0.0

----- DATA SET 8: ELEMENT INCIDENCES -----
1 1 3 4 2 -1 1 55

----- DATA SET 10: INITIAL CONDITIONS -----
1 54 2 20.0D0 -0.365D0 0.0
109 0 1 0.0 0.0 0.0
2 54 2 20.0D0 -0.365D0 0.0
110 0 1 0.0 0.0 0.0

----- DATA SET 11: TRANSIENT INTEGER PARAMETERS -----
1 2 1 2 2 1 2 1 2 1 2 1 2

----- DATA SET 12: ELEMENT WIDE SOURCE/SINK -----
0.0 2.5D-7 1.0D20 2.5D-7
1 54 1 1 0

----- DATA SET 13: DIRICHLET BOUNDARY CONDITIONS -----
0.0 20.0D0 1.0D20 20.0D0
1 2
1 1 1 1 0

----- DATA SET 14: NEUMANN BOUNDARY CONDITIONS -----
0.0 0.0 1.0D20 0.0
111 112
1 1 1 1 0

1 0 1 2 0 0

----- DATA SET 15: UPPER LEAKY AQUIFER -----
0.0 0.0 1.0D20 0.0
0.0
1 54 1 1 0

----- DATA SET 16: LOWER LEAKY AQUIFER -----
0.0 -10.0D0 1.0D20 -10.0D0
1.0D-7
1 54 1 1 0

```

For FEWA simulation, the region is discretized by 55 elements and 112 nodes (Fig. 12b). Each element has a length of 1.0 m except the element number 55 which has a length of 0.772 m. With this discretization, the input data are prepared according to Appendix A and are given in Table 4. The output is again in Appendix B. Figure 13 shows the comparison of head distribution as simulated by Eq. (112) and by FEWA. Again, almost perfect agreement is obtained. The agreement is to the fourth digit.

Up to this point, only the quadrilateral elements are used to verify the internal consistency of the FEWA computer program. To test how all subroutines involved in the computations with triangular elements behave, we discretize this example with 220 triangular elements

Fig. 13. Hydraulic head distribution as predicted by Eq. (112) (o) and FEWA (--), respectively.

and 167 nodes as given in Fig. 14. With this discretization, the input data coded according to Appendix A are given in Table 5. The output is again given in Appendix B. Practically identical results are obtained for two different discretizations. Thus, it can be stated with reasonable confidence that computations involving triangular elements in FEWA are internally consistent. In the remainder of this chapter, we will demonstrate how FEWA can be applied to realistic prototype problems with two hypothetical examples that are thought typical of the real world.

Fig. 14. Discretization of verification example 3 with triangular elements.

Table 5. Input data of verification example 3 with triangular element discretization.

```

3B. VERIFICATION, UNCONFINED AQUIFER, VITE SOURCE AND LEAKY, TRIANGULAR
----- DATA SET 2: INTEGER PARAMETERS -----
167 220 1 0 0 0 1 0 0 40 0 0 0 1
----- DATA SET 3: BASIC REAL PARAMETERS -----
60.0 0.1 600.0 9999999.0 0.000001 0.000001 1000.0 9.8
1.3D-3 1.0 1.0
----- DATA SET 4: PRINTER, DISK STORE AND TIME RESETTING CONTROL -----
33
11
1.0D20
----- DATA SET 5: MATERIAL PROPERTIES -----
0.0 0.0 0.2500 1.0D-5 0.0 0.0 0.2500
----- DATA SET 6: BOTTOM ELEVATION -----
1 166 1 0.0 0.0

----- DATA SET 7: COORDINATE -----
1 54 3 0.0 1.0D0 0.0
166 0 1 54.772D00 0.0D0 0.0
2 54 3 0.0 1.0D0 0.0
167 0 1 54.772D00 0.0D0 0.0
3 53 3 0.50D0 1.0D0 0.0
166 0 1 54.386D0 0.0D0 0.0

1 55 3 0.0 0.0 0.0
2 55 3 1.0D0 0.0 0.0
3 54 3 0.5D0 0.0 0.0

----- DATA SET 8: ELEMENT INCIDENCES
1 1 3 2 0 -1
2 1 4 3 0 -1
3 3 5 2 0 -1
4 3 8 5 0 -1
5 4 6 5 0 -1
6 4 7 6 0 -1
7 6 8 5 0 -1
8 6 7 8 0 -1
9 7 9 8 0 -1
10 7 10 9 0 -1
11 9 11 8 0 -1
12 9 10 11 0 -1
13 10 12 11 0 -1
14 10 13 12 0 -1
15 12 14 11 0 -1
16 12 13 14 0 -1
17 13 15 14 0 -1
18 13 16 15 0 -1
19 15 17 14 0 -1
20 15 16 17 0 -1
21 16 18 17 0 -1
22 16 19 18 0 -1
23 18 20 17 0 -1
24 18 19 20 0 -1
25 19 21 20 0 -1
26 19 22 21 0 -1
27 21 23 20 0 -1
28 21 22 23 0 -1
29 22 24 23 0 -1
30 22 25 24 0 -1
31 24 26 23 0 -1
32 24 25 26 0 -1
33 25 27 26 0 -1
34 25 28 27 0 -1
35 27 29 26 0 -1
36 27 28 29 0 -1
37 28 30 29 0 -1
38 28 31 30 0 -1
39 30 32 29 0 -1
40 30 31 32 0 -1

```

Table 5. (continued)

41	31	33	32	0	-1
42	31	34	33	0	-1
43	33	35	32	0	-1
44	33	34	35	0	-1
45	34	36	35	0	-1
46	34	37	36	0	-1
47	36	38	35	0	-1
48	36	37	38	0	-1
49	37	39	38	0	-1
50	37	40	39	0	-1
51	39	41	38	0	-1
52	39	40	41	0	-1
53	40	42	41	0	-1
54	40	43	42	0	-1
55	42	44	41	0	-1
56	42	43	44	0	-1
57	43	45	44	0	-1
58	43	46	45	0	-1
59	45	47	44	0	-1
60	45	46	47	0	-1
61	46	48	47	0	-1
62	46	49	48	0	-1
63	48	50	47	0	-1
64	48	49	50	0	-1
65	49	51	50	0	-1
66	49	52	51	0	-1
67	51	53	50	0	-1
68	51	52	53	0	-1
69	52	54	53	0	-1
70	52	55	54	0	-1
71	54	56	53	0	-1
72	54	55	56	0	-1
73	55	57	56	0	-1
74	55	58	57	0	-1
75	57	59	56	0	-1
76	57	58	59	0	-1
77	58	60	59	0	-1
78	58	61	60	0	-1
79	60	62	59	0	-1
80	60	61	62	0	-1
81	61	63	62	0	-1
82	61	64	63	0	-1
83	63	65	62	0	-1
84	63	64	65	0	-1
85	64	66	65	0	-1
86	64	67	66	0	-1
87	66	68	65	0	-1
88	66	67	68	0	-1
89	67	69	68	0	-1
90	67	70	69	0	-1
91	69	71	68	0	-1
92	69	70	71	0	-1
93	70	72	71	0	-1
94	70	73	72	0	-1
95	72	74	71	0	-1
96	72	73	74	0	-1
97	73	75	74	0	-1
98	73	76	75	0	-1
99	75	77	74	0	-1
100	75	76	77	0	-1
101	76	78	77	0	-1
102	76	79	78	0	-1
103	78	80	77	0	-1
104	78	79	80	0	-1
105	79	81	80	0	-1
106	79	82	81	0	-1
107	81	83	80	0	-1
108	81	82	83	0	-1
109	82	84	83	0	-1
110	82	85	84	0	-1

Table 5. (continued)

111	84	86	83	0	-1
112	84	85	86	0	-1
113	85	87	86	0	-1
114	85	88	87	0	-1
115	87	89	86	0	-1
116	87	88	89	0	-1
117	88	90	89	0	-1
118	88	91	90	0	-1
119	90	92	89	0	-1
120	90	91	92	0	-1
121	91	93	92	0	-1
122	91	94	93	0	-1
123	93	95	92	0	-1
124	93	94	95	0	-1
125	94	96	95	0	-1
126	94	97	96	0	-1
127	96	98	95	0	-1
128	96	97	98	0	-1
129	97	99	98	0	-1
130	97	100	99	0	-1
131	99	101	98	0	-1
132	99	100	101	0	-1
133	100	102	101	0	-1
134	100	103	102	0	-1
135	102	104	101	0	-1
136	102	103	104	0	-1
137	103	105	104	0	-1
138	103	106	105	0	-1
139	105	107	104	0	-1
140	105	106	107	0	-1
141	106	108	107	0	-1
142	106	109	108	0	-1
143	108	110	107	0	-1
144	108	109	110	0	-1
145	109	111	110	0	-1
146	109	112	111	0	-1
147	111	113	110	0	-1
148	111	112	113	0	-1
149	112	114	113	0	-1
150	112	115	114	0	-1
151	114	116	113	0	-1
152	114	115	116	0	-1
153	115	117	116	0	-1
154	115	118	117	0	-1
155	117	119	116	0	-1
156	117	118	119	0	-1
157	118	120	119	0	-1
158	118	121	120	0	-1
159	120	122	119	0	-1
160	120	121	122	0	-1
161	121	123	122	0	-1
162	121	124	123	0	-1
163	123	125	122	0	-1
164	123	124	125	0	-1
165	124	126	125	0	-1
166	124	127	126	0	-1
167	126	128	125	0	-1
168	126	127	128	0	-1
169	127	129	128	0	-1
170	127	130	129	0	-1
171	129	131	128	0	-1
172	129	130	131	0	-1
173	130	132	131	0	-1
174	130	133	132	0	-1
175	132	134	131	0	-1
176	132	133	134	0	-1
177	133	135	134	0	-1
178	133	136	135	0	-1
179	135	137	134	0	-1
180	135	136	137	0	-1

Table 5. (continued)

181	136	138	137	0	-1
182	136	139	138	0	-1
183	138	140	137	0	-1
184	138	139	140	0	-1
185	139	141	140	0	-1
186	139	142	141	0	-1
187	141	143	140	0	-1
188	141	142	143	0	-1
189	142	144	143	0	-1
190	142	145	144	0	-1
191	144	146	143	0	-1
192	144	145	146	0	-1
193	145	147	146	0	-1
194	145	148	147	0	-1
195	147	149	146	0	-1
196	147	148	149	0	-1
197	148	150	149	0	-1
198	148	151	150	0	-1
199	150	152	149	0	-1
200	150	151	152	0	-1
201	151	153	152	0	-1
202	151	154	153	0	-1
203	153	155	152	0	-1
204	153	154	155	0	-1
205	154	156	155	0	-1
206	154	157	156	0	-1
207	156	158	155	0	-1
208	156	157	158	0	-1
209	157	159	158	0	-1
210	157	160	159	0	-1
211	159	161	158	0	-1
212	159	160	161	0	-1
213	160	162	161	0	-1
214	160	163	162	0	-1
215	162	164	161	0	-1
216	162	163	164	0	-1
217	163	165	164	0	-1
218	163	166	165	0	-1
219	165	167	164	0	-1
220	165	166	167	0	-1

----- DATA SET 10: INITIAL CONDITIONS -----

1	54	3	20.0D0	-0.365D0	0.0
163	0	1	0.0	0.0	0.0
2	54	3	20.0D0	-0.365D0	0.0
164	0	1	0.0	0.0	0.0
3	53	3	19.635D00	-0.365D0	0.0
165	0	1	0.0	0.0	0.0

----- DATA SET 11: TRANSIENT INTEGER PARAMETERS -----

1	2	1	2	2	1	2	1	2	1	2	1	2
---	---	---	---	---	---	---	---	---	---	---	---	---

----- DATA SET 12: SOURCES/SINKS -----

0.0	2.5D-7	1.0D20	2.5D-7	
1	219	1	1	0

----- DATA SET 13: DIRICHLET BOUNDARY CONDITIONS -----

0.0	20.0D0	1.0D20	20.0D0	
1	2			
1	1	1	1	0

Table 5. (continued)

----- DATA SET 14: NEUMANN BOUNDARY CONDITIONS -----

0.0		0.0		1.0D20	0.0
166	167				
1	1	1	1	0	
1	0	1	2	0	0

----- DATA SET 15: UPPER LEAKY AQUIFER -----

0.0		0.0		1.0D20	0.0
1.0					
1	219	1	1	0	

----- DATA SET 16: LOWER LEAKY AQUIFER -----

0.0		-10.0D0		1.0D20	-10.0D0
1.0D-7					
1	219	1	1	0	

V.4 Demonstration Example 1

To demonstrate the versatility of FEWA, consider a confined aquifer bounded below and above by two aquitards (Fig. 15a). Other stratigraphic formations are also present (Fig. 15a). This aquifer is bound on the right side by a lake and on the left side by a groundwater divide (Figs. 15a and b). It is further assumed that the aquifer is bound on the back by an aquifuge (impervious formation) and on the front by another aquifer (Fig. 15b). A stream is flowing from the front and curves into the lake (Figs. 15a and 15b). We will assume that the confined aquifer is variable in thickness and has a thickness of 20 m at point A, 19 m at points D and C, and 18 m at point B (Fig. 15b). The aquifer is made of three formations, one of which is pervious, while the other two portions (one on EFGH and the other on IJKL) are less pervious. We will also assume that intermittent pumping occurs at location P. Our task is to analyze the response of this confined aquifer to the changing conditions of: (1) fluctuating lake levels, (2) temporally varying recharge from its adjacent groundwater basin through the interface AB (Fig. 15b), (3) transient leakage from the stream and through upper and lower confining layers, and (4) pumping activity at point P with time-dependent withdrawals.

To employ FEWA for investigating the behavior of the aquifer, the properties of the three formations must be obtained. Typical values for Hygiene sandstone, silt loam, and the Bait Net clay (van Genuchten 1980) will be used and are given in Table 6. The upper and lower confining beds will be given the following leakage coefficients: $K_U = 0.0 \text{ s}^{-1}$ to the left of the stream and $K_U = 5.0 \times 10^{-7} \text{ s}^{-1}$ to the right of the stream, and $K_L = 10^{-7} \text{ s}^{-1}$. We will assume that the hydraulic head in the aquifer below the lower aquitard is a constant $h_L = 50 \text{ m}$ in reference to the bottom of the lake, and the hydraulic head in the aquifer above the upper aquitard has a transient head varying linearly between the heads of the river and the lake. For demonstration purposes, a simulation of 6 d will be performed. Assume that the water levels at the lake and in the stream are the same and are time dependent during these 6 d as given in Fig. 16. The water level is measured from the datum at the bottom of the lake.

Fig. 15. Problem sketch of demonstration example 1 for flow in a confined aquifer: (a) three-dimensional view and (b) plane view.

Table 6. Typical material properties for sandstone, silt loam, and clay

Properties Medium	α' (m^{-1})	β' (m^{-1})	n_e (dimensionless)	K_{xx} (m/sec)	K_{xy} (m/sec)	K_{yy} (m/sec)	S_y (dimensionless)
ABCD- (EFGH + IJKL) Sandstone	10^{-5}	3×10^{-6}	0.10	25×10^{-5}	0.625×10^{-5}	1.25×10^{-5}	0.10
EFGH Silt loam	2×10^{-5}	3×10^{-6}	0.27	5.74×10^{-7}	2.97×10^{-7}	5.74×10^{-7}	0.27
IJKL Clay	4×10^{-5}	3×10^{-6}	0.45	9.49×10^{-9}	4.75×10^{-9}	9.49×10^{-9}	0.45

This implies that the transient head h_u above the upper aquitard is also given by Fig. 16. Two different profiles or normal fluxes (Fig. 17) are imposed on the Neumann boundary AB (Fig. 15b): profile no. 1 to the left of the river and profile no. 2 to the right of and including the river. The amount of water withdrawn by pumping at point P is assumed time-dependent and is given in Fig. 18.

With the above problem description, we are now ready to encode the input data according to Appendix A for FEWA simulation. First we must divide the region into elements and nodes. For this demonstration, we discretize the region with 200 elements and 231 nodes with the maximum grid size Δs on the order of 10 m (Fig. 19). Elements 38, 48, and 58 are composed of silt loam (material type 2). Elements 93, 103, and 114 are composed of clays (material type 3). The rest of the region is made of sandstone (material type 1). The pump is located at node 83. The input data set for this discretization is given in Table 7. The output is given in Appendix B. Figures 20 and 21 depict the distribution of hydraulic head and the vertically integrated Darcy fluxes, respectively. These figures clearly show the development of a depression core around the pumping well as expected. Furthermore, the expansion and shrinkage of the core correspond roughly to the increasing and decreasing of the pumping rate with a time delay.

Fig. 16. Transient water level at the lake and stream.

Fig. 17. Normal fluxes imposed on the Neumann boundary.

Fig. 18. Operation of the pump at point P.

Table 7. Input data set for the demonstration example 1

```

      4      INPUT FOR DEMONSTRATION EXAMPLE 1
      DATA SET 2: BASIC INTEGER PARAMETERS
231 200      3      6 208      0      1      0      1      0      0      1      1      0
      DATA SET 3: BASIC REAL PARAMETERS
      1.8D03      0.0D00      1.8D03      5.184D05      1.0D-4      1.0D-5      1.0D03      9.8D00
      1.3D-3      1.0D00      1.0D00
      DATA SET 4: LINEPRINTER, DISKOUTPUT, AND TIME STEP SETTING CONTROL
33          131          131          131 131
              131          131          131          131
31          131
1          1          1          1          1          1
              1          1          1          1
1          1
      1.0D50
      DATA SET 5: MATERIAL PROPERTIES
      1.0D-5      3.0D-6      0.1D00      1.25D-5      1.25D-5      0.631-5      0.0D00
      2.0D-5      3.0D-6      0.27D07      5.74D-7      5.74D-7      2.971-7      0.0D00
      4.0D-5      3.0D-6      0.45D00      9.49D-9      9.49D-9      4.751-9      0.0D00
      DATA SET 6: AQUIFER THICKNESS
      1      1      1      20.0000      -0.1000      0.0
      3      1      1      19.8000      -0.1000      0.0
      5      1      1      19.6000      -0.1000      0.0
      7      1      1      19.4000      -0.1000      0.0
      9      1      1      19.2000      -0.1000      0.0
      11     0      1      19.0000      0.0          0.0
      12     1      1      19.9000      -0.0900      0.0
      14     1      1      19.7200      -0.0900      0.0
      16     1      1      19.5400      -0.0900      0.0
      18     1      1      19.3600      -0.0900      0.0
      20     1      1      19.1800      -0.0900      0.0
      22     0      1      19.0000      0.0          0.0
      23     1      1      19.8000      -0.0800      0.0
      25     1      1      19.6400      -0.0800      0.0
      27     1      1      19.4800      -0.0800      0.0
      29     1      1      19.3200      -0.0800      0.0
      31     1      1      19.1600      -0.0800      0.0
      33     0      1      19.0000      0.0          0.0
      34     1      1      19.7000      -0.0700      0.0
      36     1      1      19.5600      -0.0700      0.0
      38     1      1      19.4200      -0.0700      0.0
      40     1      1      19.2800      -0.0700      0.0
      42     1      1      19.1400      -0.0700      0.0
      44     0      1      19.0000      0.0          0.0
      45     1      1      19.6000      -0.0600      0.0
      47     1      1      19.4800      -0.0600      0.0
      49     1      1      19.3600      -0.0600      0.0
      51     1      1      19.2400      -0.0600      0.0
      53     1      1      19.1200      -0.0600      0.0
      55     0      1      19.0000      0.0          0.0
      56     1      1      19.5000      -0.0500      0.0
      58     1      1      19.4000      -0.0500      0.0
      60     1      1      19.3000      -0.0500      0.0
      62     1      1      19.2000      -0.0500      0.0
      64     1      1      19.1000      -0.0500      0.0
      66     0      1      19.0000      0.0          0.0
      67     1      1      19.4000      -0.0400      0.0
      69     1      1      19.3200      -0.0400      0.0
      71     1      1      19.2400      -0.0400      0.0
      73     1      1      19.1600      -0.0400      0.0
      75     1      1      19.0800      -0.0400      0.0
      77     0      1      19.0000      0.0          0.0
      78     1      1      19.3000      -0.0300      0.0
      80     1      1      19.2400      -0.0300      0.0
      82     1      1      19.1800      -0.0300      0.0
      84     1      1      19.1200      -0.0300      0.0
      86     1      1      19.0600      -0.0300      0.0
      88     0      1      19.0000      0.0          0.0

```

Table 7. (continued)

89	1	1	19.2000	-0.0200	0.0
91	1	1	19.1600	-0.0200	0.0
93	1	1	19.1200	-0.0200	0.0
95	1	1	19.0800	-0.0200	0.0
97	1	1	19.0400	-0.0200	0.0
99	0	1	19.0000	0.0	0.0
100	1	1	19.1000	-0.0100	0.0
102	1	1	19.0800	-0.0100	0.0
104	1	1	19.0600	-0.0100	0.0
106	1	1	19.0400	-0.0100	0.0
108	1	1	19.0200	-0.0100	0.0
110	0	1	19.0000	0.0	0.0
111	1	1	19.0000	0.0	0.0
113	1	1	19.0000	0.0	0.0
115	1	1	19.0000	0.0	0.0
117	1	1	19.0000	0.0	0.0
119	1	1	19.0000	0.0	0.0
121	0	1	19.0000	0.0	0.0
122	1	1	18.9800	0.0100	0.0
124	1	1	18.9200	0.0100	0.0
126	1	1	18.9400	0.0100	0.0
128	1	1	18.9600	0.0100	0.0
130	1	1	18.9800	0.0100	0.0
132	0	1	19.0000	0.0	0.0
133	1	1	18.8000	0.0200	0.0
135	1	1	18.8400	0.0200	0.0
137	1	1	18.8800	0.0200	0.0
139	1	1	18.9200	0.0200	0.0
141	1	1	18.9600	0.0200	0.0
143	0	1	19.0000	0.0	0.0
144	1	1	18.7000	0.0300	0.0
146	1	1	18.7600	0.0300	0.0
148	1	1	18.8200	0.0300	0.0
150	1	1	18.8800	0.0300	0.0
152	1	1	18.9400	0.0300	0.0
154	0	1	19.0000	0.0	0.0
155	1	1	18.6000	0.0400	0.0
157	1	1	18.6800	0.0400	0.0
159	1	1	18.7600	0.0400	0.0
161	1	1	18.8400	0.0400	0.0
163	1	1	18.9200	0.0400	0.0
165	0	1	19.0000	0.0	0.0
166	1	1	18.5000	0.0500	0.0
168	1	1	18.6000	0.0500	0.0
170	1	1	18.7000	0.0500	0.0
172	1	1	18.8000	0.0500	0.0
174	1	1	18.9000	0.0500	0.0
176	0	1	19.0000	0.0	0.0
177	1	1	18.4000	0.0600	0.0
179	1	1	18.5200	0.0600	0.0
181	1	1	18.6400	0.0600	0.0
183	1	1	18.7600	0.0600	0.0
185	1	1	18.8800	0.0600	0.0
187	0	1	19.0000	0.0	0.0
188	1	1	18.3000	0.0700	0.0
190	1	1	18.4400	0.0700	0.0
192	1	1	18.5800	0.0700	0.0
194	1	1	18.7200	0.0700	0.0
196	1	1	18.8600	0.0700	0.0
198	0	1	19.0000	0.0	0.0
199	1	1	18.2000	0.0800	0.0
201	1	1	18.3600	0.0800	0.0
203	1	1	18.5200	0.0800	0.0
205	1	1	18.6800	0.0800	0.0
207	1	1	18.8400	0.0800	0.0
209	0	1	19.0000	0.0	0.0
210	1	1	18.1000	0.0900	0.0
212	1	1	18.2800	0.0900	0.0
214	1	1	18.4600	0.0900	0.0
216	1	1	18.6400	0.0900	0.0
218	1	1	18.8200	0.0900	0.0

Table 7. (continued)

220	0	1	19.0000	0.0	0.0
221	1	1	16.0000	0.1000	0.0
223	1	1	16.2000	0.1000	0.0
225	1	1	16.4000	0.1000	0.0
227	1	1	18.6000	0.1000	0.0
229	1	1	18.8000	0.1000	0.0
231	0	1	19.0000	0.0	0.0
DATA SET 7: MOCE COORDINATE					
1	1	1	0.0	0.0	0.0
3	1	1	0.0	0.0	0.0
5	1	1	0.0	0.0	0.0
7	1	1	0.0	0.0	0.0
9	1	1	0.0	0.0	0.0
11	0	1	0.0	0.0	0.0
12	1	1	10.0000	-0.1000	0.0
14	1	1	9.8000	-0.1000	0.0
16	1	1	9.6000	-0.1000	0.0
18	1	1	9.4000	-0.1000	0.0
20	1	1	9.2000	-0.1000	0.0
22	0	1	9.0000	0.0	0.0
23	1	1	20.0000	-0.2000	0.0
25	1	1	19.6000	-0.2000	0.0
27	1	1	19.2000	-0.2000	0.0
29	1	1	18.8000	-0.2000	0.0
31	1	1	18.4000	-0.2000	0.0
33	0	1	18.0000	0.0	0.0
34	1	1	30.0000	-0.3000	0.0
36	1	1	29.4000	-0.3000	0.0
38	1	1	28.8000	-0.3000	0.0
40	1	1	28.2000	-0.3000	0.0
42	1	1	27.6000	-0.3000	0.0
44	0	1	27.0000	0.0	0.0
45	1	1	40.0000	-0.4000	0.0
47	1	1	39.2000	-0.4000	0.0
49	1	1	38.4000	-0.4000	0.0
51	1	1	37.6000	-0.4000	0.0
53	1	1	36.8000	-0.4000	0.0
55	0	1	36.0000	0.0	0.0
56	1	1	50.0000	-0.5000	0.0
58	1	1	49.0000	-0.5000	0.0
60	1	1	48.0000	-0.5000	0.0
62	1	1	47.0000	-0.5000	0.0
64	1	1	46.0000	-0.5000	0.0
66	0	1	45.0000	0.0	0.0
67	1	1	60.0000	-0.6000	0.0
69	1	1	58.8000	-0.6000	0.0
71	1	1	57.6000	-0.6000	0.0
73	1	1	56.4000	-0.6000	0.0
75	1	1	55.2000	-0.6000	0.0
77	0	1	54.0000	0.0	0.0
78	1	1	70.0000	-0.7000	0.0
80	1	1	68.6000	-0.7000	0.0
82	1	1	67.2000	-0.7000	0.0
84	1	1	65.8000	-0.7000	0.0
86	1	1	64.4000	-0.7000	0.0
88	0	1	63.0000	0.0	0.0
89	1	1	80.0000	-0.8000	0.0
91	1	1	78.4000	-0.8000	0.0
93	1	1	76.8000	-0.8000	0.0
95	1	1	75.2000	-0.8000	0.0
97	1	1	73.6000	-0.8000	0.0
99	0	1	72.0000	0.0	0.0
100	1	1	90.0000	-0.9000	0.0
102	1	1	88.2000	-0.9000	0.0
104	1	1	86.4000	-0.9000	0.0
106	1	1	84.6000	-0.9000	0.0
108	1	1	82.8000	-0.9000	0.0
110	0	1	81.0000	0.0	0.0

Table 7. (continued)

111	1	1	100.0000	-1.0000	0.0
113	1	1	98.0000	-1.0000	0.0
115	1	1	96.0000	-1.0000	0.0
117	1	1	94.0000	-1.0000	0.0
119	1	1	92.0000	-1.0000	0.0
121	0	1	90.0000	0.0	0.0
122	1	1	110.0000	-1.1000	0.0
124	1	1	107.8000	-1.1000	0.0
126	1	1	105.6000	-1.1000	0.0
128	1	1	103.4000	-1.1000	0.0
130	1	1	101.2000	-1.1000	0.0
132	0	1	99.0000	0.0	0.0
133	1	1	120.0000	-1.2000	0.0
135	1	1	117.6000	-1.2000	0.0
137	1	1	115.2000	-1.2000	0.0
139	1	1	112.8000	-1.2000	0.0
141	1	1	110.4000	-1.2000	0.0
143	0	1	108.0000	0.0	0.0
144	1	1	130.0000	-1.3000	0.0
146	1	1	127.4000	-1.3000	0.0
148	1	1	124.8000	-1.3000	0.0
150	1	1	122.2000	-1.3000	0.0
152	1	1	119.6000	-1.3000	0.0
154	0	1	117.0000	0.0	0.0
155	1	1	140.0000	-1.4000	0.0
157	1	1	137.2000	-1.4000	0.0
159	1	1	134.4000	-1.4000	0.0
161	1	1	131.6000	-1.4000	0.0
163	1	1	128.8000	-1.4000	0.0
165	0	1	126.0000	0.0	0.0
166	1	1	150.0000	-1.5000	0.0
168	1	1	147.0000	-1.5000	0.0
170	1	1	144.0000	-1.5000	0.0
172	1	1	141.0000	-1.5000	0.0
174	1	1	138.0000	-1.5000	0.0
176	0	1	135.0000	0.0	0.0
177	1	1	160.0000	-1.6000	0.0
179	1	1	156.8000	-1.6000	0.0
181	1	1	153.6000	-1.6000	0.0
183	1	1	150.4000	-1.6000	0.0
185	1	1	147.2000	-1.6000	0.0
187	0	1	144.0000	0.0	0.0
188	1	1	170.0000	-1.7000	0.0
190	1	1	166.6000	-1.7000	0.0
192	1	1	163.2000	-1.7000	0.0
194	1	1	159.8000	-1.7000	0.0
196	1	1	156.4000	-1.7000	0.0
198	0	1	153.0000	0.0	0.0
199	1	1	180.0000	-1.8000	0.0
201	1	1	176.4000	-1.8000	0.0
203	1	1	172.8000	-1.8000	0.0
205	1	1	169.2000	-1.8000	0.0
207	1	1	165.6000	-1.8000	0.0
209	0	1	162.0000	0.0	0.0
210	1	1	190.0000	-1.9000	0.0
212	1	1	186.2000	-1.9000	0.0
214	1	1	182.4000	-1.9000	0.0
216	1	1	178.6000	-1.9000	0.0
218	1	1	174.8000	-1.9000	0.0
220	0	1	171.0000	0.0	0.0
221	1	1	200.0000	-2.0000	0.0
223	1	1	196.0000	-2.0000	0.0
225	1	1	192.0000	-2.0000	0.0
227	1	1	188.0000	-2.0000	0.0
229	1	1	184.0000	-2.0000	0.0
231	0	1	180.0000	0.0	0.0

Table 7. (continued)

1	1	1	0.0	5.5000	0.0
3	1	1	11.0000	5.5000	0.0
5	1	1	22.0000	5.5000	0.0
7	1	1	33.0000	5.5000	0.0
9	1	1	44.0000	5.5000	0.0
11	0	1	55.0000	0.0	0.0
12	1	1	0.2500	5.4750	0.0
14	1	1	11.2000	5.4750	0.0
16	1	1	22.1500	5.4750	0.0
18	1	1	33.1000	5.4750	0.0
20	1	1	44.0500	5.4750	0.0
22	0	1	55.0000	0.0	0.0
23	1	1	0.5000	5.4500	0.0
25	1	1	11.4000	5.4500	0.0
27	1	1	22.3000	5.4500	0.0
29	1	1	33.2000	5.4500	0.0
31	1	1	44.1000	5.4500	0.0
33	0	1	55.0000	0.0	0.0
34	1	1	0.7500	5.4250	0.0
36	1	1	11.6000	5.4250	0.0
38	1	1	22.4500	5.4250	0.0
40	1	1	33.3000	5.4250	0.0
42	1	1	44.1500	5.4250	0.0
44	0	1	55.0000	0.0	0.0
45	1	1	1.0000	5.4000	0.0
47	1	1	11.8000	5.4000	0.0
49	1	1	22.6000	5.4000	0.0
51	1	1	33.4000	5.4000	0.0
53	1	1	44.2000	5.4000	0.0
55	0	1	55.0000	0.0	0.0
56	1	1	1.2500	5.3750	0.0
58	1	1	12.0000	5.3750	0.0
60	1	1	22.7500	5.3750	0.0
62	1	1	33.5000	5.3750	0.0
64	0	1	44.2500	5.3750	0.0
66	0	1	55.0000	0.0	0.0
67	1	1	1.5000	5.3500	0.0
69	1	1	12.2000	5.3500	0.0
71	1	1	22.9000	5.3500	0.0
73	1	1	33.6000	5.3500	0.0
75	1	1	44.3000	5.3500	0.0
77	0	1	55.0000	0.0	0.0
78	1	1	1.7500	5.3250	0.0
80	1	1	12.4000	5.3250	0.0
82	1	1	23.0500	5.3250	0.0
84	1	1	33.7000	5.3250	0.0
86	1	1	44.3500	5.3250	0.0
88	0	1	55.0000	0.0	0.0
89	1	1	2.0000	5.3000	0.0
91	1	1	12.6000	5.3000	0.0
93	1	1	23.2000	5.3000	0.0
95	1	1	33.8000	5.3000	0.0
97	1	1	44.4000	5.3000	0.0
99	0	1	55.0000	0.0	0.0
100	1	1	2.2500	5.2750	0.0
102	1	1	12.8000	5.2750	0.0
104	1	1	23.3500	5.2750	0.0
106	1	1	33.9000	5.2750	0.0
108	1	1	44.4500	5.2750	0.0
110	0	1	55.0000	0.0	0.0
111	1	1	2.5000	5.2500	0.0
113	1	1	13.0000	5.2500	0.0
115	1	1	23.5000	5.2500	0.0
117	1	1	34.0000	5.2500	0.0
119	1	1	44.5000	5.2500	0.0
121	0	1	55.0000	0.0	0.0
122	1	1	2.7500	5.2250	0.0
124	1	1	13.2000	5.2250	0.0
126	1	1	23.6500	5.2250	0.0
128	1	1	34.1000	5.2250	0.0
130	1	1	44.5500	5.2250	0.0

Table 7. (continued)

132	0	1	55.0000	0.0	0.0
133	1	1	3.0000	5.2000	0.0
135	1	1	13.4000	5.2000	0.0
137	1	1	23.8000	5.2000	0.0
139	1	1	34.2000	5.2000	0.0
141	1	1	44.6000	5.2000	0.0
143	0	1	55.0000	0.0	0.0
144	1	1	3.2500	5.1750	0.0
146	1	1	13.6000	5.1750	0.0
148	1	1	23.9500	5.1750	0.0
150	1	1	34.3000	5.1750	0.0
152	1	1	44.6500	5.1750	0.0
154	0	1	55.0000	0.0	0.0
155	1	1	3.5000	5.1500	0.0
157	1	1	13.8000	5.1500	0.0
159	1	1	24.1000	5.1500	0.0
161	1	1	34.4000	5.1500	0.0
163	1	1	44.7000	5.1500	0.0
165	0	1	55.0000	0.0	0.0
166	1	1	3.7500	5.1250	0.0
168	1	1	14.0000	5.1250	0.0
170	1	1	24.2500	5.1250	0.0
172	1	1	34.5000	5.1250	0.0
174	1	1	44.7500	5.1250	0.0
176	0	1	55.0000	0.0	0.0
177	1	1	4.0000	5.1000	0.0
179	1	1	14.2000	5.1000	0.0
181	1	1	24.4000	5.1000	0.0
183	1	1	34.6000	5.1000	0.0
185	1	1	44.8000	5.1000	0.0
187	0	1	55.0000	0.0	0.0
188	1	1	4.2500	5.0750	0.0
190	1	1	14.4000	5.0750	0.0
192	1	1	24.5500	5.0750	0.0
194	1	1	34.7000	5.0750	0.0
196	1	1	44.8500	5.0750	0.0
198	0	1	55.0000	0.0	0.0
199	1	1	4.5000	5.0500	0.0
201	1	1	14.6000	5.0500	0.0
203	1	1	24.7000	5.0500	0.0
205	1	1	34.8000	5.0500	0.0
207	1	1	44.9000	5.0500	0.0
209	0	1	55.0000	0.0	0.0
210	1	1	4.7500	5.0250	0.0
212	1	1	14.8000	5.0250	0.0
214	1	1	24.8500	5.0250	0.0
216	1	1	34.9000	5.0250	0.0
218	1	1	44.9500	5.0250	0.0
220	0	1	55.0000	0.0	0.0
221	1	1	5.0000	5.0000	0.0
223	1	1	15.0000	5.0000	0.0
225	1	1	25.0000	5.0000	0.0
227	1	1	35.0000	5.0000	0.0
229	1	1	45.0000	5.0000	0.0
231	0	1	55.0000	0.0	0.0

Table 7. (continued)

DATA SET 8: ELEMENT INCIDENCES									
1	1	12	13	2	1	10	20		
DATA SET 9: MATERIAL CORRECTIONS FOR REGIONS IN ZFGH AND IJKL									
38	2	10	2	0					
93	2	10	3	0					
DATA SET 10: PRE-INITIAL CONDITIONS									
1	230	1		5.0D01	0.0D0	0.0D0			
DATA SET 11: TRANSIENT INTEGRAL PARAMETERS									
19	20	2	8	11	1	8	1	2	2
DATA SET 12: SOURCES/SINKS									
	0.0D00	0.0D00		1.0D50	0.0D00				
1	199	1	1	0					
	0.0D00	-1.0D-3	2.592D04	-5.0D-4	5.184D04	-2.0D-3	7.776D04	-1.0D-3	
1.728D05		-1.0D-3	2.160D05	-5.0D-4	2.592D05	-1.0D-3	5.184D05	-1.0D-3	
83									
1	0	1	1	0					
DATA SET 13: DIRICHLET BOUNDARY CONDITIONS									
	0.0D00	5.0D01	2.592D04	5.1D01	5.184D04	4.9D01	7.776D04	5.0D01	
1.728D05		5.0D01	2.160D05	5.05D01	2.592D05	5.0D01	5.184D05	5.0D01	
221	222	223	224	225	226	227	228	229	230
1	10	1	1	0					
DATA SET 14: FLUX BOUNDARY CONDITIONS									
	0.0D00	-2.0D-6	7.776D04	-2.0D-6	1.296D05	-4.0D-6	1.728D05	-2.0D-6	
2.956D05		-2.0D-6	3.456D05	-2.0D-6	4.320D05	-2.0D-6	5.184D05	-2.0D-6	
	0.0D00	-2.0D-6	8.640D04	-1.5D-6	1.296D05	-2.0D-6	1.728D05	-1.6D-6	
2.952D05		-1.55D-6	3.456D05	-1.5D-6	4.320D05	-1.75D-6	5.184D05	-2.0D-6	
1	12	23	34	45	56	67	78	89	100
177	188	199	210						
1	13	1	1	0					
15	5	1	2	0					
1	18	1	2	1	1				
DATA SET 15: UPPER LEAKING AQUIFER									
	0.0D00	5.0D01	2.592D04	5.1D01	5.184D04	4.9D01	7.776D04	5.0D01	
1.728D05		5.0D01	2.160D05	5.05D01	2.592D05	5.0D01	5.184D05	5.0D01	
	0.0D00	5.0D01	2.592D04	5.1D01	5.184D04	4.9D01	7.776D04	5.0D01	
1.728D05		5.0D01	2.160D05	5.05D01	2.592D05	5.0D01	5.184D05	5.0D01	
5.0D-7		0.0D0							
1	139	1	2	0					
141	6	1	1	0					
151	6	1	1	0					
161	6	1	1	0					
171	6	1	1	0					
181	6	1	1	0					
191	6	1	1	0					
148	5	10	2	0					
149	5	10	2	0					
150	5	10	2	0					
DATA SET 16: LOWER LEAKING AQUIFERS									
	0.0D00	5.0D01	5.184D05	5.0D01					
	0.0D00								
1	199	1	1	0

Fig. 20. Distribution of hydraulic head as simulated by FEWA for demonstration example 1 at time (a) 0.0, (b) 0.3, (c) 0.6, (d) 0.9, (e) 1.0, (f) 2.0, (g) 2.5, (h) 3.0, (i) 4.0, and (j) 5.0 d, respectively.

Fig. 20. Continued.

Fig. 20. Continued.

Fig. 20. Continued.

Fig. 20. Continued.

(a) TIME = 0.0 d

(b) TIME = 0.3 d

Fig. 21. Plot of vertically integrated Darcy fluxes as simulated by FEWA for demonstration example 1 at time (a) 0, (b) 0.3, (c) 0.6, (d) 0.9, (e) 1.0, (f) 2.0, (g) 2.5, (h) 3.0, (i) 4.0, and (j) 5.0 d, respectively.

Fig. 21. Continued.

Fig. 21. Continued.

ORNL-DWG 83-1771

(A) TIME = 3.0 d

ORNL-DWG 83-1770

(g) TIME = 2.5 d

Fig. 21. Continued.

Fig. 21. Continued.

V.5 Demonstration Example 2

This example is identical to demonstration example 1 except the aquifer is unconfined and no pumping is to be made (Fig. 22). Without an upper confining bed, the water surface is allowed to rise or fall freely due to rainfall infiltration, evapotranspiration, leakage to or from the lower aquifers, and changes in the water levels of adjacent surface water bodies. The problem becomes nonlinear because the transmissivity tensor and the generalized storage coefficient both depend on the piezometric head. The piezometric head is the dependent variable to be sought by solving Eqs. (79) through (83) with the FEWA code. The elevation of the aquifer bottom will be assumed to vary linearly from 2 m at point A to 1 m at point B and 0 m at points C and D (Fig. 22). It is further assumed that all aquifer properties and initial and boundary conditions remain the same as those in demonstration example 1. The absence of the upper confining aquitard can be specified by setting the leaking coefficient $K_y = 0.0$. The inclusion of infiltration or evapotranspiration is done by specifying sources/sinks distributed through the area. We will assume that infiltration or evapotranspiration is transient. Infiltration/evapotranspiration rates for the three geologic formations are given in Fig. 23. Positive values in Fig. 23 denote infiltration, while the negative values represent evapotranspiration.

Using the same finite element discretization as that for demonstration example 1, one can prepare the input data according to Appendix A. The input data set for this demonstration example is given in Table 8. The output is given in Appendix B. Figures 24 and 25 illustrate the distribution of free water surface and the vertically integrated Darcy fluxes, respectively, at several time instants. These figures indicate that the head distribution is of highly transient in the earlier days in response to the time-dependent infiltration/evapotranspiration, boundary head, and boundary flux. The transient head in the left part of the aquifer is more closely related to the infiltration/evapotranspiration input whereas it is more related to the changing head on the Dirichlet boundary. In the later days, the head distribution is gradually developed toward steady state since both the source/sink and boundary conditions are becoming constant with time.

ORNL-DWG 82-18453

Fig. 22. Problem sketch of demonstration example 2 for flow in an unconfined aquifer: (a) three-dimensional view and (b) plane view.

Fig. 23. Profile of net rainfall and evapotranspiration.

Table 8. Input data set for demonstration example 2

```

5      INPUT FOR DEMONSTRATION EXAMPLE 2
      DATA SET 2: BASIC INTEGER PARAMETERS
231 200      3      6      288      0      1      0      0      20      0      0      0      1      0
      DATA SET 3: BASIC REAL PARAMETERS
1.8D03      0.0D00      1.8D03      5.184D05      1.0D-4      1.0D-5      1.0D03      9.8D00
1.3D-3      1.0D00      1.0D00
      DATA SET4: LINEPRINTER, DISKOUTPUT, AND TIME STEP SETTING CONTROL
33          131          131          131      131
          131          131          131          131
31          131
1          1          1          1          1
          1          1          1          1
1          1          1          1
1.0D50
      DATA SET 5: MATERIAL PROPERTIES
1.0D-5      3.0D-6      0.10D00      1.25D-5      1.25D-5      0.63D-5      0.10D00
2.0D-5      3.0D-6      0.27D00      5.74D-7      5.74D-7      2.97D-7      0.27D00
4.0D-5      3.0D-6      0.45D00      9.49D-9      9.49D-9      4.75D-9      0.45D00
----- DATA SET 6: BOTTOM ELEVATION -----
1      1      1      2.0000      -0.1000      0.0
3      1      1      1.8000      -0.1000      0.0
5      1      1      1.6000      -0.1000      0.0
7      1      1      1.4000      -0.1000      0.0
9      1      1      1.2000      -0.1000      0.0
11     0      1      1.0000      0.0          0.0
12     1      1      1.9000      -0.0900      0.0
14     1      1      1.7200      -0.0900      0.0
16     1      1      1.5400      -0.0900      0.0
18     1      1      1.3600      -0.0900      0.0
20     1      1      1.1800      -0.0900      0.0
22     0      1      1.0000      0.0          0.0
23     1      1      1.8000      -0.0800      0.0
25     1      1      1.6400      -0.0800      0.0
27     1      1      1.4800      -0.0800      0.0
29     1      1      1.3200      -0.0800      0.0
31     1      1      1.1600      -0.0800      0.0
33     0      1      1.0000      0.0          0.0
34     1      1      1.7000      -0.0700      0.0
36     1      1      1.5600      -0.0700      0.0
38     1      1      1.4200      -0.0700      0.0
40     1      1      1.2800      -0.0700      0.0
42     1      1      1.1400      -0.0700      0.0
44     0      1      1.0000      0.0          0.0
45     1      1      1.6000      -0.0600      0.0
47     1      1      1.4800      -0.0600      0.0
49     1      1      1.3600      -0.0600      0.0
51     1      1      1.2400      -0.0600      0.0
53     1      1      1.1200      -0.0600      0.0
55     0      1      1.0000      0.0          0.0
56     1      1      1.5000      -0.0500      0.0
58     1      1      1.4000      -0.0500      0.0
60     1      1      1.3000      -0.0500      0.0
62     1      1      1.2000      -0.0500      0.0
64     1      1      1.1000      -0.0500      0.0
66     0      1      1.0000      0.0          0.0
67     1      1      1.4000      -0.0400      0.0
69     1      1      1.3200      -0.0400      0.0
71     1      1      1.2400      -0.0400      0.0
73     1      1      1.1600      -0.0400      0.0
75     1      1      1.0800      -0.0400      0.0
77     0      1      1.0000      0.0          0.0
78     1      1      1.3000      -0.0300      0.0
80     1      1      1.2400      -0.0300      0.0
82     1      1      1.1800      -0.0300      0.0
84     1      1      1.1200      -0.0300      0.0
86     1      1      1.0600      -0.0300      0.0
88     0      1      1.0000      0.0          0.0
89     1      1      1.2000      -0.0200      0.0

```

Table 8. (continued)

91	1	1	1.1600	-0.0200	0.0
93	1	1	1.1200	-0.0200	0.0
95	1	1	1.0800	-0.0200	0.0
97	1	1	1.0400	-0.0200	0.0
99	0	1	1.0000	0.0	0.0
100	1	1	1.1000	-0.0100	0.0
102	1	1	1.0800	-0.0100	0.0
104	1	1	1.0600	-0.0100	0.0
106	1	1	1.0400	-0.0100	0.0
108	1	1	1.0200	-0.0100	0.0
110	0	1	1.0000	0.0	0.0
111	1	1	1.0000	0.0	0.0
113	1	1	1.0000	0.0	0.0
115	1	1	1.0000	0.0	0.0
117	1	1	1.0000	0.0	0.0
119	1	1	1.0000	0.0	0.0
121	0	1	1.0000	0.0	0.0
122	1	1	0.9000	0.0100	0.0
124	1	1	0.9200	0.0100	0.0
126	1	1	0.9400	0.0100	0.0
128	1	1	0.9600	0.0100	0.0
130	1	1	0.9800	0.0100	0.0
132	0	1	1.0000	0.0	0.0
133	1	1	0.8000	0.0200	0.0
135	1	1	0.8400	0.0200	0.0
137	1	1	0.8800	0.0200	0.0
139	1	1	0.9200	0.0200	0.0
141	1	1	0.9600	0.0200	0.0
143	0	1	1.0000	0.0	0.0
144	1	1	0.7000	0.0300	0.0
146	1	1	0.7600	0.0300	0.0
148	1	1	0.8200	0.0300	0.0
150	1	1	0.8800	0.0300	0.0
152	1	1	0.9400	0.0300	0.0
154	0	1	1.0000	0.0	0.0
155	1	1	0.6000	0.0400	0.0
157	1	1	0.6800	0.0400	0.0
159	1	1	0.7600	0.0400	0.0
161	1	1	0.8400	0.0400	0.0
163	1	1	0.9200	0.0400	0.0
165	0	1	1.0000	0.0	0.0
166	1	1	0.5000	0.0500	0.0
168	1	1	0.6000	0.0500	0.0
170	1	1	0.7000	0.0500	0.0
172	1	1	0.8000	0.0500	0.0
174	1	1	0.9000	0.0500	0.0
176	0	1	1.0000	0.0	0.0
177	1	1	0.4000	0.0600	0.0
179	1	1	0.5200	0.0600	0.0
181	1	1	0.6400	0.0600	0.0
183	1	1	0.7600	0.0600	0.0
185	1	1	0.8800	0.0600	0.0
187	0	1	1.0000	0.0	0.0
188	1	1	0.3000	0.0700	0.0
190	1	1	0.4400	0.0700	0.0
192	1	1	0.5800	0.0700	0.0
194	1	1	0.7200	0.0700	0.0
196	1	1	0.8600	0.0700	0.0
198	0	1	1.0000	0.0	0.0
199	1	1	0.2000	0.0800	0.0
201	1	1	0.3600	0.0800	0.0
203	1	1	0.5200	0.0800	0.0
205	1	1	0.6800	0.0800	0.0
207	1	1	0.8400	0.0800	0.0
209	0	1	1.0000	0.0	0.0
210	1	1	0.1000	0.0900	0.0
212	1	1	0.2800	0.0900	0.0
214	1	1	0.4600	0.0900	0.0
216	1	1	0.6400	0.0900	0.0
218	1	1	0.8200	0.0900	0.0
220	0	1	1.0000	0.0	0.0

Table 8. (continued)

221	1	1	0.0	0.1000	0.0
223	1	1	0.2000	0.1000	0.0
225	1	1	0.4000	0.1000	0.0
227	1	1	0.6000	0.1000	0.0
229	1	1	0.8000	0.1000	0.0
231	0	1	1.0000	0.0	0.0

DATA SET 7: NODE COORDINATE

1	1	1	0.0	0.0	0.0
3	1	1	0.0	0.0	0.0
5	1	1	0.0	0.0	0.0
7	1	1	0.0	0.0	0.0
9	1	1	0.0	0.0	0.0
11	0	1	0.0	0.0	0.0
12	1	1	10.0000	-0.1000	0.0
14	1	1	9.8000	-0.1000	0.0
16	1	1	9.6000	-0.1000	0.0
18	1	1	9.4000	-0.1000	0.0
20	1	1	9.2000	-0.1000	0.0
22	0	1	9.0000	0.0	0.0
23	1	1	20.0000	-0.2000	0.0
25	1	1	19.6000	-0.2000	0.0
27	1	1	19.2000	-0.2000	0.0
29	1	1	18.8000	-0.2000	0.0
31	1	1	18.4000	-0.2000	0.0
33	0	1	18.0000	0.0	0.0
34	1	1	30.0000	-0.3000	0.0
36	1	1	29.4000	-0.3000	0.0
38	1	1	28.8000	-0.3000	0.0
40	1	1	28.2000	-0.3000	0.0
42	1	1	27.6000	-0.3000	0.0
44	0	1	27.0000	0.0	0.0
45	1	1	40.0000	-0.4000	0.0
47	1	1	39.2000	-0.4000	0.0
49	1	1	38.4000	-0.4000	0.0
51	1	1	37.6000	-0.4000	0.0
53	1	1	36.8000	-0.4000	0.0
55	0	1	36.0000	0.0	0.0
56	1	1	50.0000	-0.5000	0.0
58	1	1	49.0000	-0.5000	0.0
60	1	1	48.0000	-0.5000	0.0
62	1	1	47.0000	-0.5000	0.0
64	1	1	46.0000	-0.5000	0.0
66	0	1	45.0000	0.0	0.0
67	1	1	60.0000	-0.6000	0.0
69	1	1	58.8000	-0.6000	0.0
71	1	1	57.6000	-0.6000	0.0
73	1	1	56.4000	-0.6000	0.0
75	1	1	55.2000	-0.6000	0.0
77	0	1	54.0000	0.0	0.0
78	1	1	70.0000	-0.7000	0.0
80	1	1	68.6000	-0.7000	0.0
82	1	1	67.2000	-0.7000	0.0
84	1	1	65.8000	-0.7000	0.0
86	1	1	64.4000	-0.7000	0.0
88	0	1	63.0000	0.0	0.0
89	1	1	80.0000	-0.8000	0.0
91	1	1	78.4000	-0.8000	0.0
93	1	1	76.8000	-0.8000	0.0
95	1	1	75.2000	-0.8000	0.0
97	1	1	73.6000	-0.8000	0.0
99	0	1	72.0000	0.0	0.0
100	1	1	90.0000	-0.9000	0.0
102	1	1	88.2000	-0.9000	0.0
104	1	1	86.4000	-0.9000	0.0
106	1	1	84.6000	-0.9000	0.0
108	1	1	82.8000	-0.9000	0.0
110	0	1	81.0000	0.0	0.0

Table 8. (continued)

111	1	1	100.0000	-1.0000	0.0
113	1	1	98.0000	-1.0000	0.0
115	1	1	96.0000	-1.0000	0.0
117	1	1	94.0000	-1.0000	0.0
119	1	1	92.0000	-1.0000	0.0
121	0	1	90.0000	0.0	0.0
122	1	1	110.0000	-1.1000	0.0
124	1	1	107.8000	-1.1000	0.0
126	1	1	105.6000	-1.1000	0.0
128	1	1	103.4000	-1.1000	0.0
130	1	1	101.2000	-1.1000	0.0
132	0	1	99.0000	0.0	0.0
133	1	1	120.0000	-1.2000	0.0
135	1	1	117.6000	-1.2000	0.0
137	1	1	115.2000	-1.2000	0.0
139	1	1	112.8000	-1.2000	0.0
141	1	1	110.4000	-1.2000	0.0
143	0	1	108.0000	0.0	0.0
144	1	1	133.0000	-1.3000	0.0
146	1	1	127.4000	-1.3000	0.0
148	1	1	124.8000	-1.3000	0.0
150	1	1	122.2000	-1.3000	0.0
152	1	1	119.6000	-1.3000	0.0
154	0	1	117.0000	0.0	0.0
155	1	1	140.0000	-1.4000	0.0
157	1	1	137.2000	-1.4000	0.0
159	1	1	134.4000	-1.4000	0.0
161	1	1	131.6000	-1.4000	0.0
163	1	1	128.8000	-1.4000	0.0
165	0	1	126.0000	0.0	0.0
166	1	1	150.0000	-1.5000	0.0
168	1	1	147.0000	-1.5000	0.0
170	1	1	144.0000	-1.5000	0.0
172	1	1	141.0000	-1.5000	0.0
174	1	1	138.0000	-1.5000	0.0
176	0	1	135.0000	0.0	0.0
177	1	1	160.0000	-1.6000	0.0
179	1	1	156.8000	-1.6000	0.0
181	1	1	153.6000	-1.6000	0.0
183	1	1	150.4000	-1.6000	0.0
185	1	1	147.2000	-1.6000	0.0
187	0	1	144.0000	0.0	0.0
198	1	1	170.0000	-1.7000	0.0
190	1	1	166.6000	-1.7000	0.0
192	1	1	163.2000	-1.7000	0.0
194	1	1	159.8000	-1.7000	0.0
196	1	1	156.4000	-1.7000	0.0
198	0	1	153.0000	0.0	0.0
199	1	1	180.0000	-1.8000	0.0
201	1	1	176.4000	-1.8000	0.0
203	1	1	172.8000	-1.8000	0.0
205	1	1	169.2000	-1.8000	0.0
207	1	1	165.6000	-1.8000	0.0
209	0	1	162.0000	0.0	0.0
211	1	1	190.0000	-1.9000	0.0
212	1	1	186.2000	-1.9000	0.0
214	1	1	182.4000	-1.9000	0.0
216	1	1	178.6000	-1.9000	0.0
218	1	1	174.8000	-1.9000	0.0
220	0	1	171.0000	0.0	0.0
221	1	1	200.0000	-2.0000	0.0
223	1	1	196.0000	-2.0000	0.0
225	1	1	192.0000	-2.0000	0.0
227	1	1	188.0000	-2.0000	0.0
229	1	1	184.0000	-2.0000	0.0
231	0	1	180.0000	0.0	0.0

Table 8. (continued)

1	1	1	0.0	5.5000	0.0
3	1	1	11.0000	5.5000	0.0
5	1	1	22.0000	5.5000	0.0
7	1	1	33.0000	5.5000	0.0
9	1	1	44.0000	5.5000	0.0
11	0	1	55.0000	0.0	0.0
12	1	1	0.2500	5.4750	0.0
14	1	1	11.2000	5.4750	0.0
16	1	1	22.1500	5.4750	0.0
18	1	1	33.1000	5.4750	0.0
20	1	1	44.0500	5.4750	0.0
22	0	1	55.0000	0.0	0.0
23	1	1	0.5000	5.4500	0.0
25	1	1	11.4000	5.4500	0.0
27	1	1	22.3000	5.4500	0.0
29	1	1	33.2000	5.4500	0.0
31	1	1	44.1000	5.4500	0.0
33	0	1	55.0000	0.0	0.0
34	1	1	0.7500	5.4250	0.0
36	1	1	11.6000	5.4250	0.0
38	1	1	22.4500	5.4250	0.0
40	1	1	33.3000	5.4250	0.0
42	1	1	44.1500	5.4250	0.0
44	0	1	55.0000	0.0	0.0
45	1	1	1.0000	5.4000	0.0
47	1	1	11.8000	5.4000	0.0
49	1	1	22.6000	5.4000	0.0
51	1	1	33.4000	5.4000	0.0
53	1	1	44.2000	5.4000	0.0
55	0	1	55.0000	0.0	0.0
56	1	1	1.2500	5.3750	0.0
58	1	1	12.0000	5.3750	0.0
60	1	1	22.7500	5.3750	0.0
62	1	1	33.5000	5.3750	0.0
64	1	1	44.2500	5.3750	0.0
66	0	1	55.0000	0.0	0.0
67	1	1	1.5000	5.3500	0.0
69	1	1	12.2000	5.3500	0.0
71	1	1	22.9000	5.3500	0.0
73	1	1	33.6000	5.3500	0.0
75	1	1	44.3000	5.3500	0.0
77	0	1	55.0000	0.0	0.0
78	1	1	1.7500	5.3250	0.0
80	1	1	12.4000	5.3250	0.0
82	1	1	23.0500	5.3250	0.0
84	1	1	33.7000	5.3250	0.0
86	1	1	44.3500	5.3250	0.0
88	0	1	55.0000	0.0	0.0
89	1	1	2.0000	5.3000	0.0
91	1	1	12.6000	5.3000	0.0
93	1	1	23.2000	5.3000	0.0
95	1	1	33.8000	5.3000	0.0
97	1	1	44.4000	5.3000	0.0
99	0	1	55.0000	0.0	0.0
100	1	1	2.2500	5.2750	0.0
102	1	1	12.8000	5.2750	0.0
104	1	1	23.3500	5.2750	0.0
106	1	1	33.9500	5.2750	0.0
108	1	1	44.4500	5.2750	0.0
110	0	1	55.0000	0.0	0.0
111	1	1	2.5000	5.2500	0.0
113	1	1	13.0000	5.2500	0.0
115	1	1	23.5000	5.2500	0.0
117	1	1	34.0500	5.2500	0.0
119	1	1	44.5000	5.2500	0.0
121	0	1	55.0000	0.0	0.0
122	1	1	2.7500	5.2250	0.0
124	1	1	13.2000	5.2250	0.0
126	1	1	23.6500	5.2250	0.0
128	1	1	34.1000	5.2250	0.0
130	1	1	44.5500	5.2250	0.0

Table 8. (continued)

132	0	1	55.0000	0.0	0.0
133	1	1	3.0000	5.2000	0.0
135	1	1	13.4000	5.2000	0.0
137	1	1	23.8000	5.2000	0.0
139	1	1	34.2000	5.2000	0.0
141	1	1	44.6000	5.2000	0.0
143	0	1	55.0000	0.0	0.0
144	1	1	3.2500	5.1750	0.0
146	1	1	13.6000	5.1750	0.0
148	1	1	23.9500	5.1750	0.0
150	1	1	34.3000	5.1750	0.0
152	1	1	44.6500	5.1750	0.0
154	0	1	55.0000	0.0	0.0
155	1	1	3.5000	5.1500	0.0
157	1	1	13.8000	5.1500	0.0
159	1	1	24.1000	5.1500	0.0
161	1	1	34.4000	5.1500	0.0
163	1	1	44.7000	5.1500	0.0
165	0	1	55.0000	0.0	0.0
166	1	1	3.7500	5.1250	0.0
168	1	1	14.0000	5.1250	0.0
170	1	1	24.2500	5.1250	0.0
172	1	1	34.5000	5.1250	0.0
174	1	1	44.7500	5.1250	0.0
176	0	1	55.0000	0.0	0.0
177	1	1	4.0000	5.1000	0.0
179	1	1	14.2000	5.1000	0.0
181	1	1	24.4000	5.1000	0.0
183	1	1	34.6000	5.1000	0.0
185	1	1	44.8000	5.1000	0.0
187	0	1	55.0000	0.0	0.0
188	1	1	4.2500	5.0750	0.0
190	1	1	14.4000	5.0750	0.0
192	1	1	24.5500	5.0750	0.0
194	1	1	34.7000	5.0750	0.0
196	1	1	44.8500	5.0750	0.0
198	0	1	55.0000	0.0	0.0
199	1	1	4.5000	5.0500	0.0
201	1	1	14.6000	5.0500	0.0
203	1	1	24.7000	5.0500	0.0
205	1	1	34.8000	5.0500	0.0
207	1	1	44.9000	5.0500	0.0
209	0	1	55.0000	0.0	0.0
210	1	1	4.7500	5.0250	0.0
212	1	1	14.8000	5.0250	0.0
214	1	1	24.8500	5.0250	0.0
216	1	1	34.9000	5.0250	0.0
218	1	1	44.9500	5.0250	0.0
220	0	1	55.0000	0.0	0.0
221	1	1	5.0000	5.0000	0.0
223	1	1	15.0000	5.0000	0.0
225	1	1	25.0000	5.0000	0.0
227	1	1	35.0000	5.0000	0.0
229	1	1	45.0000	5.0000	0.0
231	0	1	55.0000	0.0	0.0

DATA SET 8: ELEMENT INCIDENCES

1	1	12	13	2	-1	19	20
---	---	----	----	---	----	----	----

DATA SET 9: MATERIAL CORRECTIONS FOR REGIONS IN EFGH AND IJKL

38	2	10	-2	0
93	2	10	-3	0

DATA SET 10: PRE-INITIAL CONDITIONS

1	230	1	5.0000	0.000	0.000
---	-----	---	--------	-------	-------

Fig. 24. Distribution of hydraulic head as simulated by FEWA for demonstration example 2 at time (a) 0, (b) 0.3, (c) 0.6, (d) 0.9 (e) 1.0, (f) 2.0, (g) 2.5, (h) 3.0, (i) 4.0, and (j) 5.0 d, respectively.

(c) TIME = 0.6d

(d) TIME = 0.9d

Fig. 24. Continued.

(a) TIME = 1.0 d

(f) TIME = 2.0 d

Fig. 24. Continued.

Fig. 24. Continued.

Fig. 24. Continued.

Fig. 25. Plot of vertically integrated Darcy fluxes as simulated by FEWA for demonstration example 2 at time (a) 0.0, (b) 0.3, (c) 0.6, (d) 0.9 (e) 1.0, (f) 2.0, (g) 2.5, (h) 3.0, (i) 4.0, and (j) 5.0 d, respectively.

(c) TIME = 0.6 d

(d) TIME = 0.9 d

Fig. 25. Continued.

ORNL-DWG 83-1835

(e) TIME = 1.0 d

ORNL-DWG 83-1836

(f) TIME = 2.0 d

Fig. 25. Continued.

ORNL-DWG 83-1837

(g) TIME = 2.5 d

ORNL-DWG 83-1838

(h) TIME = 3.0 d

Fig. 25. Continued.

ORNL-DWG 83-1839

(I) TIME = 4.0 d

ORNL-DWG 83-1840

(J) TIME = 5.0 d

Fig. 25. Continued.

V. REFERENCES

- Bear, J. 1972. Dynamics of Fluids in Porous Media. American Elsevier, New York. 764 pp.
- Bear, J. 1977. Hydraulics of Groundwater. McGraw-Hill Book Company, New York. 569 pp.
- Biot, M. 1940. General theory of three-dimensional consolidation. J. Appl. Phys. 12:155-164.
- Bredehoeft, J. D., and G. F. Pinder. 1970. Digital analysis of areal flow in multi-aquifer groundwater systems: A quasi three-dimensional model. Water Resour. Res. 6:883-888.
- Carslaw, H. S., and J. C. Jaeger. 1959. Conduction of Heat in Solids, Oxford at the Clarendon Press, London. 510 pp.
- Clough, R. W. 1971. Analysis of structural vibrations and dynamic response. IN R. H. Gallagher, Y. Yamado, and J. T. Oden (eds.), Recent Advances in Matrix Methods of Structural Analysis and Design. University of Alabama Press, Huntsville.
- Conte, S. D. 1965. Elementary Numerical Analysis. McGraw-Hill Book Company, New York. 278 pp.
- Gupta, S. K., C. R. Cole, and F. W. Bond. 1979. Methodology for release consequence analysis - Part III, Finite Element Three-dimensional Groundwater (FE3DGW) flow model. PNL-2939. Pacific Northwest Laboratory, Richland, Washington.
- Hilderbrand, F. B. 1968. Finite-Difference Equations and Simulations. Prentice-Hall, Inc., Englewood, New Jersey. 338 pp.
- Lappala, E. G. 1980. Modeling of water and solute transport under variably saturated-unsaturated conditions: State of the Art. pp. 81-137. IN Modeling and Low-Level Waste Management: An Interagency Workshop, ORO-821. Oak Ridge, Tennessee.
- Owczarek, J. A. 1964. Fundamentals of Gas Dynamics. International Textbook Company, Scranton, Pennsylvania. 648 pp.
- Pinder, G. F. 1974. Galerkin Finite Element Models for Aquifer Simulation Water Resources Program. Princeton University, Princeton.

- Pinder, G. F., and W. G. Gray. 1977. Finite Element Simulation in Surface and Subsurface Hydrology. Academic Press, New York. 295 pp.
- Polubarinova-Kochina, P. Ya. 1962. Theory of Groundwater Movement. Princeton University Press, Princeton, New Jersey. 613 pp.
- Trescott, P. C. 1975. Documentation of finite-difference model for simulation of three-dimensional groundwater flow. U.S.G.S. Open File Report 75-438. U.S.G.S., Reston, Virginia.
- Trescott, P. C., G. F. Pinder, and S. P. Larson. 1976. Finite-difference model for aquifer simulation in two dimensions with results of numerical experiments. Book 7, Chapter C1. Technique of Water-Resources Investigations of the United States Geological Survey. U.S.G.S., Reston, Virginia. 116 pp.
- van Genuchten, M. Th. 1980. A closed form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Sci. Am. J. 44:892-898.
- Verruijt, A. 1969. Elastic storage of aquifers. IN Flow through Porous Media. R. DeWiest (ed.), New York, McGraw-Hill, Chapter 8.
- 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.
- Yeh, G. T. 1983. Solution of groundwater flow equations using an orthogonal finite element scheme. IN Proc., International 83 Association for the Advancement of Modeling and Simulation Techniques in Enterprises Summer Conference, Modeling and Simulation, Nice, France, September 12-14, 1983 (in press).

VII. GLOSSARY OF NOTATIONS

A	= Area of a triangular element
$[A]$	= Mass matrix whose (i,j) -th entry is given in Eq. (62)
α	= Compressibility of the media (T^2L/M)
α'	= $\alpha\rho_f g$ = Modified compressibility of the medium ($1/L$)
b	= Aquifer thickness (L)
B	= Boundary enclosing R
$[B]$	= Stiff matrix whose (i,j) -th entry is given in Eq. (63)
B_C	= Cauchy boundary
B_D	= Dirichlet boundary
B_F	= Flux boundary
B_N	= Neumann boundary
β	= Compressibility of water (T^2L/M)
∇	= Del operator ($1/L$)
∇_H	= Del operator over horizontal coordinates ($1/L$)
$[C]$	= Coefficient matrix
$\{D\}$	= Load column vector whose i -th entry is given in Eq. (64)
δ	= Delta function
e	= Strain of the solid (dimensionless)
F	= Moving boundary
F_B	= Vertically integrated flux through the entire boundary (L/T)
F_D	= Vertically integrated flux through the entire Dirichlet boundary (L^3/T)
F_F	= Vertically integrated flux through the entire flux boundary (L^3/T)
F_L	= Vertically integrated flux through the entire unspecified boundary
F_{LA}	= Rate of water leaked through the lower aquitard for the entire region (L^3/T)
F_n	= Vertically integrated normal flux through the boundary (L^2/T)
F_p	= Accumulated rate of change of water in the region due to the change of hydraulic head (L^3/T)
F_s	= Rate of water recharged to the entire region (L^3/T)

- F_{UA} = Rate of water leaked through the upper aquitard for the region (L^3/T)
- F_V = Accumulated rate of water in the region vaulting from change of hydraulic head, leaking through upper aquitard, leaking through lower aquitard, and sources/sinks (L^3/T)
- h = Dupuit hydraulic head (L)
- $\{h\}$ = Vector representing the hydraulic head at all nodes (L)
- h_D = Value of h on the Dirichlet boundary (L)
- h_I = Initial value of h (L)
- h_j = Value of h at node point j (L)
- h_L = Hydraulic head in the lower aquitard (L)
- h_U = Hydraulic head in the upper aquitard (L)
- g = Gravity acceleration (L/T^2)
- I = Net flux of water through the moving boundary (L/T)
- $\bar{K}$ = Hydraulic conductivity tensor (L/T)
- K_L = Leaking coefficient of lower aquitard (1/T)
- K_U = Leaking coefficient of upper aquitard (1/T)
- λ_s = Lamé constants (F/L^2)
- $[M]$ = Mass matrix resulting from modifying $[A]$ with boundary conditions
- μ = Dynamic viscosity of fluid [$M/(TL)$]
- μ_s = Lamé constants (F/L^2)
- $\vec{n}$ = Unit vector normal to a surface
- n_e = Effective porosity (dimensionless)
- $\left. \begin{array}{l} N_f \\ N_g \\ N_h \\ N_i \\ N_j \\ N_k \\ N_l \\ N_m \end{array} \right\}$ = Basis functions for nodes f, g, h, i, j, k, l , and m , respectively

$\left. \begin{matrix} N_1^a \\ N_2^a \\ N_3^a \\ N_4^a \end{matrix} \right\} =$ Basis functions for nodes 1, 2, 3,
and 4, respectively, of element a

$\left. \begin{matrix} N_1^b \\ N_2^b \\ N_3^b \\ N_4^b \end{matrix} \right\} =$ Basis functions for nodes 1, 2, 3,
and 4, respectively, of element b

$\left. \begin{matrix} N_1^c \\ N_2^c \\ N_3^c \\ N_4^c \end{matrix} \right\} =$ Basis functions for nodes 1, 2, 3,
and 4, respectively, of element c

$\left. \begin{matrix} N_1^d \\ N_2^d \\ N_3^d \end{matrix} \right\} =$ Basis functions for nodes 1, 2, and 3,
respectively, of element d

$\left. \begin{matrix} N_1^e \\ N_2^e \\ N_3^e \\ N_4^e \end{matrix} \right\} =$ Basis functions for nodes 1, 2, 3, and 4,
respectively, of quadrilateral element e

$\{Q\}$ = Load column vector resulting from boundary conditions
whose i -th entry is given in Eq. (65)

Q_U = Rate of water leaked through upper aquitard ($L^3/T/L^2$)

Q_L = Rate of water leaked through lower aquitard ($L^3/T/L^2$)

ϕ = Hydraulic head (L)

ψ = Pressure head (L)

p = Pressure [$M/(T^2L) = F/L^2$]

p_0 = Reference pressure (F/L^2)

- Ψ_D = Value of Ψ on the Dirichlet boundary (L)
 Ψ_I = Initial value of Ψ (L)
 h_s = Top elevation of the aquifer in the case of confined aquifer
 or water table elevation in the case of unconfined aquifer (L)
 $\left. \begin{matrix} L_1 \\ L_2 \\ L_3 \end{matrix} \right\}$ = Area coordinates for triangular elements e
 $\left. \begin{matrix} N_1^e \\ N_2^e \\ N_3^e \end{matrix} \right\}$ = Basis functions for nodes 1, 2, and 3, respectively,
 of a triangular element e
 q = Internal source/sink of water ($L^3/T/L^3$)
 q_C = Total flux on the Cauchy boundary ($L^3/T/L^2$)
 q_F = Vertically integrated flux on the flux boundary (L^2/T)
 q_N = Gradient flux on the Neumann boundary ($L^3/T/L^2$)
 ρ_s = Density of solid grains (M/L^3)
 ρ_f = Fluid density (M/L^3)
 R = Region of interest
 R_r = Residual defined in Eq (58)
 S = Generalized storage coefficient (dimensionless)
 $[S]$ = Stiff matrix vaulting from modifying $[B]$ with
 boundary conditions
 σ = Incremental pressure of the fluid (F/L^2)
 S_p = Source/sink strength of p-th well (L^3/T)
 S_s = Specific storage (1/L)
 S_w = $W + I$ = total water sources/sinks (L/T)
 S_y = Specific yield of the aquifer (dimensionless)
 t = Time (T)
 $\bar{T}$ = Transmissibility tensor (L^2/T)
 Γ = Surface enclosing the volume V (L^2)
 U = Displacement vector of the media (L/T)
 U_f = Velocity vector of the moving boundary (L/T)

v	= Volume (L^3)
V_{fs}	= Relative velocity of fluid to the solid skeleton (L/T)
V_s	= Velocity of solid skeleton (L/T)
W	= Vertically integrated internal sources/sinks (L/T)
W_i	= Weighting function for node point i
x, y	= Horizontal Cartesian coordinates
x_p	= x-coordinate of p-th well (L)
ξ, η	= Local coordinates for isoparametric elements
$\{Y\}$	= Load vector incorporating the boundary conditions $\{D\}$ and $\{Q\}$
y_p	= y-coordinate of p-th well (L)
z	= Potential head (L)
z_0	= Bottom elevation of the aquifer (L)

APPENDIX A

DATA INPUT GUIDE OF FEWA

APPENDIX A: DATA INPUT GUIDE OF FEMA

1. TITLE: FORMAT(15,948,1,1) - One card per problem.

SPROB	TITLE	IDUC
5	77	86

SPROB = Problem number.

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

IDUC = Integer indicating if the diagnostic output is desired? 0 = No, 1 = Yes.

2. BASIC INTEGER PARAMETERS: Two cards per problem.

Card 1 - FORMAT(20A4)

DATNAM
80

DATNAM = Data name to describe this data set.

Card 2 - FORMAT(16I5)

NUP	NEL	NMAT	NEN	NTI	NSS	NSIB	NCF
5	10	15	20	25	30	35	40
NSTPT	NITER	ILUMP	INIB	ICONFI	NDTCRC	IFNYS	
45	50	55	60	65	70		80

NUP = Number of nodal points.

NEL = Number of elements.

NMAT = Number of material types.

NEN = Number of elements with IE(N,5) correction.

NTI = Number of time steps or time increments.

APPENDIX A. (continued)

ESS = Steady state control,
 0 = steady state solution desired,
 1 = transient state or transient solutions.

ESFR = Auxiliary storage output control;
 0 = no storage,
 1 = output stored in Logical Unit 1.

KP = Permeability input control;
 0 = input saturated hydraulic conductivity,
 1 = input saturated permeability.

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

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

ILUMP = Mass lumping control;
 0 = no lumping,
 1 = lumping.

IMD = Mid-difference integration? 0 = No, 1 = Yes.

ICOWI = Confined aquifer? 0 = No, 1 = Yes.

NTCHG = No. of times to reset step size to initial
 time step size.

IPWIS = Is pointwise iterative method used?
 0 = No, 1 = yes.

*** NOTE: NTI can be computed by $NTI = I_1 + 1 + I_2 + 1$,
 where I_1 = largest integer not exceeding
 $\log(DLNAX/DELT) / \log(1+CHNG)$,
 I_2 = largest integer not exceeding
 $(RTIME-DF.T*((1+CHNG)**(I_1+1)-1)/CHNG)/DLNAX$,
 RTIME = real simulation time,
 DLNAX, DELT, and CHNG are defined in Data Set 3.

APPENDIX A. (continued)

3. BASIC REAL PARAMETERS

Three cards per problem. Use of an E-, D-, or another P-type field specification in the input card overrides any of the b10.3 field.

Card 1 - FORNAT(20A4)

DATNAM

DATNAM = Data name to describe this data set.
It may contain up to 80 characters.

Card 2 - FORNAT(8D10.3).

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

DELT = Initial time step size, (T).

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

DELMAX = Maximum value of DELT, (T).

THAX = Maximum simulation time, (T).

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

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

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

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

Card 3 - FORNAT(8D10.3).

VISC	W	ONE
10	20	30

VISC = Dynamic viscosity of water, (M/L/T).

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

ONE = Iteration parameter;
0.0 ~ 1.0 = under-relaxation,
1.0 ~ 1.0 = exact relaxation,
1.0 ~ 2.0 = over-relaxation.

APPENDIX A. (continued)

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

Card 1 - **FORMAT(20A4)**

DATNAM

80

DATNAM = Data name to describe this data set. It may contain up to 80 characters.

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

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

KPRO = Printer control for steady state and initial conditions;

0 = print nothing,
 1 = print FLOW, FRATE, and TFLOW,
 2 = print above (1) plus piezometric head h,
 3 = print above (2) plus integrated flux.

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

Card $[(NTI/80+1)+2]$ to Card $[(NTI/80+1)+2+1]$ - **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 $[(NTI/80+1)*2 + 2]$ to card $[(NTI/80)*2 + (NDTCHG/8+1) + 1]$

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

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

APPENDIX A. (continued)

5. MATERIAL PROPERTIES

A total of (NHAT+1) cards are required for this data set.

Card 1 - FORMAT(20A4)

DATNAM	80
--------	----

DATNAM = Data name for this data set. It may contain up to 80 characters.

Card 2 to Card [NHAT+1] - FORMAT(8D10.3)

PROP(1,1)	PROP(1,2)	----	PROP(1,7)	80
10	20		70	

PROP(I,1)	PROP(I,2)	----	PROP(I,7)	80
10	20		70	

PROP(NHAT,1)	PROP(NHAT,2)	----	PROP(NHAT,7)	80
10	20		70	

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

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

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

PROP(I,4) = xx-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L²).

PROP(I,5) = yy-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L²).

PROP(I,6) = xy-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L²).

PROP(I,7) = Specific yield of medium I, (dimensionless in decimal point).

APPENDIX A. (continued)

NOTE: DATA SETS 6 THROUGH 10 ARE NOT NEEDED IF NSTRT .GT. 0.

6. BOTTOM ELEVATION OF UNCONFINED AQUIFER OR THICKNESS OF CONFINED AQUIFER
A Group of cards are needed per problem. The first card is used to describe this data set.

Card 1 - FORMAT(20A4)

DATAH

80

DATAH = Data name to describe this data set. It may contain up to 80 characters.

Card 2 on - FORMAT(3I5,5X,4D10.3)

NI	NSEQ	NAD	BNI	BAD	
5	10	15	20	30	40

80

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

NSEQ = NSEQ subsequent node numbers will be automatically generated.

NAD = Node number increment for each of the NSEQ nodes.

BNI = Aquifer bottom elevation (unconfined case) or aquifer thickness (confined case) of node NI, (1).

BAD = Increment of aquifer bottom elevation or aquifer thickness for each of the NSEQ subsequent nodes, (1).

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

7. MODAL POINT COORDINATE

Usually a total of $(2 \cdot \text{NWP} + 1)$ cards are required. One for the description of this data set, NWP cards for x-coordinate, and NWP cards for y-coordinate. However, if a group of subsequent cards appears in regular pattern, automatic generation may be made.

Card 1 - FORMAT(20A4)

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

**YAD = Increment of y-coordinate for each of the
NSBQ subsequent nodes, (L).**

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

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

Usually a total of (NEL+1) cards are needed. The first card is used to describe the data set and the second card on contains element incidence and material type for each element. However, if a group of elements appears in regular pattern, automatic generation is made.

Card 1 - FORMAT (20A4)

DATA

DATNAM = 80 column of characters are used to describe this data set.

Card 2 on - FORNAT (16IS)

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

NI = Global element number.

IZ (NI, I) = Global node number of the first node of element NI.

IZ(NI,2) = Global node number of the second node of element NI.

IE(NI,3) = Global node number of the third node of element NI.

IE(NI,4) = Global node number of the fourth node of element NI.

APPENDIX A. (continued)

IE(NI,5) = Material type to be applied to element block **MODL** X **NLAY**. If the element block is confined, use positive number. If the element block is unconfined, use negative number.

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(NI,1) ~ IE(NI,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. The following figure provides an example. The object is considered 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 element 2 through 15

ORNL-DWG 81-3524 ESD

automatically, including both the incidence and material type, only one card is necessary.

APPENDIX A. (continued)

1	2	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, HTYP, this situation can easily be changed by using material-correction facility.

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

9. MATERIAL TYPE CORRECTION

This data set is required only if NCH > 0. Normally (NCH+1) cards are required, the first one is for the description of data set and the rest for the material type correction, one for each correction.

Card 1 - FORMAT(20A0)

DATNAM
80

DATNAM = Data name used to describe this data set.
It may contain up to 80 characters.

Card 2 on - FORMAT(5I5)

N	NSEQ	HAD	HTYP	HTYPAD	
5	10	15	20	25	80

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

NSEQ = NSEQ subsequent elements will automatically generate material types.

HAD = Increment of element number for each of the NSEQ subsequent elements.

HTYP = Material type for element N.

HTYPAD = Increment of HTYP for each of the NSEQ subsequent elements.

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

APPENDIX A. (continued)

10. CARD INPUT FOR INITIAL OR PRE-INITIAL CONDITIONS

This data set is read in similar to that in card data set 6.

Card 1 - FORNAT(20A0)

DATAH = 80 columns of characters are used to describe this data set.

Card 2 on - FORNAT(3I5,5X,4F10.3)

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.

NWI = Initial or pre-initial piezometric head of node NI, (L).

NAD = 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, head distribution for NSTR different times have been generated and written on disk or magnetic tape. If NSTR > 0, these distribution will be read from Logical Unit 2, and NSTR-th distribution will be used as the initial condition for current calculation. If NSTR is greater than zero, the head 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 NSTR = 0.

APPENDIX A. (continued)

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

NSEQ = NSEQ elements will contain the source type to be generated automatically.

NAD = Increment of element number for each of the NSEQ elements.

NTYP = Source type in element N.

NTYPAD = Increment of NTYP for each of the NSEQ subsequent elements.

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

- (d) Well Source/Sink Profile: **FORMAT(8D10.3)**.
Number of cards depends on **NUPR** and **NUDP**. This sub-data set is read in similar to Data Set 12 (b).

First Profile

TWSSP(1,1)	WSSP(1,1)	TWSSP(1,2)	WSSP(1,2)	---
10	20	30	40	80
:				
---	TWSSP(1,NUDP)	WSSP(1,NUDP)		
80				

Second Profile

TWSSP (2, 1)	WSSP (2, 1)	TWSSP (2, 2)	WSSP (2, 2)	---
10	20	30	40	80
:				
---	TWSSP (2, NWDP)	WSSP (2, NWDP)		
				80

I-th Profile

TWSSP(I,1)	WSSP(I,1)	TWSSP(I,2)	WSSP(I,2)	---
10	20	30	40	80

APPENDIX A. (continued)

NOTE ON AUXILIARY STORAGE UNITS: Logical Unit 1 is used to store output if $KSTN > 0$, and Logical Unit 2 is used for input if $WSTNT > 0$. 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 $WTI = 0$, and in the latter case $KSS = 0$ and $WTI > 0$. If $KSS > 0$, there will be no steady state calculation.

11. INTEGER PARAMETERS FOR TRANSIENT SOURCE, BOUNDARY CONDITIONS, AND LEAKING UPPER AND LOWER AQUIFERS.
Two cards per problem are required.

Card 1 - FORMAT(20A4)

DATBAN

80

DATBAN = 80 columns of characters for describing this data set.

Card 2 - FORMAT(16I5)

NBEL	NBUP	NBPR	NBDF	NBDP	NBDR	NBDF	NBSP
5	10	15	20	25	30	35	40

NBDF	NBUPR	NBUPR	NBUPR	NBUPR	NBUPR	NBUPR	NBUPR
45	50	55	60	65	70	75	80

NBEL = No. of Neumann element-sides.

NBUP = No. of Neumann boundary node points.

NBPR = No. of Neumann flux profiles.

NBDF = No. of data points in each of the NBUPR profiles.

NBDP = No. of Dirichlet node points, should be .GE. 1.

NBDR = No. of Dirichlet head profiles, should be .GE. 1.

NBDF = No. of data points in each of the NBDR profiles, should be .GE. 2.

APPENDIX A. (continued)

NSPR = No. of source/sink profiles, should be .GE. 1.

NSDP = No. of data points in each of the NSPR profiles, should be .GE. 2.

NSUPR = No. of head profiles in upper leaking aquifers, should be .GE. 1.

NSUDP = No. of data points in each of the NSUPR profiles, should be .GE. 2.

NLPR = No. of head profiles in lower leaking aquifers, should be .GE. 1.

NLDP = No. of data points in each of the NLPR profiles, should be .GE. 2.

NSNP = No. of well source/sink nodes

NSPP = No. of well source/sink profiles.

NSDP = No. of data points in each well s/s profile.

12. SOURCES/SINKS

- (a) Data Set Description: **FORMAT(20A4)** - One card per problem.

DATNAM

80

DATNAM = 80 columns of characters for describing the nature of this data set.

- (b) Element-wide Source/Sink Profiles: **FORMAT(8D10.3)**
Number of cards depends on NSPR and NSDP. This sub-data set is read in NSPR-wisely. Each card contains four data points.

First profile

TSOS (1,1)	SOS (1,1)	TSOS (1,2)	SOS (1,2)	---
10	20	30	40	80
:				
---	TSOS (1,NSDP)	SOS (1,NSDP)		
80				

80

APPENDIX A. (continued)

Second profile

TSOS (2, 1)	SOS (2, 1)	TSOS (2, 2)	SOS (2, 2)	---
10	20	30	40	50
.				
.				
---	TSOS (2, NSOP)	SOS (2, NSOP)		
60				

I-th profile

TSOS(I,1)	SOS(I,1)	TSOS(I,2)	SOS(I,2)	---
10	20	30	40	50
.				
..				
---	TSOS(I,NSOP)	SOS(I,NSOP)		
60				

NSPR-th profile

TSOS (NSPR, 1)	SOS (NSPR, 1)	TSOS (NSPR, 2)	SOS (NSPR, 2)	---
10	20	30	40	50
:				
:				
---	TSOS (NSPR, NSDP)	SOS (NSPR, NSDP)		
60				

TSOS(I,J) = Time of J-th data point in I-th profile, (").

SOS(I,J) = Source/sink value of J-th data point in I-th profile, (L=3/I/L=2)

(c) Source type in Each Element: FORMAT(5I5) - Usually one card per element. However, automatic generation can be made.

S	NSRC	NAD	STYP	STYPAD	
5	10	15	20	25	30

APPENDIX A. (continued)

---	TWSSP(I,NWDP)	WSSP(I,NWDP)	
-----	---------------	--------------	--

80

⋮

NWDP-th Profile

TWSSP(NWDP,1)	TWSSP(NWDP,1)	TWSSP(NWDP,2)	WSSP(NWDP,2)	---
---------------	---------------	---------------	--------------	-----

10

20

30

40

80

⋮

---	TWSSP(NWDP,NWDP)	WSSP(NWDP,NWDP)	
-----	------------------	-----------------	--

80

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

WSSP(I,J) = Source/sink value of J-th data point in I-th well source/sink profile, (L*3/T).

- (e) Global Nodal Number of Compressed Well Source/Sink Nodes. The number of cards required in this sub data set depends on NWNP. Each card contains 16 nodes with FORNAT(16I5).

NPN(1)	NPN(2)	NPN(3)	---	NPN(I)	---	NPN(NWNP)	
--------	--------	--------	-----	--------	-----	-----------	--

5

10

15

80

NPN(I) = Global node number of I-th well source/sink node.

- (f) Type of Well Source/Sink Nodes: FORNAT(5I5)
Normally one card per well node, i.e. a total of NWNP cards. However, if the well nodes appear in regular pattern, automatic generation may be made.

NI	NSEQ	NILE	NITYP	NITYPA	
----	------	------	-------	--------	--

5

10

15

20

25

80

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

APPENDIX A. (continued)

NSEQ = NSEQ subsequent well nodes will be generated automatically.

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

NIITP = Type of well source/sink profile assigned to node NI.

NIITPA = Increment of NIITP for each of the NSEQ subsequent nodes.

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

13. **DIRICHLET BOUNDARY CONDITIONS:** This data set is required if **NDNF** > 0.

(a) Data Description: **FORMAT(20A4)** - One card per problem.

DATNAM

80

DATNAM = 80 columns of characters for describing the nature of this data set.

- (b) **Dirichlet-head Profiles:** **FORMAT(8D10.3)** - This sub-data set is read in similar to that in sub-data set 12 (b).

First profile

THED(1,1)	HED(1,1)	THED(1,2)	HED(1,2)	---
10	20	30	40	80

---	THED(1,NDDP)	HED(1,NDDP)	
			80

Second profile

THED(2,1)	HED(2,1)	THED(2,2)	HED(2,2)	---
10	20	30	40	80

APPENDIX A. (continued)

---	THED(2,NDNP)	HED(2,NDNP)	
-----	--------------	-------------	--

80

⋮

I-th profile

THED(I,1)	HED(I,1)	THED(I,2)	HED(I,2)	---
-----------	----------	-----------	----------	-----

10

20

30

40

80

⋮

---	THED(I,NDNP)	HED(I,NDNP)	
-----	--------------	-------------	--

80

⋮

NDNP-th profile

THED(NDNP,1)	HED(NDNP,1)	THED(NDNP,2)	HED(NDNP,2)	---
--------------	-------------	--------------	-------------	-----

10

20

30

40

80

⋮

---	THED(NDNP,NDNP)	HED(NDNP,NDNP)	
-----	-----------------	----------------	--

80

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

HED(I,J) = Head value of J-th data point in I-th
Dirichlet-head profile, (L).

- (c) Dirichlet Nodes: FORMAT(16I5) - The number of cards in this sub-data set depends on NDNP. Each card contains 16 nodes.

NPD(1)	NPD(2)	NPD(3)	---	NPD(I)	---	NPD(NDNP)	
--------	--------	--------	-----	--------	-----	-----------	--

5

10

15

80

NPD(I) = Global node number of I-th Dirichlet
node.

APPENDIX A. (continued)

- (d) Type of Dirichlet Node: FORNAT(5I5) - 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.

NI	NSEQ	NIAD	NIITYP	NIITYPA	
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.

NIAD = Increment of compressed Dirichlet node number for each of the NSEQ nodes.

NIITYP = Type of Dirichlet-head profile for node NI.

NIITYPA = Increment of NIITYP for each of the NSEQ subsequent nodes.

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

14. NEUMANN BOUNDARY CONDITIONS: This data set is required if and only if NDNP > 0.

- (a) Data Description: FORNAT(20A4) - One card per problem.

DATNAM
80

DATNAM = 80 columns of characters for describing the nature and purpose of this data set.

- (b) Prescribed Neumann Flux Profiles: FORNAT(8D10.3) - This sub-data set is read in similar to that in sub-data set (2) (b).

First profile

TRNF(1,1)	RNF(1,1)	TRNF(1,2)	RNF(1,2)	---
10	20	30	40	80

APPENDIX A. (continued)

---	TRNF(1,NNDP)	RNF(1,NNDP)	
-----	--------------	-------------	--

80

Second profile

TRNF(2,1)	RNF(2,1)	TRNF(2,2)	RNF(2,2)	---
10	20	30	40	80

---	TRNF(2,NNDP)	RNF(2,NNDP)	
-----	--------------	-------------	--

80

I-th profile

TRNF(I,1)	RNF(I,1)	TRNF(I,2)	RNF(I,2)	---
10	20	30		80

---	TRNF(I,NNDP)	RNF(I,NNDP)	
-----	--------------	-------------	--

80

NNPR-th profile

TRNF(NNPR,1)	RNF(NNPR,1)	TRNF(NNPR,2)	RNF(NNPR,2)	---
10	20	30	40	80

---	TRNF(NNPR,NNDP)	RNF(NNPR,NNDP)	
-----	-----------------	----------------	--

80

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

RNF(I,J) = Value of Neumann flux of J-th data
point in I-th Neumann flux profile,
($L \cdot 3/T/L \cdot 2$)

APPENDIX A. (continued)

- (c) Neumann Nodes: **FORMAT(1615)** - The number of cards required in this sub-data set depends on **NNNP**. Each card contains 16 nodes.

NNN(1)	NNN(2)	NNN(3)	---	NNN(I)	---	NNN(NNNP)	---
5	10	15					80

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

- (d) Type of Neumann Node: **FORMAT(515)**

NI	NSEQ	NIAD	NI1YP	NI1PAD	
5	10	15	20	25	80

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

NSEQ = The type of **NSEQ** subsequent Neumann flux profiles will be generated automatically.

NIAD = Increment of compressed Neumann node number in each of the **NSEQ** nodes.

NI1YP = Type of Neumann flux profile for node **NI**.

NI1PAD = Increment of **NI1YP** for each of the **NSEQ** subsequent nodes.

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

- (e) Neumann Boundary Element-sides: **FORMAT(1615)**

NI	NSEQ	IS1	IS2	IS1AD	IS2AD	
5	10	15	20	25	30	80

NI = Compressed element-side number of the first element-side in the sequence.

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

IS1 = Compressed Neumann node number of the first node in the element-side **NI**.

APPENDIX A. (continued)

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

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

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

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

15. UPPER LEAKING AQUIFERS

(a) Data Description: FORMAT(20A8)

DATNAM = 80 columns of characters for describing the name and purpose of this data set.

(b) Head Profiles in the Upper Leaking Aquifers: FORMAT(8D10.3)
This sub-data set is read in similar to that in sub-data set 12 (b). The number of cards depends on NHUPR and NHUDP. The data is read in NHUPR-wisely.

First profile

Second profile

APPENDIX A. (continued)

-----	THUTAB (2, NHUP)	HUTAB (2, NHUP)	
-------	------------------	-----------------	--

80

.
 .
 .

I-th profile

THUTAB (I, 1)	HUTAB (I, 1)	THUTAB (I, 2)	-----
---------------	--------------	---------------	-------

10

20

30

80

.
 .

-----	THUTAB (I, NHUP)	HUTAB (I, NHUP)	
-------	------------------	-----------------	--

80

.
 .

NHUP-th profile

THUTAB (NHUP, 1)	HUTAB (NHUP, 1)	THUTAB (NHUP, 2)	-----
------------------	-----------------	------------------	-------

10

20

30

80

.
 .

-----	THUTAB (NHUP, NHUP)	HUTAB (NHUP, NHUP)	
-------	---------------------	--------------------	--

80

THUTAB (I, J) = Time of J-th data point in I-th upper-aquifer-head vs time profile, (I).

HUTAB (I, J) = Head in the upper leaking aquifer of J-th data point in I-th profile, (I).

- (c) Leaking Coefficient in the Upper Aquifer: FORMAT (8D10.3)
The number of cards depends on NHUP. Each card contains 8 points of the NHUP values.

KKU (1)	KKU (2)	---	KKU (I)	---	KKU (NHUP)	
---------	---------	-----	---------	-----	------------	--

10

20

80

KKU (I) = Leaking coefficient of I-th upper leaking aquifer type, (I/I).

APPENDIX A. (continued)

(d) Type of Upper Leaking Aquifers: FORMAT(5I5)

N	NSEQ	NAD	NTYP	NTYPAD	
5	10	15	20	25	80

N = Global element number of the first element in the sequence that will have the upper leaking aquifer type automatically generated.

NSEQ = NSEQ subsequent elements will have their overlaying leaky aquifer type generated automatically.

NAD = Increment of element number for each of the NSEQ elements.

NTYP = Upper leaking aquifer type for element N.

NTYPAD = Increment of NTYP for each of the NSEQ subsequent elements.

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

16. LOWER LEAKING AQUIFERS: This data set is read in identical to that in Data Set 15.

(a) Data Description: FORMAT(20A8)

DATNAM
80

DATNAM = 80 columns of characters for describing the data set.

(b) Head Profiles in the Lower Leaking Aquifers: FORMAT(8D10.3)

First profile

THLTAB(1,1)	HLTAB(1,1)	THLTAB(1,2)	-----
10	20	30	80

:

APPENDIX A. (continued)

-----	THLTAB (1, NMLDP)	HLTAB (1, NMLDP)	
			80

Second profile

THLTAB (2, 1)	HLTAB (2, 1)	THLTAB (2, 2)	-----
10	20	30	80

:

-----	THLTAB (2, NMLDP)	HLTAB (2, NMLDP)	
			80

:

I-th profile

THLTAB (I, 1)	HLTAB (I, 1)	THLTAB (I, 2)	-----
10	20	30	80

:

-----	THLTAB (I, NMLDP)	HLTAB (I, NMLDP)	
			80

:

NMLPR-th profile

THLTAB (NMLPR, 1)	HLTAB (NMLPR, 1)	THLTAB (NMLPR, 2)	-----
10	20	30	80

:

-----	THLTAB (NMLPR, NMLDP)	HLTAB (NMLPR, NMLDP)	
			80

THLTAB (I, J) = Time of J-th data point in I-th
lower-aquifer-head vs time profile,
(T).

HLTAB (I, J) = Head in the lower leaking aquifer of
J-th data point in I-th profile.

APPENDIX A. (continued)

(c) Leaking Coefficients in the Lower Aquifers: FORMAT(8D10.3)

RKL(1)	RKL(2)	-----	RKL(I)	-----	RKL(NHLPB)	
10	20					80

RKL(I) = Leaking coefficient of I-th profile, (1/T).

(d) Type of Lower Leaking Aquifer: FORMAT(5I5)

N	NSEQ	NAD	NTYP	NTYPAD	
5	10	15	20	25	80

N = Global element number of the first element in the sequence that will have the lower leaking aquifer type generated automatically.

NSEQ = NSEQ subsequent elements will have their underlying leaky aquifer type generated automatically.

NAD = Increment of elements for each of the NSEQ subsequent elements.

NTYP = Lower leaking aquifer type for element N.

NTYPAD = Increment of NTYP for each of the NSEQ subsequent elements.

*** 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.

APPENDIX B

OUTPUT

APPENDIX B: OUTPUT

This appendix contains the outputs (see microfiche, inside back cover) of all five examples presented in the report. The correspondence between the microfiche title and example output is included in the following list.

Microfiche Title		Example Output	
TGYXFEW1	DATA 05/07/83 TIME 14.01.41	16662	Verification Example 1
TGYXFEW2	DATA 05/07/83 TIME 14.01.34	16662	Verification Example 2
TGYXFEW3	DATA 05/07/83 TIME 14.01.31	16662	Verification Example 3
TGYXFEW3	DATA 05/10/83 TIME 04.47.07	16662	Verification Example 3 with Triangular Element
TGYXFEW4	DATA 05/07/83 TIME 14.01.46	16662	Demonstration Example 1
TGYXFEW5	DATA 05/07/83 TIME 14.01.55	16662	Demonstration Example 2

APPENDIX C**LISTING OF FORTRAN IV SOURCE PROGRAM**

APPENDIX C: LISTING OF FORTRAN IV SOURCE PROGRAM

```

C C HAIN 005
C * HAIN 010
C * HAIN 015
C * HAIN 020
C * HAIN 025
C * HAIN 030
C * HAIN 035
CC HAIN 040
C HAIN 045
CC HAIN 050
C YEH, G. T. AND D. E. HOPE, 1983. "FEWA: A FINITE ELEMENT MODEL OF HAIN 055
C WATER FLOW THROUGH AQUIFERS", ORNL- , OAK RIDGE NATIONAL LAB., HAIN 060
C OAK RIDGE, TENNESSEE 37830. HAIN 065
CC HAIN 070
C HAIN 075
CC HAIN 080
C ***** HAIN 085
C * HAIN 090
C * DEFINITIONS OF VARIABLES HAIN 095
C * HAIN 100
C * MAXNP = MAXIMUM NO. OF NODAL POINTS HAIN 105
C * MAXEL = MAXIMUM NO. OF ELEMENTS HAIN 110
C * MAXBWP = MAXIMUM NO. OF HALF BAND-WIDTH PLUS ONE HAIN 115
C * MAXBEL = MAXIMUM NO. OF BOUNDARY ELEMENT SIDES HAIN 120
C * MAXBNP = MAXIMUM NO. OF BOUNDARY NODAL POINTS HAIN 125
C * MAXMAM = MAXIMUM NO. OF MATERIALS HAIN 130
C * MAXMTI = MAXIMUM NO. OF TIME INCREMENTS HAIN 135
C * MAXDTC = MAXIMUM NO. OF TIMES TO RESET INITIAL TIME STEP SIZE HAIN 140
C * HAIN 145
C * HAIN 150
C * MNP = NO. CF NODAL PCINTS HAIN 155
C * NEL = NO. OF ELEMENTS HAIN 160
C * IBAND = BAND-WIDTH OF THE GLOBAL MATRIX HAIN 165
C * INBP = HALF BAND-WIDTH PLUS ONE HAIN 170
C * IHALFP = HALF BAND-WIDTH OF THE GLOBAL MATRIX HAIN 175
C * NBEL = NO. CF BOUNDARY ELEMENT SIDES HAIN 180
C * MBN = NO. CF BOUNDARY NODES HAIN 185
C * HAIN 190
C * HAIN 195
C * MTI = NO. OF TIME INCREMENTS OR NO. OF TIME STEPS HAIN 200
C * MMAT = NO. OF MATERIALS HAIN 205
C * MSTRT = NO. OF RESTART RECORDS TO BE READ VIA LOGICAL UNIT 2 HAIN 210
C * NITER = NO. OF ITERATIONS ALLOWED FOR SOLVING NONLINEAR EQUATIONS* HAIN 215
C * KSTR = CONTROL STORAGE ON LOGICAL UNIT 1, 0=NO/, 1=YES HAIN 220
C * HAIN 225
C * KPRO = LINE PRINTER CONTROL FOR THE STEADY STATE OR PRE-INITIAL HAIN 230
C * CONDITIONS HAIN 235
C * 0= PRINT NOTHING, 1=PRINT FLUXES THROUGH ALL BOUNDARIES HAIN 240
C * 2=PRINT PRESSURE HEAD ALSO, 3=PRINT DARCY VELOCITY ALSO HAIN 245
C * HAIN 250
C * KDSK = DISK STORAGE CONTROL FOR THE STEADY STATE OR PRE-INITIAL HAIN 255
C * CONDITIONS, 0=MC STORAGE, 1=YES HAIN 260
C * HAIN 265
C * KSS = STEADY STATE SOLUTION DESIRED? 0=YES, 1=NO HAIN 270
C * NDTCNG = NO. OF TIMES TO RESET TIME STEP SIZE HAIN 275
C * HAIN 280
C * HAIN 285
C * TOLA = ERROR TOLERANCE FOR ITERATING STEADY STATE SOLUTION HAIN 290
C * TOLB = ERROR TOLERANCE FOR ITERATING TRANSIENT SOLUTION HAIN 295
C * DELT = INITIAL TIME STEP SIZE. (T) HAIN 300

```

APPENDIX C. (continued)

C * CHNG	= PERCENTAGE CHANGE IN TIME STEP SIZE, (DECIMAL VALUES)	*	MAIN 305
C * DELHAX	= MAXIMUM VALUE OF TIME STEP SIZE ALLOWED, (T)	*	MAIN 310
C * THAX	= MAXIMUM SIMULATION TIME, (T)	*	MAIN 315
C *		*	MAIN 320
C *		*	MAIN 325
C * MXNBP	= MAXIMUM NO. OF NEUMANN BOUNDARY NODAL POINTS	*	MAIN 330
C * MXNEL	= MAXIMUM NO. OF NEUMANN BOUNDARY ELEMENT SIDES	*	MAIN 335
C * MXNBP	= MAXIMUM NO. OF NEUMANN BOUNDARY PROFILES	*	MAIN 340
C * MXNDP	= MAXIMUM NO. OF NEUMANN BOUNDARY PARAMETERS FOR EACH PROF.	*	MAIN 345
C * MNBP	= NO. OF NEUMANN BOUNDARY NODAL POINTS	*	MAIN 350
C * MNEL	= NO. OF NEUMANN BOUNDARY ELEMENT SIDES	*	MAIN 355
C * MNBP	= NO. OF NEUMANN BOUNDARY PROFILES	*	MAIN 360
C * MNDP	= NO. OF NEUMANN BOUNDARY PARAMETERS FOR EACH PROFILE	*	MAIN 365
C *		*	MAIN 370
C * MXDNP	= MAXIMUM NO. OF DIRICHLET BOUNDARY NODAL POINTS	*	MAIN 375
C * MXDPE	= MAXIMUM NO. OF DIRICHLET BOUNDARY PROFILES	*	MAIN 380
C * MXDDP	= MAXIMUM NO. OF DIRICHLET BOUNDARY PARAMETERS FOR EACH PR	*	MAIN 385
C * MDNP	= NO. OF DIRICHLET BOUNDARY NODAL POINTS	*	MAIN 390
C * MDPE	= NO. OF DIRICHLET BOUNDARY PROFILES	*	MAIN 395
C * MDDP	= NO. OF DIRICHLET BOUNDARY PARAMETERS FOR EACH PROFILE	*	MAIN 400
C *		*	MAIN 405
C * MXSPR	= MAXIMUM NO. OF SOURCE PROFILES	*	MAIN 410
C * MXSDP	= MAXIMUM NO. OF SOURCE PARAMETERS FOR EACH PROFILE	*	MAIN 415
C * MSPR	= NO. OF SOURCE PROFILES	*	MAIN 420
C * MSDP	= NO. OF SOURCE PARAMETERS FOR EACH PROFILE	*	MAIN 425
C * MXWNP	= MAXIMUM NO. OF WELL NODAL POINTS	*	MAIN 430
C * MWNP	= NO. OF WELL NODAL POINTS	*	MAIN 435
C * MXWPR	= MAXIMUM NO. OF WELL SOURCE PROFILES	*	MAIN 440
C * MXWDP	= MAXIMUM NO. OF DATA POINT IN EACH WELL SOURCE PROFILE	*	MAIN 445
C * MWPR	= NO. OF WELL SOURCE/SINK PROFILES	*	MAIN 450
C * MWDP	= NO. OF DATA POINT IN EACH OF THE WELL SOURCE/SINK PROFILE	*	MAIN 455
C *		*	MAIN 460
C * MXHUPR	= MAXIMUM NO. OF HEAD-IN-UPPER-AQUIFER (HU) PROFILES	*	MAIN 465
C * MXHUDP	= MAXIMUM NO. OF DATA POINTS IN EACH HU-TIME(T) PROFILE	*	MAIN 470
C * MHUPR	= NO. OF HU-T PROFILES	*	MAIN 475
C * MHUDP	= NO. OF DATA POINTS IN EACH HU-T PROFILE	*	MAIN 480
C * MXHLP	= MAXIMUM NO. OF HEAD-IN-LOWER-AQUIFER (HL) PROFILES	*	MAIN 485
C * MXHLP	= MAXIMUM NO. OF DATA POINTS IN EACH HL-TIME(T) PROFILE	*	MAIN 490
C * MHLPR	= NO. OF HL-T PROFILES	*	MAIN 495
C * MHLP	= NO. OF DATA POINTS IN EACH HL-T PROFILE	*	MAIN 500
C *		*	MAIN 505
C * ILUMP	= MASS MATRIX LUMPING OPTIONS, 0=NO LUMPING, 1=YES	*	MAIN 510
C * INID	= MID-DIFFERENCE OPTION, 0=NO MID-DIFFERENCE, 1=YES	*	MAIN 515
C * ICONFI	= IS CONFINED AQUIFER? 0=NO, 1=YES	*	MAIN 520
C * IPNTS	= IS POINTWISE SOLUTION TO BE USED; 0 = NO, 1 = YES.	*	MAIN 525
C *		*	MAIN 530
C *		*	MAIN 535
C * I(N)	= X-COORDINATE OF N-TH NODE, (L)	*	MAIN 540
C * Z(N)	= Z-COORDINATE OF N-TH NODE, (L)	*	MAIN 545
C * B(N)	= BED ROCK ELEVATION OF N-TH NODE, (L)	*	MAIN 550
C * IE(N,1)	= GLOBAL NODE NUMBER OF THE FIRST NODE OF N-TH ELEMENT	*	MAIN 555
C * IE(N,2)	= GLOBAL NODE NUMBER OF THE SECOND NODE OF N-TH ELEMENT	*	MAIN 560
C * IE(N,3)	= GLOBAL NODE NUMBER OF THE THIRD NODE OF N-TH ELEMENT	*	MAIN 565
C * IE(N,4)	= GLOBAL NODE NUMBER OF THE FOURTH NODE OF N-TH ELEMENT	*	MAIN 570
C *	IF N-TH ELEMENT IS A TRIANGULAR ELEMENT, IE(N,4)=0	*	MAIN 575
C * IE(N,5)	= INTEGER TO INDICATE THE MATERIAL TYPE OF N-TH ELEMENT	*	MAIN 580
C *	IF N-TH ELEMENT IS UNCONFINED, IE(N,5) SHOULD BE (-)	*	MAIN 585
C *		*	MAIN 590
C * C(N,I)	= AN ARRAY TO STORE THE ASSEMBLED GLOBAL MATRIX	*	MAIN 595
C * LLNODE(I,N)	= GLOBAL NODE NUMBER OF THE I-TH COMPRESSED LOWER-LEFT	*	MAIN 600

APPENDIX C. (continued)

```

C *      NODE TO THE N-TH GLOBAL NODE *      MAIN 605
C * R(N) = 1N ARRAY TO STORE THE ASSEMBLED GLOBAL LOAD VECTOR *      MAIN 610
C * H(N) = PRESSURE HEAD AT N-TH NODE, (L) *      MAIN 615
C * HP(N) = PRESSURE HEAD OF N-TH NODE AT PREVIOUS TIME STEP, (L) *      MAIN 620
C * HW(N) = PRESSURE HEAD OF N-TH NODE AT ITERATE, (L) *      MAIN 625
C * NPCNV(I) = AN ARRAY TO STORE THE NODE NUMBER OF I-TH *      MAIN 630
C *      NON-CONVERGENT POINT *      MAIN 635
C * VX(N) = X-COMPONENT VERTICALLY INTEGRATED FLUX AT NODE N, (L**2/T) *      MAIN 640
C * VZ(N) = Z-COMPONENT VERTICALLY INTEGRATED FLUX AT NODE N, (L**2/T) *      MAIN 645
C * AKX(N,I) = IX-COMPONENT OF THE TRANSMISSIBILITY TENSOR AT *      MAIN 650
C *      I-TH NODE OF N-TH ELEMENT, (L**2/T) *      MAIN 655
C * AKZ(N,I) = ZZ-COMPONENT OF THE TRANSMISSIBILITY TENSOR AT *      MAIN 660
C *      I-TH NODE OF N-TH ELEMENT, (L**2/T) *      MAIN 665
C * AKXZ(N,I) = XZ- OR ZX- COMPONENT OF THE TRANSMISSIBILITY *      MAIN 670
C *      TENSOR AT I-TH NODE OF N-TH ELEMENT, (L**2/T) *      MAIN 675
C * *      MAIN 680
C * DLB(I) = LENGTH OF I-TH BOUNDARY ELEMENT SIDE, (L) *      MAIN 685
C * DCOSXB(I) = X-DIRECTIONAL COSINE OF I-TH BOUNDARY ELEMENT SIDE *      MAIN 690
C * DCOSZB(I) = Z-DIRECTIONAL COSINE OF I-TH BOUNDARY ELEMENT SIDE *      MAIN 695
C * BFLX(I) = BOUNDARY FLUX AT I-TH BOUNDARY NODE, (L**3/SEC/L) *      MAIN 700
C * BFLXK(I) = BOUNDARY FLUX AT PREVIOUS TIME OF I-TH BOUNDARY NODE *      MAIN 705
C * NBE(I) = ELEMENT NUMBER OF I-TH BOUNDARY ELEMENT SIDE *      MAIN 710
C * ISB(I,1) = GLOBAL NODE NUMBER OF THE FIRST NODE OF I-TH BOUNDARY *      MAIN 715
C *      ELEMENT SIDE *      MAIN 720
C * ISB(I,2) = GLOBAL NODE NUMBER OF THE SECOND NODE OF I-TH BOUNDARY *      MAIN 725
C *      ELEMENT SIDE *      MAIN 730
C * ISB(I,3) = LOCAL NODE NUMBER OF THE FIRST NODE OF I-TH BOUNDARY *      MAIN 735
C *      ELEMENT SIDE *      MAIN 740
C * ISB(I,4) = LOCAL NODE NUMBER OF THE SECOND NODE OF I-TH BOUNDARY *      MAIN 745
C *      ELEMENT SIDE *      MAIN 750
C * NPB(I) = GLOBAL NODE NUMBER OF I-TH BOUNDARY NODE *      MAIN 755
C * *      MAIN 760
C * *      MAIN 765
C * NPN(I) = GLOBAL NODAL POINT NUMBER OF I-TH NEUMANN BOUNDARY NODE *      MAIN 770
C * INTYP(I) = TYPE OF NEUMANN FLUX ASSIGNED TO I-TH NEUMANN NODE *      MAIN 775
C * RNF(I,J) = NEUMANN FLUX OF J-TH DATA POINT IN I-TH TYPE *      MAIN 780
C *      --OUT FROM THE REGION, --INTO THE REGION, (L/T) *      MAIN 785
C * TRNF(I,J) = TIME OF J-TH DATA POINT IN I-TH NEUMANN FLUX PROFILE *      MAIN 790
C * RFLX(I) = NEUMANN FLUX OF I-TH TYPE AT THE PRESENT TIME, (L/T) *      MAIN 795
C * IS(I,1) = COMPRESSED NEUMANN BOUNDARY NODE NUMBER OF FIRST NODE *      MAIN 800
C *      OF I-TH NEUMANN BOUNDARY ELEMENT SIDE *      MAIN 805
C * IS(I,2) = COMPRESSED NEUMANN BOUNDARY NODE NUMBER OF SECOND NODE *      MAIN 810
C *      OF I-TH NEUMANN BOUNDARY ELEMENT SIDE *      MAIN 815
C * *      MAIN 820
C * NPD(I) = GLOBAL NODAL POINT NUMBER OF I-TH DIRICHLET NODE *      MAIN 825
C * IDTYP(I) = TYPE OF DIRICHLET HEAD ASSIGNED TO I-TH DIRICHLET NODE *      MAIN 830
C * RED(I,J) = DIRICHLET HEAD OF J-TH DATA POINT IN I-TH TYPE, (L) *      MAIN 835
C * THED(I,J) = TIME OF J-TH DATA POINT IN I-TH DIRICHLET HEAD TYPE *      MAIN 840
C * GHED(I) = GIVEN DIRICHLET HEAD OF I-TH TYPE AT THE PRESENT TIME *      MAIN 845
C *      OBTAINED BY INTERPOLATING WITH TIME FROM THED AND RED *      MAIN 850
C * *      MAIN 855
C * ISTYP(J) = TYPE OF SOURCE PROFILE ASSIGNED TO N-TH ELEMENT *      MAIN 860
C * SOS(I,J) = SOURCE OF J-TH DATA POINT IN I-TH PROFILE, (L**3/T/L**2) *      MAIN 865
C * TSOS(I,J) = TIME OF J-TH DATA POINT IN I-TH SOURCE PROFILE, (T) *      MAIN 870
C * SOJS(I) = VALUE OF THE SOURCE OF I-TH TYPE AT THE PRESENT TIME *      MAIN 875
C *      OBTAINED BY INTERPOLATING WITH TIME FROM TSOS AND SOS *      MAIN 880
C * *      MAIN 885
C * NPW(I) = GLOBAL NODE NUMBER OF I-TH WELL NODE *      MAIN 890
C * INTYP(I) = TYPE OF WELL SOURCE/SINK PROFILE ASSIGNED TO I-TH WELL *      MAIN 895
C * WSS(I) = VALUE OF I-TH WELL SOURCE/SINK AT PRESENT TIME, (L**3/T) *      MAIN 900

```

APPENDIX C. (continued)

```

C * WSSP(I,J) = WELL SOURCE/SINK FUNCTION OF J-TH DATA POINT OF I-TH *      NAIN 905
C *   PROFILE, (L**3/T). *      NAIN 910
C * TWSSP(I,J) = TIME OF J-TH DATA POINT OF I-TH WELL SOURCE/SINK *      NAIN 915
C *   PROFILE, (T) *      NAIN 920
C * *      NAIN 925
C * THUTAB(I,J) = TIME OF J-TH DATA POINT IN I-TH HU-T PROFILE, (T) *      NAIN 930
C * BUTAB(I,J) = HU VALUE OF J-TH DATA POINT IN I-TH HU-T PROFILE, (L) *      NAIN 935
C * HU(I) = HU VALUE IN I-TH HU-T PROFILE AT THE PRESENT TIME, (L) *      NAIN 940
C * RLU(I) = LEAKING COEFFICIENT IN UPPER AQUIFER OF I-TH TYPE, (1/T) *      NAIN 945
C * IHUTYP(N) = TYPE OF HU-T PROFILE ASSIGNED TO N-TH ELEMENT *      NAIN 950
C * *      NAIN 955
C * THL(I,J) = TIME OF J-TH DATA POINT IN I-TH HL-T PROFILE, (T) *      NAIN 960
C * HLTAB(I,J) = HL VALUE OF J-TH DATA POINT IN I-TH HL-T PROFILE, (L) *      NAIN 965
C * HL(I) = HL VALUE IN I-TH HL-T PROFILE AT THE PRESENT TIME, (L) *      NAIN 970
C * RKL(I) = LEAKING COEFFICIENT IN LOWER AQUIFER OF I-TH TYPE, (1/T) *      NAIN 975
C * IHUTYP(N) = TYPE OF HL-T PROFILE ASSIGNED TO N-TH ELEMENT *      NAIN 980
C * *      NAIN 985
C * *      NAIN 990
C * *      NAIN 995
C * PROP(I,J) = J-TH MATERIAL PROPERTY OF I-TH MATERIAL *      NAIN 1000
C *   PROP(I,1) = MODIFIED COMPRESSIBILITY OF MEDIA, (1/L) *      NAIN1000
C *   PROP(I,2) = MODIFIED COMPRESSIBILITY OF WATER, (1/L) *      NAIN1005
C *   PROP(I,3) = POROSITY, (DIMENSIONLESS IN DECIMAL POINT) *      NAIN1010
C *   PROP(I,4) = XX-HYDRAULIC CONDUCTIVITY OR PERMEABILITY *      NAIN1015
C *   PROP(I,5) = ZZ-HYDRAULIC CONDUCTIVITY OR PERMEABILITY *      NAIN1020
C *   PROP(I,6) = YZ-HYDRAULIC CONDUCTIVITY OR PERMEABILITY *      NAIN1025
C *   PROP(I,7) = SPECIFIC YIELD OF UNCONFINED AQUIFER, *      NAIN1030
C * *      NAIN1035
C * *      NAIN1040
C * KPR(I) = LINE PRINTING FOR I-TH TIME STEP? *      NAIN1045
C *   0= PRINT NOTHING, 1=PRINT FLOW AND FLUX, 2=ALSO PRESSURE *      NAIN1050
C *   >3=ALSO DARCY VELOCITY *      NAIN1055
C * *      NAIN1060
C * KDSK(I) = STORE RESULTS ON LOGICAL UNIT I? 0=NO, 1=YES *      NAIN1065
C * TDTC(I) = TIME OF I-TH TIME TO RESET TIME STEP SIZE = DELTO, (T) *      NAIN1070
C *   SHOULD BE DIMENSIONED AS TDTC(MXNDTC) *      NAIN1075
C * *      NAIN1080
C * *      NAIN1085
C * PRATE(I) = FLOW RATE THROUGH I-TH BOUNDARY, (L**3/T) *      NAIN1090
C * FLOW(I) = AMOUNT OF FLOW THROUGH I-TH BOUNDARY IN PRESENT TIME *      NAIN1095
C *   STEP, (L**3) *      NAIN1100
C * FLOW(I) = ACCUMULATED FLOW VOLUME THROUGH I-TH BOUNDARY, (L**3) *      NAIN1105
C * WHERE I=1 IS THE DIRICHLET BOUNDARY, I=2 IS THE NEUMANN BOUNDARY, *      NAIN1110
C *   I=3 IS THE UPPER LEAKY AQUIFERS, I=4 IS THE LOWER LEAKY *      NAIN1115
C *   AQUIFERS, I=5 IS THE NUMERICAL LOSS, I=6 IS THE WHOLE *      NAIN1120
C *   BOUNDARY, I=7 IS THE ACCUMULATED IN THE REGION *      NAIN1125
C *   I=8 IS THE ARTIFICIAL SOURCES AND SINK *      NAIN1130
C * *      NAIN1135
C * *      NAIN1140
C * ***** *      NAIN1145
C * *      NAIN1150
C * *      NAIN1155
C * MAIN PROGRAM *      NAIN1160
C *   IMPLICIT REAL*8(A-H,O-Z) *      NAIN1165
C * *      NAIN1170
C * COMMON /SIZE/ MAXNP,MAXEL,MAXHP,MAXBL,MAXBP,MAXMT,MAXNTI,MXDTC *      NAIN1175
C * COMMON /GEOM/ NNP,NEL,IBAND,INBP,IBALPB,NBEL,NBN *      NAIN1180
C * COMMON /CTRL/ NTI,WHAT, WITER,WSTRT,KSTR,KPRO,KDSKO,KSS,WDTCG *      NAIN1185
C * COMMON /TOTLS/ TOLA,TOLB *      NAIN1190
C * COMMON /PARAM/ DELT,CBNG,DELHAX,THAX,DELTO *      NAIN1195
C * COMMON /CN/ HXNBP,HXNEL,HXNBP,HXNDP,HNBP,HNEL,HNBP,HNDP *      NAIN1200

```


APPENDIX C. (continued)

```

COMMON /CD/ HXDHP,HXDPR,HXDDP,HDXP,HDXR,HDDP
COMMON /CS/ HXSHP,HXSDP,HSPR,HSDP,HXWHP,HXWPR,HXWDP,HWWP,HWPR,HWDP
COMMON /OPT/ ILHNP,IBID,ICCNFI,IPNTS
C
COMMON /KUH/ HXHUPR,HXHUDP,HXUPR,HXUDP
COMMON /KLHL/ HXLPR,HXLDP,HHLPR,HHLDP
C
DIMENSION X(595),Z(595),B(595),IE(528,5)
C
DIMENSION C(595,16),R(595),H(595),HP(595),HW(595),HPCNV(595),
> VX(595),VZ(595),AKI(528,4),AKZ(528,4),AKXZ(528,4)
DIMENSION ILNDE(5,595)
C
DIMENSION DLE(199),DCCSXB(199),DCOSZE(199),BFLX(200),BFLXP(200),
> HBE(199),ISB(199,4),HFB(200)
C
DIMENSION TRNP(10,8),RNP(10,8),RFLX(10)
DIMENSION INTYP(21),HFN(21),IS(20,2)
C
DIMENSION THED(20,8),HED(20,8),GHED(20)
DIMENSION IDTYP(30),HFD(30)
C
DIMENSION ISOS(6,8),SCS(6,8),SOUS(6),ISTYP(528)
DIMENSION TWSSP(6,8),WSSP(6,8),WSS(6),INTYP(10),HFN(10)
C
DIMENSION THUTAB(2,8),HUTAB(2,8),ZT(2),RKU(2),IHUTYP(528)
DIMENSION THLTAB(2,8),HLTAB(2,8),HL(2),RKL(2),IHLTYP(528)
C
DIMENSION PROP(3,7)
C
DIMENSION KPR(500),KDSK(500),TDTCH(20)
C
DIMENSION PRATE(10),FLOW(10),TFLOW(10)
C
----- INITIATE ARRAYS FOR NODAL POINTS
C
DO 110 NP=1,MAXNP
  X(NP)=0.0
  Z(NP)=0.0
  R(NP)=0.0
  H(NP)=0.0
  HP(NP)=0.0
  HW(NP)=0.0
  B(NP)=0.0
  VX(NP)=0.0
  VZ(NP)=0.0
DO 100 IB=1,MAXHBP
  100 C(NP,IB)=0.0
DO 105 I=1,5
  105 LLNDE(I,NP)=0
  110 CONTINUE
C
----- INITIATE ARRAYS FOR ELEMENTS
C
DO 120 NP=1,MAXEL
  ISTYP(NP)=0
  IHUTYP(NP)=0
  IHLTYP(NP)=0
DO 120 IQ=1,5
  120 IE(NP,IQ)=0

```

MAIN1205
 MAIN1210
 MAIN1215
 MAIN1220
 MAIN1225
 MAIN1230
 MAIN1235
 MAIN1240
 MAIN1245
 MAIN1250
 MAIN1255
 MAIN1260
 MAIN1265
 MAIN1270
 MAIN1275
 MAIN1280
 MAIN1285
 MAIN1290
 MAIN1295
 MAIN1300
 MAIN1305
 MAIN1310
 MAIN1315
 MAIN1320
 MAIN1325
 MAIN1330
 MAIN1335
 MAIN1340
 MAIN1345
 MAIN1350
 MAIN1355
 MAIN1360
 MAIN1365
 MAIN1370
 MAIN1375
 MAIN1380
 MAIN1385
 MAIN1390
 MAIN1395
 MAIN1400
 MAIN1405
 MAIN1410
 MAIN1415
 MAIN1420
 MAIN1425
 MAIN1430
 MAIN1435
 MAIN1440
 MAIN1445
 MAIN1450
 MAIN1455
 MAIN1460
 MAIN1465
 MAIN1470
 MAIN1475
 MAIN1480
 MAIN1485
 MAIN1490
 MAIN1495
 MAIN1500

APPENDIX C. (continued)

```

C      DO 140 IQ=1,4
C      AKX(HF,IQ)=0.0
C      AKZ(HF,IQ)=0.0
C      AKYZ(HF,IQ)=0.0
140 CONTINUE
C
C      150 CONTINUE
C
C      ----- INITIATE ARRAYS FOR BOUNDARY ELEMENT SIDES
C
C      DO 180 NP=1,NAXBEL
C      DLB(NP)=0.0
C      DCOSXB(NP)=0.0
C      DCOSZB(NP)=0.0
C      WBF(NP)=0
C      DO 180 IQ=1,4
C      ISB(NP,IQ)=0
180 CONTINUE
C
C      ----- INITIATE ARRAYS FOR BOUNDARY NODAL POINTS
C
C      DO 190 NP=1,NAXBNP
C      BFLX(NP)=0.0
C      BFLXE(NP)=0.0
190 NPB(NP)=0
C
C      ----- INITIATE ARRAYS FOR NEUMANN BOUNDARY ELEMENT SIDES
C
C      DO 240 NP=1,NXNBL
C      DO 240 IQ=1,2
240 IS(NP,IQ)=0
C
C      ----- INITIATE ARRAYS FOR NEUMANN BOUNDARY NODES
C
C      DO 250 I=1,NXNBP
C      NPN(I)=0
C      INTYE(I)=0
250 CONTINUE
C
C      ----- INITIATE ARRAYS FOR NEUMANN FLUX CONDITIONS
C
C      DO 290 I=1,NXNFB
C      RFLX(I)=0.0
C      DO 280 J=1,NXNDP
C      RNF(I,J)=0.0
C      TRNF(I,J)=0.0
280 CONTINUE
290 CONTINUE
C
C      ----- INITIATE ARRAYS FOR DIRICHLET BOUNDARY NODES
C
C      DO 350 I=1,NXDNP
C      NPD(I)=0
C      IDTYF(I)=0
350 CONTINUE
C
C      ----- INITIATE ARRAYS FOR DIRICHLET HEAD CONDITIONS

```

```

NAIN1505
NAIN1510
NAIN151
NAIN1520
NAIN1525
NAIN1530
NAIN1535
NAIN1540
NAIN1545
NAIN1550
NAIN1555
NAIN1560
NAIN1565
NAIN1570
NAIN1575
NAIN1580
NAIN1585
NAIN1590
NAIN1595
NAIN1600
NAIN1605
NAIN1610
NAIN1615
NAIN1620
NAIN1625
NAIN1630
NAIN1635
NAIN1640
NAIN1645
NAIN1650
NAIN1655
NAIN1660
NAIN1665
NAIN1670
NAIN1675
NAIN1680
NAIN1685
NAIN1690
NAIN1695
NAIN1700
NAIN1705
NAIN1710
NAIN1715
NAIN1720
NAIN1725
NAIN1730
NAIN1735
NAIN1740
NAIN1745
NAIN1750
NAIN1755
NAIN1760
NAIN1765
NAIN1770
NAIN1775
NAIN1780
NAIN1785
NAIN1790
NAIN1795
NAIN1800

```

APPENDIX C. (continued)

C	DO 390 I=1,NIDFB	NAIN1805
	GHED(I)=0.0	NAIN1810
	DO 380 J=1,NIDDP	NAIN1815
	HED(I,J)=0.0	NAIN1820
	THED(I,J)=0.0	NAIN1825
	380 CONTINUE	NAIN1830
	390 CONTINUE	NAIN1835
C		NAIN1840
C	----- INITIATE ARRAYS FOR SOURCE CONDITIONS	NAIN1845
C		NAIN1850
	DO 410 I=1,NINWP	NAIN1855
	NPW(I)=0	NAIN1860
	INTYF(I)=0	NAIN1865
410	CONTINUE	NAIN1870
	DO 440 I=1,NINWP	NAIN1875
	WSS(I)=0.0	NAIN1880
	DO 440 J=1,NINWP	NAIN1885
	TWSSF(I,J)=0.0	NAIN1890
	WSSF(I,J)=0.0	NAIN1895
440	CONTINUE	NAIN1900
C		NAIN1905
	DO 490 I=1,NISFE	NAIN1910
	SOUS(I)=0.0	NAIN1915
	DO 480 J=1,NISDP	NAIN1920
	SOS(I,J)=0.0	NAIN1925
	TSOS(I,J)=0.0	NAIN1930
480	CONTINUE	NAIN1935
490	CONTINUE	NAIN1940
C		NAIN1945
C	----- INITIATE ARRAYS FOR MATERIAL PROPERTIES	NAIN1950
C		NAIN1955
	DO 650 I=1,MAXMAT	NAIN1960
	DO 610 J=1,7	NAIN1965
610	PROP(I,J)=0.0	NAIN1970
C		NAIN1975
650	CONTINUE	NAIN1980
C		NAIN1985
	DO 660 I=1,NXHUPB	NAIN1990
	HU(I)=0.0	NAIN1995
	RKU(I)=0.0	NAIN2000
	DO 655 J=1,NXHUDP	NAIN2005
	HUTAB(I,J)=0.0	NAIN2010
	THUTAB(I,J)=0.0	NAIN2015
655	CONTINUE	NAIN2020
660	CONTINUE	NAIN2025
C		NAIN2030
	DO 680 I=1,NXHLPB	NAIN2035
	HL(I)=0.0	NAIN2040
	RKL(I)=0.0	NAIN2045
	DO 675 J=1,NXHIDP	NAIN2050
	HLTAB(I,J)=0.0	NAIN2055
	THLTAB(I,J)=0.0	NAIN2060
675	CONTINUE	NAIN2065
680	CONTINUE	NAIN2070
C		NAIN2075
C		NAIN2080
C		NAIN2085
C	----- INITIATE ARRAYS FOR FLOW THROUGH VARIOUS TYPES OF BOUNDARIES	NAIN2090
C		NAIN2095
		NAIN2100

APPENDIX C. (continued)

DO 700 I=1,10	MAIN2105
PRATE(I)=0.0	MAIN2110
FLOW(I)=0.0	MAIN2115
700 TFLOW(I)=0.0	MAIN2120
C	MAIN2125
C	MAIN2130
C ----- PASS THE PROGRAM TO GW2D	MAIN2135
C	MAIN2140
CALL GW2D(X,Z,B,IZ, C,B,H,HP,HV, V1,V2,AX,AY,AZ,AXZ,	MAIN2145
1 WPCNV, DLB,DCOSIB,DCOSZB,BFLI,BFLXP,UBE,ISB,WPB,	MAIN2150
2 WPN,INTYP,WNF,TRNF,BFLI,IS, WPD,ICTYP,HED,THED,GHED,	MAIN2155
3 ISTYP,SOS,ISOS,SOUS, PHOP,KPR,KDSK, PRATE,FLOW,TFLOW,	MAIN2160
4 THUTAB,HUTAB,HU,RKU,INHUTIP, THLTAB,HLTAB,HL,RKL,INLTYP,TDTCN,	MAIN2165
5 WPN,INTYP,WSS,THSSP,WSSP, LLNODE)	MAIN2170
C	MAIN2175
STOP	MAIN2180
END	MAIN2185

APPENDIX C. (continued)

BLOCK DATA	BLOC 005
IMPLICIT REAL*8 (A-H,O-Z)	BLOC 010
COMMON /SIZE/ HAXHP,HAXEL,HAXHBP,HAXBEL,HAXBNP,HAXHAT,HAXHTI,HIDTC	BLOC 015
COMMON /CH/ HXHP,HXEL,HXHPR,HXHPF,HXHP,HHEL,HHPR,HHDP	BLOC 020
COMMON /CD/ HXDHP,HXDPR,HXDDP,HDXF,HXDP,HDDP	BLOC 025
COMMON /CS/ HXSPR,HXSDP,HXSP,HXSD,HXWHP,HXWPR,HXWDP,HXWP,HXWP,HHDP	BLOC 030
COMMON /KXHP/ HXHPR,HXHPD,HXHP,HHDP	BLOC 035
COMMON /KLEL/ HXELPR,HXELDP,HXELPR,HHDP	BLOC 040
DATA HAXHP,HAXEL,HAXHBP,HAXBEL,HAXBNP/595,528,16,199,200/	BLOC 045
DATA HAXHAT,HAXHTI,HIDTC/3,500,20/	BLOC 050
DATA HXHP,HXEL,HXHPF,HXHP/21,20,10,8/	BLOC 055
DATA HXDHP,HXDPR,HXDDP/30,20,8/	BLOC 060
DATA HXSPR,HXSDP,HXWHP,HXWPR,HXWDP/6,8,10,6,8/	BLOC 065
DATA HXHPR,HXHPD,HXELPR,HXELDP/2,8,2,8/	BLOC 070
END	BLOC 075

APPENDIX C. (continued)

```

SUBROUTINE GW2D(X,Z,B,IE, C,R,H,HP,HV, VX,VZ,AKI,AKZ,AKIZ,
1  WPCNV, DLB,DCOSXB,DCOSZB,BFLX,BFLXP,WBE,ISB,WPB,
2  WPH,INTYP,WNP,TRNP,BFLX,IS, WPD,INTYP,HED,TRED,CHED,
3  ISTYP,SOS,TSOS,SOUS, PROP,KPR,KDSK, PRATE,FLOW,TPLCN,
4  THUTAB,HUTAB,HU,HKU,INUTYP, THLTAB,HLTAB,HL,BKL,IBLTYP,TDTCH,
5  WPH,INTYP,WSS,TWSSP,WSSP, LLNODE)
C
IMPLICIT REAL*8 (A-H,O-Z)
REAL*4 SUBHD
C
COMMON /SIZE/ HAXNP,HAXEL,HAXHNP,HAXBEL,HAXBNP,HAXHAT,HAXHTI,HIDTC
COMMON /GEOM/ HNP,HKL,IBAND,IRBP,IBALPB,WBEL,WBN
COMMON /CTRL/ WTI,HAT, NITER,NSTRT,KSTR,KPRO,KDSKO,KSS,NDTCNG
COMMON /TOLLS/ TOLA,TOLB
COMMON /PARAM/ DELT,CHNG,DELMAX,THAX,DELTO
COMMON /CN/ HXNHP,HXNEL,HXNHP,HXNDP,HXNP,HXEL,HXPR,HNDP
COMMON /CD/ HXDNP,HXDPR,HXDDP,HXNP,HXPR,HDDP
COMMON /CS/ HXSPR,HXSDP,HXSPR,HXSDP,HXNHP,HXNPR,HXNDP,HXNP,HXPR,HDDP
COMMON /KUNP/ HXNUPR,HXNDPR,HXUPR,HXUDP
COMMON /KLHL/ HXHLPR,HXHLDP,HXHLPR,HXHLDP
COMMON /OPT/ ILNP,IBID,ICCNFI,IPNTS
C
DIMENSION TITLE(9)
DIMENSION I (HAXNP),Z (HAXNP),B (HAXNP),IE (HAXEL,5)
C
DIMENSION C (HAXNP,HAXNP),R (HAXNP),H (HAXNP),HP (HAXNP),HV (HAXNP),
> VX (HAXNP),VZ (HAXNP),AKI (HAXEL,4),AKZ (HAXEL,4),AKIZ (HAXEL,4),
> WPCNV (HAXNP)
DIMENSION LLNODE (5,HAXNP)
C
DIMENSION DLB (HAXBEL),DCOSXB (HAXBEL),DCOSZB (HAXBEL),BFLX (HAXBNP),
> BFLXP (HAXBNP),WBE (HAXBEL),ISB (HAXBEL,4),WPB (HAXBNP)
C
DIMENSION WPH (HXNHP),INTYP (HXNHP), IS (HXNEL,2)
DIMENSION WNP (HXNPR,HXNDP), TRNP (HXNPR,HXNDP), BFLX (HXNPR)
C
DIMENSION WPD (HXDNP),INTYP (HXDNP)
DIMENSION HED (HXDPR,HXDDP), TRED ("HXDP,HXDDP), CHED (HXDPR)
C
DIMENSION ISTYP (HAXEL)
DIMENSION SOS (HXSPR,HXSDP), TSOS (HXSPR,HXSDP), SOUS (HXSPR)
DIMENSION WPH (HXNHP),INTYP (HXNHP)
DIMENSION WSS (HXNPR),WSSP (HXNPR,HXNDP),TWSSP (HXNPR,HXNDP)
C
DIMENSION THUTAB (HXHUPR,HXHUPR),HUTAB (HXHUPR,HXHUPR)
DIMENSION HU (HXHUPR),HKU (HXHUPR),INUTYP (HAXEL)
DIMENSION THLTAB (HXHLPR,HXHLDP),HLTAB (HXHLPR,HXHLDP)
DIMENSION HL (HXHLPR),BKL (HXHLPR),IBLTYP (HAXEL)
C
DIMENSION PROP (HAXHAT,7)
C
DIMENSION KPR (HAXHTI),KDSK (HAXHTI),TDTCH (HXDTCH)
C
DIMENSION PRATE (10),FLOW (10),TFLOW (10)
DIMENSION SUBHD (8,3)
C
DATA SUBHD/4HNP,4HT IN,4HTIA,4EL CO,4HNDIT,4HIONS,2*4H
> 4HSTEA,4HDI-S,4HTATE,4H INI,4HTIAL,4H CON,4HNDITI,4HONS, 8*
> 4H /
C

```

```

GW2D 005
GW2D 010
GW2D 015
GW2D 020
GW2D 025
GW2D 030
GW2D 035
GW2D 040
GW2D 045
GW2D 050
GW2D 055
GW2D 060
GW2D 065
GW2D 070
GW2D 075
GW2D 080
GW2D 085
GW2D 090
GW2D 095
GW2D 100
GW2D 105
GW2D 110
GW2D 115
GW2D 120
GW2D 125
GW2D 130
GW2D 135
GW2D 140
GW2D 145
GW2D 150
GW2D 155
GW2D 160
GW2D 165
GW2D 170
GW2D 175
GW2D 180
GW2D 185
GW2D 190
GW2D 195
GW2D 200
GW2D 205
GW2D 210
GW2D 215
GW2D 220
GW2D 225
GW2D 230
GW2D 235
GW2D 240
GW2D 245
GW2D 250
GW2D 255
GW2D 260
GW2D 265
GW2D 270
GW2D 275
GW2D 280
GW2D 285
GW2D 290
GW2D 295
GW2D 300

```

APPENDIX C. (continued)

C		GW2D 305
C	***** DATA SET 1: PROBLEM IDENTIFICATION AND DESCRIPTION	GW2D 310
C		GW2D 315
C	100 READ 10, NPROB, (TITLE(1),I=1,9), IDBC	GW2D 320
C		GW2D 325
C	IF (NPROB-LE.0) GO TO 990	GW2D 330
C		GW2D 335
C	PRINT 1100, NPL-3, (TITLE(1),I=1,9)	GW2D 340
C		GW2D 345
C	----- READ AND PRINT INPUT DATA BY CALLING DATAX	GW2D 350
C		GW2D 355
C	KOUT=0	GW2D 360
C		GW2D 365
C	CALL DATAX(X,Z,B,IE,H, V1,V2, DLB,DCOSXB,DCOSZB,NBZ,ISB,NPB,	GW2D 370
C	1 NPN,INTYP,NRP,TRNF,IS, NPD,INTYP,NED,THED, ISTYP,SOS,TSOS,	GW2D 375
C	2 PROF,KPR,KDSE, ISTOP,MAXDIF, V,ONE, TIME, NPN,INTYP,NSSP,TWSSP,	GW2D 380
C	3 THUTAB,UTAB,NB,RKO,INTYP, TULTAB,ELTAB,HL,NKL,INTYP,TDICH)	GW2D 385
C		GW2D 390
C	CALL LUDGEN(LLNODE,IE,MAXBP,MAXEL,NBP,NEL)	GW2D 395
C		GW2D 400
C	----- COMPUTE TRANSMISSIBILITY TENSOR COMPONENTS BASED ON CONDUCTIVITY	GW2D 405
C	----- AND THICKNESS AND STORE THEM IN AKX,AKZ,AKYZ IF IT IS CONFIGURED	GW2D 410
C	----- ACQUIFER	GW2D 415
C		GW2D 420
C	DO 120 N=1,NEL	GW2D 425
C	NTYP=IE(N,5)	GW2D 430
C	IF (NTYP-LE.0) GO TO 120	GW2D 435
C	NQ=4	GW2D 440
C	IEHQ=IE(N,4)	GW2D 445
C	IF (IEHQ-EQ.0) NQ=3	GW2D 450
C	DO 110 IQ=1,NQ	GW2D 455
C	NP=IE(N,IQ)	GW2D 460
C	AKX(N,IQ)=PROP(NTYP,4)*B(NP)	GW2D 465
C	AKZ(N,IQ)=PROP(NTYP,5)*B(NP)	GW2D 470
C	AKYZ(N,IQ)=PROP(NTYP,6)*B(NP)	GW2D 475
C	110 CONTINUE	GW2D 480
C	120 CONTINUE	GW2D 485
C		GW2D 490
C	KDIG=NSTRT	GW2D 495
C	IF (ISTOP-GT.0) GO TO 990	GW2D 500
C		GW2D 505
C	----- COMPUTE BAND-WIDTH VARIABLES	GW2D 510
C		GW2D 515
C	INHALPB=MAXDIF	GW2D 520
C	ISANC=2*INHALPB+1	GW2D 525
C	IRBP=INHALPB+1	GW2D 530
C		GW2D 535
C	IF (IRBP-GT.MAXIRBP .AND. IPNTS-EQ.0) GO TO 910	GW2D 540
C		GW2D 545
C	----- PREPARE INITIAL OR PRE-INITIAL VARIABLES	GW2D 550
C		GW2D 555
C	IF (NPNP-NE.0) CALL INTERP(RPLX,TRNF,NRP,TIME,	GW2D 560
C	1 NPNR,NINDP, NPNR,NNDP)	GW2D 565
C	IF (NNDP-NE.0) CALL INTERP(GHED,THED,NED,TIME,	GW2D 570
C	1 NIDPR,NIDDP, NIDPR,NDDP)	GW2D 575
C	IF (NPNR-NE.0) CALL INTERP(SOUS,TSOS,SOS,TIME,	GW2D 580
C	1 NISPR,NISDP, NISPR,NSDP)	GW2D 585
C	IF (NPNP-NE.0) CALL INTERP(NSSP,TWSSP,NSSP,TIME,NPNR,NINDP,	GW2D 590
C	1 NPNR,NNDP)	GW2D 595
C		GW2D 600

APPENDIX C. (continued)

```

C ----- INTERPOLATE FOR THE HU AND HL VALUE FOR THE PRESENT TIME      GV2D 605
C                                                                           GV2D 610
C   CALL INTERP(HU,THTAB,HUTAB,TIME,HXHUFR,HXHUFP,HUFR,HUFP)             GV2D 615
C   CALL INTERP(HL,THLTAB,HLTAB,TIME,HXLPR,HXLDP,HLPF,HLPD)             GV2D 620
C                                                                           GV2D 625
C ----- APPLY DIRICHLET BOUNDARY CONDITIONS TO INITIAL CONDITIONS      GV2D 630
C                                                                           GV2D 635
C   DO 150 I=1,NBND                                                       GV2D 640
C     NF=NEB(I)                                                            GV2D 645
C     ITYP=IDTYP(I)                                                        GV2D 650
C     H(NF)=CHED(ITYP)                                                    GV2D 655
150 CONTINUE                                                                GV2D 660
C                                                                           GV2D 665
C   IF(ICONFI.EQ.0) CALL SPBOP(IE,H,B,AKX,AKZ,AKYZ,PROP,HAXEL,HAXFP,    GV2D 670
C   > HAXHAT, HXI)                                                         GV2D 675
C                                                                           GV2D 680
C   CALL VELT(X,Z,IE,C,H,   VX,VZ,AKX,AKZ,AKYZ, HAXEL,HAXFP,HAXHAT)    GV2D 685
C   KFLOW=-1                                                                GV2D 690
C                                                                           GV2D 695
C   CALL SFLOW(X,Z,IE,VX,VZ,DLB,DCOSXB,DCOSZB,BFLX,BFLFP,ISB,NFB,NPB,    GV2D 700
C   1 NPH,NPD, PRATE,FLOW,TFLOW, HAXFP,HXKEL,HXBEL,HXBFP,              GV2D 705
C   2 KFLOW, DELT,H,NP, PRCP,HAXHAT, B, SOVS,ISTYP, NPV,WSS,INTYP,        GV2D 710
C   3 HU,HXU,INUTYP,HL,HXL,INLTP,HXHUFR,HXHUFP)                          GV2D 715
C                                                                           GV2D 720
C   DO 160 I=1,8                                                           GV2D 725
C     IF(I.EQ.7) GO TO 160                                                 GV2D 730
C     FLOW(I)=0.0                                                          GV2D 735
C     TFLOW(I)=0.0                                                        GV2D 740
160 CONTINUE                                                                GV2D 745
C     FLOW(7)=0.0                                                         GV2D 750
C                                                                           GV2D 755
C                                                                           GV2D 760
C ----- PRINT INITIAL OR PRE-INITIAL VARIABLES                        GV2D 765
C                                                                           GV2D 770
C   KDIAG=0                                                                GV2D 775
C                                                                           GV2D 780
C   CALL PRINTT(VX,VZ,H,NFB,BFLX, PRATE,FLOW,TFLOW, HAXFP,HAXEL,        GV2D 785
C   > HXBFP, HNF,HXL,   TIME,DELT,SUBHD(1,1),ISAND,KPRO,KOUT,          GV2D 790
C   > KDIAG,-1)                                                            GV2D 795
C                                                                           GV2D 800
C   IF(KSTR.EQ.1 .AND. KSS.EQ.1 .AND. NSTRT.EQ.0 .AND. KDSKO.EQ.1)      GV2D 805
C   > CALL STORE(X,Z,IE, H,B,   VX,VZ,DLB,DCOSXB,DCOSZB,NFB,ISB,NPB,    GV2D 810
C   1 TITLE,TIME,HAXFP,HAXEL,HAXBFP,HAXEL,NPROB,NPF,HXL,NB,NDEL,NTI,    GV2D 815
C   2 NSTRT, HU,HXU,INUTYP,HL,HXL,INLTP,HXHUFR,HXHUFP,HUFR,HUFP)        GV2D 820
C                                                                           GV2D 825
C   IF (KSS.NE.0) GO TO 500                                               GV2D 830
C                                                                           GV2D 835
C ----- PERFORM STEADY-STATE CALCULATION                             GV2D 840
C                                                                           GV2D 845
C   170 DO 180 NP=1,NBP                                                    GV2D 850
C   180 H(NF)=H(NP)                                                         GV2D 855
C                                                                           GV2D 860
C   KDIG=KDIG+1                                                            GV2D 865
C   IF(ISUG.NE.0) PRINT 1110, KDIG,TIME,DELT                            GV2D 870
C                                                                           GV2D 875
C                                                                           GV2D 880
C   DO 220 NP=1,NBP                                                       GV2D 885
C   220 H(NF)=ONE*H(NP)+(1.0D0-ONE)*EP(NP)                                GV2D 890
C                                                                           GV2D 895
C ----- ITERATION LOOP ON THE NON-LINEAR EQUATION BEGINS             GV2D 900

```


APPENDIX C. (continued)

```

C      IF (IBUG.NE.0) PRINT 1120
C      DO 350 IT=1,NIITER
C      ----- EVALUATE SOIL PROPERTIES FOR PREVIOUS ITERATE
C      IF (ICONFI.EQ.0) CALL SPROP (IP,HW,B,AKX,AKZ,AKYZ,PROP,
C      >      MAXEL,MAXNP, MAXMAT, MEL)
C      ----- ASSEMBLE STEAD-STATE ELEMENT MATRICES QA AND QB INTO THE
C      ----- GLOBAL MATRIX C AND CONSTRUCT GLOBAL LOAD VECTOR R FROM
C      ----- ELEMENT LOAD VECTOR EQ.
C      CALL ASSEMBL (X,Z,IP, C,R, HP,HW,B, AKX,AKZ,AKYZ, PROP,
C      1      ISTYP,SOUS, MAXNP,MAXEL,MAXNDP,MAXMAT,      KSS,V,DELT,
C      2      HW,HW,INTYP,HL,RKL,INTYP,MAXNPA,MAXLPA,
C      3      WSS,NPA,INTYP,LLMODE)
C      ----- APPLY STEADY-STATE BOUNDARY CONDITIONS
C      CALL BC (C,R,X,Z,RPLX,INTYP,NPA,IS,GNED,INTYP,NPD,MAXNP,MAXNDP)
C      ***** DETERMINE IF MATRIX SOLUTION IS TO BE USED?
C      IF (IFMTS.EQ.1) GO TO 250
C      ***** USE MATRIX SOLUTION
C      CALL BANSOL (1,C,R,NPA,INBP,MAXNP,MAXNDP)
C      CALL BANSOL (2,C,R,NPA,INBP,MAXNP,MAXNDP)
C      GO TO 300
C      ***** USE POINTWISE ITERATION
C      250 DO 265 NP=1,NNP
C      DO 260 I=1,4
C      LMODE=LLMODE (I,NP)
C      IF (LMODE.EQ.0) GO TO 260
C      R (NP)=R (NP)-C (NP,I)*R (LMODE)
C      260 CONTINUE
C      R (NP)=ONE*(R (NP)/C (NP,5)) + (1.0D0-ONE)*HW (NP)
C      265 CONTINUE
C      ----- OBTAIN MAXIMUM RELATIVE DEVIATION FROM PREVIOUS ITERATE
C      300 CONTINUE
C      NPP=0
C      RD=-1.0D0
C      RES=-1.0D0
C      DO 320 NP=1,NNP
C      RESNP=DABS (R (NP)-R (NP))
C      RES=CHAI (RES,RESNP)
C      IF (R (NP).NE.0.0D0) RD=DMAX1 (RD,DABS (RESNP/R (NP)))
C      IF (RESNP.LE.TOLA) GO TO 320
C      NPP=NPP+1
C      NPCNV (NPP)=NP
C      320 CONTINUE
C      NUCNV=NPP
C

```

GN2D 905
 GN2D 910
 GN2D 915
 GN2D 920
 GN2D 925
 GN2D 930
 GN2D 935
 GN2D 940
 GN2D 945
 GN2D 950
 GN2D 955
 GN2D 960
 GN2D 965
 GN2D 970
 GN2D 975
 GN2D 980
 GN2D 985
 GN2D 990
 GN2D 995
 GN2D 1000
 GN2D 1005
 GN2D 1010
 GN2D 1015
 GN2D 1020
 GN2D 1025
 GN2D 1030
 GN2D 1035
 GN2D 1040
 GN2D 1045
 GN2D 1050
 GN2D 1055
 GN2D 1060
 GN2D 1065
 GN2D 1070
 GN2D 1075
 GN2D 1080
 GN2D 1085
 GN2D 1090
 GN2D 1095
 GN2D 1100
 GN2D 1105
 GN2D 1110
 GN2D 1115
 GN2D 1120
 GN2D 1125
 GN2D 1130
 GN2D 1135
 GN2D 1140
 GN2D 1145
 GN2D 1150
 GN2D 1155
 GN2D 1160
 GN2D 1165
 GN2D 1170
 GN2D 1175
 GN2D 1180
 GN2D 1185
 GN2D 1190
 GN2D 1195
 GN2D 1200

APPENDIX C. (continued)

```

C ----- UPDATE PRESSURE WITH CURRENT ITERATE
C
  IF(IPWIS.EQ.1) GO TO 335
  DO 330 NF=1,NMF
    HW(NF)=ONE*R(HF)+(1.000-ONE)*HW(NF)
    HW(NF)=ONE*R(HF)+(1.000-ONE)*HW(NF)
    H(NF)=R(HF)
  330 CONTINUE
  GO TO 345
  335 DO 340 NF=1,NMF
    HW(NF)=R(HF)
    H(NF)=R(HF)
  340 CONTINUE
C ----- ESCAPE FROM ITERATION LOOP IF THE MAXIMUM RESIDUAL IS
C ----- SUFFICIENTLY SMALL
C
  345 CONTINUE
  IF(1806.NE.0) PRINT 1200, IT,RES,HD,HUCVH
  IF(IT.EQ.1) GO TO 350
  IF(RES.IT.TOLA) GO TO 360
  IF(WITER.EQ.1) GO TO 360
  350 CONTINUE
C ----- END OF ITERATION LOOP ON THE NON-LINEAR EQUATIONS
C
  PRINT 1210, IT,WITER
C ----- PRINT NONCONVERGING MODES
C
  IF(1806.EQ.0) GO TO 360
  PRINT 1220
  PRINT 1230, (HPCNV(WFF),WFF=1,HUCVH)
C ----- CALCULATE DANCY'S VELOCITY
C
  360 IF(1807.EQ.0) CALL SPROP(IE,H,B,AK1,AK2,AK3,PROP,
    > HAXEL,HAYDE,HAYHAT,SEL)
C
  CALL VZLT(X,2,IE,C,H, VX,VZ,AK1,AK2,AK3, HAXEL,HAYDP,HAYHDP)
C
  440 KFLOW=0
  CALL SFLOW(X,2,IE,VX,VZ,DLS,DCOSIS,DCOSIS,BFLX,BFLIP,ISB,VBE,NPB,
    1 NFB,NFD, FRATE,FLOW,TFLOW, HAYDP,HAXEL,HAYDEL,HAYHDP,
    2 KFLOW, DELT,H,NP, PROP,HAYHAT, B, SOUS,ISTYP, NPW,SS,INTYP,
    3 NU,RKU,INUTYP,BL,BLL,INLTYP,NXHUPR,NXHLP)
C
  DO 450 I=1,8
    IF(I.EQ.7) GO TO 450
    FLOW(I)=0.0
    TFLC7(I)=0.0
  450 CONTINUE
  FLOW(7)=0.
C ----- PRINT STEADY-STATE VARIABLES
C
  CALL PRINTT(VX,VZ,H,NFB,BFLX, FRATE,FLOW,TFLOW, HAYDP,HAXEL,
    1 HAYHDP, NFB,NEL, TIME,DELT,SUBED(1,2),IBAND,KPRO,KOUT,
    2 KDIAG,0)

```

GU2D1205
 GU2D1210
 GU2D1215
 GU2D1220
 GU2D1225
 GU2D1230
 GU2D1235
 GU2D1240
 GU2D1245
 GU2D1250
 GU2D1255
 GU2D1260
 GU2D1265
 GU2D1270
 GU2D1275
 GU2D1280
 GU2D1285
 GU2D1290
 GU2D1295
 GU2D1300
 GU2D1305
 GU2D1310
 GU2D1315
 GU2D1320
 GU2D1325
 GU2D1330
 GU2D1335
 GU2D1340
 GU2D1345
 GU2D1350
 GU2D1355
 GU2D1360
 GU2D1365
 GU2D1370
 GU2D1375
 GU2D1380
 GU2D1385
 GU2D1390
 GU2D1395
 GU2D1400
 GU2D1405
 GU2D1410
 GU2D1415
 GU2D1420
 GU2D1425
 GU2D1430
 GU2D1435
 GU2D1440
 GU2D1445
 GU2D1450
 GU2D1455
 GU2D1460
 GU2D1465
 GU2D1470
 GU2D1475
 GU2D1480
 GU2D1485
 GU2D1490
 GU2D1495
 GU2D1500

APPENDIX C. (continued)

```

C      IF (KSTR.EQ.1 .AND. KDSKO.EQ.1) CALL STORE(X,Z,IE,
1      H,B,V1,V2, DLD,DCOSIB,DCOS2B,HBZ,ISB,NPB,TITLE,TIME,HAKNP,
2      HAKEL,HAKBHP,HAKBEL,BPBON,BBP,BEL,BBN,BDEL,BTI, BSEXT,
3      BU,BKU,INUTYP,BL,BKL,INLTYP,BKHUPB,BKHLPB,BHUPB,BHLPB)
C
C      IF (BTI.EQ.0) GO TO 100
C      KSS=1
C
C      ----- PERFORM TRANSIENT-STATE CALCULATION
C
500 CONTINUE
    TIME=TIME+DELT
    W1=0
    W2=1.-W
    KFLON=1
    TFLON(7)=0.0
C
C
C      ----- BEGIN THE TIME-MARCHING LOOP
C
    IDELT=0
    DO 890 ITH=1,BTI
C      ----- INTERPOLATE FOR SOURCE AND BOUNDARY CONDITION AT THE PRESENT
C
    IF (BUNP.NE.0) CALL INTERP(RFLY,TRNF,BNP,TIME,
1    HIBER,HXNDP, HUPR,HNDP)
    IF (BNDP.NE.0) CALL INTERP(GRED,THED,BED,TIME,
1    HXDP,HXDDP, NDP,HDDP)
    IF (BSPR.NE.0) CALL INTERP(SOUS,TSOS,SOS,TIME,
1    HXSPB,HXSDP, BSPR,BSDP)
    IF (BUNP.NE.0) CALL INTERP(WSS,TWSSP,WSSP,TIME,HXWPR,HXWDP,
1    HUPB,HNDP)
C
C      ----- INTERPOLATE FOR BU AND BL AT THE PRESENT TIME
C
    CALL INTERP(BU,THUTAB,BUTAB,TIME,HXHUFB,HXHUDP,BHUPB,BHUDP)
    CALL INTERP(BL,THLTAB,BLTAB,TIME,HXHLFB,HXHLDP,BHLPB,BHLPD)
C
    DO 510 NP=1,NBP
510 NP(BE)=B(NP)
C
    KDIG=KDIG+1
    IF (IBUG.NE.0) PRINT 1110, KDIG,TIME,DELT
C
    IF (IBUG.NE.0) PRINT 1120
C
    GO 520 NP=1,NBF
520 BW(BE)=B1*(ONE*B(NP)+(1.000-ONE)*BP(NP))+W2*B(NP)
C
C      ----- BEGIN ITERATION LOOP ON THE NON-LINEAR EQUATION
C
    DO 650 IT=1,NITER
C
C      ----- EVALUATE SOIL PROPERTIES FOR PREVIOUS ITERATE
C
    IF (ICONFL.EQ.0) CALL SPROP(IE,BU,B,AKX,AKZ,AKXZ,PROP,
>    HAKEL,HAKBHP,HAKBET, BEL)
C

```

GW2D1505
 GW2D1510
 GW2D1515
 GW2D1520
 GW2D1525
 GW2D1530
 GW2D1535
 GW2D1540
 GW2D1545
 GW2D1550
 GW2D1555
 GW2D1560
 GW2D1565
 GW2D1570
 GW2D1575
 GW2D1580
 GW2D1585
 GW2D1590
 GW2D1595
 GW2D1600
 GW2D1605
 GW2D1610
 GW2D1615
 GW2D1620
 GW2D1625
 GW2D1630
 GW2D1635
 GW2D1640
 GW2D1645
 GW2D1650
 GW2D1655
 GW2D1660
 GW2D1665
 GW2D1670
 GW2D1675
 GW2D1680
 GW2D1685
 GW2D1690
 GW2D1695
 GW2D1700
 GW2D1705
 GW2D1710
 GW2D1715
 GW2D1720
 GW2D1725
 GW2D1730
 GW2D1735
 GW2D1740
 GW2D1745
 GW2D1750
 GW2D1755
 GW2D1760
 GW2D1765
 GW2D1770
 GW2D1775
 GW2D1780
 GW2D1785
 GW2D1790
 GW2D1795
 GW2D1800

APPENDIX C. (continued)

```

C ----- ASSEMBLE ELEMENT MATRICES Q1 AND Q3 INTO THE GLOBAL MATRIX C
C ----- AND CONSTRUCT THE GLOBAL LOAD VECTOR R FROM ELEMENT LOAD
C ----- VECTOR RQ.
C
      CALL ASSEMBL(X,Z,IE,C,R,HP,HWP,B,AK1,AK2,AK3,PROP,
1      ISTD,SOUS,HAINP,HAIEL,HAHBP,HAHBT,KSS,W,DELT,
2      HU,HU,INTYP,HL,REL,INTYP,HXNUP,HXNLP,
3      USS,HWP,INTYP,LLMODE)
C
C ----- APPLY BOUNDARY CONDITIONS TO MODIFY THE GLOBAL MATRIX C AND
C ----- THE LOAD VECTOR R.
C
      CALL BC(C,R,X,Z,RPLX,INTYP,HWP,IS,CHED,INTYP,WPD,HAINP,HAHBP)
C
C ***** DETERMINE IF MATRIX SOLUTION IS TO BE USED?
C
      IF(IEHIS.EQ.1) GO TO 550
C
C ***** USE MATRIX SOLUTION
C
      CALL BANSOL(1,C,R,HWP,INBP,HAINP,HAHBP)
      CALL BANSOL(2,C,R,HWP,INBP,HAINP,HAHBP)
      GO TO 600
C
C ***** USE POINTWISE ITERATION
C
550 DO 565 NP=1,HWP
      DO 560 I=1,4
        LMODE=ILMODE(X,NP)
        IF(LMODE.EQ.0) GO TO 560
        R(NP)=R(NP)-C(NP,I)*R(LMODE)
560 CONTINUE
        R(NP)=ONE*(R(NP)/C(NP,5))+(1.00-ONE)*HW(NP)
565 CONTINUE
C
C ----- OBTAIN MAXIMUM RELATIVE DEVIATION FROM PREVIOUS ITERATE
C
600 CONTINUE
      WPP=0
      RD=-1.000
      RES=-1.000
      DO 620 NP=1,HWP
        RESN=DABS(R(NP)-H(NP))
        RES=MAX1(RES,RESN)
        IF(H(NP).NE.0.000) RD=MAX1(RD,DABS(RESN/H(NP)))
        IF(RESN.LE.TOLB) GO TO 620
        WPP=WPP+1
        WPCNV(WPP)=NP
620 CONTINUE
C
      WPCNV=WPP
C
C ----- UPDATE PRESSURE WITH CURRENT ITERATE
C
      IF(IEHIS.EQ.1) GO TO 635
      DO 630 NP=1,HWP
        HN(NP)=W1*(ONE*R(NP)+(1.00-ONE)*HW(NP))+W2*HP(NP)
        HW(NP)=W1*(ONE*R(NP)+(1.00-ONE)*H(NP))+W2*HP(NP)
        H(NP)=R(NP)
630 CONTINUE

```

GW2D1805
 GW2D1810
 GW2D1815
 GW2D1820
 GW2D1825
 GW2D1830
 GW2D1835
 GW2D1840
 GW2D1845
 GW2D1850
 GW2D1855
 GW2D1860
 GW2D1865
 GW2D1870
 GW2D1875
 GW2D1880
 GW2D1885
 GW2D1890
 GW2D1895
 GW2D1900
 GW2D1905
 GW2D1910
 GW2D1915
 GW2D1920
 GW2D1925
 GW2D1930
 GW2D1935
 GW2D1940
 GW2D1945
 GW2D1950
 GW2D1955
 GW2D1960
 GW2D1965
 GW2D1970
 GW2D1975
 GW2D1980
 GW2D1985
 GW2D1990
 GW2D1995
 GW2D2000
 GW2D2005
 GW2D2010
 GW2D2015
 GW2D2020
 GW2D2025
 GW2D2030
 GW2D2035
 GW2D2040
 GW2D2045
 GW2D2050
 GW2D2055
 GW2D2060
 GW2D2065
 GW2D2070
 GW2D2075
 GW2D2080
 GW2D2085
 GW2D2090
 GW2D2095
 GW2D2100

APPENDIX C. (continued)

```

      GO TO 645
635 DO 640 NP=1,NMP
      HW(NP)=R(NP)
      H(NP)=R(NP)
640 CONTINUE
C
C ----- ESCAPE FROM ITERATION LOOP IF THE MAXIMUM RESIDUAL IS
C ----- SUFFICIENTLY SMALL.
C
645 CONTINUE
      IF(IBUG.NE.0) PRINT 1200, IT,RES,RD,WNCVN
      IF(IT.EQ.1 .AND. ITN.EQ.1) GO TO 650
      IF(RES.IT.TOLB) GO TO 660
      IF(WITER.EQ.1) GO TO 660
650 CONTINUE
C
C ----- END THE ITERATION LOOP ON THE NON-LINEAR EQUATION
C
      PRINT 1400, ITN, IT,WITER
C
      IF(IEUG.EQ.0) GO TO 660
C
C ----- PRINT NONCONVERGING NODES
C
      PRINT 1220
      PRINT 1230, (WPCNV(NPP),NPP=1,WNCVN)
C
C ----- CALCULATE DARCY'S VELOCITY
C
660 IF(ICONFL.EQ.0) CALL SPROP(IE,H,B,AKX,AKZ,AKYZ,PROP,
>      HAKEL,HAXNP,HAXHAT,NEL)
C
      CALL VELT(X,Z,IE,C,H, VX,VZ,AKX,AKZ,AKYZ, HAXEL,HAXNP,HAXHBP)
C
C
710 IF(INID.EQ.0) GO TO 740
C
      DO 720 I=1,NMP
720 H(I)=2.0D0*H(I)-HP(I)
C
      DO 730 I=1,NMNE
      NP=NED(I)
      ITYP=IDTYP(I)
      H(NP)=GBED(ITYP)
730 CONTINUE
C
C ----- COMPUTE FLUXES THROUGH ALL TYPES OF BOUNDARIES
C
740 CALL SFLOW(X,Z,IE,VX,VZ,DLB,DCOSXB,DCOSZB,BFLX,BFLYP,ISB,WBE,
1      WPB,WEN,WPB, PRATE,FLOW,TFLOW, HAXNP,HAKEL,HAXBEL,HABNP,
2      KFLOW, DELT,H,HP, PROP,HAXHAT, B, SQUS,ISTYP, WPM,WSS,IWTYP,
3      HU,RKU,INUTYP,HL,BKL,IHLTYP,HXNUPR,HXHLPR)
C
C ----- PRINT VARIABLES AT EACH TIME STEP
C
      CALL PRINTT(VX,VZ,H,WPB,BFLX, PRATE,FLOW,TFLOW, HAXNP,HAKEL,
1      HAXBEL, WPM,WEL, TIME,DELT,SUBND(1,3),IBAND,KPR(ITN),ROUT,
2      KDIAG,ITN)
C

```

```

GW2D2105
GW2D2110
GW2D2115
GW2D2120
GW2D2125
GW2D2130
GW2D2135
GW2D2140
GW2D2145
GW2D2150
GW2D2155
GW2D2160
GW2D2165
GW2D2170
GW2D2175
GW2D2180
GW2D2185
GW2D2190
GW2D2195
GW2D2200
GW2D2205
GW2D2210
GW2D2215
GW2D2220
GW2D2225
GW2D2230
GW2D2235
GW2D2240
GW2D2245
GW2D2250
GW2D2255
GW2D2260
GW2D2265
GW2D2270
GW2D2275
GW2D2280
GW2D2285
GW2D2290
GW2D2295
GW2D2300
GW2D2305
GW2D2310
GW2D2315
GW2D2320
GW2D2325
GW2D2330
GW2D2335
GW2D2340
GW2D2345
GW2D2350
GW2D2355
GW2D2360
GW2D2365
GW2D2370
GW2D2375
GW2D2380
GW2D2385
GW2D2390
GW2D2395
GW2D2400

```

APPENDIX C. (continued)

```

      IF (KSTR.EQ.1 .AND. KDSK(ITH).EQ.1) CALL STORE(X,Z,IE,H,B,VX,VZ,
>      DLB,DCOSXB,DCOSZB,WBE,ISB,WPB,TITLE,TIME,MAXWP,MAXEL,MAXBWP,
>      MAXDEL,NEFROB,WNP,NEL,NBN,NBEL,NTI,NSTRT,
>      BU,RKU,INUTYP,HL,NKL,IHLTYP,NKHUPR,NKHLPR,NHUPR,NHLPR)
C
C ----- PREPARE FOR NEXT TIME STEP
C
      IF (TIME.GT.TMAX) GO TO 100
      DELT=DELT*(1.0D0+CHNG)
      DELT=DMIN1(DELT,DELMAX)
      IF (TIME.EQ.TDTCN(IDELT+1)) DELT=DELTO
      TIME=TIME+DELT
      IF (TIME.LE.TDTCN(IDELT+1)) GO TO 890
      IDELT=IDELT+1
      TIME=TIME-DELT
      DELT=TCTCN(IDELT)-TIME
      IF (DELT.LE.0.0) DELT=DELTO
      TIME=TIME+DELT
890 CONTINUE
C
C ----- END OF TIME-MARCHING LOOP
C
      GO TO 100
C
910 PRINT 2000, INBP,MAXBWP
C
990 RETURN
C
10  FORMAT(15,9A8,I3)
1100 FORMAT(/8H1PROBLEM,15,3H..,9A8/)
1110 FORMAT(1H1,52H*****
> 62H*****
> 5H****//17H DIAGNOSTIC TABLE,14,12H.. AT TIME =,1PD12.4,
> 9H ,(DELT =,1PD12.4,1H))
1120 FORMAT(///30H TABLE OF ITERATIVE PARAMETERS// 6X,
> 9HITERATION,7X,8HRESIDUAL,6X,9HDEVIATION,6X,
> 19HNO. NON-CONV. NODES)
1200 FORMAT(5X,110,3X,E12.4,3X,E12.4,15X,110)
1210 FORMAT(1H1/5X,68H WARNING: NON-CONVERGENCE OCCUR DURING STEADY STA
>TE SOLUTION AT IT =,13,13H .GT. MAXIT=,13)
1220 FORMAT(//30H TABLE OF NON-CONVERGING NODES)
1230 FORMAT(/(5X,2015))
1400 FORMAT(1H1/5X,35H WARNING: NON-CONVERGENCE OCCUR AT ,15, 8H-TH STE
>P//1H ,5X,5H IT =,13,15H .GT. MAXIT =,13)
2010 FORMAT(///26H HALF-BANDWIDTH-PLUS-ONE =,14,
> 25H EXCEEDS MAX. ALLOWABLE =,14)
      END

```

GW2D2405
 GW2D2410
 GW2D2415
 GW2D2420
 GW2D2425
 GW2D2430
 GW2D2435
 GW2D2440
 GW2D2445
 GW2D2450
 GW2D2455
 GW2D2460
 GW2D2465
 GW2D2470
 GW2D2475
 GW2D2480
 GW2D2485
 GW2D2490
 GW2D2495
 GW2D2500
 GW2D2505
 GW2D2510
 GW2D2515
 GW2D2520
 GW2D2525
 GW2D2530
 GW2D2535
 GW2D2540
 GW2D2545
 GW2D2550
 GW2D2555
 GW2D2560
 GW2D2565
 GW2D2570
 GW2D2575
 GW2D2580
 GW2D2585
 GW2D2590
 GW2D2595
 GW2D2600
 GW2D2605
 GW2D2610
 GW2D2615
 GW2D2620
 GW2D2625
 GW2D2630
 GW2D2635

APPENDIX C. (continued)

```

SUBROUTINE DATANH(I,Z,B,IE,H,VI,VZ,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB, DATA 005
1 NPN,INTYP,RNF,TRNF,IS,NPD,IDTYP,HED,THED,ISTYP,SOS,TSOS, DATA 010
2 PROP,KPR,KDSK,ISTOP,HAXDIP,W,ONE,TIME,NPH,INTYP,WSSP,TWSSP, DATA 015
3 THUTAB,HUTAB,HU,RKU,INHUTYP,THLTAB,HLTAB,HL,RKL,IHLTYP,TDTC) DATA 020
C DATA 025
C DATA 030
C FUNCTION OF SUBROUTINE--TO READ, PRINT, AND CHECK VARIABLES DATA 035
C PERTAINING TO SIMULATION TIME, GEOMETRY OF THE SYSTEM, ITS SOIL DATA 040
C PROPERTIES, BOUNDARY-INITIAL CONDITIONS FOR BOTH STEADY-STATE AND DATA 045
C TRANSIENT CASES, AND NUMERICAL CONVERGENCE CRITERIA. DATA 050
C DATA 055
C DATA 060
C DATA 065
C IMPLICIT REAL*8(A-H,O-Z) DATA 070
REAL*4 PHAT,DATNAM DATA 075
C DATA 080
COMMON /SIZE/ MAXNP,MAXEL,MAXHEP,MAXBEL,MAXBNP,MAXNAT,MAXNTI,MAXDTC DATA 085
COMMON /GEOM/ NNP,NEL,IBAND,IBBP,IBALPB,NBEL,NBN DATA 090
COMMON /CTRL/ NTI,NHAT,NITER,NSTRT,KSTR,KPRO,KDSKO,KSS,WDTCNG DATA 095
COMMON /CN/ NXNPP,NXNEL,NXNPR,NXNDP,NWNP,NWEL,NWPR,NWDP DATA 100
COMMON /CD/ NXDNP,NXDPR,NXDDP,NNDP,NDPR,NDDP DATA 105
COMMON /CS/ NISPR,NISDP,NSPR,NSDP,NXWNP,NXWPR,NXWDF,NWNP,NWPR,NWDP DATA 110
COMMON /OPT/ ILUMP,INID,ICCNFI,IPNTS DATA 115
COMMON /KURU/ NXHUPR,NXHUPD,NHUPR,NHUPD DATA 120
COMMON /KLHL/ NXHLPR,NXHLDF,NHLPR,NHLDP DATA 125
COMMON /TOTLNS/ TOLA,TOLB DATA 130
COMMON /PARAM/ DELT,CHNG,DELMAX,TMAX,DELTO DATA 135
C DATA 140
C DIMENSION TITLE(9) DATA 145
C DIMENSION I(MAXNP),Z(MAXNP),B(MAXNP),IE(MAXEL,5) DATA 150
C DATA 155
C DIMENSION H(MAXNP),VI(MAXNP),VZ(MAXNP) DATA 160
C DATA 165
C DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),NBE(MAXBEL), DATA 170
> ISB(MAXBEL,4),NPB(MAXBNP) DATA 175
C DATA 180
C DIMENSION NPN(MXNPP),INTYP(MXNPP),IS(MXNEL,2) DATA 185
C DIMENSION RNF(MXNPR,MXNDP),TRNF(MXNPR,MXNDP) DATA 190
C DIMENSION NPD(MXDNP),IDTYP(MXDNP) DATA 195
C DIMENSION HED(MXDPR,NXDDP),THED(MXDPR,MXDDP) DATA 200
C DIMENSION ISTYP(MAXEL) DATA 205
C DIMENSION SOS(NISPR,NISDP),TSOS(NISPR,NISDP) DATA 210
C DIMENSION NPH(MXNPP),INTYP(MXNPP) DATA 215
C DIMENSION WSSP(MXWPR,MXWDP),TWSSP(MXWPR,MXWDP) DATA 220
C DATA 225
C DIMENSION THUTAB(MXHUPR,MXHUPD),HUTAB(MXHUPR,MXHUPD) DATA 230
C DIMENSION HU(MXHUPR),RKU(MXHUPR),INHUTYP(MAXEL) DATA 235
C DIMENSION THLTAB(MXHLPR,MXHLDP),HLTAB(MXHLPR,MXHLDP) DATA 240
C DIMENSION HL(MXHLPR),RKL(MXHLPR),IHLTYP(MAXEL) DATA 245
C DATA 250
C DIMENSION PROP(MAXNAT,7) DATA 255
C DIMENSION KPR(MAXNTI),KDSK(MAXNTI),TDTC(MXDTC) DATA 260
C DIMENSION PHAT(3,7),DATNAM(20) DATA 265
C DATA 270
C DATA PHAT/4H ,4H ALP,4H ,4H B,4H ETAP,4H ,4H , DATA 275
> 4H FOR,4H ,4H K,4H SX/P,4H SX ,4H K,4H SZ/P,4H SZ ,4H KS,4H DATA 280
> IZ/P,4H SXZ ,4H ,4H SY ,4H / DATA 285
C DATA 290
C DATA 295
C ISTOP=0 DATA 300

```

APPENDIX C. (continued)

C		DATA 305
C	***** DATA SET 2: BASIC INTEGER PARAMETERS	DATA 310
C		DATA 315
	READ 9, (DATNAN(I),I=1,20)	DATA 320
	READ 10, NWP,NEL,NHAT,NCH,NTI,KSS,KSTR,KCP,NSTRT,NITER,ILUMP,INID,	DATA 325
	> ICONFI,NDTCHG,IPNTS	DATA 330
	IF (NDTCHG.LE.0) NDTCHG=1	DATA 335
C		DATA 340
C	***** DATA SET 3: BASIC REAL PARAMETERS	DATA 345
C		DATA 350
	READ 9, (DATNAN(I),I=1,20)	DATA 355
	READ 20, DELT,CHNG,DELMAX,THAX,TOLA,TOLB,RHO,GRAV,VISC,W,ONE	DATA 360
	DELTO=DELT	DATA 365
C		DATA 370
C	***** DATA SET 4: LINE-PRINT AND DISK STORE CONTROL	DATA 375
C		DATA 380
C	----- READ LINE-PRINTER CONTROLS	DATA 385
C		DATA 390
	READ 9, (DATNAN(I),I=1,20)	DATA 395
	READ 30, KPRO, (KPR(ITH),ITH=1,NTI)	DATA 400
C		DATA 405
C	----- READ AUXILIARY STORAGE CONTROL	DATA 410
C		DATA 415
	READ 30, KDSKO, (KDSK(ITH),ITH=1,NTI)	DATA 420
C		DATA 425
C	----- READ TIME FOR RESETTNG TIME STEP SIZE TO INITIAL SIZE	DATA 430
C		DATA 435
	READ 20, (TDTCH(I),I=1,NDTCHG)	DATA 440
C		DATA 445
	IF (THAX.LE.0.0) THAX=1.0E50	DATA 450
C		DATA 455
	PRINT 1000, NWP,NEL,NHAT,NCH,NTI,KSS,KSTR,KCP,NSTRT,NITER,	DATA 460
	> ILUMP,INID,ICONFI,NDTCHG,IPNTS	DATA 465
	PRINT 1100, DELT,CHNG,DELMAX,THAX,TOLA,TOLB,RHO,GRAV,VISC,W,ONE	DATA 470
	PRINT 1210, KPRO, (KPR(ITH),ITH=1,NTI)	DATA 475
	PRINT 1220, KDSKO, (KDSK(ITH),ITH=1,NTI)	DATA 480
	PRINT 1230, (TDTCH(I),I=1,NDTCHG)	DATA 485
C		DATA 490
C	***** DATA SET 5: MATERIAL PROPERTIES	DATA 495
C		DATA 500
	READ 9, (DATNAN(I),I=1,20)	DATA 505
	PRINT 1400, ((FNAT(I,J),I=1,3),J=1,7)	DATA 510
	DO 110 I=1,NHAT	DATA 515
	READ 20, (PROP(I,J),J=1,7)	DATA 520
	110 PRINT 1410, I, (PROP(I,J),J=1,7)	DATA 525
C		DATA 530
	IF (KCF.EQ.0) GO TO 140	DATA 535
C		DATA 540
C	----- CONVERT PERMEABILITY TO CONDUCTIVITY	DATA 545
C		DATA 550
	DO 130 I=1,NHAT	DATA 555
	PKCF=RHO*GRAV/VISC	DATA 560
	PROP(I,4)=PROP(I,4)*PKCF	DATA 565
	PROP(I,5)=PROP(I,5)*PKCF	DATA 570
	PROP(I,6)=PROP(I,6)*PKCF	DATA 575
	130 CONTINUE	DATA 580
C		DATA 585
C	----- IF RESTART, DATA SETS 6 THROUGH 10 ARE NOT NEEDED SINCE THEY	DATA 590
C	----- ARE READ IN VIA LOGICAL UNIT 2.	DATA 595
C		DATA 600

APPENDIX C. (continued)

```

140 IF (NSTRT.GT.0) GO TO 450
C
C ***** DATA SET 6: BED ROCK ELEVATION OR AQUIFER THICKNESS
C
      READ 9, (DATNAM(I),I=1,20)
      CALL READR(B,MAXBP,NBP)
C
C ***** DATA SET 7: NODAL POINT COORDINATE
C
      READ 9, (DATNAM(I),I=1,20)
      CALL READR(X,MAXNP,NNP)
      CALL READR(Z,MAXNP,NNP)
C
C ----- PRINT NODE COORDINATES AND BED ELEVATION OR AQUIFER THICKNESS
C
      KLINE=-1
      DO 265 NI=1,NBP,3
        NJNN=NJ
        NJNX=NINO(NI+2,NBP)
        KLINE=KLINE+1
        IF (MOD(KLINE,50).EQ.0) PRINT 2100
        PRINT 2110, (NJ,X(NJ),Z(NJ),B(NJ),NJ=NJNN,NJNX)
265 CONTINUE
C
C ***** DATA SET 8: ELEMENT INCIDENCE AND MATERIAL TYPES
C
C ----- ALSO COMPUTE MAXIMUM NODAL DIFFERENCE FOR EACH ELEMENT
C
      READ 9, (DATNAM(I),I=1,20)
      MAXDIF = 0
      NJ = 0
270 READ 10, NI, (IE(NI,I),I=1,5), MODL,NLAY
      NTYP=IE(NI,5)
C
      NQ=4
      NQ1=3
      IEN4=IE(NI,4)
      IF (IEN4.EQ.0) NQ=3
      IF (IEN4.EQ.0) NQ1=2
      DO 280 IQ=1,NQ1
        IQ1=IQ+1
        DO 280 JQ=IQ1,NQ
          ND=IAES(IE(NI,IQ)-IE(NI,JQ))
280 MAXDIF=MAX0(ND,MAXDIF)
C
290 NJ = NJ + 1
      IF (NI-NJ) 300,330,310
300 PRINT 3000, NI
      ISTOP = ISTOP + 1
310 DO 320 IQ=1,NQ
320 IE(NJ,IQ)=IE(NJ-1,IQ) + 1
      IE(NJ,5) = IE(NJ-1,5)
330 IF (NJ.LI.NI) GO TO 290
      IF (NJ.EQ.NEI) GO TO 370
      IF (NOLL.LE.0) GO TO 270
C
      DO 360 I=1,NLAY
      LL=2
      DO 360 J=1,MODL

```

DATA 605
 DATA 610
 DATA 615
 DATA 620
 DATA 625
 DATA 630
 DATA 635
 DATA 640
 DATA 645
 DATA 650
 DATA 655
 DATA 660
 DATA 665
 DATA 670
 DATA 675
 DATA 680
 DATA 685
 DATA 690
 DATA 695
 DATA 700
 DATA 705
 DATA 710
 DATA 715
 DATA 720
 DATA 725
 DATA 730
 DATA 735
 DATA 740
 DATA 745
 DATA 750
 DATA 755
 DATA 760
 DATA 765
 DATA 770
 DATA 775
 DATA 780
 DATA 785
 DATA 790
 DATA 795
 DATA 800
 DATA 805
 DATA 810
 DATA 815
 DATA 820
 DATA 825
 DATA 830
 DATA 835
 DATA 840
 DATA 845
 DATA 850
 DATA 855
 DATA 860
 DATA 865
 DATA 870
 DATA 875
 DATA 880
 DATA 885
 DATA 890
 DATA 895
 DATA 900

APPENDIX C. (continued)

```

      IF (NJ.EQ.NI) GO TO 350
      DO 340 KQ=1,NQ
340  IE(NJ,KQ)=IE(NJ-1,KQ) + LL
      IE(NJ,5)=IE(NJ-1,5)
350  LL=1
360  NJ=NJ+1
      NJ = NJ - 1
      IF (NJ.IT.NEL) GO TO 270
370  READ 9, (DATNAM(I),I=1,20)
C ***** DATA SET 9: MATERIAL CORRECTIONS
C
      IF (NCH.LE.0) GO TO 410
      READ 9, (DATNAM(I),I=1,20)
      CALL READM(IE(1,5),NAXEL,NCH)
C ----- PRINT ELEMENT INCIDENCE AND MATERIAL TYPES FOR EACH ELEMENT
C
410  KLINE=-1
      DO 415 NI=1,NEL,3
      NJNN=NI
      NJNX=NINO(NI+2,NEL)
      KLINE=KLINE+1
      IF (MOD(KLINE,50).EQ.0) PRINT 2700
      PRINT 3100, (NJ, (IE(NJ,IQ),IQ=1,5), NJ=NJNN, NJNX)
415  CONTINUE
C ----- CHECK IF MATERIAL TYPE FOR EACH ELEMENT IS CORRECT
C
      DO 420 N=1,NEL
      NTYP=IE(N,5)
      HTYP=IABS(NTYP)
      IF (HTYP.GT.0 .AND. HTYP.LE.NNAT) GO TO 420
      PRINT 4000, N
      ISTOP=ISTOP+1
420  CONTINUE
      IF (ISTOP.EQ.0) GO TO 430
      PRINT 4100, ISTOP
      STOP
C ***** DATA SET 10: INITIAL CONDITIONS
C
430  TIME=0.0D0
      READ 9, (DATNAM(I),I=1,20)
      CALL READR(H,NAXNP,NUP)
C ----- IDENTIFY BOUNDARY ELEMENT SIDES AND COMPUTE DIRECTIONAL COSINES
C ----- OF EACH BOUNDARY SIDE
C
      CALL SURF(X,Z,IE, DLB,DCOSIB,DCCSZB,NBE,ISB,NPB,
      > NAXNP,NAXEL, NAXBEL,NAXBNP)
C
      GO TO 500
C ----- READ DATA SETS 6 THROUGH 10 VIA LOGICAL UNIT 2 IF NSTRT > 0
C
450  CONTINUE
      REWIND 1
      REWIND 2
      READ (2) (DUN,I=1,9), IDUN,NPT,NET,NBP,NBEL, IDUN,NRUPR,NRLPR

```

DATA 905
 DATA 910
 DATA 915
 DATA 920
 DATA 925
 DATA 930
 DATA 935
 DATA 940
 DATA 945
 DATA 950
 DATA 955
 DATA 960
 DATA 965
 DATA 970
 DATA 975
 DATA 980
 DATA 985
 DATA 990
 DATA 995
 DATA1000
 DATA1005
 DATA1010
 DATA1015
 DATA1020
 DATA1025
 DATA1030
 DATA1035
 DATA1040
 DATA1045
 DATA1050
 DATA1055
 DATA1060
 DATA1065
 DATA1070
 DATA1075
 DATA1080
 DATA1085
 DATA1090
 DATA1095
 DATA1100
 DATA1105
 DATA1110
 DATA1115
 DATA1120
 DATA1125
 DATA1130
 DATA1135
 DATA1140
 DATA1145
 DATA1150
 DATA1155
 DATA1160
 DATA1165
 DATA1170
 DATA1175
 DATA1180
 DATA1185
 DATA1190
 DATA1195
 DATA1200

APPENDIX C. (continued)

```

      IF (KSTR.EQ.1) WRITE(1) (TITLE(I),I=1,9),NPROB,NWP,NEL,NBN,NBEL,
      > NTL,NHUPR,NHLPR
C
      READ(2) (X(NP),NP=1,NPT),(Z(NP),NP=1,NPT),(B(NP),NP=1,NPT),((IE(N,
      1IQ),N=1,NPT),IQ=1,5),(DLB(N),N=1,NBEL),(DCOSXB(N),N=1,NBEL),
      2 (DCOSZB(N),N=1,NBEL),(NBE(N),N=1,NBEL),((ISB(N,IQ),N=1,NBEL),IQ=
      3 1,4),(NPB(NF),NP=1,NBN),(RKU(I),I=1,NHUPR),(IHUTYP(N),N=1,NEL),
      4 (RKL(I),I=1,NHLPR),(IHLTYF(N),N=1,NEL)
C
      IF (KSTR.EQ.1) WRITE(1) (X(NP),NP=1,NWP),(Z(NP),NP=1,NBN),
      1 (B(NP),NP=1,NWP),((IE(N,IQ),N=1,NEL),IQ=1,5),(DLB(N),N=1,NBEL),
      2 (DCOSXB(N),N=1,NBEL),(DCOSZB(N),N=1,NBEL),(NBE(N),N=1,NBEL),
      3 ((ISB(N,IQ),N=1,NBEL),IQ=1,4),(NPB(NP),NP=1,NBN),
      4 (RKU(I),I=1,NHUPR),(IHUTYP(N),N=1,NEL),
      5 (RKL(I),I=1,NHLPR),(IHLTYF(N),N=1,NEL)
C
      DO 470 ITH=1,NSTRT
      READ(2) TIME,(H(NP),NP=1,NPT),(VX(NP),NP=1,NPT),(VZ(NP),NP=1,NPT),
      1 (HU(I),I=1,NHUPR),(HL(I),I=1,NHLPR)
C
      IF(KSTR.EQ.0) GO TO 470
      WRITE(1) TIME,(H(NP),NP=1,NWP),(VX(NP),NP=1,NWP),(VZ(NP),NP=1,
      1 NWP),(HU(I),I=1,NHUPR),(HL(I),I=1,NHLPR)
      470 CONTINUE
C
      500 CONTINUE
C
      ***** DATA SET 11: TRANSIENT CONTROLLING INTEGERS
C
      READ 9, (DATNAM(I),I=1,20)
      READ 10, NHEL,NWNP,NWPR,NWDP,NDMP,NDPR,NDDP,NSPR,NSDP,
      1 NHUPR,NHUPR,NHLPR,NHLDP,NWNP,NWPR,NWDP
C
      PRINT 5100, NHEL,NWNP,NWPR,NWDP,NDMP,NDPR,NDDP,NSPR,NSDP,
      1 NHUPR,NHUPR,NHLPR,NHLDP,NWNP,NWPR,NWDP
C
      ***** DATA SET 12: SOURCES/SINKS
C
      ----- READ AND PRINT SOURCE PROFILES
C
      READ 9, (DATNAM(I),I=1,20)
      PRINT 5300
      DO 540 I=1,NSPE
      READ 20, (TSOS(I,J),SOS(I,J),J=1,NSDP)
      PRINT 5500, I
      PRINT 5510, (TSOS(I,J),SOS(I,J),J=1,NSDP)
      510 CONTINUE
C
      ----- READ SOURCE TYPE ASSIGNED TO EACH ELEMENT
C
      CALL READN(ISTYP,NAXEL,NEL)
      PRINT 5600, (ISTYP(I),I=1,NEL)
C
      IF(NWNP.EQ.0) GO TO 600
C
      ----- READ AND WRITE WELL SOURCE/SINK PROFILES
C
      PRINT 5700
      DO 570 I=1,NWPR
      READ 20, (TWSSP(I,J),WSSP(I,J),J=1,NWDP)

```

DATA 1205
 DATA 1210
 DATA 1215
 DATA 1220
 DATA 1225
 DATA 1230
 DATA 1235
 DATA 1240
 DATA 1245
 DATA 1250
 DATA 1255
 DATA 1260
 DATA 1265
 DATA 1270
 DATA 1275
 DATA 1280
 DATA 1285
 DATA 1290
 DATA 1295
 DATA 1300
 DATA 1305
 DATA 1310
 DATA 1315
 DATA 1320
 DATA 1325
 DATA 1330
 DATA 1335
 DATA 1340
 DATA 1345
 DATA 1350
 DATA 1355
 DATA 1360
 DATA 1365
 DATA 1370
 DATA 1375
 DATA 1380
 DATA 1385
 DATA 1390
 DATA 1395
 DATA 1400
 DATA 1405
 DATA 1410
 DATA 1415
 DATA 1420
 DATA 1425
 DATA 1430
 DATA 1435
 DATA 1440
 DATA 1445
 DATA 1450
 DATA 1455
 DATA 1460
 DATA 1465
 DATA 1470
 DATA 1475
 DATA 1480
 DATA 1485
 DATA 1490
 DATA 1495
 DATA 1500

APPENDIX C. (continued)

```

      PRINT 5710, I
      PRINT 5510, (TWSSP(I,J),WSSP(I,J),J=1,NWDP)
570 CONTINUE
C ----- READ WELL SOURCE/SINK MODES AND TYPE OF PROFILES ASSIGNED TO
C
      READ 10, (NWP(I),I=1,NWNP)
      CALL READW(INTYP,NWNP,NWNP)
C ----- PRINT GLOBAL WELL NODE NUMBERS AND PROFILE TYPE OF WELL NODE
C
      LINE=0
      DO 590 I=1,NWNP,5
      LINE=LINE+1
      IF (MOD(LINE-1,50).EQ.0) PRINT 5800
      NJNN=I
      NJNI=MIN0(I+4,NWNP)
      PRINT 5850, (J,NWP(J),INTYP(J),J=NJNN,NJNI)
590 CONTINUE
C ***** DATA SET 13: DIRICHLET BOUNDARY CONDITIONS
C ----- READ AND PRINT DIRICHLET HEAD PROFILES
C
600 IF (NNDP.EQ.0) GO TO 700
      READ 9, (DATNAB(I),I=1,20)
      PRINT 6000
      DO 610 I=1,NDDP
      READ 20, (THED(I,J),HED(I,J),J=1,NDDP)
      PRINT 6100, I
      PRINT 5510, (THED(I,J),HED(I,J),J=1,NDDP)
610 CONTINUE
C ----- READ DIRICHLET MODES AND TYPE OF DIRICHLET HEAD ASSIGNED TO
C ----- EACH CP THE DIRICHLET MODES.
C
      READ 10, (NPD(I),I=1,NNDP)
      CALL READN(IDTYP,NNDP,NNDP)
C ----- PRINT GLOBAL MODAL NUMBERS AND PROFILE TYPE OF DIRICHLET MODES
C
      LINE=0
      DO 620 I=1,NNDP,5
      LINE=LINE+1
      IF (MOD(LINE-1,50).EQ.0) PRINT 6200
      NJNN=I
      NJNI=MIN0(I+4,NNDP)
      PRINT 6250, (J,NPD(J),IDTYP(J),J=NJNN,NJNI)
620 CONTINUE
C ***** DATA 14: NEUMANN BOUNDARY CONDITIONS
C ----- READ AND PRINT NEUMANN FLUX PROFILES
C
700 IF (NUNP.EQ.0) GO TO 800
      READ 9, (DATNAB(I),I=1,20)
      PRINT 7000
      DO 710 I=1,NUNP
      READ 20, (TRNF(I,J),RNF(I,J),J=1,NUNDP)
      PRINT 7100, I
      PRINT 5510, (TRNF(I,J),RNF(I,J),J=1,NUNDP)

```

DATA1505
 DATA1510
 DATA1515
 DATA1520
 DATA1525
 DATA1530
 DATA1535
 DATA1540
 DATA1545
 DATA1550
 DATA1555
 DATA1560
 DATA1565
 DATA1570
 DATA1575
 DATA1580
 DATA1585
 DATA1590
 DATA1595
 DATA1600
 DATA1605
 DATA1610
 DATA1615
 DATA1620
 DATA1625
 DATA1630
 DATA1635
 DATA1640
 DATA1645
 DATA1650
 DATA1655
 DATA1660
 DATA1665
 DATA1670
 DATA1675
 DATA1680
 DATA1685
 DATA1690
 DATA1695
 DATA1700
 DATA1705
 DATA1710
 DATA1715
 DATA1720
 DATA1725
 DATA1730
 DATA1735
 DATA1740
 DATA1745
 DATA1750
 DATA1755
 DATA1760
 DATA1765
 DATA1770
 DATA1775
 DATA1780
 DATA1785
 DATA1790
 DATA1795
 DATA1800

APPENDIX C. (continued)

```

710 CONTINUE
C ----- READ NEUBANN NODES AND TYPE OF NEUBANN FLUX ASSIGNED TO THEM
C
      READ 10, (NPN(I),I=1,NNNP)
      CALL READN(INTYP,NXNNP,NNNP)
C ----- PRINT GLOBAL NCDAL NUMBERS AND PROFILE TYPES OF NEUBANN NODES
C
      LINE=0
      DO 720 I=1,NNNP,5
      LINE=LINE+1
      IF (MOD(LINE-1,50).EQ.0) PRINT 7200
      NJNN=I
      NJNX=NINO(I+4,NNNP)
      PRINT 6250, (J,NPN(J), INTYP(J),J=NJNN,NJNX)
720 CONTINUE
C ----- READ NEUBANN BOUNDARY ELEMENT-SIDES
C
      NODES=0
730 READ 10, NI,NSEQ,IS1,IS2,IS1AD,IS2AD
      IF (NI.EQ.0) GO TO 740
      NJ=NI+NSEQ
      DO 735 NP=NI,NJ
      IS(NP,1)=IS1+(NP-NI)*IS1AD
      IS(NP,2)=IS2+(NP-NI)*IS2AD
      NODES=NODES+1
735 CONTINUE
      GO TO 730
740 IF (NODES.EQ.NNEL) GO TO 750
      PRINT 7300
      STOP
C
750 CONTINUE
C ----- PRINT THE READ-IN NEUBANN BOUNDARY ELEMENT SIDES
C
      LINE=0
      DO 760 NP=1,NNEL,5
      LINE=LINE+1
      IF (MOD(LINE-1,50).EQ.0) PRINT 7600
      NJNN=NP
      NJNX=NINO(NP+4,NNEL)
      PRINT 7650, (J,IS(J,1),IS(J,2),J=NJNN,NJNX)
760 CONTINUE
C ***** DATA SET 15: UPPER LEAKING AQUIFERS
C ----- READ AND PRINT HEAD PROFILE IN THE UPPER LEAKING AQUIFER
C
800 CONTINUE
      READ 9, (DATNAN(I),I=1,20)
      PRINT 8000
      DO 810 I=1,NHUPR
      READ 20, (THUTAB(I,J),HUTAB(I,J),J=1,NHODP)
      PRINT 8100, I
      PRINT 5510, (THUTAB(I,J),HUTAB(I,J),J=1,NHODP)
810 CONTINUE

```

DATA1805
 DATA1810
 DATA1815
 DATA1820
 DATA1825
 DATA1830
 DATA1835
 DATA1840
 DATA1845
 DATA1850
 DATA1855
 DATA1860
 DATA1865
 DATA1870
 DATA1875
 DATA1880
 DATA1885
 DATA1890
 DATA1895
 DATA1900
 DATA1905
 DATA1910
 DATA1915
 DATA1920
 DATA1925
 DATA1930
 DATA1935
 DATA1940
 DATA1945
 DATA1950
 DATA1955
 DATA1960
 DATA1965
 DATA1970
 DATA1975
 DATA1980
 DATA1985
 DATA1990
 DATA1995
 DATA2000
 DATA2005
 DATA2010
 DATA2015
 DATA2020
 DATA2025
 DATA2030
 DATA2035
 DATA2040
 DATA2045
 DATA2050
 DATA2055
 DATA2060
 DATA2065
 DATA2070
 DATA2075
 DATA2080
 DATA2085
 DATA2090
 DATA2095
 DATA2100

APPENDIX C. (continued)

```

C ----- READ AND PRINT UPPER AQUIFER LEAKING COEFFICIENT
C
C      READ 20, (RKU(I),I=1,NHUPR)
C      PRINT 8200, (RKU(I),I=1,NHUPR)
C
C ----- READ TYPE OF UPPER LEAKING AQUIFER ASSIGNED TO EACH ELEMENT
C
C      CALL READN (IHUTYP,MAXEL,NEL)
C      PRINT 8300, (IHUTYP(I),I=1,NEL)
C
C ***** DATA SET 16: LOWER LEAKING AQUIFERS
C
C ----- READ AND PRINT HEAD PROFILE IN THE LOWER LEAKING AQUIFER
C
C      READ 9, (DATMAH(I),I=1,20)
C      PRINT 9000
C      DO 910 I=1,NHLEK
C      READ 20, (THLTAB(I,J),HLTAB(I,J),J=1,NHLDLP)
C      PRINT 9100, I
C      PRINT 5510, (THLTAB(I,J),HLTAB(I,J),J=1,NHLDLP)
C 910 CONTINUE
C
C ----- READ AND PRINT LOWER AQUIFER LEAKING COEFFICIENT
C
C      READ 20, (RKL(I),I=1,NHLPK)
C      PRINT 9200, (RKL(I),I=1,NHLPK)
C
C ----- READ TYPE OF LOWER LEAKING AQUIFER ASSIGNED TO EACH ELEMENT
C
C      CALL READN (IHLTYP,MAXEL,NEL)
C      PRINT 9300, (IHLTYP(I),I=1,NEL)
C
C
C      IF (ISTOP.EQ.0) GO TO 990
C      PRINT 9900, ISTOP
C
C 990 CONTINUE
C
C      RETURN
C
C      9 FORMAT (20A4)
C      10 FORMAT (16I5)
C      20 FORMAT (1D10.3)
C      30 FORMAT (80I1)
C      40 FORMAT (3I5,5X,4D10.3)
C
C 1000 FORMAT (1H0/5X,35H **** BASIC INTEGER PARAMETERS ****/5X,
C      > 40H NUMBER OF NODAL POINTS. . . . .,I5/ 5X,
C      > 40H NUMBER OF ELEMENTS. . . . .,I5/ 5X,
C      > 40H NUMBER OF DIFFERENT MATERIALS. . . . .,I5/ 5X,
C      > 40H NUMBER OF CORRECTION MATERIALS. . . . .,I5/ 5X,
C      > 40H NUMBER OF TIME INCREMENTS. . . . .,I5/5X,
C      > 40H STEADY-STATE I.C. CONTROL. . . . .,I5/ 5X,
C      > 40H AUXILIARY STORAGE CONTROL. . . . .,I5/ 5X,
C      > 40H CONDUCTIVITY-PERMEABILITY CONTROL. . . . .,I5/ 5X,
C      > 40H RESTART PARAMETER. . . . .,I5/ 5X,
C      > 40H NO. OF ITERATION CYCLE. . . . .,I5/5X,
C      > 40H LUMPING INDICATOR, ILUMP. . . . .,I5/ 5X,
C      > 40H MID-DIFFERENCE CONTROL, IMID. . . . .,I5/ 5X,
C      > 40H CONFINED AQUIFER CONTROL, ICONFI. . . . .,I5/ 5X,

```

DATA2105
 DATA2110
 DATA2115
 DATA2120
 DATA2125
 DATA2130
 DATA2135
 DATA2140
 DATA2145
 DATA2150
 DATA2155
 DATA2160
 DATA2165
 DATA2170
 DATA2175
 DATA2180
 DATA2185
 DATA2190
 DATA2195
 DATA2200
 DATA2205
 DATA2210
 DATA2215
 DATA2220
 DATA2225
 DATA2230
 DATA2235
 DATA2240
 DATA2245
 DATA2250
 DATA2255
 DATA2260
 DATA2265
 DATA2270
 DATA2275
 DATA2280
 DATA2285
 DATA2290
 DATA2295
 DATA2300
 DATA2305
 DATA2310
 DATA2315
 DATA2320
 DATA2325
 DATA2330
 DATA2335
 DATA2340
 DATA2345
 DATA2350
 DATA2355
 DATA2360
 DATA2365
 DATA2370
 DATA2375
 DATA2380
 DATA2385
 DATA2390
 DATA2395
 DATA2400

APPENDIX C. (continued)

```

> 40H NO. OF TIMES TO RESET TIME STEP SIZE . . ,15/ 5X, DATA2405
> 40H IS POINTWISE SOLUTION TO BE USED . . . ,15//) DATA2410
1100 FORMAT(5X,40H TIME INCREMENT. . . . . ,F10.3/ 5X, DATA2415
> 40H FRACTION FOR INCREASING TIME STEP SIZE ,F10.6/ 5X, DATA2420
> 40H MAXIMUM VALUE OF DELT . . . . . ,D10.3/ 5X, DATA2425
> 40H MAXIMUM VALUE OF TIME . . . . . ,D10.3//5X, DATA2430
> 40H STEADY-STATE TOLERANCE. . . . . ,F10.6/ 5X, DATA2435
> 40H TRANSIENT-STATE TOLERANCE . . . . . ,F10.6//5X, DATA2440
> 40H DENSITY OF WATER. . . . . ,F10.3/ 5X, DATA2445
> 40H ACCELERATION OF GRAVITY . . . . . ,F10.3/ 5X, DATA2450
> 40H VISCOSITY OF WATER. . . . . ,F10.6//5X, DATA2455
> 40H TIME-INTEGRATION PARAMETER. . . . . ,F10.6/ 5X, DATA2460
> 40H ITERATION PARAMETER . . . . . ,F10.6/) DATA2465
1210 FORMAT(1H//6X,26H **** LINE PRINT CONTROL **/(7X,50I2)) DATA2470
1220 FORMAT(1H0/6X,26H **** STORAGE CONTROL **/(7X,50I2)) DATA2475
1230 FORMAT(1H0/6X,29H **** TIME TO RESET DELT **/(10X,8D12.4)) DATA2480
1400 FORMAT(1H1/5X,26H *** MATERIAL PROPERTIES **//9H MAT. NO.,9(3A4)) DATA2485
1410 FORMAT(1H,9D12.4) DATA2490
2100 FORMAT(1H1/5X,27H **** NODAL POINT DATA ****//
1 3(2X,5H NODE,9X,2H X,9X,2H Z,9X,2H P)/ 3(2X,5H ----,9X,2H -,9X,
2 2H -,9X,2H -)) DATA2495
2110 FORMAT( 3(2X,15,3D11.3)) DATA2500
2700 FORMAT(1H1/5X,55H **** ELEMENT DATA: GLOBAL INDICES OF ELEMENT MOD DATA2505
1E ****/3(5X,5H ELN,5H MOD1,5H MOD2,5H MOD3,5H MOD4,5H MTP)/ DATA2510
2 3(5X,5H ---,5H ----,5H ----,5H ----,5H ----)) DATA2515
3000 FORMAT(////26H ERROR IN ELEMENT CARD NO.,15//) DATA2520
3100 FORMAT(3(5X,6I5)) DATA2525
4000 FORMAT(////40H ERROR IN MATERIAL TYPE CODE FOR ELEMENT,15//) DATA2530
4100 FORMAT(////28H EXECUTION HALTED BECAUSE OF,15,13H FATAL ERRORS//) DATA2535
5100 FORMAT(1H1/5X,39H **** TRANSIENT INTEGER PARAMETERS ****//5X, DATA2540
> 40H NO. OF NEUMANN ELEMENT SIDE . . . . . ,15/ 5X, DATA2545
> 40H NO. OF NEUMANN NODAL POINTS . . . . . ,15/ 5X, DATA2550
> 40H NO. OF NEUMANN CONDITION PROFILES . . . . ,15/ 5X, DATA2555
> 40H NO. OF NEUMANN CONDITION PARAMETERS . . ,15/ 5X, DATA2560
> 40H NO. OF DIRICHLET NODAL POINTS . . . . . ,15/ 5X, DATA2565
> 40H NO. OF DIRICHLET CONDITION PROFILES . . ,15/ 5X, DATA2570
> 40H NO. OF DIRICHLET CONDITION PARAMETERS . ,15/ 5X, DATA2575
> 40H NO. OF SOURCE PROFILES . . . . . ,15/ 5X, DATA2580
> 40H NO. OF SOURCE PARAMETERS . . . . . ,15/ 5X, DATA2585
> 40H NO. OF HU PROFILES . . . . . ,15/ 5X, DATA2590
> 40H NO. OF DATA POINTS IN HU PROFILE . . . . ,15/ 5X, DATA2595
> 40H NO. OF HL PROFILES . . . . . ,15/ 5X, DATA2600
> 40H NO. OF DATA POINTS IN HL PROFILE . . . . ,15/ 5X, DATA2605
> 40H NO. OF WELL SOURCE/SINK MODES . . . . . ,15/ 5X, DATA2610
> 40H NO. OF WELL SOURCE/SINK PROFILES . . . . ,15/ 5X, DATA2615
> 40H NO. OF DATA POINTS IN WELL PROFILE . . ,15/) DATA2620
5300 FORMAT(1H1/5X,27H *** SOURCE INFORMATION ***) DATA2625
5500 FORMAT(1H0/5X,12H PROFILE NO.,12,/ 4(4X,4HTIME,6X,6HSOURCE,2X)/ DATA2630
> 4(4X,4H----,6X,6H----,2X)) DATA2635
5510 FORMAT(1H ,4(2D11.3)) DATA2640
5600 FORMAT(1H0//5X,60H LIST OF SOURCE PROFILE TYPE ASSIGNED TO ELEMENT DATA2645
1 1,2,....,NEL// (4X,30I3)) DATA2650
5700 FORMAT(1H0//5X,37H *** WELL SOURCE/SINK INFORMATION ***) DATA2655
5710 FORMAT(1H0/5X,12H PROFILE NO.,12/ 4(4X,4HTIME,6X,6HSOURCE,2X)/ DATA2660
1 4(4X,4H----,6X,6H----,2X)) DATA2665
5800 FORMAT(1H0//10X,65H GLOBAL NODAL NUMBER AND PROFILE TYPE OF WELL S DATA2670
OURCE/SINK MODES //5X,5(5H 1,5H MPN,5H MTP,5X)) DATA2675
5850 FORMAT(1H ,4X,5(3I5,5X)) DATA2680
6000 FORMAT(1H1/5X,40H **** DIRICHLET BOUNDARY CONDITIONS ****) DATA2685
6100 FORMAT(1H0/5X,12H PROFILE NO.,12,/ 4(4X,4HTIME,6X,6H HEAD ,2X)/ DATA2690
DATA2695
DATA2700
DATA2705
DATA2710
DATA2715
DATA2720
DATA2725
DATA2730
DATA2735
DATA2740
DATA2745
DATA2750
DATA2755
DATA2760
DATA2765
DATA2770
DATA2775
DATA2780
DATA2785
DATA2790
DATA2795
DATA2800
DATA2805
DATA2810
DATA2815
DATA2820
DATA2825
DATA2830
DATA2835
DATA2840
DATA2845
DATA2850
DATA2855
DATA2860
DATA2865
DATA2870
DATA2875
DATA2880
DATA2885
DATA2890
DATA2895
DATA2900
DATA2905
DATA2910
DATA2915
DATA2920
DATA2925
DATA2930
DATA2935
DATA2940
DATA2945
DATA2950
DATA2955
DATA2960
DATA2965
DATA2970
DATA2975
DATA2980
DATA2985
DATA2990
DATA2995

```

APPENDIX C. (continued)

```

      > 4(4X,4H----,6X,6H ---- ,2X)
6200 FORMAT(1H0//10X,65H GLOBAL NODAL NUMBER AND PROFILE TYPE OF DIRICH
      1LET BOUNDARY NODES//5X,5(5H I,5H NPDB,5H DTYP,5X))
6250 FORMAT(1H ,4X,5(3I5,5X))
7000 FORMAT(1H1/5X,36H **** NEUMAN BOUNDARY CONDITIONS ****)
7100 FORMAT(1H0,5X,12H PROFILE NO.,12//4(4X,4HTIME,6X,6H FLUX ,2X)/
      > 4(4X,4H----,6X,6H ---- ,2X))
7200 FORMAT(1H0//10X,63H GLOBAL NODAL NUMBER AND PROFILE TYPE OF NEUMAN
      1H BOUNDARY NODES//5X,5(5H I,5H NPDB,5H DTYP,5X))
7300 FORMAT(/// 36H ERROR IN READING NEUMAN ELEMENT SIDE//)
7600 FORMAT(1H0//10X,27H NEUMAN ELEMENT SIDE DATA //5X,
      1 5(5H N,5H CNP1,5H CNP2,2X))
7650 FORMAT(1H ,4X,5(3I5,2X))
8000 FORMAT(1H1/5X,43H **** UPPER AQUIFER LEAKING PARAMETERS ****)
8100 FORMAT(1H0/10X,43H --- UPPER LEAKY AQUIFER HEAD PROFILE NO.,12,
      1 4H ----//4(4X,4HTIME,6X,6H HEAD ,2X)/ 4(4X,4H----,6X,6H ---- ,2X))
8200 FORMAT(1H0/10X,77H --- LIST OF UPPER AQUIFER LEAKING COEFFICIENT
      1 FOR PROFILES 1, 2, .... ---/ (4D20.3))
8300 FORMAT(1H0//10X,75H --- LIST OF UPPER LEAKY AQUIFER TYPE ASSIGNED
      1 TO ELEMENT 1,2,.....,NEL ---/(5X,30I3))
9000 FORMAT(1H1/5X,43H **** LOWER AQUIFER LEAKING PARAMETERS ****)
9100 FORMAT(1H0/10X,43H --- LOWER LEAKY AQUIFER HEAD PROFILE NO.,12,
      1 4H ----//4(4X,4HTIME,6X,6H HEAD ,2X)/ 4(4X,4H----,6X,6H ---- ,2X))
9200 FORMAT(1H0/10X,77H --- LIST OF LOWER AQUIFER LEAKING COEFFICIENT
      1 FOR PROFILES 1, 2, .... ---/ (4D20.3))
9300 FORMAT(1H0//10X,75H --- LIST OF LOWER LEAKY AQUIFER TYPE ASSIGNED
      1 TO ELEMENT 1,2,.....,NEL ---/(5X,30I3))
9900 FORMAT(///45H ASSEMBLY AND SOLUTION WILL NOT BE PERFORMED ,15,
      > 19H FATAL CARD ERRORS///)
C
      END
DATA2705
DATA2710
DATA2715
DATA2720
DATA2725
DATA2730
DATA2735
DATA2740
DATA2745
DATA2750
DATA2755
DATA2760
DATA2765
DATA2770
DATA2775
DATA2780
DATA2785
DATA2790
DATA2795
DATA2800
DATA2805
DATA2810
DATA2815
DATA2820
DATA2825
DATA2830
DATA2835
DATA2840
DATA2845
DATA2850
DATA2855

```


APPENDIX C. (continued)

```

      SUBROUTINE SURF(X,Z,IE, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,
      > NAXNP,NAXEL, NAYBEL,NAXBNP)
C
C
C FUNCTION OF SUBROUTINE--TO IDENTIFY BOUNDING SIDES THROUGH THE ARRAY
C ISB(NP,4), TO CALCULATE THEIR LENGTHS DLB(NP), AND TO DETERMINE THE
C DIRECTION COSINES DCOSX(NP) AND DCOSZ(NP) OF THE OUTWARDLY DIRECTED
C UNIT NORMAL VECTOR FOR EACH BOUNDARY ELEMENT NBE(NP).
C
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      COMMON /GEOM/ NNP,NEL,IBAND,INBP,IBALPB, NBE1,NBN
C
C      DIMENSION X(NAXNP),Z(NAXNP),IE(NAXEL,5)
C      DIMENSION DLB(NAXBEL),DCOSXB(NAXBEL),DCOSZB(NAXBEL),NBE(NAXBEL),
C      > ISB(NAXBEL,4),NPB(NAXBNP)
C
C ----- FIND SURFACE SIDES BY LOCATING NONDUPLICATED SIDES
C
C      NBE1=0
C      NBN=0
C      DO 39C NI=1,NEL
C      NQ=4
C      IEN4=IE(NI,4)
C      IF (IEN4.EQ.0) NQ=3
C      DO 380 IQ=1,NQ
C      IQ1=IQ+1
C      IF (IQ.EQ.NQ) IQ1=1
C
C      DO 190 NJ=1,NEL
C      IF (NJ.EQ.NI) GO TO 190
C      NQ=4
C      IEN4=IE(NJ,4)
C      IF (IEN4.EQ.0) NQ=3
C      DO 18C JQ=1,NQ
C      JQ1=JQ+1
C      IF (JQ.EQ.NQ) JQ1=1
C      IF (IE(NI,IQ).EQ.IE(NJ,JQ).AND.IE(NI,IQ1).EQ.IE(NJ,JQ1)) GO TO 380
C      IF (IE(NI,IQ).EQ.IE(NJ,JQ1).AND.IE(NI,IQ1).EQ.IE(NJ,JQ)) GO TO 380
C
C 180 CONTINUE
C 190 CONTINUE
C
C ***** A BOUNDARY SIDE HAS BEEN FOUND.
C
C      NI=IE(NI,IQ)
C      NJ=IE(NI,IQ1)
C      NBE1=NBE1+1
C      NBE(NBE1)=NI
C      ISB(NBE1,1)=NI
C      ISB(NBE1,2)=NJ
C      ISB(NBE1,3)=IQ
C      ISB(NBE1,4)=IQ1
C
C      IF (NBE1.GT.1) GO TO 200
C
C      NBN=NBN+1
C      NPB(NBN)=NI
C      NBN=NBN+1
C      NPB(NBN)=NJ

```

```

SURF 005
SURF 010
SURF 015
SURF 020
SURF 025
SURF 030
SURF 035
SURF 040
SURF 045
SURF 050
SURF 055
SURF 060
SURF 065
SURF 070
SURF 075
SURF 080
SURF 085
SURF 090
SURF 095
SURF 100
SURF 105
SURF 110
SURF 115
SURF 120
SURF 125
SURF 130
SURF 135
SURF 140
SURF 145
SURF 150
SURF 155
SURF 160
SURF 165
SURF 170
SURF 175
SURF 180
SURF 185
SURF 190
SURF 195
SURF 200
SURF 205
SURF 210
SURF 215
SURF 220
SURF 225
SURF 230
SURF 235
SURF 240
SURF 245
SURF 250
SURF 255
SURF 260
SURF 265
SURF 270
SURF 275
SURF 280
SURF 285
SURF 290
SURF 295
SURF 300

```

APPENDIX C. (continued)

C		SURF 305
200	DO 210 I=1,NBN	SURF 310
	IJ=NFB(I)	SURF 315
	IF(IJ.EQ.NI) GC TO 250	SURF 320
210	CONTINUE	SURF 325
	NBN=NBN+1	SURF 330
	NFB(NBN)=NI	SURF 335
C		SURF 340
250	DO 260 J=1,NBN	SURF 345
	IJ=NFB(J)	SURF 350
	IF(IJ.EQ.NJ) GC TO 290	SURF 355
260	CONTINUE	SURF 360
	NBN=NBN+1	SURF 365
	NFB(NBN)=NJ	SURF 370
C		SURF 375
290	CONTINUE	SURF 380
C		SURF 385
380	CONTINUE	SURF 390
390	CONTINUE	SURF 395
C		SURF 400
C	----- CALCULATE SIDE LENGTH AND DIRECTION COSINES	SURF 405
C		SURF 410
	DO 490 NP=1,NBEL	SURF 415
	NI=ISB(NP,1)	SURF 420
	NJ=ISB(NP,2)	SURF 425
	DX=X(NJ)-X(NI)	SURF 430
	DZ=Z(NJ)-Z(NI)	SURF 435
	DLB(NP)=DSQRT(DX*DX+DZ*DZ)	SURF 440
C		SURF 445
	BETA=ATAN2(DZ,DX)	SURF 450
	DCOSXB(NP)=DSIN(BETA)	SURF 455
	DCOSZB(NP)=-DCCS(BETA)	SURF 460
490	CONTINUE	SURF 465
C		SURF 470
C		SURF 475
C	----- PRINT BOUNDARY SIDE INFORMATION	SURF 480
C		SURF 485
	LINE=0	SURF 490
	DO 695 NP=1,NBEL,2	SURF 495
	LINE=LINE+1	SURF 500
	IF(MOD(LINE-1,50).EQ.0) PRINT 6900	SURF 505
	NJNN=NF	SURF 510
	NJNX=MINO(NP+1,NBEL)	SURF 515
	PRINT 6950, (J, NBE(J), (ISB(J,I), I=1,4), DLB(J), DCOSXB(J), DCOSZB(J)	SURF 520
	1, J=NJNN, NJNX)	SURF 525
	695 CONTINUE	SURF 530
C		SURF 535
6900	FORMAT (1H1//10X,51H *** COMPUTED BOUNDARY ELEMENT SIDE INFORMATION	SURF 540
	1 ***//5X,2(5H NP,5H N,5H NP1,5H NP2,3H L1,3H L2,11H DLB(B	SURF 545
	2P) ,11H DCOSXB(NP),11H DCOSZB(NP),1X))	SURF 550
6950	FORMAT (1H ,4X,2(4I5,2I3,3D11.3,1X))	SURF 555
C		SURF 560
	RETURN	SURF 565
	END	SURF 570

APPENDIX C. (continued)

```

SUBROUTINE LHDGEN(LLNODE,IE,HAXNP,HAXEL,NWP,NEL)
IMPLICIT REAL*8(A-H,O-Z)
C
C   DIMENSION LLNODE(5,HAXNP),IE(HAXEL,5)
C
C   DO 490 NP=1,NWP
C
C   DO 110 I=1,4
110  LLNODE(I,NP)=0
C
C   NLNOD=0
C   KOUNT=0
C
C   DO 390 N=1,NEL
C     IEN1=IE(N,1)
C     IEN2=IE(N,2)
C     IEN3=IE(N,3)
C     IEN4=IE(N,4)
C     NQ=4
C     IF(IEN4.EQ.0) NQ=3
C
C     IF(IEN4.EQ.0) GO TO 210
C
C     ----- FOR THE CASE OF QUADRILATERAL ELEMENT
C
C     IF(IEN1.NE.NP .AND. IEN2.NE.NP .AND. IEN3.NE.NP .AND. IEN4.NE.NP)
1  GO TO 390
C     GO TO 220
C
C     ----- FOR THE CASE OF TRIANGULAR ELEMENT
C
C     210 IF(IEN1.NE.NP .AND. IEN2.NE.NP .AND. IEN3.NE.NP) GO TO 390
C
C     ----- MERGE THE QUADRILATERAL AND TRIANGULAR ELEMENT BRANCH
C
C     220 DO 250 IQ=1,NQ
C       NI=IE(N,IQ)
C
C       ----- WHEN NI IS TO THE RIGHT OF NP, ONE SKIP THE COUNT
C
C       IF(NI.GE.NP) GO TO 290
C
C       ----- WHEN NI IS TO THE LEFT OF NP, ONE COMPRESS THE NODE
C
C       NLNOD=NLNOD+1
C       IF(NLNOD.GT.1) GO TO 230
C
C       ----- THE FIRST LOWER-LEFT NODE IS ENCOUNTERED
C
C       KOUNT=KOUNT+1
C       LLNODE(KOUNT,NP)=NI
C       GO TO 290
C
C       ----- IF NLNOD IS GREATER THAN 1, WE HAVE TO CHECK IF IT IS THE
C       ----- NODE ALREADY COMPRESSED? IF YES, SKIP. IF NOT INCREASE THE
C       ----- KOUNT.
C
C     230 DO 240 J=1,KOUNT
C       NJ=LLNODE(J,NP)
C       IF(NI.EQ.NJ) GO TO 290

```

```

LWDG 005
LWDG 010
LWDG 015
LWDG 020
LWDG 025
LWDG 030
LWDG 035
LWDG 040
LWDG 045
LWDG 050
LWDG 055
LWDG 060
LWDG 065
LWDG 070
LWDG 075
LWDG 080
LWDG 085
LWDG 090
LWDG 095
LWDG 100
LWDG 105
LWDG 110
LWDG 115
LWDG 120
LWDG 125
LWDG 130
LWDG 135
LWDG 140
LWDG 145
LWDG 150
LWDG 155
LWDG 160
LWDG 165
LWDG 170
LWDG 175
LWDG 180
LWDG 185
LWDG 190
LWDG 195
LWDG 200
LWDG 205
LWDG 210
LWDG 215
LWDG 220
LWDG 225
LWDG 230
LWDG 235
LWDG 240
LWDG 245
LWDG 250
LWDG 255
LWDG 260
LWDG 265
LWDG 270
LWDG 275
LWDG 280
LWDG 285
LWDG 290
LWDG 295
LWDG 300

```

APPENDIX C. (continued)

240 CONTINUE	LNDG 305
C	LNDG 310
C ----- THE NODE NI HAS NOT BEEN COMPRESSED YET. HENCE COMPRESSED IT	LNDG 315
C	LNDG 320
KOUNT=KOUNT+1	LNDG 325
LLNODE(KOUNT,NF)=NI	LNDG 330
C	LNDG 335
290 CONTINUE	LNDG 340
C	LNDG 345
390 CONTINUE	LNDG 350
C	LNDG 355
IF(KOUNT.LE.4) GO TO 470	LNDG 360
WRITE(6,1000) NF	LNDG 365
STOP	LNDG 370
C	LNDG 375
470 LLNODE(5,NF)=NF	LNDG 380
C	LNDG 385
490 CONTINUE	LNDG 390
C	LNDG 395
LINE=0	LNDG 400
DO 590 NP=1,NPF,4	LNDG 405
NJHN=NF	LNDG 410
NJHX=NINO(NF+3,NPF)	LNDG 415
LINE=LINE+1	LNDG 420
IF(EOB(LINE-1,50).EQ.0) PRINT 5000	LNDG 425
PRINT 5100, (NJ,(LLNODE(1,NJ),I=1,5),NJ=NJHN,NJHX)	LNDG 430
590 CONTINUE	LNDG 435
C	LNDG 440
1000 FORMAT(1H1/5X,'*** ',14,'-TH NODE HAS MORE THAN 4 NODES TO ITS LO	LNDG 445
WER-LEFT SIDE: STOP ***')	LNDG 450
5000 FORMAT(1H1/5X,46H ** GENERATED LOWER-LEFT NODES ALL NODES **,//	LNDG 455
1 4(2X,5H NF,5H 1,5H 2,5H 3,5H 4,5H 5) /	LNDG 460
2 4(2X,5H --,5H -,5H -,5H -,5H -,5H -) //)	LNDG 465
5100 FORMAT(1H ,4(2X,6I5))	LNDG 470
C	LNDG 475
RETURN	LNDG 480
END	LNDG 485

APPENDIX C. (continued)

```

      SUBROUTINE VELT(X,Z,IE, C,H,  VI,VZ,AKX,AKZ,AKYZ,
      > HAXEL,HAXNP,HAXHBP)
C
C  FUNCTION OF SUBROUTINE TC COMPUTE DARCY VELOCITY VI AND VZ
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /GEOM/ NNP,NEL,IBAND,IBBP,IBALFB, NEEL,NBN
      COMMON /OPT/ ILUMP,IBID,ICCNFI,IPNTS
C
      DIMENSION X(HAXNP),Z(HAXNP),IE(HAXEL,5)
      DIMENSION C(HAXNP,HAXHBP),H(HAXNP),VI(HAXNP),VZ(HAXNP),
      > AKX(HAXEL,4),AKZ(HAXEL,4),AKYZ(HAXEL,4)
C
      DIMENSION QQ(4,4),RQ(4),IQ(4),ZQ(4),HQ(4), AKYG(4),AKZG(4),
      > AKXZG(4)
C
C  ----- INITIATE DARCY VELOCITY VI AND VZ.
C
      DO 100 NP=1,NNP
        VX(NP)=0.0
      100 VZ(NP)=0.0
C
C  ----- COMPUTE VERTICALLY INTEGRATED VOLUMETRIC FLUX BY APPLYING FEM
C  ----- TO VERTICALLY INTEGRATED DARCY EQUATIONS.
C
      DO 390 IXZ=1,2
C
C  ----- INITIATE THE MATRIX C(NP,IB)
C
      IF(IPNTS.EQ.1) GO TO 150
C
      DO 110 NP=1,NNP
        DO 110 IB=1,IBBP
      110 C(NP,IB)=0.0
      GO TO 190
C
      150 DO 155 NP=1,NNP
      155 C(NP,1)=0.0
      190 CONTINUE
C
C  ----- COMPUTE ELEMENT MATRIX QQ(IQ,JQ) AND RQ(IQ)
C
      DO 290 H=1,NEL
        HQ=4
        IEN4=IE(H,4)
        IF(IEN4.EQ.0) HQ=3
C
      DO 210 IQ=1,HQ
        NP=IE(H,IQ)
        IQ(IQ)=X(NP),
        ZQ(IQ)=Z(NP)
      210 HQ(IQ)=H(NP)
C
      DO 215 KG=1,HQ
        AKYG(KG)=AKX(H,KG)
        AKXZG(KG)=AKYZ(H,KG)
      215 AKZG(KG)=AKZ(H,KG)
C
      IF(HQ.EQ.3) CALL Q3D(CQ,RQ,IQ,ZQ,AKYG,AKZG,AKXZG,HQ, IYZ)

```

```

VELT 005
VELT 010
VELT 015
VELT 020
VELT 025
VELT 030
VELT 035
VELT 040
VELT 045
VELT 050
VELT 055
VELT 060
VELT 065
VELT 070
VELT 075
VELT 080
VELT 085
VELT 090
VELT 095
VELT 100
VELT 105
VELT 110
VELT 115
VELT 120
VELT 125
VELT 130
VELT 135
VELT 140
VELT 145
VELT 150
VELT 155
VELT 160
VELT 165
VELT 170
VELT 175
VELT 180
VELT 185
VELT 190
VELT 195
VELT 200
VELT 205
VELT 210
VELT 215
VELT 220
VELT 225
VELT 230
VELT 235
VELT 240
VELT 245
VELT 250
VELT 255
VELT 260
VELT 265
VELT 270
VELT 275
VELT 280
VELT 285
VELT 290
VELT 295
VELT 300

```

APPENDIX C. (continued)

IF(NQ.EQ.4) CALL Q4D(QQ,IQ,IQ,ZQ,AKXG,AKZG,AKIZG,HQ, IIZ)	VELT 305
C	VELT 310
C ----- ASSEMBLE QQ(IQ,JQ) INTO THE GLOBAL MATRIX C(NP,IB) AND	VELT 315
C ----- FORM THE LOAD VECTOR VI(NP) OR VZ(NP).	VELT 320
C	VELT 325
DO 280 IQ=1,NQ	VELT 330
NI=IF(N,IQ)	VELT 335
DO 240 JQ=1,NQ	VELT 340
NJ=IF(N,JQ)	VELT 345
C	VELT 350
IF(IFN1S.EQ.1) GO TO 230	VELT 355
C	VELT 360
IF(NJ.LT.NI) GC TO 240	VELT 365
IB=NJ-NI+1	VELT 370
C(NI,IB)=C(NI,IB)+QQ(IQ,JQ)	VELT 375
GO TC 240	VELT 380
C	VELT 385
230 C(NI,1)=C(NI,1)+QQ(IQ,JQ)	VELT 390
240 CONTINUE	VELT 395
C	VELT 400
IF(IIZ.EQ.2) GC TO 260	VELT 405
VI(NI)=VI(NI)+BQ(IQ)	VELT 410
GO TC 280	VELT 415
260 VZ(NI)=VZ(NI)+BQ(IQ)	VELT 420
280 CONTINUE	VELT 425
C	VELT 430
290 CONTINUE	VELT 435
C	VELT 440
IF(IFN1S.EQ.1) GO TO 360	VELT 445
C	VELT 450
C ----- SOLVE THE MATRIX EQUATION: CX = B	VELT 455
C	VELT 460
IF(IIZ.EQ.2) GC TO 350	VELT 465
CALL BANSOL(1,C,VI,NMF,INBP,MAXNP,MAXHBP)	VELT 470
CALL BANSOL(2,C,VI,NMF,INBP,MAXNP,MAXHBP)	VELT 475
GO TC 390	VELT 480
C	VELT 485
350 CALL BANSOL(1,C,VZ,NMF,INBP,MAXNP,MAXHBP)	VELT 490
CALL BANSOL(2,C,VZ,NMF,INBP,MAXNP,MAXHBP)	VELT 495
GO TC 390	VELT 500
C	VELT 505
360 IF(IIZ.EQ.2) GC TO 370	VELT 510
DO 365 NP=1,NMF	VELT 515
365 VI(NP)=VI(NP)/C(NP,1)	VELT 520
GO TC 380	VELT 525
C	VELT 530
370 DO 375 NP=1,NMF	VELT 535
375 VZ(NP)=VZ(NP)/C(NP,1)	VELT 540
C	VELT 545
390 CONTINUE	VELT 550
C	VELT 555
RETURN	VELT 560
END	VELT 565

APPENDIX C. (continued)

```

      SUBROUTINE Q4D(QQ,RQ,IQ,ZQ,AKIG,AKZG,AKIZG,HQ,IND)
C
C FUNCTION OF SUBROUTINE-TC EVALUATE THE MATRIX QUADRATURE OVER THE
C AREA OF ONE ELEMENT. THESE INTEGRALS ARISE THROUGH THE
C APPLICATION OF THE GALERKIN INTEGRATION SCHEME
C
      IMPLICIT REAL*8 (A-H,C-Z)
      REAL*8 N(4)
C
      DIMENSION QQ(4,4),RQ(4),IQ(4),ZQ(4),HQ(4), AKIG(4),AKZG(4),
      > AKIZG(4)
      DIMENSION S(4),T(4),DNX(4),DNZ(4)
      DIMENSION PJAB(2,2),DNSS(4),DNIT(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 QQ(IQ,JQ) AND RQ(IQ)
C
      DO 110 IQ=1,4
      RQ(IQ)=0.0
      DO 110 JQ=1,4
      QQ(IQ,JQ)=0.0
110 CONTINUE
C
C ----- SUMMATION OF THE INTEGRAND OVER ALL GAUSSIAN POINTS
C
C
      DO 490 KG=1,4
C
C ----- DETERMINE LOCAL COORDINATE (SS,TT) OF GAUSSIAN POINT KG
C
      SS=P*S(KG)
      TT=P*T(KG)
C
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,DNIT,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)*DNIT(1)
      PJAB(1,2)=PJAB(1,2)-ZQ(I)*DNSS(1)
      PJAB(2,1)=PJAB(2,1)-IQ(I)*DNIT(1)
      PJAB(2,2)=PJAB(2,2)+IQ(I)*DNSS(1)
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

```

```

Q4D 005
Q4D 010
Q4D 015
Q4D 020
Q4D 025
Q4D 030
Q4D 035
Q4D 040
Q4D 045
Q4D 050
Q4D 055
Q4D 060
Q4D 065
Q4D 070
Q4D 075
Q4D 080
Q4D 085
Q4D 090
Q4D 095
Q4D 100
Q4D 105
Q4D 110
Q4D 115
Q4D 120
Q4D 125
Q4D 130
Q4D 135
Q4D 140
Q4D 145
Q4D 150
Q4D 155
Q4D 160
Q4D 165
Q4D 170
Q4D 175
Q4D 180
Q4D 185
Q4D 190
Q4D 195
Q4D 200
Q4D 205
Q4D 210
Q4D 215
Q4D 220
Q4D 225
Q4D 230
Q4D 235
Q4D 240
Q4D 245
Q4D 250
Q4D 255
Q4D 260
Q4D 265
Q4D 270
Q4D 275
Q4D 280
Q4D 285
Q4D 290
Q4D 295
Q4D 300

```

APPENDIX C. (continued)

C	DO 260 I=1,4	Q4D 305
	DNX(I)=DNSS(I)*PJAB(1,1)+DNIT(I)*PJAB(1,2)	Q4D 310
	DNZ(I)=DNSS(I)*PJAB(2,1)+DNIT(I)*PJAB(2,2)	Q4D 315
260	CONTINUE	Q4D 320
C		Q4D 325
	AKIK=AKIG(KG)	Q4D 330
	AKIK=AKIG(KG)	Q4D 335
	AKIZF=AKIZG(KG)	Q4D 340
C		Q4D 345
C	----- ACCUMULATE THE SUMS TO OBTAIN THE MATRIX INTEGRALS QQ(IQ,JQ)	Q4D 350
C	----- AND EQ(IQ) .	Q4D 355
C		Q4D 360
	DO 390 IQ=1,4	Q4D 365
	DO 390 JQ=1,4	Q4D 370
	QQ(IQ,JQ)=QQ(IQ,JQ)+N(IQ)*N(JQ)*DJAC	Q4D 375
	IF(IND.EQ.2) GC TO 350	Q4D 380
	RQ(IC)=RQ(IQ)-N(IQ)*RQ(JQ)*(AKIK*DNX(JQ)+AKIZK*DNZ(JQ))*DJAC	Q4D 385
	GO TO 390	Q4D 390
350	RQ(IC)=RQ(IQ)-N(IQ)*RQ(JQ)*(AKIZK*DNX(JQ)+AKIK*DNZ(JQ))*DJAC	Q4D 395
390	CONTINUE	Q4D 400
C		Q4D 405
	490 CONTINUE	Q4D 410
C		Q4D 415
	RETURN	Q4D 420
	END	Q4D 425
		Q4D 430

APPENDIX C. (continued)

	SUBROUTINE Q3D (QQ,BQ,IQ,ZQ,AKXG,AKZG,AKXZG,HQ, IND)	Q3D 005
C		Q3D 010
C	FUNCTION OF SUBROUTINE TO EVALUATE THE MATRIX OVER A TRIANGULAR	Q3D 015
C	ELEMENT.	Q3D 020
C		Q3D 025
	IMPLICIT REAL*8 (A-H,O-Z)	Q3D 030
C		Q3D 035
	DIMENSION QQ (4,4), BQ (4), IQ (4), ZQ (4), HQ (4), AKXG (4), AKZG (4),	Q3D 040
	> AKXZG (4)	Q3D 045
	DIMENSION A (3), B (3), C (3)	Q3D 050
	DATA OSIX/0.16666666666666667D0/	Q3D 055
	DATA OTWLV/0.8333333333333333D-1/	Q3D 060
C		Q3D 065
	DO 110 IQ=1,4	Q3D 070
	BQ (IQ)=0.0	Q3D 075
	DO 110 JQ=1,4	Q3D 080
	QQ (IQ,JQ)=0.0	Q3D 085
110	CONTINUE	Q3D 090
C		Q3D 095
	X1=XQ (1)	Q3D 100
	X2=XQ (2)	Q3D 105
	X3=XQ (3)	Q3D 110
	Z1=ZQ (1)	Q3D 115
	Z2=ZQ (2)	Q3D 120
	Z3=ZQ (3)	Q3D 125
C		Q3D 130
	A (1)=X2*Z3 - X3*Z2	Q3D 135
	A (2)=X3*Z1 - X1*Z3	Q3D 140
	A (3)=X1*Z2 - X2*Z1	Q3D 145
C		Q3D 150
	AREA2=ABS (A (1)+A (2)+A (3))	Q3D 155
	AREA=AREA2/2.0D0	Q3D 160
C		Q3D 165
	B (1)=(Z2-Z3)/AREA2	Q3D 170
	B (2)=(Z3-Z1)/AREA2	Q3D 175
	B (3)=(Z1-Z2)/AREA2	Q3D 180
	C (1)=(X3-X2)/AREA2	Q3D 185
	C (2)=(X1-X3)/AREA2	Q3D 190
	C (3)=(X2-X1)/AREA2	Q3D 195
C		Q3D 200
	AOSIX=AREA*OSIX	Q3D 205
	AOTWLV=AREA*OTWLV	Q3D 210
	QQ (1,1)=AOSIX	Q3D 215
	QQ (1,2)=AOTWLV	Q3D 220
	QQ (1,3)=AOTWLV	Q3D 225
	QQ (2,1)=AOTWLV	Q3D 230
	QQ (2,2)=AOSIX	Q3D 235
	QQ (2,3)=AOTWLV	Q3D 240
	QQ (3,1)=AOTWLV	Q3D 245
	QQ (3,2)=AOTWLV	Q3D 250
	QQ (3,3)=AOSIX	Q3D 255
C		Q3D 260
	DHDX=0.0	Q3D 265
	DHDZ=0.0	Q3D 270
	DO 200 IQ=1,3	Q3D 275
	DHDX=DHDX+HQ (IQ)*B (IQ)	Q3D 280
200	DHDZ=DHDZ+HQ (IQ)*C (IQ)	Q3D 285
C		Q3D 290
	IF (IND.EQ.2) GO TO 400	Q3D 295
C		Q3D 300

APPENDIX C. (continued)

	RQ(1) = -AREA*DHDX*(AKZG(1)*OSIX+(AKZG(2)+AKZG(3))*OTWLVE) -	Q3D 305
	1 AREA*DHDZ*(AKZG(1)*OSIX+(AKZG(2)+AKZG(3))*OTWLVE)	Q3D 310
	RQ(2) = -AREA*DHDY*((AKZG(1)+AKZG(3))*OTWLVE+AKZG(2)*OSIX) -	Q3D 315
	1 AREA*DHDZ*((AKZG(1)+AKZG(3))*OTWLVE+AKZG(2)*OSIX)	Q3D 320
	RQ(3) = -AREA*DHDY*((AKZG(1)+AKZG(2))*OTWLVE+AKZG(3)*OSIX) -	Q3D 325
	1 AREA*DHDZ*((AKZG(1)+AKZG(2))*OTWLVE+AKZG(3)*OSIX)	Q3D 330
C	RETURN	Q3D 335
		Q3D 340
		Q3D 345
C	*00 RQ(1) = -AREA*DHDZ*(AKZG(1)*OSIX+(AKZG(2)+AKZG(3))*OTWLVE) -	Q3D 350
	1 AREA*DHDY*(AKZG(1)*OSIX+(AKZG(2)+AKZG(3))*OTWLVE)	Q3D 355
	RQ(2) = -AREA*DHDZ*(AKZG(1)+AKZG(3))*OTWLVE+AKZG(2)*OSIX) -	Q3D 360
	1 AREA*DHDY*((AKZG(1)+AKZG(3))*OTWLVE+AKZG(2)*OSIX)	Q3D 365
	RQ(3) = -AREA*DHDZ*(AKZG(1)+AKZG(2))*OTWLVE+AKZG(3)*OSIX) -	Q3D 370
	1 AREA*DHDY*((AKZG(1)+AKZG(2))*OTWLVE+AKZG(3)*OSIX)	Q3D 375
C	RETURN	Q3D 380
	END	Q3D 385
		Q3D 390

APPENDIX C. (continued)

```

      SUBROUTINE SFROP(IZ,H, B,      AKX,AKZ,AKXZ, PROP, HAXEL,HAXNP,
      > HAXHAT,NEL)
C
C
C FUNCTION OF SUBROUTINE--TO CALCULATE SOIL PROPERTIES, I.E. THE
C THE TRANSIMIBILITY FOR UNCONFINED PROBLEM
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 H(4)
C
      DIMENSION IE(HAXEL,5),H(HAXNP),B(HAXNP),
      > AKX(HAXEL,4),AKZ(HAXEL,4),AKXZ(HAXEL,4)
C
      DIMENSION PROP(HAXHAT,7)
C
      DIMENSION HQ(4),BQ(4), S(4),T(4), BKG(4),BKG(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 ----- COMPUTE AKX, AKZ, AND AKXZ
C
      DO 900 H=1,NEL
      NTYP=IE(H,5)
      IF(NTYP.GT.0) GO TO 900
      NTYP=IABS(NTYP)
C
C ----- EVALUATE HEAD AND BED ELEVATION AT GAUSSIAN POINTS FOR THE
C ----- ELEMENT
C
      HQ=4
      IEN4=IE(H,4)
      IF(IEN4.EQ.0) HQ=3
      DO 110 IQ=1,HQ
      NP=IE(H,IQ)
      BQ(IQ)=E(NP)
110 HQ(IQ)=H(NP)
C
      IF(HQ.EQ.3) GO TO 200
C
      DO 150 KG=1,HQ
      SS=P*S(KG)
      TT=P*T(KG)
      SH=1.0D0-SS
      SP=1.0D0+SS
      TH=1.0D0-TT
      TP=1.0D0+TT
C
      N(1)=0.25D0*SH*TH
      N(2)=0.25D0*SP*TH
      N(3)=0.25D0*SP*TP
      N(4)=0.25D0*SH*TP
C
      BKG(KG)=0.0
      HKG(KG)=0.0
      DO 120 IQ=1,HQ
      BKG(KG)=BKG(KG)+BQ(IQ)*N(IQ)
120 HKG(KG)=HKG(KG)+HQ(IQ)*N(IQ)
150 CONTINUE

```

```

SFRO 005
SFRO 010
SFRO 015
SFRO 020
SFRO 025
SFRO 030
SFRO 035
SFRO 040
SFRO 045
SFRO 050
SFRO 055
SFRO 060
SFRO 065
SFRO 070
SFRO 075
SFRO 080
SFRO 085
SFRO 090
SFRO 095
SFRO 100
SFRO 105
SFRO 110
SFRO 115
SFRO 120
SFRO 125
SFRO 130
SFRO 135
SFRO 140
SFRO 145
SFRO 150
SFRO 155
SFRO 160
SFRO 165
SFRO 170
SFRO 175
SFRO 180
SFRO 185
SFRO 190
SFRO 195
SFRO 200
SFRO 205
SFRO 210
SFRO 215
SFRO 220
SFRO 225
SFRO 230
SFRO 235
SFRO 240
SFRO 245
SFRO 250
SFRO 255
SFRO 260
SFRO 265
SFRO 270
SFRO 275
SFRO 280
SFRO 285
SFRO 290
SFRO 295
SFRO 300

```

APPENDIX C. (continued)

C	GO TO 400	SERO 305
C		SERO 310
	200 DO 220 IQ=1,NQ	SERO 315
	BKG(IQ)=BQ(IQ)	SERO 320
	220 BKG(IQ)=BQ(IQ)	SERO 325
C		SERO 330
	400 CONTINUE	SERO 335
	SATK1=EROP(HTYP,4)	SERO 340
	SATK2=EROP(HTYP,5)	SERO 345
	SATK3=EROP(HTYP,6)	SERO 350
C		SERO 355
	DO 500 KG=1,NQ	SERO 360
	USKPCT=BKG(KG)-BKG(KG)	SERO 365
	AKX(H,KG)=SATK1*USKPCT	SERO 370
	AKZ(H,KG)=SATK2*USKPCT	SERO 375
	AKIZ(H,KG)=SATK3*USKPCT	SERO 380
C		SERO 385
	500 CONTINUE	SERO 390
C		SERO 395
	900 CONTINUE	SERO 400
	RETURN	SERO 405
	END	SERO 410
		SERO 415

APPENDIX C. (continued)

```

SUBROUTINE ASEHBL(X,Z,IE, C,R, HP, HW,B, AKX,AKZ,AKXZ,
1 PROF, ISTYP,SOUS, HAXMP,HAXEL,HAXHBP,HAXHAT, KSS,W,DELT,
2 HU,RKU,IHUTYP,HL,BKL,IHLTYP,HXHUPR,HXHLPR, WSS,WPW,IWTP,LLMODE)
C
C
C FUNCTION OF SUBROUTINE--TO ASSEMBLE THE TOTAL COEFFICIENT MATRIX
C C(WP,IB) AND LOAD VECTOR R(WP) FROM THE ELEMENT MATRICES QA(IQ,JQ),
C CB(IQ,JQ), AND RQ(IQ).
C
C
C IMPLICIT REAL*8(A-H,O-Z)
C
COMMON /GEOM/ NWP,NEL,IBAND,IHBP,IHALPR, NBEL,NBN
COMMON /CS/ HXSPR,HXSDP,WSPL,WSDP,HXWMP,HXWPR,HXWDP,NWMP,NWPR,NWDP
COMMON /OPT/ ILUMP,ILID,ICCNPI,IPNTS
C
DIMENSION I(HAXMP),Z(HAXMP),IE(HAXEL,5)
DIMENSION C(HAXMP,HAXHBP),R(HAXMP), HP(HAXMP)
DIMENSION LLMODE(5,HAXMP)
DIMENSION HW(HAXMP),B(HAXMP)
DIMENSION AKX(HAXEL,4),AKZ(HAXEL,4),AKXZ(HAXEL,4)
DIMENSION PROF(HAXHAT,7)
DIMENSION ISTYP(HAXEL),SOUS(HXSPR)
DIMENSION WSS(HXWPR), WPW(HXWMP),IWTP(HXWMP)
DIMENSION HU(HXHUPR),RKU(HXHUPR),IHUTYP(HAXEL)
DIMENSION HL(HXHLPR),BKL(HXHLPR),IHLTYP(HAXEL)
C
DIMENSION QA(4,4),QB(4,4),BQ(4), AKXG(4),AKZG(4),AKXZG(4)
DIMENSION IQ(4),ZQ(4),IEH(4), HNBQ(4)
C
DELT1=1./DELT
W1=W
W2=1.-W
IF (KSS.GT.0) GO TO 100
DELT1=0.
W1=1.
W2=0.
C
C ----- INITIATE MATRICES C(WP,IB) AND R(WP)
C
100 IF(IPNTS.EQ.1) GO TO 140
C
DO 110 NP=1,NWP
R(NP)=0.0
DO 110 IP=1,IHBP
C(NP,IE)=0.0
110 CONTINUE
GO TO 190
C
140 DO 145 NP=1,NWP
R(NP)=0.0
DO 145 I=1,5
C(NP,I)=0.0
145 CONTINUE
C
C ----- START TO ASSEMBLE OVER ALL ELEMENTS
C
190 CONTINUE
DO 690 N=1,NEL
NQ=4

```

```

ASEH 005
ASEH 010
ASEH 015
ASEH 020
ASEH 025
ASEH 030
ASEH 035
ASEH 040
ASEH 045
ASEH 050
ASEH 055
ASEH 060
ASEH 065
ASEH 070
ASEH 075
ASEH 080
ASEH 085
ASEH 090
ASEH 095
ASEH 100
ASEH 105
ASEH 110
ASEH 115
ASEH 120
ASEH 125
ASEH 130
ASEH 135
ASEH 140
ASEH 145
ASEH 150
ASEH 155
ASEH 160
ASEH 165
ASEH 170
ASEH 175
ASEH 180
ASEH 185
ASEH 190
ASEH 195
ASEH 200
ASEH 205
ASEH 210
ASEH 215
ASEH 220
ASEH 225
ASEH 230
ASEH 235
ASEH 240
ASEH 245
ASEH 250
ASEH 255
ASEH 260
ASEH 265
ASEH 270
ASEH 275
ASEH 280
ASEH 285
ASEH 290
ASEH 295
ASEH 300

```

APPENDIX C. (continued)

```

      IEN4=IE(N,4)
      IF(IEN4.EQ.0) NQ=3
C ----- COMPUTE MATRICES QA(IQ,JQ), QB(IQ,JQ), AND RQ(IQ) FOR EACH N
C
      HCONFI=IE(N,5)
      HTYP=IAPS(HCONFI)
      ALP=PROP(HTYP,1)
      BETA=PROP(HTYP,2)
      PCA=PROP(HTYP,3)
      SY=PROP(HTYP,7)
C
      I=ISTYP(N)
      WSOSH=SOUS(I)
C
      I=IHOTYP(N)
      HUM=HU(I)
      RKUM=EKU(I)
      I=IHITYP(N)
      HLM=HL(I)
      RKLH=RKL(I)
C
      DO 210 IQ=1,NQ
      NP=IE(N,IQ)
      IEN(IQ)=NP
      IQ(IQ)=I(NP)
      ZQ(IQ)=Z(NP)
      HMBQ(IQ)=HW(NP)-B(NP)
      IF(HCONFI.GT.0) HMBQ(IQ)=B(NP)
210 CONTINUE
C
      DO 220 KG=1,NQ
      AKIG(KG)=AKI(N,KG)
      AKZG(KG)=AKZ(N,KG)
      AKZG(KG)=AKZ(N,KG)
220 CONTINUE
C
      IF(NQ.EQ.3) CALL Q3(QA,QB,RQ,AKIG,AKZG,AKZG,IQ,ZQ,HMBQ,
1 ALP,BETAP,PCR,SY, WSOSH,HUM,RKUM,HLM,RKLH)
      IF(NQ.EQ.4) CALL Q4(QA,QB,RQ,AKIG,AKZG,AKZG,IQ,ZQ,HMBQ,
1 ALP,BETAP,PCR,SY, WSOSH,HUM,RKUM,HLM,RKLH)
C ----- ASSEMBLE QA(IQ,JQ) AND QB(IQ,JQ) INTO THE GLOBAL MAT'IX
C ----- C(NP,IB) = B + A/DELT AND FORM THE GLOBAL LOAD VECTOR R(NP).
C ----- SINCE C IS SYMMETRIC, ONLY THE UPPER HALF BAND IS STORED.
C
      IF(INID.EQ.1) GO TO 400
C
C ----- FOR THE CASES OF NON MID-DIFFERENCES
C
      DO 390 IQ=1,NQ
      NI=IEN(IQ)
      R(NI)=R(NI)+RQ(IQ)
      DO 340 JQ=1,NQ
      NJ=IEN(JQ)
      QA(IQ,JQ)=QA(IQ,JQ)*DELT1
      R(NI)=R(NI)+(QA(IQ,JQ)-B2*QB(IQ,JQ))*NP(NJ)
C
      IF(IENIS.EQ.1) GO TO 320
C

```

```

      ASEN 305
      ASEN 310
      ASEN 315
      ASEN 320
      ASEN 325
      ASEN 330
      ASEN 335
      ASEN 340
      ASEN 345
      ASEN 350
      ASEN 355
      ASEN 360
      ASEN 365
      ASEN 370
      ASEN 375
      ASEN 380
      ASEN 385
      ASEN 390
      ASEN 395
      ASEN 400
      ASEN 405
      ASEN 410
      ASEN 415
      ASEN 420
      ASEN 425
      ASEN 430
      ASEN 435
      ASEN 440
      ASEN 445
      ASEN 450
      ASEN 455
      ASEN 460
      ASEN 465
      ASEN 470
      ASEN 475
      ASEN 480
      ASEN 485
      ASEN 490
      ASEN 495
      ASEN 500
      ASEN 505
      ASEN 510
      ASEN 515
      ASEN 520
      ASEN 525
      ASEN 530
      ASEN 535
      ASEN 540
      ASEN 545
      ASEN 550
      ASEN 555
      ASEN 560
      ASEN 565
      ASEN 570
      ASEN 575
      ASEN 580
      ASEN 585
      ASEN 590
      ASEN 595
      ASEN 600

```

APPENDIX C. (continued)

IF(NJ.II.NI) GO TO 340	ASEN 605
IB=NJ-NI+1	ASEN 610
C(NI,IB)=C(NI,IB)+QA(IQ,JQ)+W1*QB(IQ,JQ)	ASEN 615
GO TC 340	ASEN 620
C	ASEN 625
320 IF(NJ.IE.NI) GO TO 330	ASEN 630
R(NI)=R(NI)-(QA(IQ,JQ)+W1*CB(IQ,JQ))*HW(NJ)	ASEN 635
GO TC 340	ASEN 640
C	ASEN 645
330 DO 335 I=1,5	ASEN 650
LNODE=LLNODE(I,NI)	ASEN 655
IF(NJ.EQ.LNODE) GO TO 337	ASEN 660
335 CONTINUE	ASEN 665
C	ASEN 670
WRITE(6,1000) NI,N,JQ	ASEN 675
STOP	ASEN 680
C	ASEN 685
337 C(NI,I)=C(NI,I)+QA(IQ,JQ)+W1*QB(IQ,JQ)	ASEN 690
C	ASEN 695
340 CONTINUE	ASEN 700
C	ASEN 705
350 CONTINUE	ASEN 710
GO TC 690	ASEN 715
C	ASEN 720
C ----- FOR THE CASES OF MID-DIFFERENCES	ASEN 725
C	ASEN 730
400 CONTINUE	ASEN 735
DO 490 IQ=1,NQ	ASEN 740
NI=IPN(IQ)	ASEN 745
R(NI)=R(NI)+RQ(IQ)	ASEN 750
DO 440 JQ=1,NQ	ASEN 755
NJ=IPN(JQ)	ASEN 760
QA(IQ,JQ)=2.0D0*QA(IQ,JQ)*DELTI	ASEN 765
R(NI)=R(NI)+CA(IQ,JQ)*HF(NJ)	ASEN 770
C	ASEN 775
IF(IPN15.EQ.1) GO TO 420	ASEN 780
C	ASEN 785
IF(NJ.II.NI) GO TO 440	ASEN 790
IB=NJ-NI+1	ASEN 795
C(NI,IB)=C(NI,IB)+QA(IQ,JQ)+QB(IQ,JQ)	ASEN 800
GO TC 440	ASEN 805
C	ASEN 810
420 IF(NJ.IE.NI) GO TO 430	ASEN 815
R(NI)=R(NI)-(QA(IQ,JQ)+QB(IQ,JQ))*HW(NJ)	ASEN 820
GO TC 440	ASEN 825
C	ASEN 830
430 DO 435 I=1,5	ASEN 835
LNODE=LLNODE(I,NI)	ASEN 840
IF(NJ.EQ.LNODE) GO TO 437	ASEN 845
435 CONTINUE	ASEN 850
C	ASEN 855
WRITE(6,1000) NI,N,JQ	ASEN 860
STOP	ASEN 865
C	ASEN 870
437 C(NI,I)=C(NI,I)+QA(IQ,JQ)+QB(IQ,JQ)	ASEN 875
C	ASEN 880
440 CONTINUE	ASEN 885
490 CONTINUE	ASEN 890
C	ASEN 895
690 CONTINUE	ASEN 900

APPENDIX C. (continued)

C		ASEH 905
C	----- INCORPORATE WELL SOURCE/SINK	ASEH 910
C		ASEH 915
	IF (NWNF.EQ.0) GO TO 990	ASEH 920
	DO 790 I=1,NWNF	ASEH 925
	NI=NEN(I)	ASEH 930
	ITYP=INTYP(I)	ASEH 935
	R(NI)=R(NI)+WSS(ITYP)	ASEH 940
	790 CONTINUE	ASEH 945
C		ASEH 950
	1000 FORMAT(1H1/5X,'*** WARNING: NONE OF THE LOWER-LEFT NODE IN EQUATIO	ASEH 955
	1H',I3,/5X,'*** IS CORRESPONDING TO ',I5,'-TH ELEMENT-S',I2,	ASEH 960
	2'-TH NODE; STOP *****)	ASEH 965
C		ASEH 970
	990 RETURN	ASEH 975
	END	ASEH 980

	SUBROUTINE Q4(QA,QB,RQ,AKIG,AKZG,AKIZG,XQ,ZQ,HBBQ,	005
	> ALP,BETAP,POR,SY,WSOSH,HUN,RKUN,HLN,RELB)	010
C		015
C		020
C	FUNCTION OF SUBROUTINE--TO EVALUATE THE MATRIX QUADRATURES OVER THE	025
C	AREA OF ONE ELEMENT OF WATER CONTENT AND COMPRESSIBILITY QA(IQ,JQ)	030
C	AND OF CONDUCTIVITY QB(IQ,JQ) AND RQ(IQ), THE LATTER ARISING FROM THE	035
C	GRAVITY TERM IN THE MOISTURE-FLOW EQUATION. THESE INTEGRALS ARISE	040
C	THROUGH APPLICATION OF THE GALERKIN INTEGRATION SCHEME.	045
C		050
C		055
	IMPLICIT REAL*8 (A-H,O-Z)	060
	REAL*8 N(4)	065
C		070
	COMMON /OPT/ ILUMP,INID,ICCNFI,IPNTS	075
C		080
	DIMENSION QA(4,4),QB(4,4),RQ(4),AKIG(4),AKZG(4),AKIZG(4),	085
	> XQ(4),ZQ(4),HBBQ(4)	090
	DIMENSION S(4),T(4),DNX(4),DNZ(4)	095
	DIMENSION PJAB(2,2),DNSS(4),DNIT(4)	100
C		105
	DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-	110
	> 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /	115
C		120
C		125
C	----- INITIATE MATRICES QA, QB, AND RQ	130
	DO 110 IQ=1,4	135
	RQ(IQ)=0.0	140
	DO 110 JQ=1,4	145
	QA(IQ,JQ)=0.0	150
	QB(IQ,JQ)=0.0	155
	110 CONTINUE	160
C		165
C		170
C	----- SUMMATION OF THE INTEGRAND OVER ALL GAUSSIAN POINTS	175
C		180
	DO 490 KG=1,4	185
C		190
C	----- DETERMINE LOCAL COORDINATE (SS,TT) OF GAUSSIAN POINT KG	195
C		200
	SS=P*S(KG)	205
	TT=P*T(KG)	210
C		215
C	----- CALCULATE VALUES OF BASIS FUNCTIONS N(IQ) AND THEIR	220
C	----- DERIVATIVES DNX(IQ) AND DNZ(IQ) W.R.T. X AND Z, RESPECTIVELY	225
C	----- AT THE GAUSSIAN POINT KG.	230
C		235
	CALL BASE(N,DNSS,DNIT,SS,TT)	240
C		245
	DO 210 I=1,2	250
	DO 210 J=1,2	255
	210 PJAB(I,J)=0.0	260
C		265
	DO 220 I=1,4	270
	PJAB(1,1)=PJAB(1,1)+ZQ(I)*DNIT(I)	275
	PJAB(1,2)=PJAB(1,2)-ZQ(I)*DNSS(I)	280
	PJAB(2,1)=PJAB(2,1)-XQ(I)*DNIT(I)	285
	PJAB(2,2)=PJAB(2,2)+XQ(I)*DNSS(I)	290
	220 CONTINUE	295
C		300

APPENDIX C. (continued)

DJAC=PJAB(2,2)*PJAB(1,1)-PJAB(1,2)*PJAB(2,1)	QA	305
DJACI=1.0D0/DJAC	QA	310
C	QA	315
DO 230 I=1,2	QA	320
DO 230 J=1,2	QA	325
PJAB(I,J)=PJAB(I,J)*DJACI	QA	330
230 CONTINUE	QA	335
C	QA	340
DO 260 I=1,4	QA	345
DNX(I)=DNSS(I)*PJAB(1,1)+DNIT(I)*PJAB(1,2)	QA	350
DNZ(I)=DNSS(I)*PJAB(2,1)+DNIT(I)*PJAB(2,2)	QA	355
260 CONTINUE	QA	360
C	QA	365
C ----- SET HYDRAULIC CONDUCTIVITIES EQUAL TO THOSE AT THE GAUSSIAN	QA	370
C ----- POINTS.	QA	375
AKXQP=AKXG(KG)	QA	380
AKXZQP=AKXZG(KG)	QA	385
AKZQP=AKZG(KG)	QA	390
C	QA	395
C ----- INTERPOLATE TO OBTAIN (H-B) AT THE GAUSSIAN POINT.	QA	400
C	QA	405
HMBQF=0.0	QA	410
DO 280 IQ=1,4	QA	415
280 HMBQF=HMBQF+HMBG(IQ)*N(IQ)	QA	420
C	QA	425
C	QA	430
C	QA	435
C ----- ACCUMULATE THE SUMS TO OBTAIN THE MATRIX INTEGRALS QA, QB,	QA	440
C ----- AND RQ	QA	445
C	QA	450
FHP=(ALP+POR*BETAP)*HMBQF+SY	QA	455
FHP=FHP+DJAC	QA	460
AKXQP=AKXQP+DJAC	QA	465
AKXZQP=AKXZQP+DJAC	QA	470
AKZQP=AKZQP+DJAC	QA	475
WSOSK=WSOSH+DJAC	QA	480
RKUNK=RKUN+DJAC	QA	485
RKLHF=RKLH+DJAC	QA	490
C	QA	495
DO 390 IQ=1,4	QA	500
RQ(IQ)=RQ(IQ)+(WSOSK+RKUNK*HUN+RKLHK*RLH)*N(IQ)	QA	505
DO 390 JQ=1,4	QA	510
QA(IQ,JQ)=QA(IQ,JQ)+FHP*N(IQ)*N(JQ)	QA	515
QB(IQ,JQ)=QB(IQ,JQ)+DNX(IQ)*(AKXQP*DNX(JQ)+AKXZQP*DNZ(JQ))+	QA	520
1 DNZ(IQ)*(AKXZQP*DNX(JQ)+AKZQP*DNZ(JQ))+ (RKUNK+RKLHK)*N(IQ)*N(JQ)	QA	525
390 CONTINUE	QA	530
C	QA	535
490 CONTINUE	QA	540
C	QA	545
IF(ILUMP.NE.0) GO TO 600	QA	550
C	QA	555
C ----- IF MASS MATRIX IS NOT TO BE LUMPED, RETURN	QA	560
C	QA	565
RETURN	QA	570
C	QA	575
C ----- FOR THE CASES WHEN THE MASS MATRIX IS TO BE LUMPED	QA	580
C	QA	585
600 CONTINUE	QA	590
DO 690 I=1,4	QA	595
SUM=0.0	QA	600

APPENDIX C. (continued)

```
      DO 640 J=1,4  
        SUM=SUM+QA(I,J)  
640   QA(I,J)=0.0  
      QA(I,I)=SUM  
690   CONTINUE  
C  
      RETURN  
      END
```

```
QA  605  
QA  610  
QA  615  
QA  620  
QA  625  
QA  630  
QA  635  
QA  640
```

APPENDIX C. (continued)

	SUBROUTINE Q3(QA,QB,RQ,AKXC,AKZG,AKXZG,XQ,ZQ,HNBQ,	Q3	005
	> ALP,BETAP,POR,SY,WSOSH,HON,REUN,FLH,BKLA)	Q3	010
C		Q3	015
C	FUNCTION OF SUBROUTINE-TO EVALUATE MATRIX OVER A TRIANGULAR ELEMENT	Q3	020
C		Q3	025
	IMPLICIT REAL*8(A-H,O-Z)	Q3	030
C		Q3	035
	COMMON /OPT/ ILUMP,INID,ICOEPI,IPNTS	Q3	040
C		Q3	045
	DIMENSION QA(4,4),QB(4,4),RQ(4),AKXC(4),AKZG(4),AKXZG(4),	Q3	050
	> XQ(4),ZQ(4),HNBQ(4)	Q3	055
	DIMENSION A(3),B(3),C(3)	Q3	060
C		Q3	065
	DATA OSIX/0.1666666666666667D0/	Q3	070
	DATA OTULVE/0.8333333333333333D-1/	Q3	075
	DATA OTHRTY/0.3333333333333333D-1/	Q3	080
	DATA USIXTY/0.1666666666666667D-1/	Q3	085
C		Q3	090
	DO 110 IQ=1,4	Q3	095
	RQ(IQ)=0.0	Q3	100
	DO 110 JQ=1,4	Q3	105
	QA(IQ,JQ)=0.0	Q3	110
	QB(IQ,JQ)=0.0	Q3	115
110	CONTINUE	Q3	120
C		Q3	125
	X1=XQ(1)	Q3	130
	X2=XQ(2)	Q3	135
	X3=XQ(3)	Q3	140
	Z1=ZQ(1)	Q3	145
	Z2=ZQ(2)	Q3	150
	Z3=ZQ(3)	Q3	155
C		Q3	160
	A(1)=X2*Z3 - X3*Z2	Q3	165
	A(2)=X3*Z1 - X1*Z3	Q3	170
	A(3)=X1*Z2 - X2*Z1	Q3	175
C		Q3	180
	AREA2=0ABS(A(1)+A(2)+A(3))	Q3	185
	AREA=AREA2/2.0D0	Q3	190
C		Q3	195
	B(1)=(Z2-Z3)/AREA2	Q3	200
	B(2)=(Z3-Z1)/AREA2	Q3	205
	B(3)=(Z1-Z2)/AREA2	Q3	210
	C(1)=(X3-X2)/AREA2	Q3	215
	C(2)=(X1-X3)/AREA2	Q3	220
	C(3)=(X2-X1)/AREA2	Q3	225
C		Q3	230
	F1=(ALP+POR*BETAP)*HNBQ(1) + SY	Q3	235
	F2=(ALP+POR*BETAP)*HNBQ(2) + SY	Q3	240
	F3=(ALP+POR*BETAP)*HNBQ(3) + SY	Q3	245
C		Q3	250
	QA(1,1)=AREA*(F1*0.1D0 + F2*OTHRTY + F3*OTHRTY)	Q3	255
	QA(1,2)=AREA*(F1*OTHRTY + F2*OTHRTY + F3*OSIXTY)	Q3	260
	QA(1,3)=AREA*(F1*OTHRTY + F2*OSIXTY + F3*OTHRTY)	Q3	265
	QA(2,1)=AREA*(F1*OTHRTY + F2*OTHRTY + F3*OSIXTY)	Q3	270
	QA(2,2)=AREA*(F1*OTHRTY + F2*0.1D0 + F3*OTHRTY)	Q3	275
	QA(2,3)=AREA*(F1*OSIXTY + F2*OTHRTY + F3*OTHRTY)	Q3	280
	QA(3,1)=AREA*(F1*OTHRTY + F2*OSIXTY + F3*OTHRTY)	Q3	285
	QA(3,2)=AREA*(F1*OSIXTY + F2*OTHRTY + F3*OTHRTY)	Q3	290
	QA(3,3)=AREA*(F1*OTHRTY + F2*OTHRTY + F3*0.1D0)	Q3	295
C		Q3	300

APPENDIX C. (continued)

SUBROUTINE BC(C,R,X,Z,RPLX,INTYP,NPN,IS,GRD,INTYP,NPD,	BC	005
1 HAXNF,HAXHBP)	BC	010
C	BC	015
C	BC	020
C FUNCTION OF SUBROUTINE--TO APPLY BOTH CONSTANT AND TIME-VARYING	BC	025
C (RAINFALL-SEEPAGE) FLUX-TYPE NEWMAN AND PRESSURE-TYPE DIRICHLET	BC	030
C BOUNDARY CONDITIONS.	BC	035
C	BC	040
C	BC	045
IMPLICIT REAL*8 (A-H,O-Z)	BC	050
C	BC	055
COMMON /GEOM/ NNP,NEL,IBAND,IBBP,IBALPB, NSEL,NBN	BC	060
COMMON /CH/ HXNPP,HXNEL,HXNP3,HXNDP,HXNPP,HXEL,HXNP5,HXNDP	BC	065
COMMON /CD/ HXNDP,HXDEF,HXDDP,HXNP,HXDP,HXNDP	BC	070
COMMON /OPT/ ILNPP,ILND,ICONFI,IPWIS	BC	075
C	BC	080
DIMENSION C(HAXNF,HAXHBP),R(HAXNP)	BC	085
DIMENSION X(HAXNP),Z(HAXNP)	BC	090
DIMENSION RPLX(HXNPP),INTYP(HXNPP),NPN(HXNPP), IS(HXNEL,2)	BC	095
DIMENSION GRD(HXNDP),INTYP(HXNDP),NPD(HXNDP)	BC	100
C	BC	105
C ***** APPLY DIRICHLET BOUNDARY CONDITIONS	BC	110
C	BC	115
IF (NPN.EQ.0) GO TO 400	BC	120
C	BC	125
IF (IPWIS.EQ.1) GO TO 200	BC	130
DO 180 NPP=1,NPN	BC	135
C	BC	140
C ----- MODIFY LOAD VECTOR FOR NON-ZERO DIRICHLET BOUNDARY VALUES	BC	145
C	BC	150
NI=NED(NPP)	BC	155
INTYP=INTYP(NPP)	BC	160
BB=GRD(INTYP)	BC	165
IF (BB.EQ.0.0) GO TO 140	BC	170
DO 110 IB=1,IBALPB	BC	175
NJ=NI-IB	BC	180
IF (NJ.LT.1) GO TO 120	BC	185
JB=IE+1	BC	190
R(NJ)=R(NJ) - BB*C(NJ,JB)	BC	195
110 CONTINUE	BC	200
120 DO 130 IB=1,IBALPB	BC	205
NJ=NI+IB	BC	210
IF (NJ.GT.NPN) GO TO 140	BC	215
JB=IE+1	BC	220
R(NJ)=R(NJ)+BB*C(NJ,JB)	BC	225
130 CONTINUE	BC	230
C	BC	235
140 R(NI)=R2	BC	240
C	BC	245
C ----- ZERO COLUMN NPD(NPP)	BC	250
C	BC	255
DO 150 IB=1,IBALPB	BC	260
NJ=NI-IB	BC	265
IF (NJ.LT.1) GO TO 160	BC	270
JB=IE+1	BC	275
C(NJ,JB)=0.0	BC	280
150 CONTINUE	BC	285
C	BC	290
C ----- MODIFY ROW NPD(NPP)	BC	295
C	BC	300

APPENDIX C. (continued)

160 DO 170 KB=1,INBP	BC 305
C(NI,KB)=0.0	BC 310
170 CONTINUE	BC 315
C(NI,1)=1.000	BC 320
180 CONTINUE	BC 325
GO TO 400	BC 330
C	BC 335
200 DO 210 NBP=1,NDNP	BC 340
NI=NED(NBP)	BC 345
ITYP=IDITYP(NBP)	BC 350
NB=GNED(ITYP)	BC 355
R(NI)=RE	BC 360
C(NI,5)=1.000	BC 365
DO 210 I=1,4	BC 370
C(NI,I)=0.0	BC 375
210 CONTINUE	BC 380
C	BC 385
C ***** APPLY NEUMANN OR CAUCHY BOUNDARY CONDITIONS	BC 390
C	BC 395
400 IF (NDNP.EQ.0) GO TO 500	BC 400
DO 490 BP=1,NDNP	BC 405
I=IS(BP,1)	BC 410
NI=NEP(I)	BC 415
J=IS(BP,2)	BC 420
NJ=NEP(J)	BC 425
DX=X(NI)-X(NJ)	BC 430
DZ=Z(NI)-Z(NJ)	BC 435
EL=DSQRT(DX*DX+DZ*DZ)	BC 440
ITYPI=INTYP(I)	BC 445
ITYPJ=INTYP(J)	BC 450
R(NI)=R(NI)-EL*(RFLX(ITYPI)/3.000+RFLX(ITYPJ)/6.000)	BC 455
R(NJ)=R(NJ)-EL*(RFLX(ITYPI)/6.000+RFLX(ITYPJ)/3.000)	BC 460
490 CONTINUE	BC 465
C	BC 470
500 RETURN	BC 475
END	BC 480

APPENDIX C. (continued)

AREA=AREA/3.000	Q3	305
AKIQS=AKIG(1)+AKIG(2)+AKIG(3)	Q3	310
AKZQS=AKZG(1)+AKZG(2)+AKZG(3)	Q3	315
AKXZQS=AKXZG(1)+AKXZG(2)+AKXZG(3)	Q3	320
C	Q3	325
DO 210 IQ=1,3	Q3	330
DO 210 JQ=1,3	Q3	335
QB(IQ,JQ)=AREA*(3(IQ)*B(JQ)*AKIQS+C(IQ)*C(JQ)*AKZQS +	Q3	340
1 (B(IQ)*C(JQ)+B(JQ)*C(IQ))*AKXZQS)	Q3	345
210 CONTINUE	Q3	350
C	Q3	355
AKULCS=AREA*(RKUH+KCLB)*OSIX	Q3	360
AKULCT=AREA*(RKUH+KCLM)*OTULVE	Q3	365
DO 220 IQ=1,3	Q3	370
DO 220 JQ=1,3	Q3	375
IF(IQ.EQ.JQ) QB(IQ,JQ)=QB(IQ,JQ)+AKULCS	Q3	380
IF(IQ.NE.JQ) QB(IQ,JQ)=QB(IQ,JQ)+AKULOT	Q3	385
220 CONTINUE	Q3	390
C	Q3	395
RQIQ=AREA*(RUCSH+RKUH+RHH+RKLH+RLH)	Q3	400
DO 310 IQ=1,3	Q3	405
310 RQ(IQ)=RQIQ	Q3	410
C	Q3	415
IF(ILUMP.EQ.0) RETURN	Q3	420
C	Q3	425
C ----- FOR THE CASES WHEN THE MASS MATRIX IS TO BE LUMPED	Q3	430
C	Q3	435
DO 690 I=1,3	Q3	440
SUH=0.0	Q3	445
DO 640 J=1,3	Q3	450
SUH=SUH+QA(I,J)	Q3	455
640 QA(I,J)=0.0	Q3	460
QA(I,I)=SUH	Q3	465
690 CONTINUE	Q3	470
C	Q3	475
RETURN	Q3	480
END	Q3	485

APPENDIX C. (continued)

	SUBROUTINE BASE(N,DWSS,DWTT,SS,TT)	BASE 005
C		BASE 010
C	FUNCTION OF THE SUBROUTINE-TO COMPUTE THE VALUES OF BASIS FUNCTIONS	BASE 015
C		BASE 020
	IMPLICIT REAL*8 (A-H,O-Z)	BASE 025
	REAL*8 N(4)	BASE 030
C		BASE 035
	DIMENSION DWSS(4),DWTT(4)	BASE 040
C		BASE 045
	SH=1.000-SS	BASE 050
	SP=1.000+SS	BASE 055
	TH=1.000-TT	BASE 060
	TP=1.000+TT	BASE 065
	N(1)=0.2500*SH*TH	BASE 070
	N(2)=0.2500*SP*TH	BASE 075
	N(3)=0.2500*SP*TP	BASE 080
	N(4)=0.2500*SH*TP	BASE 085
	DWSS(1)=-0.2500*TH	BASE 090
	DWSS(2)=0.2500*TH	BASE 095
	DWSS(3)=0.2500*TP	BASE 100
	DWSS(4)=-0.2500*TP	BASE 105
	DWTT(1)=-0.2500*SH	BASE 110
	DWTT(2)=-0.2500*SP	BASE 115
	DWTT(3)=0.2500*SP	BASE 120
	DWTT(4)=0.2500*SH	BASE 125
	RETURN	BASE 130
	END	BASE 135

APPENDIX C. (continued)

```

SUBROUTINE BANSOL(KKK,C,N,NBP,INBP,MAXNP,MAXNBP)
C
C
C FUNCTION OF SUBROUTINE--TO SOLVE THE MATRIX EQUATION  $CX = R$ ,
C RETURNING THE SOLUTION X IN R. IT IS ASSUMED THAT THE ARRAY
C C(NP,IB) CONTAINS ONLY THE UPPER HALF BAND OF A SYMMETRIC MATRIX.
C
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      DIMENSION C(MAXNP,MAXNBP),R(MAXNP)
C
C      INHALF=INBP-1
C      NNP1=NNP-1
C
C ***** IF KKK=1, TRIANGULARIZE THE BAND MATRIX C(NP,IB).
C ***** IF KKK=2, SIMPLY SOLVE WITH THE NEW RIGHT-HAND SIDE R(NP)
C
C      IF (KKK.EQ.2) GO TO 500
C
C ----- TRIANGULARIZE MATRIX C
C
C      NU=N1      ALPB
C      DO 120 I=1,NU
C      NJ=NI-1
C      PIVOTI=1.0D0/C(NI,I)
C      DO 120 LB=2,INBP
C      A=C(NI,IB)*PIVOTI
C      NK=NJ+LB
C      JB=0
C      DO 110 KB=LB,INBP
C      JB=JB+1
C      C(NK,JB)=C(NK,JB)-A*C(NI,KB)
C 110 C(NK,JB)=C(NK,JB)-A*C(NI,KB)
C 120 C(NI,IB)=A
C
C      NI=NU+1
C      DO 140 NI=NI,NNP1
C      NJ=NI-1
C      NB=NNP-NJ
C      PIVOTI=1.0D0/C(NI,I)
C      DO 140 LB=2,NB
C      A=C(NI,LB)*PIVOTI
C      NK=NJ+LB
C      JB=0
C      DO 130 KB=LP,NB
C      JB=JB+1
C      C(NK,JB)=C(NK,JB)-A*C(NI,KB)
C 130 C(NK,JB)=C(NK,JB)-A*C(NI,KB)
C 140 C(NI,LB)=A
C      RETURN
C
C ----- MODIFY LOAD VECTOR R
C
C 500 NU=NNP-INHALF
C      DO 610 NI=1,NU
C      NJ=NI-1
C      A=R(NI)
C      R(NI)=A/C(NI,1)
C      DO 610 LB=2,INBP
C      NK=NJ+LB
C      R(NK)=R(NK)-C(NI,LB)*A
C 610 R(NK)=R(NK)-C(NI,LB)*A

```

BANS 005
 BANS 010
 BANS 015
 BANS 020
 BANS 025
 BANS 030
 BANS 035
 BANS 040
 BANS 045
 BANS 050
 BANS 055
 BANS 060
 BANS 065
 BANS 070
 BANS 075
 BANS 080
 BANS 085
 BANS 090
 BANS 095
 BANS 100
 BANS 105
 BANS 110
 BANS 115
 BANS 120
 BANS 125
 BANS 130
 BANS 135
 BANS 140
 BANS 145
 BANS 150
 BANS 155
 BANS 160
 BANS 165
 BANS 170
 BANS 175
 BANS 180
 BANS 185
 BANS 190
 BANS 195
 BANS 200
 BANS 205
 BANS 210
 BANS 215
 BANS 220
 BANS 225
 BANS 230
 BANS 235
 BANS 240
 BANS 245
 BANS 250
 BANS 255
 BANS 260
 BANS 265
 BANS 270
 BANS 275
 BANS 280
 BANS 285
 BANS 290
 BANS 295
 BANS 300

APPENDIX C. (continued)

C	HL=HU+1	BANS 305
	DO 620 HI=HL,HUP1	BANS 310
	HJ=HI-1	BANS 315
	HB=HUP-HJ	BANS 320
	A=R(HI)	BANS 325
	R(HI)=A/C(HI,1)	BANS 330
	DO 620 LB=2,HB	BANS 335
	HK=HJ+LB	BANS 340
	620 R(HK)=R(HK)-C(HI,LB)*A	BANS 345
C		BANS 350
C	----- BACK-SOLVE	BANS 355
C		BANS 360
	S(HUP)=R(HUP)/C(HUP,1)	BANS 365
	DO 630 IE=1,INALPE	BANS 370
	HI=HUP-IE	BANS 375
	HJ=HI-1	BANS 380
	HB=IE+1	BANS 385
	DO 630 KB=2,HB	BANS 390
	HK=HJ+KB	BANS 395
	630 R(HI)=R(HI)-C(HI,KB)*S(HK)	BANS 400
C		BANS 405
	DO 640 IB=INBP,HUP1	BANS 410
	HI=HUP-IB	BANS 415
	HJ=HI-1	BANS 420
	DO 640 KB=2,INBP	BANS 425
	HK=HJ+KB	BANS 430
	640 R(HI)=R(HI)-C(HI,KB)*S(HK)	BANS 435
C		BANS 440
	RETURN	BANS 445
	END	BANS 455

APPENDIX C. (continued)

```

SUBROUTINE SPLOW(X,I,IE,VI,VZ,ELB,DCOSXB,DCOSXB,BFLI,BFLIP,ISB,
1 NBP,NFB,NFB,NPD,PRATE,FLOW,TFLOW,MAXBP,MAXEL,MAXDEL,MAXBSP,
2 KFLOW,DELT,H,HP,PROP,MAXHAT,B,SQUS,ISTYP,NPW,YSS,IUTYP,
3 NU,BKU,INUTYP,NL,NKL,INLTYP,NXNUP,NXNLP)
C
C
C FUNCTION OF SUBROUTINE--TO COMPUTE BOUNDARY FLUXES, FLOW RATES,
C INCREMENTAL FLOWS OCCURRING DURING TIME DELT, TOTAL FLOWS SINCE
C TIME ZERO, AND THE CHANGE IN MOISTURE CONTENT FOR THE ENTIRE
C SYSTEM DURING TIME DELT.
C
C
C IMPLICIT REAL*8(A-H,O-Z)
C
COMMON /GEOM/ NBP,NEL,INAD,INBP,INALFE, NDEL,NBN
COMMON /CN/ NXNUP,NXNLP,NXNUP,NXNLP,NXNUP,NXNLP,NXNLP,NXNLP
COMMON /CD/ NXNUP,NXNLP,NXNLP,NXNLP,NXNLP,NXNLP,NXNLP,NXNLP
COMMON /CS/ NXNUP,NXNLP,NXNLP,NXNLP,NXNLP,NXNLP,NXNLP,NXNLP
COMMON /CPT/ ILNUP,INID,ICCNFI,IPNTS
C
C
C DIMENSION I(MAXBP),Z(MAXBP),IE(MAXEL,5)
C DIMENSION VI(MAXBP),VZ(MAXBP),H(MAXBP),HP(MAXBP),B(MAXBP)
C DIMENSION DLE(MAXDEL),DCOSXB(MAXDEL),DCOSXB(MAXDEL),BFLI(MAXBSP),
C > BFLIP(MAXBSP),NFB(MAXDEL),ISB(MAXDEL,4),NPD(MAXBSP)
C DIMENSION SQUS(NXNUP),ISTYP(MAXEL)
C DIMENSION NSS(NXNUP),NPN(NXNUP),IUTYP(NXNUP)
C DIMENSION NPB(NXNUP),NPD(NXNUP)
C DIMENSION NU(NXNUP),PKU(NXNUP),INUTYP(MAXEL)
C DIMENSION NL(NXNLP),NKL(NXNLP),INLTYP(MAXEL)
C DIMENSION PRCP(MAXHAT,7)
C DIMENSION PRATE(10),FLOW(10),TFLOW(10)
C DIMENSION IQ(4),ZQ(4),DBQ(4),HBPQ(4),H01HQ(4),HLHQ(4)
C
C
C DATA NULEAK,NLLEAK,SOURCE/0.000,0.000,0.000/
C
C
C ----- CALCULATE NODAL FLOW RATES
C
C
C DO 110 NP=1,NBN
C   BFLIF(NP)=BFLI(NP)
C 110 BFLIP(NP)=0.0
C
C ***** CALCULATE VOLUMETRIC FLOW RATE
C
C
C DO 170 NP=1,NBNI
C   N=NBP(NP)
C   NI=ISB(NP,1)
C   NJ=ISB(NP,2)
C   DO 120 I=1,NBN
C     IJ=NFB(I)
C     IF(IJ.NE.NI) GO TO 120
C     NII=I
C     GO TO 130
C 120 CONTINUE
C 130 DO 140 J=1,NPN
C   IJ=NFB(J)
C   IF(IJ.NE.NJ) GO TO 140
C   NJJ=J
C   GO TO 150
C 140 CONTINUE

```

SPLO 005
 SPLO 010
 SPLO 015
 SPLO 020
 SPLO 025
 SPLO 030
 SPLO 035
 SPLO 040
 SPLO 045
 SPLO 050
 SPLO 055
 SPLO 060
 SPLO 065
 SPLO 070
 SPLO 075
 SPLO 080
 SPLO 085
 SPLO 090
 SPLO 095
 SPLO 100
 SPLO 105
 SPLO 110
 SPLO 115
 SPLO 120
 SPLO 125
 SPLO 130
 SPLO 135
 SPLO 140
 SPLO 145
 SPLO 150
 SPLO 155
 SPLO 160
 SPLO 165
 SPLO 170
 SPLO 175
 SPLO 180
 SPLO 185
 SPLO 190
 SPLO 195
 SPLO 200
 SPLO 205
 SPLO 210
 SPLO 215
 SPLO 220
 SPLO 225
 SPLO 230
 SPLO 235
 SPLO 240
 SPLO 245
 SPLO 250
 SPLO 255
 SPLO 260
 SPLO 265
 SPLO 270
 SPLO 275
 SPLO 280
 SPLO 285
 SPLO 290
 SPLO 295
 SPLO 300

APPENDIX C. (continued)

```

150 CONTINUE
C ----- COMPUTE THE CONTRIBUTION OF SIDE (NI,NJ) TO THE TOTAL FLUX
C ----- THROUGH POINTS NI AND NJ, RESPECTIVELY.
C
      PNI=(VI(NI)*DCOSXB(NP)+VZ(NI)*DCOSZB(NP))*DLB(NP)
      PNJ=(VI(NJ)*DCOSXB(NP)+VZ(NJ)*DCOSZB(NP))*DLB(NP)
C ----- DISTRIBUTE THE ABOVE FLOUES TO TOW END POINTS OF THE SIDE
C
      BFLI(NII)=BFLI(NII)+PNI/3.000+PNJ/6.000
      BFLI(NJJ)=BFLI(NJJ)+PNJ/3.000+PNI/6.000
170 CONTINUE
C
      IF (KPLCB.GT.0) GO TO 200
      DO 180 NP=1,NBN
180 BFLIF(NP)=BFLI(NP)
C
      DO 190 I=1,8
190 TPLOM(I)=0.0
C ***** DETERMINE TOTAL FLOWS AND TOTAL FLOW RATES THROUGH VARIOUS
C ***** TYPES OF BOUNDARIES, STARTING WITH THE NET FLOWS THROUGH THE
C ***** ENTIRE BOUNDARY.
200 SUM=C.
      SUMP=0.
      DO 210 NP=1,NBN
      SUM=SUM+BFLI(NP)
210 SUMP=SUMP+BFLIF(NP)
C
      PRATE(6)=SUM
      FLOW(6)=0.500*(SUM+SUMP)*DELT
C ***** THE DIRICHLET BOUNDARY
C
      PRATE(1)=0.
      FLOW(1)=0.
      IF (NINF.LE.0) GO TO 400
      SUM=0.
      SUMP=0.
      DO 330 NPP=1,NBNP
      NP=NED(NPP)
      DO 310 I=1,NBN
      IJ=NED(I)
      IF (IJ.NE.NP) GO TO 310
      NII=I
      GO TO 320
310 CONTINUE
320 CONTINUE
      SUM=SUM+BFLI(NII)
      SUMP=SUMP+BFLIF(NII)
330 CONTINUE
      PRATE(1)=SUM
      FLOW(1)=0.500*(SUM+SUMP)*DELT
C ***** THE NEUMANN OR THE CAUCHY BOUNDARY
C
400 PRATE(2)=0.0

```

```

SPLO 305
SPLO 310
SPLO 315
SPLO 320
SPLO 325
SPLO 330
SPLO 335
SPLO 340
SPLO 345
SPLO 350
SPLO 355
SPLO 360
SPLO 365
SPLO 370
SPLO 375
SPLO 380
SPLO 385
SPLO 390
SPLO 395
SPLO 400
SPLO 405
SPLO 410
SPLO 415
SPLO 420
SPLO 425
SPLO 430
SPLO 435
SPLO 440
SPLO 445
SPLO 450
SPLO 455
SPLO 460
SPLO 465
SPLO 470
SPLO 475
SPLO 480
SPLO 485
SPLO 490
SPLO 495
SPLO 500
SPLO 505
SPLO 510
SPLO 515
SPLO 520
SPLO 525
SPLO 530
SPLO 535
SPLO 540
SPLO 545
SPLO 550
SPLO 555
SPLO 560
SPLO 565
SPLO 570
SPLO 575
SPLO 580
SPLO 585
SPLO 590
SPLO 600

```

APPENDIX C. (continued)

FLOW(2)=0.0	SPLO 605
IF (NNNP.LE.0) GO TO 600	SPLO 610
SUN=0.	SPLO 615
SUNP=0.	SPLO 620
DO 530 NPP=1,NNNP	SPLO 625
NP=NNP(NPP)	SPLO 630
DO 510 I=1,NNN	SPLO 635
IX=NNP(I)	SPLO 640
IF (IX.NE.NP) GO TO 510	SPLO 645
NI1=I	SPLO 650
GO TO 520	SPLO 655
510 CONTINUE	SPLO 660
520 CONTINUE	SPLO 665
SUN=SUN+BFLX(NI1)	SPLO 670
SUNP=SUNP+BFLXF(NI1)	SPLO 675
530 CONTINUE	SPLO 680
FRATE(2)=SUN	SPLO 685
FLOW(2)=0.500*(SUN+SUNP)*DELT	SPLO 690
C	SPLO 695
600 FRATE(3)=0.	SPLO 700
FLOW(3)=0.	SPLO 705
FRATE(4)=0.	SPLO 710
FLOW(4)=0.	SPLO 715
C	SPLO 720
C ***** THE UNSPECIFIED BOUNDARY, I. E. BOUNDARY WITH ZERO TOTAL FLUX	SPLO 725
C	SPLO 730
SUN=0.	SPLO 735
SUNP=0.	SPLO 740
DO 710 I=1,4	SPLO 745
SUN=SUN+FRATE(I)	SPLO 750
710 SUNP=SUNP+FLOW(I)	SPLO 755
FRATE(5)=FRATE(6)-SUN	SPLO 760
FLOW(5)=FLOW(6)-SUNP	SPLO 765
C	SPLO 770
C ***** CALCULATE THE INCREASE IN THE WATER CONTENT AND THE SOURCE	SPLO 775
C	SPLO 780
HJLEAK=HJLEAK	SPLO 785
HLLEAK=HLLEAK	SPLO 790
SOURCE=SOURCE	SPLO 795
QTH=0.	SPLO 800
HJLEAK=0.0	SPLO 805
HLLEAK=0.0	SPLO 810
SOURCE=0.0	SPLO 815
DO 830 N=1,NEL	SPLO 820
NQ=4	SPLO 825
IEH4=IE(N,4)	SPLO 830
IF (IEH4.EQ.0) LC=3	SPLO 835
C	SPLO 840
NCONFI=IE(N,5)	SPLO 845
NTYP=IAES(NCONFI)	SPLO 850
C	SPLO 855
ALP=PROP(NTYP,1)	SPLO 860
BETAP=PROP(NTYP,2)	SPLO 865
POF=PROP(NTYP,3)	SPLO 870
SY=PROP(NTYP,7)	SPLO 875
C	SPLO 880
IXY=ISTYP(N)	SPLO 885
SCUSLP=SCUS(ITYP)	SPLO 890
C	SPLO 895
ITYP=INOTYP(N)	SPLO 900

APPENDIX C. (continued)

	HUE=HU(ITYP)	SFLO 905
	RKUE=RKU(ITYP)	SFLO 910
	ITYF=IHITYF(H)	SFLO 915
	HIE=HL(ITYP)	SFLO 920
	RKLE=RKL(ITYP)	SFLO 925
C		SFLO 930
	DO 810 IQ=1,NQ	SFLO 935
	NP=IE(N,IQ)	SFLO 940
	IQ(IQ)=I(NP)	SFLO 945
	ZQ(IQ)=Z(NP)	SFLO 950
	DBQ(IQ)=H(NP)-HP(NP)	SFLO 955
	HNBQ(IQ)=H(NP)-E(NP)	SFLO 960
	IF(RCONF1.GT.0) HNBQ(IQ)=B(NP)	SFLO 965
	HNBQ(IQ)=RKUE*(HUE-H(NP))	SFLO 970
	HLNBQ(IQ)=RKLE*(HLE-H(NP))	SFLO 975
	810 CONTINUE	SFLO 980
C		SFLO 985
C		SFLO 990
	IF(NQ.EQ.3) CALL Q3TH(QTH,UPLEKH,DNLEKH,DBQ,IQ,ZQ,HNBQ,	SFLO 995
	1 HNBQ,HLNBQ,ALP,BETAP,POR,SY,SOUSH,SOUSQP,KFLOW)	SFLO1000
	IF(NQ.EQ.4) CALL Q4TH(QTH,UPLEKH,DNLEKH,DBQ,IQ,ZQ,HNBQ,	SFLO1005
	1 HNBQ,HLNBQ,ALP,BETAP,POR,SY,SOUSH,SOUSQP,KFLOW)	SFLO1010
C		SFLO1015
	QTH=QTH+QTHH	SFLO1020
	HULEAK=HULEAK+UPLEKH	SFLO1025
	HLLEAK=HLLEAK+DNLEKH	SFLO1030
	SOURCE=SOURCE+SOUSH	SFLO1035
	830 CONTINUE	SFLO1040
C		SFLO1045
	IF(MWNP.EQ.0) GO TO 850	SFLO1050
	DO 840 I=1,MWNP	SFLO1055
	ITYF=JWITYF(I)	SFLO1060
	840 SOURCE=SOURCE-MSS(ITYF)	SFLO1065
C		SFLO1070
	850 FRATE(3)=HULEAK	SFLO1075
	FRATE(4)=HLLEAK	SFLO1080
	FRATE(6)=SOURCE	SFLO1085
C		SFLO1090
	IF(KFLOW.GT.0) GO TO 880	SFLO1095
	HULEKP=HULEAK	SFLO1100
	HLLEKP=HLLEAK	SFLO1105
	SORCIP=SOURCE	SFLO1110
C		SFLO1115
	880 FLOW(3)=0.5D0*(HULEAK+HULEKP)*DELT	SFLO1120
	FLOW(4)=0.5D0*(HLLEAK+HLLEKP)*DELT	SFLO1125
	FLOW(8)=0.5D0*(SOURCE+SORCIP)*DELT	SFLO1130
C		SFLO1135
	FLOW(7)=QTH	SFLO1140
	FRATE(7)=FLOW(7)/DELT	SFLO1145
	IF(KFLOW.LE.0) FRATE(7)=-(FRATE(6)+FRATE(8)+FRATE(3)+FRATE(4))	SFLO1150
C		SFLO1155
	DO 910 I=1,8	SFLO1160
	910 TFLOW(I)=TFLOW(I)+FLOW(I)	SFLO1165
C		SFLO1170
	RETURN	SFLO1175
	END	SFLO1180

APPENDIX C. (continued)

```

SUBROUTINE Q4TH(QTHH,UPLKX,DNLEKH,DHQ,IQ,ZQ,HMBQ,HUMHQ,HLNHQ,
1      AIP,BETAF,FOR,SY, SOUSH,SOUSQP, KKFLOW)
C
C  FUNCTION OF SUBROUTINE--TO EVALUATE THE WATER-CONTENT INTEGRAL
C  OVER THE AREA OF ONE ELEMENT.
C
      IMPLICIT REAL*8 (A-H,C-Z)
      REAL*8 N(4)
C
      DIMENSION S(4),T(4),IQ(4),ZQ(4),DHQ(4), HMBQ(4),HUMHQ(4),HLNHQ(4)
      DIMENSION PJAB(2,2),DNSS(4),DNIT(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
      QTHH=0.
      UPLKX=0.0
      DNLEKH=0.0
      SOUSH=0.0
C
C  ----- SUMMATION OF THE INTEGRAND OVER ALL GAUSSIAN POINTS
C
      DO 490 KG=1,4
C
C  ----- DETERMINE LOCAL 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) AT THE GAUSSIAN
C  ----- POINT KG.
C
      CALL BASE(N,DNSS,DNIT,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)*DNIT(I)
      PJAB(1,2)=PJAB(1,2)-ZQ(I)*DNSS(I)
      PJAB(2,1)=PJAB(2,1)-XQ(I)*DNIT(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)
C
C  ----- INTERPOLATE TO OBTAIN VALUES OF INTEGRAND AT THE GAUSSIAN
C  ----- POINTS KG.
C
      DHQP=0.0
      HMBQP=0.0
      HUMHQP=0.0
      HLNHQP=0.0
      DO 280 IQ=1,4
      DHQP=DHQP+DHQ(IQ)*N(IQ)
      HMBQP=HMBQP+HMBQ(IQ)*N(IQ)
      HUMHQP=HUMHQP+HUMHQ(IQ)*N(IQ)
      HLNHQP=HLNHQP+HLNHQ(IQ)*N(IQ)
280 CONTINUE

```

Q4TH 005
Q4TH 010
Q4TH 015
Q4TH 020
Q4TH 025
Q4TH 030
Q4TH 035
Q4TH 040
Q4TH 045
Q4TH 050
Q4TH 055
Q4TH 060
Q4TH 065
Q4TH 070
Q4TH 075
Q4TH 080
Q4TH 085
Q4TH 090
Q4TH 095
Q4TH 100
Q4TH 105
Q4TH 110
Q4TH 115
Q4TH 120
Q4TH 125
Q4TH 130
Q4TH 135
Q4TH 140
Q4TH 145
Q4TH 150
Q4TH 155
Q4TH 160
Q4TH 165
Q4TH 170
Q4TH 175
Q4TH 180
Q4TH 185
Q4TH 190
Q4TH 195
Q4TH 200
Q4TH 205
Q4TH 210
Q4TH 215
Q4TH 220
Q4TH 225
Q4TH 230
Q4TH 235
Q4TH 240
Q4TH 245
Q4TH 250
Q4TH 255
Q4TH 260
Q4TH 265
Q4TH 270
Q4TH 275
Q4TH 280
Q4TH 285
Q4TH 290
Q4TH 295
Q4TH 300

APPENDIX C. (continued)

C	THQP=DBQP*((ALP+POR*BETAP)*HNBQP+SY)	Q4TH 305
	IF(KBPLOW.LE.0) THQP=FOR*HNBQP	Q4TH 310
C		Q4TH 315
C	----- ACCUMULATE THE SUM TO EVALUATE THE INTEGRAL.	Q4TH 320
C		Q4TH 325
	QTHH=QTHH+THQP*DJAC	Q4TH 330
	UPLKHH=UPLKHH-HUNHQP*DJAC	Q4TH 335
	DNLEHH=DNLEHH-RLNHQP*DJAC	Q4TH 340
	SOUSH=SOUSH-SOUSQP*DJAC	Q4TH 345
490	CONTINUE	Q4TH 350
C		Q4TH 355
	RETURN	Q4TH 360
	END	Q4TH 365
		Q4TH 370

APPENDIX C. (continued)

```

SUBROUTINE Q3TH(QTHH,UPLEKH,DNLEKH,DHQ,IQ,ZQ,HNBQ,HNBHQ,HLNHQ,
1      ALP,BETAF,POR,SY, SOUSH,SOUSQP, KKFLOW)
C
C FUNCTION OF SUBROUTINE--TO EVALUATE THE WATER-CONTENT INTEGRAL
C OVER THE AREA OF ONE ELEMENT.
C
      IMPLICIT REAL*8 (A-H,C-Z)
      DIMENSION IQ(4),ZQ(4),DHQ(4),HNBQ(4),HNBHQ(4),HLNHQ(4)
      DIMENSION A(3),STRGE(3)
C
      DATA CSIX/0.1666666666666667D0/
      DATA OTWLVE/0.8333333333333333D-1/
C
      X1=XQ(1)
      X2=XQ(2)
      X3=XQ(3)
      Z1=ZQ(1)
      Z2=ZQ(2)
      Z3=ZQ(3)
      A(1)=X2*Z3 - X3*Z2
      A(2)=X3*Z1 - X1*Z3
      A(3)=X1*Z2 - X2*Z1
C
      AREA2=DABS(A(1)+A(2)+A(3))
      AREA=AREA2/2.0E0
C
      AREAL=AREA/3.0D0
      UPLEKH=0.0
      DNLEKH=0.0
      DO 310 IQ=1,3
      UPLEKH=UPLEKH-AREAL*HNBHQ(IQ)
      DNLEKH=DNLEKH-AREAL*HLNHQ(IQ)
310 CONTINUE
C
      DO 320 IQ=1,3
      STRGE(IQ)=(ALP+POR*BETAF)*HNBQ(IQ)+SY
      IF(KKFLOW.LE.0) STRGE(IQ)=POR*HNBQ(IQ)
320 CONTINUE
      QTHH=STRGE(1)*(DHQ(1)*CSIX+(DHQ(2)+DHQ(3))*OTWLVE)*AREA +
1      STRGE(2)*(DHQ(2)*CSIX+(DHQ(3)+DHQ(1))*OTWLVE)*AREA +
2      STRGE(3)*(DHQ(3)*CSIX+(DHQ(1)+DHQ(2))*OTWLVE)*AREA
C
      SOUSH=-SOUSQP*AREA
C
      RETURN
      END

```

```

Q3TH 005
Q3TH 010
Q3TH 015
Q3TH 020
Q3TH 025
Q3TH 030
Q3TH 035
Q3TH 040
Q3TH 045
Q3TH 050
Q3TH 055
Q3TH 060
Q3TH 065
Q3TH 070
Q3TH 075
Q3TH 080
Q3TH 085
Q3TH 090
Q3TH 095
Q3TH 100
Q3TH 105
Q3TH 110
Q3TH 115
Q3TH 120
Q3TH 125
Q3TH 130
Q3TH 135
Q3TH 140
Q3TH 145
Q3TH 150
Q3TH 155
Q3TH 160
Q3TH 165
Q3TH 170
Q3TH 175
Q3TH 180
Q3TH 185
Q3TH 190
Q3TH 195
Q3TH 200
Q3TH 205
Q3TH 210
Q3TH 215
Q3TH 220
Q3TH 225
Q3TH 230
Q3TH 235

```

APPENDIX C. (continued)

```

SUBROUTINE PRINT(VI,VZ,H,NPB,BPLX, PRATE, FLOW, TFLOW, MAXNP, MAXEL,
> MAXBNP, NNP, NEL, TIME, DELT, S, RD, IBAND, KPB, KOUT, KDIAG, ITIM)
C
C
C FUNCTION OF SUBROUTINE--TO OUTPUT FLOWS, PRESSURE HEADS, TOTAL
C HEADS, WATER CONTENTS, AND DARCY VELOCITIES AS SPECIFIED BY
C PARAMETER KPB.
C
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      REAL*4 SUBHD
C
C      DIMENSION VI(MAXNP), VZ(MAXNP), H(MAXNP)
C      DIMENSION NPB(MAXBNP), BPLX(MAXBNP)
C      DIMENSION PRATE(10), FLOW(10), TFLOW(10)
C      DIMENSION SUBHD(8)
C
C      IF (KPB.EQ.0) RETURN
C
C ----- PRINT SYSTEM FLOW INFORMATION
C
100 KDIAG=KDIAG+1
   KDIAG=KDIAG-1
   PRINT 1000, KDIAG, TIME, DELT, ITIM, (PRATE(I), FLOW(I), TFLOW(I), I=1,8)
130 IF (KPB.EQ.1) RETURN
C
C ----- PRINT PRESSURE HEAD
C
   KOUT=KOUT+1
   KLINE=-1
   DO 290 NI=1,NNP,8
     NJNN=NI
     NJMX=NINO(NI+7,NNP)
     KLINE=KLINE+1
     IF (MOD(KLINE,50).EQ.0) PRINT 2000, KOUT, TIME, DELT, IBAND, ITIM,
> (SUBHD(I), I=1,8)
     PRINT 2100, NI, (H(NJ), NJ=NJNN, NJMX)
290 CONTINUE
   IF (KPB.EQ.2) RETURN
C
C ----- PRINT VERTICALLY INTEGRATED VOLUMETRIC FLUX
C
   KOUT=KOUT+1
   KLINE=-1
   DO 390 NI=1,NNP,4
     NJNN=NI
     NJMX=NINO(NI+3,NNP)
     KLINE=KLINE+1
     IF (MOD(KLINE,50).EQ.0) PRINT 3000, KOUT, TIME, DELT, IBAND, ITIM,
> (SUBHD(I), I=1,8)
     PRINT 3100, (NJ, VI(NJ), VZ(NJ), NJ=NJNN, NJMX)
390 CONTINUE
C
   RETURN
C
1000 FORMAT(1H1,32H TABLE OF SYSTEM-FLOW PARAMETERS,2X,7HTABLE: ,I4,
> 12H.. AT TIME =,1PD11.4,9H , (DELT = 1PD12.4,1H),6H ITIM=,I4//5X,

```

```

PRIN 005
PRIN 010
PRIN 015
PRIN 020
PRIN 025
PRIN 030
PRIN 035
PRIN 040
PRIN 045
PRIN 050
PRIN 055
PRIN 060
PRIN 065
PRIN 070
PRIN 075
PRIN 080
PRIN 085
PRIN 090
PRIN 095
PRIN 100
PRIN 105
PRIN 110
PRIN 115
PRIN 120
PRIN 125
PRIN 130
PRIN 135
PRIN 140
PRIN 145
PRIN 150
PRIN 155
PRIN 160
PRIN 165
PRIN 170
PRIN 175
PRIN 180
PRIN 185
PRIN 190
PRIN 195
PRIN 200
PRIN 205
PRIN 210
PRIN 215
PRIN 220
PRIN 225
PRIN 230
PRIN 235
PRIN 240
PRIN 245
PRIN 250
PRIN 255
PRIN 260
PRIN 265
PRIN 270
PRIN 275
PRIN 280
PRIN 285
PRIN 290
PRIN 295
PRIN 300

```

APPENDIX C. (continued)

```

> 13H TYPE OF FLOW,35X,12HRATE(L**3/T),6Y,15HINC. PLGW(L**3),7Y,      PRIN 305
> 16HTOTAL PLCH(L**3)/5X,                                              PRIN 310
1 40H 1. DIRICHLET NODE FLOW . . . . . ,3(E12.4,11X)/5X,          PRIN 315
2 40H 2. FLUX NODE PLGW . . . . . ,3(E12.4,11X)/5X              PRIN 320
3 40H 3. UPPER AQUIFER LEAKING FLOW . . . . ,3(E12.4,11X)/5X    PRIN 325
4 40H 4. LOWER AQUIFER LEAKING FLOW . . . . ,3(E12.4,11X)/5X    PRIN 330
5 40H 5. FLOW THROUGH UNSPECIFIED MODES . . . ,3(E12.4,11X)/5X   PRIN 335
6 40H 6. NET FLOW . . . . . ,3(E12.4,11X)/5X                    PRIN 340
7 40H 7. INCREASE IN WATER VOLUME . . . . . ,3(E12.4,11X)/5X    PRIN 345
8 40H 8. RECHARGING (-) OR PUMPING (+) ... ,3(E12.4,11X)/5X     PRIN 350
9 50H *** NOTE: (+) = OUT FROM, (-) = INTO THE REGION. //        PRIN 355
2000 FORMAT(13H OUTPUT TABLE,I4,31H.. HYDRAULIC HEAD (L) AT TIME =, PRIN 360
> 1PD12.4,9H ,(DELTA=,1PD12.4,15H),(BAND WIDTH=,I4,1H),6H IT =, PRIN 365
> I5//I1,8A4//I1,7H NODE I,5X,36HYDRAULIC HEAD OF NODE I,I+1,...,I+ PRIN 370
> 7 /) PRIN 375
2100 FORMAT(I7,8(1PD15.4)) PRIN 380
3000 FORMAT(13H OUTPUT TABLE,I4,37H.. INTEGRATED FLUX (L**2/T) AT TIME PRIN 385
>, ,E12.4,9H ,(DELTA=,1PD12.4,15H),(BAND WIDTH=,I4,1H),6H IT =, PRIN 390
> I5//I1, ,A4//I1,4HNODE,9X,2HVZ,9X,2HVZ,4X,4HNODE,9X,2HVZ,9X,2HVZ? PRIN 395
> 4 *RHC * 9X,2HVZ,9X,2HVZ,4X,4HNODE,9X,2HVZ,9X,2HVZ/ PRIN 400
> 27(***** PRIN 405
> 3X,27H***** PRIN 410
>*) PRIN 415
3100 FORMAT('H ,I5,2D11.3,3X,I5,2D11.3,3X,I5,2D11.3,3X,I5,2D11.3) PRIN 420
C PRIN 425
END PRIN 430

```

APPENDIX C. (continued)

```

SUBROUTINE STORE(X,Z,IE,H,B,VI,VZ,DLB,DCOSXB,DCOSZB,WBE,ISB,WPB,
1 TITLE,TIME,HAXCP,HAXEL,HAXBWP,HAXBEL,PROB,WNP,WEL,WBN,WBZL,WTI,
2 NSTAT, N",RKU,INHUTP,HL,RKL,INLTYP,HXHUPR,HXHLPR,WHUPR,WHLPR)
C
C
C FUNCTION OF SUBROUTINE--TO STORE PERTINENT QUANTITIES ON AUXILIARY
C DEVICE FOR FUTURE USE BY EITHER PLOTTING OR MATERIAL-TRANSPORT
C CODES. WHAT DEVICE IS TO BE USED MUST BE SPECIFIED BY APPROPRIATE
C JOB-CONTROL CARDS.
C
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION TITLE(9)
C      DIMENSION X(HAXWP),Z(HAXWP),IE(HAXEL,5)
C      DIMENSION H(HAXWP),VI(HAXWP),VZ(HAXWP),B(HAXWP)
C      DIMENSION DLE(HAXBEL),DCOSXB(HAXBEL),DCOSZB(HAXBEL),WBE(HAXBEL),
C      > ISB(HAXBEL,4),WPB(HAXWP)
C      DIMENSION HU(HXHUPR),RKU(HXHUPR),INHUTP(HAXEL)
C      DIMENSION HL(HXHLPR),RKL(HXHLPR),INLTYP(HAXEL)
C
C
C      DATA WFEROB/-1/
C
C      IF (NSTAT.GT.0) GO TO 10
C      IF (WFEROB.EQ.(-1)) RETURN 1
C      IF (WFEROB.EQ.WPROB) GO TO 10
C      WRITE(1) (TITLE(I),I=1,9),WPROB,WNP,WEL,WBN,WBZL,WTI,WHUPR,WHLPR
C      WRITE(1) (X(H),H=1,WHF),(Z(H),H=1,WHF),(B(H),H=1,WHF),
C      1 ((IE(H,IQ),H=1,WEL,IQ=1,5),(DLB(H),H=1,WBZL),(DCOSXB(H),H=1,
C      2 WBZL),(DCOSZB(H),H=1,WBZL),(WBE(H),H=1,WBZL),(ISB(H,IQ),H=1,
C      3 WBZL,IQ=1,4),(WPB(H),H=1,WBN),(RKU(I),I=1,WHUPR),
C      4 (INHUTP(H),H=1,WEL),(RKL(I),I=1,WHLPR),(INLTYP(H),H=1,WEL)
C      WFEROB=WPROB
C
C
C 10 WRITE(1) TIME,(H(H),H=1,WHF),(VI(H),H=1,WHF),(VZ(H),H=1,WHF),
C      1 (HU(I),I=1,WHUPR),(HL(I),I=1,WHLPR)
C      RETURN
C      END

```

```

STOR 005
STOR 010
STOR 015
STOR 020
STOR 025
STOR 030
STOR 035
STOR 040
STOR 045
STOR 050
STOR 055
STOR 060
STOR 065
STOR 070
STOR 075
STOR 080
STOR 085
STOR 090
STOR 095
STOR 100
STOR 105
STOR 110
STOR 115
STOR 120
STOR 125
STOR 130
STOR 135
STOR 140
STOR 145
STOR 150
STOR 155
STOR 160
STOR 165
STOR 170
STOR 175
STOR 180
STOR 185
STOR 190

```

APPENDIX C. (continued)

SUBROUTINE READR(P,MAXMOD,NRP)	REDR 005
IMPLICIT REAL*8(A-H,O-Z)	REDR 010
DIMENSION P(MAXMOD)	REDR 015
NODES=0	REDR 020
150 READ 40, NI,NSEQ,NAD,FNI,FAD,PRD	REDR 025
IF(NI.EQ.0) GO TO 170	REDR 030
NJ=NI+NSEQ	REDR 035
DO 160 N=NI,NJ	REDR 040
NODES=NODES+1	REDR 045
I=NI+(N-NI)*NAD	REDR 050
IF(PED.NE.0.0) GO TO 155	REDR 055
F(I)=FNI+FAD*DFLOAT(N-NI)	REDR 060
GO TO 160	REDR 065
155 I1=I-NAD	REDR 070
IF(N.EQ.NI) F(I)=FNI	REDR 075
IF(N.EQ.N1) DINC=1.0D0	REDR 080
IF(N.GT.NI) DINC=DINC*(1.0D0+PRD)	REDR 085
IF(N.GT.NI) F(I)=F(I1)+FAD*DINC	REDR 090
160 CONTINUE	REDR 095
GO TO 150	REDR 100
170 IF(NODES.EQ.NRP) GO TO 180	REDR 105
PRINT 1100	REDR 110
STOP	REDR 115
180 IF(NRP.LE.MAXMOD) GO TO 190	REDR 120
PRINT 1200	REDR 125
STOP	REDR 130
190 CONTINUE	REDR 135
C 40 FORMAT(3I5,5X,4D10.3)	REDR 140
1100 FORMAT(1E1/1E ,64H *** ERROR IN EXECUTING READR SINCE NODES .NE.	REDR 145
> NRP: STOP ***)	REDR 150
1200 FORMAT(1H1/1E ,64H *** NRP .GT. MAXMOD IN EXECUTING READR: STOP)	REDR 155
C RETURN	REDR 160
END	REDR 165
	REDR 170
	REDR 175

APPENDIX C. (continued)

SUBROUTINE READN(INDTYP,NTTYP,NTYPE)	REDN 005
IMPLICIT REAL*8(A-H,O-Z)	REDN 010
DIMENSION INDTYP(NTTYP)	REDN 015
C	REDN 020
NTYPES=0	REDN 025
110 READ 30,NI,NSEQ,NAD,NTTYP,NTYPAD	REDN 030
IF(NI.EQ.0) GO TO 130	REDN 035
NJ=NI+NSEQ	REDN 040
DO 120 N=NI,NJ	REDN 045
I=NI+(N-NI)*NAD	REDN 050
INDTYP(I)=NTTYP + NTYPAD*(N-NI)	REDN 055
NTYPES=NTYPES+1	REDN 060
120 CONTINUE	REDN 065
GO TO 110	REDN 070
130 IF(NTYPES.EQ.NTTYPE) GO TO 140	REDN 075
PRINT 1100	REDN 080
STOP	REDN 085
140 IF(NTYPE.LE.NTTYPE) GO TO 150	REDN 090
PRINT 1200	REDN 095
STOP	REDN 100
150 CONTINUE	REDN 105
C	REDN 110
30 FORMAT(5I5)	REDN 115
1100 FORMAT(1H1/1H ,64H *** ERROR IN EXECUTING READN SINCE NTYPES .NE.	REDN 120
> NTTYPE: STOP ***)	REDN 125
1200 FORMAT(1H1/1H ,46H *** NTYPE .GT. NTTYP IN EXECUTING READN: STOP)	REDN 130
C	REDN 135
RETURN	REDN 140
END	REDN 145

APPENDIX C. (continued)

	SUBROUTINE INTERP (FV,T,P,TV,NKPR,NKDP,NPR,NDP)	INTZ 005
	IMPLICIT REAL*8 (A-H,O-Z)	INTZ 010
	DIMENSION FV(NKPR),T(NKPR,NKDP),P(NKPR,NKDP)	INTZ 015
C		INTZ 020
	DO 160 I=1,NPR	INTZ 025
	DO 140 J=2,NDP	INTZ 030
	IF (I(I,J-1).LE.TV .AND. TV.LE.T(I,J)) GO TO 120	INTZ 035
	GO TC 140	INTZ 040
	120 FV(I)=P(I,J-1)*(TV-T(I,J-1))*(P(I,J)-P(I,J-1))/(T(I,J)-T(I,J-1))	INTZ 045
	GO TC 160	INTZ 050
	140 CONTINUE	INTZ 055
	PRINT 1000, I	INTZ 060
	FV(I)=0.0	INTZ 065
	160 CONTINUE	INTZ 070
C		INTZ 075
	1000 FORMAT (1H1/1H ,16H *** WARNING:T(12,55H) IS OUT OF RANGE, FV IS	INTZ 080
	> SET TO 0 IN SUBROUTINE INTERP)	INTZ 085
C		INTZ 090
	RETURN	INTZ 095
	END	INTZ 100

INTERNAL DISTRIBUTION

1. S. I. Auerbach	20. D. E. Reichle
2. W. J. Boegly, Jr.	21. E. R. Rothschild
3. R. O. Chester	22. T. H. Row
4. N. H. Cutshall	23. B. P. Spalding
5. E. C. Davis	24-33. L. E. Stratton
6. D. L. DeAngelis	34. F. G. Taylor
7. J. R. Gissel	35. D. A. Webster
8. C. S. Haase	36-40. G. T. Yeh
9-13. D. D. Huff	41. Central Research Library
14. J. R. Jones	42-54. ESD Library
15. I. L. Larsen	55-56. Laboratory Records Dept.
16. S. Y. Lee	57. Laboratory Records, ORNL-RC
17. C. A. Little	58. ORNL Y-12 Technical Library
18. T. W. Oakes	59. ORNL Patent Section
19. W. W. Pitt	

EXTERNAL DISTRIBUTION

60. G. V. Abi-Ghanem, McComb-Knutson Associates, Inc.,
12800 Industrial Park Blvd, Plymouth, MN 55441
61. M. U. Ahmad, Department of Geology, Ohio University, Athens,
OH 45701
62. E. L. Albenesius, Savannah River Laboratory, Aiken, SC 29801
63. Margaret Asgain, Dames & Moore, 1626 Cole Blvd., Golden,
CO 80401
64. M. Barainca, 550 Second Street, Department of Energy,
Idaho Falls, ID 83401
65. Dennis Beckmann, Amoco Oil, P.O. box 400, Naperville, IL 60566
66. T. W. Bergen, Computer Application Engineers, Inc.,
P.O. Box 408, Exton, PA 19341
67. M. Bergeron, USGS, Water Resources Division, 521 West Seneca
Street, Ithaca, NY 14850
68. Eshel Bresler, Soil Physicist, Agricultural Research
Organization, The Volcanic Center, Institute of Soil and Water,
P.O. Box 6, Bet Dagan, 50-250, Israel
69. R. W. Brocksen, Electric Power Research Institute,
3412 Hillview Ave., P.O. Box 10412, Palo Alto, CA 94303
70. D. R. Brown, Oak Ridge Operations, P.O. Box E, Department of
Energy, Oak Ridge, TN 37831
71. T. W. Broyd, Atkins Research and Development, Woodcote Grove
Ashley Rd., Epsom Surrey KT18 5BW England
72. W. Brutsaert, School of Civil Engineering, Cornell University,
Ithaca, NY 14850
73. T. Burby, Desert Research Institute, P.O. Box 60220, Reno,
NV 89506

74. S. T. Chau, Albert Environmental Protection Service,
14th Floor, Standard Life Center, 10405 Jasper Ave., Edmonton,
Alberta, Canada
75. S. W. Childs, Department of Soil Science, Oregon State
University, Corvallis, OR 97331
76. C. G. Clyde, Utah Water Resources Laboratory, Utah State
University, Logan, UT 84322
77. C. R. Cole, Pacific Northwest Laboratory, P.O. Box 999,
Richland, WA 99352
78. Peter Colombo, Brookhaven National Laboratory, Bldg. 830,
Upton, NY 11973
79. D. Cooper, School of Public Health, Harvard University,
665 Huntington Ave., Boston, MA 02115
80. R. C. Dahlman, Carbon Cycle Program Manager, Carbon Dioxide
Research Division, Office of Energy Research, Room J-3111,
ER-12, Department of Energy, Washington, DC 20545
81. A. K. Deb, Roy F. Weston, Inc., Weston Way, West Center,
PA 19380
82. J. L. Deichman, Rockwell Hanford Operations, 2750 E. Bldg.,
200 E Area, P.O. Box 800, Richland, WA 99353
83. D. G. de Marsily, Ecole Nationale, Supérieure des Mines,
Centre d'Informatique Géologique, 35 Rue Saint-Honore,
77305 Fontainebleau, France
84. G. DePoorter, Los Alamos Scientific Laboratory, P.O. Box 1663,
Los Alamos, NM 87545
85. David B. Dunbar, Woodward-Clyde Consultants, Three Embarcadero
Center, Suite 700, San Francisco, CA 94111
86. E. Fray, EG&G Idaho, P.O. Box 1625, Idaho Falls, ID 83401
87. P. A. Freeze, Dept. of Geological Sciences, University of
British Columbia, Vancouver, B.C., Canada
88. W. J. Ghurek, U.S. Department of Agriculture, Northeast
Watershed Research Center, 110 Research Building A, University
Park, PA 16802
89. Lynn W. Gelhar, Department of Geosciences, New Mexico Tech,
Socorro, NM 87801
90. Robert W. Gilliam, Department of Earth Sciences, University of
Waterloo, Waterloo, Ontario, Canada 21361
91. D. Goode, Environmental Engineering Department,
GCA Corporation, 213 Burlington Road, Bedford, MA 01730
92. Stephen Gorelick, USGS, Water Resources Division, MS-21,
345 Middlefield Road, Menlo Park, CA 94205
93. D. D. Gray, School of Civil Engineering, Purdue University,
West Lafayette, IN 47907
94. A. Berge Gureghian, Project Management Division, Performance
Analysis Department, Battelle Columbus, 505 King Avenue,
Columbus, OH 43201
95. David Hamilton, Water Management Division, P.O. Box 30028,
Lansing, MI 48909
96. L. A. Hermann, Dept. of Geosciences, Louisiana Tech.
University, Ruston, LA 71272

97. Beverly Herzog, Illinois State Geological Survey, Natural Resources Bldg., 615 E. Peabody Drive, Champaign, IL 61820
98. Richard Heystee, Ontario Hydro, 700 University Avenue, Toronto, Ontario M5G 1X6 Canada
99. Frank F. Hooper, Ecology, Fisheries and Wildlife Program, School of Natural Resources, The University of Michigan, Ann Arbor, MI 48109
100. J. A. Hoopes, 1261B Engineering Bldg., Department of Civil and Environmental Engineering, University of Wisconsin-Madison, Madison, WI 53706
101. J. Horyna, Department of Radiation Protection, Nuclear Research Institute, 250-68 Rez, Prague, CSSR
102. Chi-hua Huang, Dept. of Hydrology & Water Resources, University of Arizona, Tucson, AZ 85721
103. J. C. Hwang, Department of Civil Engineering, Drexel University, Philadelphia, PA 19104
104. D. Jacobs, Evaluation Research Corporation, Oak Ridge, TN 37830
105. Elizabeth Jordan, Operations Division, Office of Waste Operations and Technology, Department of Energy, MS B-107, Washington, DC 20545
106. J. Jeffcoate, Scicon Computer Services Limited, Brick Close, Kiln Farm, Milton Keynes, MK11 3EJ, Great Britain
107. R. Johnson, Department of Environmental Sciences, Oregon Graduate Center, 19600 NW Walker Road, Beaverton, OR 97006
108. A. Jonasson, Department of Geology, Chalmers University of Technology, S-412, 96 Gothenburg, Sweden
109. D. B. Jones, O'Brien & Gere Engineers, Inc., Box 4873, 1304 Buckley Rd., Syracuse, NY 13221
110. W. A. Jury, Soils Department, University of California, Riverside, CA 92501
111. D. Kent, Whitehurst Hall, Oklahoma University, Stillwater, OK 74078
112. J. T. Kuo, Dept. of Civil Engineering, National Taiwan University, Taipei, Taiwan
113. W. J. Kutz, Harza Engineering Company, 150 S. Wacker Drive, Chicago, IL 60606
114. David Langseth, Metcalf & Eddy, Inc., 50 Staniford Street, Boston, MA 02114
115. E. G. Lapalla, Extec Western, Inc., 3777 Long Beach Blvd., Long Beach, CA 90807
116. John Latkovich, Pacific Northwest Laboratory, Richland, WA 99352
117. D. E. Large, Oak Ridge Operations, P.O. Box E, Department of Energy, Oak Ridge, TN 37830
118. R. T. Lawson, Environmental Sciences Division, Australian Atomic Energy Commission, New Illanarra Rd., Lucas Heights, N.S.W., Australia
119. Denis R. LeBlanc, USGS, Water Resources Division, 150 Causeway Street, Suite 1001, Boston, MA 02114

120. Timothy C. Lederman, Radiological Sciences Institute, State of New York, Dept. of Health, The Governor Nelson A. Rockefeller Empire State Plaza, Albany, NY 12201
121. Joan Lefler, Environmental Review Division, Office of Testing and Evaluation, Office of Toxic Substances, Environmental Protection Agency, 401 M Street, SW, Washington, DC 20460
- 122-123. J. A. Lenhard, Oak Ridge Operations, P.O. Box E, Department of Energy, Oak Ridge, TN 37830
124. John J. Lentz, Applied Physics Laboratory, the Johns Hopkins University, Johns Hopkins Rd., Laurel, MD 20810
125. C. C. Leser, Department of Civil Engineering, University of Notre Dame, Notre Dame, IN 46556
126. David Lester, Science Applications, Inc., 1200 Prospect, La Jolla, CA 92038
127. G. B. Levin, EG&G Idaho, P.O. Box 1625, Idaho Falls, ID 83401
128. Richard Lo, Anderson-Nichols, 2666 East Bayshore Rd., Palo Alto, CA 94303
129. Roger L. Long, General Software Corp., Metro-Park, 8401 Corporate Dr., Lanover, MD 20785
130. Y. C. Li, Dept. of Civil Engineering, National Taiwan University, Taipei, Taiwan
131. P. R. Manam, Sargent & Lundy Engineers, 55 East Monroe Street, Chicago, IL 60603
132. A. Marples, Safety and Medical Services, British Nuclear Fuels Limited, Sellafield Seascale, Cumbria CA20 1PG, England
133. 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
134. Marry K. McGuire, 684 Mary Reed Rd., Bader, PA 15005
135. Charles F. McLane III, Dept. of Environmental Sciences, Clark Hall, University of Virginia, Charlottesville, VA 22903
136. Daene C. McKinney, Department of Environmental Resources Engineering, Hornbonds State University, Arcata, CA 95521
137. J. W. Mercer, GeoTrans Inc., P.O. Box 2550, Reston, VA 22090
138. G. L. Meyer, Environmental Protection Agency, 401 M Street SW, MS-ANR459, Washington, DC 20460
139. Fred H. Molz, Director, School of Engineering and Engineering Experiment Station, Auburn University, Auburn, AL 36849
140. Harold A. Mooney, Department of Biological Sciences, Stanford University, Stanford, CA 94305
141. Marvin K. Moss, Director, Program Integration Analysis Division, Office of Health and Environmental Research, Office of Energy Research, MS-G256, Century 21 Bldg. #20010, Room A14020, Department of Energy, Washington, DC 20545
142. Haydn H. Murray, Director, Department of Geology, Indiana University, Bloomington, IN 47405
143. T. N. Narasimhan, Lawrence Berkeley Laboratory, University of California, Berkeley, CA 94720

144. Jerry D. Nason, NEBG Installation Representative, Mail Code 311, General Electric Company, 175 Curtner Ave., San Jose, CA 95125
145. R. W. Nelson, Battelle Northwest, Richland, WA 99352
146. G. K. Oertel, Director, Defense Waste and By-Products Division, Department of Energy, DP-121, MS-B107, Headquarters, Washington, DC 20545
147. Peter M. Ornstein, MS-697, U.S. Nuclear Regulatory Commission, 7915 Eastern Ave., Silver Springs, MD 20910
148. William S. Osburn, Jr., Division of Ecological Research, Office of Health and Environmental Research, Office of Energy Research, Department of Energy, Washington, DC 20545
149. C. A. Oster, Energy Systems Department, Pacific Northwest Laboratories, Richland, WA 99352
150. Frank Parker, Dept. of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN 37235
151. Robert L. Powell, Dept. of Civil Engineering, Remote Sensing System Laboratory, University of Maryland, College Park, MD 20742
152. David Prudic, USGS, Water Resources Division, Room 229, Federal Building, Carson City, NV 89701
153. Ellen J. Quinn, Northeast Utilities, P.O. Box 270, Hartford, CT 06141
154. P.A.C. Raats, Institute for Soil Fertility, P.O. Box 30003, 9750 Ra Haren (Gr), The Netherlands
155. Paul G. Risser, Office of the Chief, Illinois Natural History Survey, Natural Resources Building, 607 E. Peabody Ave., Champaign, IL 61820
156. Jackson Robertson, USGS, 410 National Center, Reston, VA 22092
157. J. C. Rodgers, Group LS-6, Environmental Sciences Group, Los Alamos National Laboratory, Los Alamos, NM 87546
158. William L. Rohrer, Pope-Reid Associates, Inc., 245 E. 6th Street, Suite 813, St. Paul, MN 55101
159. E. M. Romney, University of California, Los Angeles, 900 Veteran Avenue, Los Angeles, CA 90024
160. P. W. Root, Jr., Savannah River Laboratory, Aiken, SC 29801
161. Jerry L. Schnoor, Division of Energy Engineering, The University of Iowa, Iowa City, IA 52242
162. Thomas R. Schultz, Wahler Associates, 12477 W. Cedar Dr., Suite 206, Lakewood, CO 80228
163. Genevieve Segol, Bechtel Civil and Minerals, Inc., P.O. Box 3965, San Francisco, CA 94119
164. Marios Sophocleous, Kansas Geological Survey, 1930 Avenue A, Campus West, The University of Kansas, Lawrence, KS 66044
165. Jack Sosebee, Environmental Sciences and Engineering, Inc., 7332 S. Alton Way, Suite H-1, Englewood, CO 80112
166. Ricky H. Stacks, Department of Pollution Control & Ecology, State of Arkansas, 8001 National Drive, P.O. Box 9583, Little Rock, AR 72209

167. J. G. Steger, Environmental Sciences Group, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87545
168. Daniel B. Stephens, New Mexico Tech., Department of Geosciences, Socorro, NM 87801
169. David Swan, Point Road, Wilson Point, S. Norwalk, CT 06854
170. Steve Y. Tsai, Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439
171. M. van Genuchten, U.S. Salinity Laboratory, 4500 Glenwood Drive, Riverside, CA 92501
172. John J. Vasconcelos, James M. Montgomery-Engineers, 250 North Madison Avenue, Pasadena, CA 91101
173. Jan Wagner, School of Chemical Engineering, Oklahoma State University, Stillwater, OK 74078
174. Robert L. Watters, Division of Ecological Research, Office of Health and Environmental Research, Office of Energy Research, Department of Energy, Washington, DC 20545
175. Tom C. Wells, Dappolonia Consulting Engineers, Inc., 10 Duff Road, Pittsburgh, PA 15235
176. D. E. Wessol, Reactor Physics Branch, EG&G Idaho, Inc., P.O. Box 1625, Idaho Falls, ID 83415
177. J. B. Whitsett, Chief, Nuclear Fuel Cycle and Waste Management Branch, Department of Energy, Idaho Operations Office, 550 Second Street, Idaho Falls, ID 83401
178. P. J. Wierenga, Department of Agronomy, New Mexico State University, P.O. Box 30, Las Cruces, NM 88003
179. Frank J. Wobber, Division of Ecological Research, Office of Health and Environmental Research, Office of Energy Research, MS-E201, Department of Energy, Washington, DC 20545
180. Robert W. Wood, Director, Division of Pollutant Characterization and Safety Research, Department of Energy, Washington, DC 20545
181. Steve Yabusaki, Geophysical Fluid Dynamics Section, Pacific Northwest Laboratories, Richland, WA 99352
182. Feng-quing Zhuang, Institute of Mechanics, Chinese Academy of Sciences, Beijing, Peoples' Republic of China
- 183-209. Technical Information Center, Oak Ridge, TN 37830