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

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

ENVIRONMENTAL SCIENCES DIVISION

FEMA: A Finite Element Model of Material
Transport Through Aquifers

G. T. Yeh and D. D. Huff

Environmental Sciences Division
Publication No. 2428

NUCLEAR and CHEMICAL WASTE PROGRAMS
(Activity No. AR 05 15 15 0; ONLWL14)

Date of Issue: January 1985

Prepared for the
Office of Defense Waste and Byproduct Management

NOTICE This document contains information of a preliminary nature.
It is subject to revision or correction and therefore does not represent a
final report.

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

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.

NOTICE
COPIES OF THIS REPORT ARE AVAILABLE
It has been reproduced from the best
available copy to permit the broadest
possible availability.

CONTENTS

LIST OF FIGURES	v
LIST OF TABLES	vii
ABSTRACT	ix
I. INTRODUCTION	1
II. MATHEMATICAL STATEMENTS	7
II.1 Three-dimensional Transport	7
II.2 Vertically Averaged Transport	14
II.3 Initial and Boundary Conditions	16
III. FINITE ELEMENT APPROXIMATION	19
III.1 Derivation of Ordinary Differential Equations with FEM	19
III.2 Incorporation of Boundary Conditions	32
III.3 Time Intergration	35
III.4 Mass Balance Computation	38
IV. COMPUTER PROGRAM STRUCTURE	41
IV.1 Purpose of FEMA Program	41
IV.2 Structure and Function of FEMA Program	42
V. VERIFICATION AND DEMONSTRATION APPLICATIONS	55
V.1 Verification Example 1	55
V.2 Verification Example 2	58
V.3 Verification Example 3	64
V.4 Demonstration Example 1	65
V.5 Demonstration Example 2	69
V.6 Summary and Conclusion	74
VI. REFERENCES	78
APPENDIX A: DATA INPUT GUIDE	A-1
APPENDIX B: INPUTS	B-1
APPENDIX C: LISTING OF FORTRAN IV SOURCE PROGRAM	C-1

List of Figures

Figure 1. Sketch of an arbitrary material volume and its enclosing surface	8
Figure 2. Sketch of a region of interest and its three types of boundaries	13
Figure 3. Example discretization by 80 nodes and 69 elements (with 59 quadrilateral and 10 triangular elements)	20
Figure 4. Relationships between globally defined and element-wise defined basis functions	20
Figure 5. Global coordinate vs local coordinate for a typical quadrilateral element	23
Figure 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	24
Figure 7. Plots of four upstream weighting functions for a quadrilateral element with $\alpha_1 = \beta_1 = \alpha_2 = \beta_2 = 1.0$	27
Figure 8. Optimum weighting factors as functions of Peclet number	28
Figure 9. Plots of three upstream weighting functions for a triangular element with $\alpha_1 = -1.0$, $\alpha_2 = -1.0$ and $\alpha_3 = 1.0$	29
Figure 10. Incorporation of Dirichlet boundary conditions	33
Figure 11. Program structure of FEMA	43
Figure 12. Concentration profiles at time = 4.0 for (a) $P_e = 0.05$, (b) $P_e = 1.0$, (c) $P_e = 50.0$, and (d) $P_e = 10.0$, respectively	59
Figure 13. Sketch of the region of interest for example 2	60
Figure 14. Concentration profiles at time = 4.0 along $y = 2.5$ for (a) $P_e = 0.05$, (b) $P_e = 1.0$, and (c) $P_e = 50.0$, respectively	62
Figure 15. Problem sketch of example 4 for flow in a confined aquifer: (a) three-dimensional view and (b) plane view	67
Figure 16. Finite element discretization for examples 4 and 5	68
Figure 17. (a) Head distribution and (b) vertically integrated Darcy's fluxes at steady state.	70

Figure 18. Concentration distribution at time = (a) 386, (b) 1,830, and (c) 3,750 days, respectively	71
Figure 19. Problem sketch of example 5 for flow in an unconfined aquifer: (a) three-dimensional view and (b) plane view	73
Figure 20. (a) head distribution and (b) vertically integrated Darcy's fluxes at steady state	75
Figure 21. Concentration distributions at time = (a) 386, (b) 1,830, and (c) 3,750 days, respectively	21

List of Tables

Table 1. Optional finite element schemes	37
Table 2. Comparison of CPU time in sec/time step required by direct elimination and pointwise iteration solution strategies for example 1	58
Table 3. Comparison of CPU time in sec/time step required by direct elimination and pointwise iteration solution strategies for example 2	63
Table 4. Comparison of CPU time in sec/time step required by direct elimination and pointwise iteration solution strategies for example 3	65
Table 5. Typical material properties for sand stone, silt loam, and clay	68
Table 6. Dispersion and media-chemical interaction properties	72

Abstract

YEH, G. T. and D. D. Huff. 1984. FEMA: A Finite Element Model of Material Transport through Aquifers. ORNL-6063. Oak Ridge National Laboratory, Oak Ridge, Tennessee. 210 pp.

This report documents the construction, verification, and demonstration of a Finite Element Model of Material Transport through Aquifers (FEMA). The particular features of FEMA are its versatility and flexibility to deal with as many real-world problems as possible. Mechanisms included in FEMA are: carrier fluid advection, hydrodynamic dispersion and molecular diffusion, radioactive decay, sorption, source/sinks, and degradation due to biological, chemical as well as physical processes. Three optional sorption models are embodied in FEMA. These are linear isotherm and Freundlich and Langmuir nonlinear isotherms. Point as well as distributed source/sinks are included to represent artificial injection/withdrawals and natural infiltration of precipitation. All source/sinks can be transient or steady state. Prescribed concentration on the Dirichlet boundary, given gradient on the Neumann boundary, and known fluxes on the Cauchy boundary can be time-dependent or constant. Source/sink strength over each element or node, concentration at each Dirichlet boundary node, gradient at each Neumann boundary segment, and flux at each Cauchy boundary segment can vary independently of each other. The aquifer may consist of as many formations as desired. Either completely confined or completely unconfined or partially confined and partially unconfined aquifers can be dealt with effectively. FEMA also includes transient leakage to or from the aquifer of interest through confining beds from or to aquifers lying below and/or above. Discretization of a compound region with irregular curved boundaries is made easy by including both quadrilateral and triangular elements in the formulation. Large-field problems can be solved without excessive requirement of CPU memory and CPU time by including pointwise solution strategies as optional alternatives to the direct elimination solution method for the matrix equation approximating the partial differential equation of material transport. FEMA is verified against two

simple cases to which analytical solutions are available. It is illustrated by one moderately sized problem on how CPU memory and CPU time can be saved with pointwise iteration solution strategies. It is then demonstrated by two examples on how the model can be applied to heterogeneous and anisotropic aquifers with transient boundary conditions and time-dependent source/sinks. One of the demonstration examples is for a confined aquifer of variable thickness with leakage from or to both upper and lower aquifers. The other is for an unconfined aquifer with distributed sources from surface deposition.

I. INTRODUCTION

Numerous models utilizing finite difference and finite element methods have been reported in the literature for simulating contaminant transport in aquifers (Bachmat et al, 1980). Models differ in: (1) how many processes are included and how each process is formulated to yield governing partial and integro-differential equations, (2) how initial and boundary conditions are treated, (3) what methods are used to solve the resulting partial differential equations with associated initial and boundary conditions, (4) how the spatial and temporal variabilities are considered, and (5) how many constituents are included in a single computer code.

The major processes determining the contaminant distribution in aquifers are: (1) advection and convection, (2) dispersion and diffusion, (3) compaction, (4) sorption, (5) radioactive decay, (6) bio-degradation, (7) complexation or speciation, and (8) source/sinks. Practically all models of contaminant transport in aquifers include advection and convection because the transport in such media is almost entirely dependent on the flow. Without flow, the transport can only be by diffusion, which is an extremely slow process. Since these mechanisms are completely determined by the flow velocity, models differ on how the flow velocity is obtained. Many models simply assume the flow velocity as input. These classes of models are only applicable to the laboratory conditions where the flow velocity can be controlled easily. For prototype applications, the flow field must be obtained either by continuous measurements over the region of interest or by hydrodynamic simulation of groundwater flow and thermal transport. Measurements of flow field under continuously transient conditions, which may result from man-made perturbations or natural causes, are relatively impractical and almost impossible because of astronomical cost or technological difficulties. Instead, simulation of well-defined hydrodynamic and thermal transport boundary value problems offers a practical approach. In contrast to continuously measuring the

flow at all points considerations are then directed to the measurements of the fixed properties of the porous media such as the hydraulic conductivity or permeability, porosity, pore-size distribution, compressibility of both soil matrix and water, bulk density of the media, thermal diffusivity, heat conductivity, heat capacity of water and soil grains, and heat of vaporization. The companion groundwater flow model (Yeh and Huff 1983) is then used to simulate the flow variables including hydraulic head and Darcy's flux.

The dispersion and diffusion combine all mass transport not carried by the mean water velocity. They may be caused by velocity fluctuations with time, the nonuniform movement of water through random meandering (hydrodynamic dispersion) and the random Brownian motion of the molecules (diffusion). Essentially, all models employ the classical approach of assuming that the flux due to dispersion and diffusion is proportional to the gradient of concentration. The proportionality is called the dispersion-diffusion coefficient tensor, which is related to the effective porosity in the case of saturated media, the diffusion coefficient, the tortuosity of the media, and the flow velocity and diffusivity tensor. This approach has become a well-accepted formalism. While this approach has met with some success in practical applications, the controversies and disputes over the values of the dispersion-diffusion coefficient tensor will undoubtedly continue.

Most of the existing models neglect the transport of contaminants by the compaction of the soil matrix. Only a few model includes this transport mechanism which is characterized by the plastic-elastic deformation of the soil matrix. The deformation, of course, depends on the compressibility of the soil matrix and on the rate of change of pressure within the media. Since the temporal spatial distribution of pressure is obtained by hydrodynamic and thermal transport simulations, which are required to provide the velocity field, no additional efforts are needed to obtain the input for this mechanism. Thus, the transport by compaction, especially that by vertical consolidation, can be handled easily.

Advection, convection, dispersion, and diffusion do not directly reduce the amount of radionuclides/chemicals dissolved in the water, but they do affect the concentration distribution.

The attenuation of dissolved radionuclides/chemicals occurs through transfer of the dissolved phase to the adsorbed phase by sorptive-related processes and through transformation of radionuclides into stable isotopes by decay or of chemicals into new species by physical, chemical, and biological processes. The transfer of radionuclides/chemicals from dissolved into adsorbed phase or vice versa is generally modeled by adsorption-desorption functions, that describe the relationships between the concentration of radionuclides/chemicals dissolved in water and that adsorbed by medium matrix. Numerous adsorption-desorption models are available (van Genuchten, 1978; Travis, 1978). These models can be divided into two categories: equilibrium models which assume instantaneous adsorption of radionuclides/chemicals and kinetic models which consider the rate of approach toward equilibrium. A list of models of these two categories can be found elsewhere (van Genuchten, 1978; Travis, 1978). These models require one to five parameters such as distribution coefficient, bonding strength, maximum adsorbed concentration, initial adsorbed concentration, rate constant or in some models purely empirical constants without physical meaning (e.g. the coefficient and power index in the Freundlich nonlinear isotherm model). This report includes three equilibrium models to deal with sorption processes: (1) linear isotherm, (2) Langmuir isotherm, and (3) Freundlich isotherm.

For radionuclides, all models assume that the decay rate is proportional to the total amount of radionuclide mass present. The proportionality is termed the decay constant which is known for each of almost all isotopes. The bio-degradation of chemicals, on the other hand, has diversified approaches. An extensive survey of modeling the degradation of chemicals in the porous media can be found in the Literature (Browman, et al., 1981). As a first approximation, a linear kinetic model can be assumed in which the amount of chemicals degraded is proportional to the amount of chemicals present. The proportionality is the first order rate constant which is analogous to the decay constant in radionuclide modeling. Complexation is to study the chemicals which appears in various forms. This requires the simulation of multi-chemical species. This mechanism is not simulated by the model in its present form.

Linking all the processes (advection/convection, dispersion/diffusion, compaction, sorption, decay, bio-degradation, and source/sink) with their corresponding formulation is the partial differential or partial integro-differential equations, one for each radionuclide/chemical species. The set of governing equations are then augmented by auxiliary conditions and the initial and boundary conditions imposed on the system (geometry of the region under contamination, aquifer characteristics, precipitation-evapotranspiration flux, location of wells, or open surface water bodies, etc).

Most of the transient contaminant transport models require the prescription of initial conditions. Some models provide the option of obtaining initial conditions by simulating the steady state version of the transient governing equations under time-invariant boundary conditions. Both options are provided by the model presented in this report.

For the most general case, four types of boundary conditions may be prescribed: (1) Dirichlet, (2) Neumann, (3) Cauchy, and (4) Variable conditions. For Dirichlet conditions, the concentration is prescribed on the boundary. For Neumann conditions, the normal gradient of the concentration is prescribed on the boundary. For Cauchy conditions, the total flux (advection and dispersion fluxes) is prescribed on the boundary. On the Variable boundary, either the Neumann or the Cauchy condition is applied depending on the flow direction. If the flow is directed into the region of interest, the Cauchy condition is normally applied. On the other hand, if the flow is directed out of the region, the Neumann condition is used to allow the contaminant to seek its own concentration on the boundary. Most of the existing models do not have the capabilities to treat all these four types of boundary conditions. Furthermore, most of the models were not designed to handle boundary values that are time-dependent and vary along the boundary. The model presented in this report is designed to include all four types of boundary conditions. Each type of boundary conditions allows the arbitrary variations of its values with time and over the entire boundary surfaces (segments) so that real-world problems can be dealt with effectively.

Once the governing equations and the associated initial and boundary conditions are defined, next important task in modeling contaminant distributions is to obtain solutions for the system to

yield numerical results. Without the solution, the model is incomplete and of little use no matter how many processes are included and how sophisticated and accurately each process is formulated in the derivation of the governing equations. After all, modeling is nothing but the computation of assumed processes. Thus, devising appropriate computational algorithms is of paramount importance in any physics-based modeling. Basically, two approaches are currently available to solve the governing equations of contaminant distributions: analytical techniques and numerical methods.

In order to obtain an analytical solution, one generally must assume constant system parameters, constant flow field, and highly simplified geometry for the region of interest. Exact explicit expressions for the spatial-temporal distributions of concentration are then generated by the use of integral and differential calculus. The advantages of analytical solutions are numerous, e.g., stability and convergence are no longer a problem, accuracy is warranted, convenience is a certainty, efficiency is assured, and the programming can be done with minimum effort. However, the necessity of having to make various simplifying assumptions severely restricts their applicability to real-world problems. In spite of these restrictions, it appears that some of the available two- and three dimensional analytical solutions (Kuo, 1976; Cleary, 1978; Yeh, 1981; van Genuchten, 1982) could be applied to well defined geohydrological systems or could be used for rough estimations, and hence should not be excluded a priori from consideration. Furthermore, these well constructed analytical solutions could serve for verification of numerical methods.

While analytical solutions may be suited for some situations, most field problems have such complex physical, chemical, and biological characteristics that the flexibility of numerical approaches are required. The purpose of numerical techniques is to reduce the governing partial differential equations to a set of approximating algebraic equations, which subsequently are solved using methods of linear algebra. The most commonly used numerical techniques are finite difference methods (FDM) and finite element methods (FEM). The fundamental distinction between FEM and FDM is that the former is based directly upon approximating the function

(Pinder and Gray, 1977), whereas the latter is based on approximation of derivatives (Forsythe and Wasow, 1960; Hildebrand, 1968). Finite element methods are adopted here to transform the governing differential equations to the matrix equation. To increase the applicability of the model in terms of stability and convergence, we have included twelve options for the derivation of the matrix equation. The resulting matrix equation is solved with either the direct elimination or successive relaxation technique.

II. MATHEMATICAL STATEMENTS

A formal mathematical derivation for the transient, vertically integrated material transport in the horizontal plane will be presented based on the general three-dimensional transport 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, of course, that boundary conditions and source/sinks can be prescribed.

II.1. Three-dimensional Transport

The mathematical formulation of the conservation of mass in three-dimensional space has been described in detail elsewhere (Duguid and Reeves 1976). Based on: (1) the continuity of mass, (2) the flux laws of hydrological transport, (3) the consolidation of media, (4) chemical fact of sorption, (5) the principle of physico-chemical decay, and (6) hypotheses of bio-chemical decomposition, one can derive the governing equations for the distribution of contaminant concentration in aquifers. The following derivation follows very closely the one given by Duguid and Reeves (1976).

The continuity of mass is expressed in integral form as:

$$\frac{D}{Dt} \int_V (n_e c + \rho_b s) dv = - \int_V \vec{n} \cdot \vec{q} dT + \int_V m dv - \int_V R dv - \int_V B dv \quad (1)$$

where t is the time; $D()/DT$ is the material derivation of () with respect to time; n_e is the effective porosity; c is the dissolved concentration in mass of materials per unit volume of water; ρ_b

is the bulk density in mass of solid matrix per unit volume of the bulk medium; s is the absorbed concentration in mass of materials per unit weight of solid matrix; $\vec{n}$ is a unit vector normal to the deforming surface T (Fig. 1); $\vec{q}$ is the mass flux through the deforming surface T ; V is the volume containing constant amount of solids; m is the internal source/sink of chemical species under consideration; R is radioactive decay of the chemicals in mass per unit time per unit volume of the media; B is the biochemical decomposition in mass per unit time unit volume of the media; and T is the surface enclosing the volume V . It should be noted the volume V will change with time because of consolidation of the solid matrix. By the Reynolds' transport theorem (Owczarek 1964), Eq. (1) can be written as:

$$\int_V \frac{\partial(n_e c + \rho_b s)}{\partial t} dv + \int_T \vec{n} \cdot [(n_e c + \rho_b s) \vec{v}_s] dT + \int_T \vec{n} \cdot \vec{q} dT = \int_V m dv - \int_V R dv - \int_V B dv \quad (2)$$

in which $\vec{v}_s$ is the velocity of the deformable surface T due to the consolidation of the media.

ORNL-DWG 82-18425

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

Applying the Gaussian divergence theorem to Eq. (2) and using the fact that V is arbitrary, we can obtain the following transport equation in differential form:

$$\frac{\partial(n_e c + \rho_b s)}{\partial t} + \nabla \cdot [(n_e c + \rho_b s) \vec{v}_s] + \nabla \cdot \vec{q} = m - R - B \quad (3)$$

The mass flux q relative to the solid matrix may be split into two parts:

$$\vec{q} = \vec{q}_a + \vec{q}_d \quad (4)$$

where $\vec{q}_a$ is the mass flux due to the mean velocity of the carrier fluid relative to the solid and $\vec{q}_d$ is due to hydrodynamic dispersion and molecular diffusion. It is evident that q_a can be written as

$$\vec{q}_a = \vec{v}c \quad (5)$$

where $\vec{v}$ is the Darcy's velocity which can be obtained from field measurement or from the simulation of groundwater flow (Yeh and Huff 1983). Analogous to the Fickian diffusion theory, $\vec{q}_d$ can be expressed as

$$\vec{q}_d = -n_e \bar{D} \nabla c \quad (6)$$

where $\bar{D}$ is the dispersion coefficient tensor. In a porous medium which is assumed to be isotropic with respect to the dispersivity (note: not isotropic with respect to the dispersion coefficient), the hydrodynamic dispersion coefficient tensor may be written as (Scheidegger 1961):

$$n_e \bar{D} = a_T |\vec{v}| \bar{\delta} + (a_L - a_T) \vec{v} \vec{v} / |\vec{v}| + a_m \tau n_e \bar{\delta} \quad (7)$$

where a_T is the transverse dispersivity, v is the magnitude of the Darcy's velocity vector $\vec{v}$, a_L is the longitudinal dispersivity, a_m is the molecular diffusion coefficient, τ is the tortuosity, and $\bar{\delta}$ is the Kronecker delta tensor of the second order.

The second term in Eq. (3) may be expanded as follows:

$$\vec{v} \cdot [(n_e c + \rho_b s) \vec{v}_i] = \vec{v}_i \cdot \nabla (n_e c + \rho_b s) + (n_e c + \rho_b s) \nabla \cdot \vec{v}_i \quad (8)$$

where the first term on the right-hand side is the product of two small terms and can be neglected. If the consolidation of the medium is due mainly to the vertical displacement, the last term in Eq. (8) becomes

$$\nabla \cdot \vec{v}_s = \frac{\partial e}{\partial t} = \alpha \rho_f g \frac{\partial h}{\partial t} = \alpha' \frac{\partial h}{\partial t} \quad (9)$$

where e is the dilatation of the medium, α is the coefficient of compressibility, ρ_f is the density of the fluid, g is the acceleration of gravity, h is the pressure head of the fluid, and α' is the modified coefficient of compressibility.

The radioactive decay term R can be written as

$$R = \lambda(n_e c + \rho_b s) \quad (10)$$

where λ is the decay rate constant. This decay constant is related to the physical half-life $T_{1/2}$ of the contaminant by

$$\lambda = \ln(2)/T_{1/2} \quad (11)$$

A thorough discussion on the biological decomposition can be found in Browman, et al (1982). A linear degradation model can be depicted for illustrative purposes:

$$B = K_w n_e c + K_s \rho_b s \quad (12)$$

where K_w is the decomposition rate constant through the dissolved phase and K_s is that through the adsorbed phase. It must be emphasized that Eq. (12) to represent biochemical degeneration processes must be replaced by a more appropriate formulation if dictated by experimental evidence.

Substituting Eqs. (4) through (12) into Eq. (3), we obtain the governing partial differential equation describing the temporal-spatial distribution of the dissolved and adsorbed materials:

$$\begin{aligned} \frac{\partial(n_e c + \rho_b s)}{\partial t} + \nabla \cdot \vec{v} c = \nabla \cdot (n_e \bar{D} \nabla c) - \lambda(n_e c + \rho_b s) - \alpha'(n_e c + \rho_b s) \frac{\partial h}{\partial t} \\ - K_w n_e c - K_s \rho_b s + m \end{aligned} \quad (13)$$

Equation (13) is not a complete formulation of the problem since it involves two dependent variables c and s in one equation. The relationship between c and s must be prescribed. In other words, the manner in which c is transferred to s or vice versa must be established in terms of a mathematical equation based on physico-chemical evidence. A substantial review on this topic has been presented by a number of investigators (van Genuchten 1978, Travis 1978). Three widely accepted isotherms are included here:

$$s = K_d c \quad \text{for linear isotherm} \quad (14)$$

$$s = K_f c^n \quad \text{for Freundlich isotherm} \quad (15)$$

and

$$s = \frac{K_L s_m c}{1 + K_L c} \quad \text{for Langmuir isotherm} \quad (16)$$

where K_d is the distribution coefficient, K_f is the Freundlich coefficient, n is Freundlich power index, K_L is the Langmuir coefficient representing strength of the bond holding the adsorbed solute on the soil surface, and s_m is the maximum amount of solute that can be adsorbed by unit weight of

soil. It is seen that the linear isotherm and Freundlich isotherms do not imply maximum sorption capacity of the soil. Hence, it can be surmised that they are more appropriate for cases of low concentration.

Substituting Eq. (14) or (15) or (16) into Eq. (13), we have the following equation:

$$n_e R_d \frac{\partial c}{\partial t} + \nabla \cdot (\vec{v} c) = \nabla \cdot (n_e \bar{D} \nabla c) - \lambda(n_e c + \rho_b s) - \alpha' (n_e c + \rho_b s) \frac{\partial h}{\partial t} - K_w n_e c - K_s \rho_b s + m \quad (17)$$

where R_d , the retardation factor, is defined as

$$R_d = \left(1 + \frac{\rho_b}{n_e} \frac{ds}{dc} \right) \quad (18)$$

This factor is physically the ratio of the pore fluid velocity to the material migration velocity. The pore velocity is defined as the Darcy's velocity $\vec{v}$ divided by the effective porosity n_e .

Equation (5) is a universal concept and is true for all aquifer systems. Equations (6), (7), and (10) are well-accepted hypotheses and have been found to be reasonable in practically all types of aquifers that can be considered as continuum material. The adoption of Eqs. (8), (9), (12), and (14) or (15) or (16) will depend on the geohydrochemical condition of the aquifer and the characteristics of the wastes under investigation. For any given site, these equations must be carefully constructed to reflect the geohydrochemical properties and the possible interactions between the waste and burial environment.

Equation (17) and Eq. (14) or (15) or (16) contain two variables, c and s . The number of equations is equal to the number of unknowns. Thus, the system is complete and a mathematical statement is posed. However, to completely define the problem, Eq. (17) must be constrained by

initial and boundary conditions. This can be done using physical considerations. It is assumed that an initial distribution of concentration can be described:

$$c = c_I \text{ in } R \quad (19)$$

where R is the region of interest (Fig. 2). The initial concentration c_I can be obtained from field measurements or from solving of the steady state version of Eq. (17).

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

Three types of boundary conditions can be prescribed for Eq. (17). The first type is the Dirichlet condition on which the concentration is given as

$$c = c_D(x, y, z, t) \text{ on } B_D(x, y, z) = 0 \quad (20)$$

where c_D is the Dirichlet concentration, B_D is part of the boundary $B(x, y, z) = 0$ (Fig. 2), and x , y , and z are the spatial coordinates. The second type is the Neumann boundary on which the gradient flux is given

$$-n \cdot (\vec{n} \cdot \vec{D} \nabla c) = q_N(x, y, z, t) \text{ on } B_N(x, y, z) = 0 \quad (21)$$

where q_N is the prescribed Neumann flux and B_N is part of the boundary B . The third type is the Cauchy boundary on which the total flux is given

$$-\bar{n}[\bar{v}c - n_e(\bar{D}\nabla c)] = q_C(x,y,z,t) \text{ on } B_C(x,y,z) = 0 \quad (22)$$

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

Equations (17) through (22) describe a general transient problem in aquifers. Unfortunately, because of the difficulties associated with computational time and CPU storage requirement for large field problems, a general transient three-dimensional computer model is not practical, although it is much needed in the near source simulation for applications. Instead, a vertically integrated transport model will be developed, which can be applied to most aquifers except in the region close to the source.

II.2. Vertically Averaged Transport

In general, transport through a porous medium is transient and three-dimensional and is described by Eqs. (17) through (22). However, since the geometry of most aquifers is such that they are thin in the vertical direction relative to their horizontal dimension, a simple approach can be employed. According to this approach, we assume that the transport in the aquifer is primarily horizontal everywhere and the vertical component of the flux 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 flux has a large vertical component, for example, in the vicinity of source/sinks. However, even in these cases, at some distance from the source/sink, the assumption of essentially horizontal transport is still valid.

Let x - and y - coordinates be on a horizontal plane and the z -coordinate be in the vertical direction. The governing equations are integrated to eliminate the z -dependence. Beginning with Eq. (17), we may formally write

$$\int_{z_o}^{h_s} \left[n_e R_d \frac{\partial c}{\partial t} + \nabla \cdot (\bar{v} c) - \nabla \cdot (n \bar{D} \nabla c) \right] dz =$$

$$- \int_{z_o}^{h_s} \left[\left(\lambda + \alpha' \frac{\partial h}{\partial t} \right) (n_e c + \rho_b S) + K_w n_e c + K_s \rho_b S - m \right] dz \quad (23)$$

where z_o and h_s are the lower and upper boundaries, respectively, of the aquifer. Both z_o 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. Under this circumstance, h_s is then a function of time as well.

Using the Leibnitz rule, we can carry out the integration of Eq. (23) to yield:

$$n_e R_d b \frac{\partial c}{\partial t} + \nabla_H \cdot (\bar{V} C) = \nabla_H \cdot (n_e \bar{D} b \cdot \nabla_H C) - (\lambda + \alpha') \frac{\partial h}{\partial t} b (n_e c + \rho_b S)$$

$$- n_e K_w b C - \rho_b K_s b S - Q_U - Q_L + M \quad (24)$$

where $b = h_s - z_o$ is the aquifer thickness; ∇_H is the Del operator on the horizontal coordinates x and y ; C and S are the vertically averaged dissolved and sorbed concentration, respectively; $\bar{V}$ is the vertically integrated Darcy's flux; and M is the vertically integrated internal source/sink. It should be noted that many terms, involving the vertical integration of the products and/or functions of deviation quantities, have been dropped in arriving at Equation (24).

In Eq. (24), Q_U and Q_L represent the quantities defined by the following equations:

$$Q_U = K \left[n_e D \cdot \nabla c - \bar{v} c \right] \Big|_{z=h_s} \quad (25)$$

and

$$Q_L = K \left[n_e \bar{D} \cdot \nabla c - \bar{v} c \right] \Big|_{z=z_o} \quad (26)$$

where $\vec{F}$ is the unit vector along the vertical direction. These quantities result from material fluxes through upper and lower confining beds, if any, respectively, to or from the aquifer of interest. They will be approximated by

$$Q_U = [(1 + r_U)K_U(h - h_U)C + (1 - \gamma_U)K_U(h - h_U)C_U]/2 \quad (27)$$

$$Q_L = [(1 + \gamma_L)K_L(h - h_L)C + (1 - \gamma_L)K_L(h - h_L)C_L]/2 \quad (28)$$

where K_U and K_L are the leakage coefficients for upper and lower confining beds, respectively; h_U and h_L are hydraulic head in the aquifers lying above or below the confining beds, respectively; C_U and C_L are the dissolved concentration in the upper and lower aquifers lying above or below the confining beds, respectively; and γ_U and γ_L are given by

$$\gamma_U = 1.0 \text{ if } h \geq h_U, \quad \gamma_U = -1.0 \text{ if } h < h_U \quad (29)$$

and

$$\gamma_L = 1.0 \text{ if } h \geq h_L, \quad \gamma_L = -1.0 \text{ if } h < h_L \quad (30)$$

II.3. Initial and Boundary Conditions

The initial and boundary conditions for Eq. (24) can be obtained by integrating Eqs. (19) through (22) with respect to the vertical coordinate z . Starting with Eq. (19), we obtain

$$C = C_I(x, y) \text{ in } R \quad (31)$$

where C_I is the prescribed initial condition, which can be defined as input or obtained by solving the steady state version of Eq. (24) using time-invariant boundary conditions.

Similarly, the integration of Eqs. (20) through (22) yield the Dirichlet, Neumann, and Cauchy boundary conditions in terms of the vertically averaged quantities;

$$C = C_D(x, y, t) \text{ on } B_D(x, y) = 0 \quad (32)$$

$$-\vec{n} \cdot (n_e b \vec{D} \nabla_H C) = Q_N(x, y, t) \text{ on } B_N(x, y) = 0 \quad (33)$$

and

$$\vec{n} \cdot (\vec{V} C - n_e b \vec{D} \nabla_H C) = Q_C(x, y, t) \text{ on } B_C(x, y) = 0 \quad (34)$$

where C_I , Q_N , and Q_C , are given by

$$C_I = - \int_{z_1}^{z_2} c_I dz \quad (35)$$

$$Q_N = \int_{z_1}^{z_2} q_N dz \quad (36)$$

and

$$Q_C = \int_{z_1}^{z_2} q_C dz \quad (37)$$

Equations (33) and (34) are usually applied to flow-through boundaries. While Eq. (33) is employed for cases when the flow is directed out from the region of interest, Eq. (34) is used when the flow is directed into the region. Under transient flow conditions, we cannot specify a priori which boundary condition should be applied. Hence, Eqs. (33) and (34) will be combined to form a variable flux boundary condition. Stated mathematically, this combined condition can be written as:

$$-\bar{n} \cdot (n_e b \bar{D} \nabla_H C) = 0 \text{ on } B_F(x, y) = 0 \text{ if } \bar{n} \cdot \bar{V} > 0 \quad (38a)$$

and

$$\bar{n} \cdot (\bar{V} C - n_e b \bar{D} \nabla_H C) = (\bar{n} \cdot \bar{V}) C_{in} \text{ on } B_F(x, y) = 0 \text{ if } \bar{n} \cdot \bar{V} < 0 \quad (38b)$$

where C_{in} is the concentration of incoming fluid and $B_F = B_N \cup B_C$. It should be noted that the Neumann flux on the variable boundary is set to zero here. In other words, one should layout the variable boundary far enough away from the source/sinks that the concentration gradient therein becomes very small. If this is not the case, the subroutines Q4RS and Q3RS in the computer code should be modified to deal with non-zero Neumann flux.

Equations (24), (31), (32), and (38) constitute a general mathematical statement for the initial-boundary value problem of vertically integrated contaminant transport in aquifers. Analytical solutions for the general system do not exist. Numerical algorithms will have to be devised to solve the problem. The finite element method (FEM) will be used here because of its ability to treat complex curved boundaries and its simplicity in handling boundary conditions.

III. FINITE ELEMENT APPROXIMATIONS

III.1. Derivation of Ordinary Differential Equations with FEM

Equations (24), (31), (32), and (38) 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 and transport 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 contaminant transport 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 a system of ordinary differential equations with respect to time, (8) incorporate boundary conditions, 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 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

ORNL-DWG P-20016

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

ORNL-DWG 82-10307A

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

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.

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:

$$N_3^a = N_j \text{ over element } a \quad (39a)$$

$$N_2^b = N_j \text{ over element } b \quad (39b)$$

$$N_4^c = N_j \text{ over element } c \quad (39c)$$

$$N_1^d = N_j \text{ over element } d \quad (39d)$$

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

$$N_1^a = N_f, N_2^a = N_i, \text{ and } N_4^a = N_g \text{ over element } a \quad (39e)$$

$$N_1^b = N_g, N_3^b = N_k, \text{ and } N_4^b = N_h \text{ over element } b \quad (39f)$$

$$N_1^c = N_i, N_2^c = N_l, \text{ and } N_3^c = N_m \text{ over element } c \quad (39g)$$

$$N_2^d = N_m \text{ and } N_3^d = N_k \text{ over element } d \quad (39h)$$

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, 2, 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:

$$\begin{aligned} N_1^e &= (1-\xi)(1-\eta)/4, \quad N_2^e = (1+\xi)(1-\eta)/4 \\ N_3^e &= (1+\xi)(1+\eta)/4, \quad N_4^e = (1-\xi)(1+\eta)/4 \end{aligned} \quad (40)$$

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

$$x = \sum_{j=1}^4 x_j N_j^e(\xi, \eta) \text{ and } y = \sum_{j=1}^4 y_j N_j^e(\xi, \eta) \quad (41)$$

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

$$N_1^e = L_1, N_2^e = L_2, \text{ and } N_3^e = L_3 \quad (42)$$

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

where N_j^e ($j=1, 2$, 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

$$L_1 = (a_1 + b_1x + c_1y)/(2A) \quad (43a)$$

$$L_2 = (a_2 + b_2x + c_2y)/(2A) \quad (43b)$$

$$L_3 = (a_3 + b_3x + c_3y)/(2A) \quad (43c)$$

where

$$a_1 = x_2y_3 - x_3y_2, \quad b_1 = y_2 - y_3, \quad c_1 = x_3 - x_2 \quad (43d)$$

$$a_2 = x_3y_1 - x_1y_3, \quad b_2 = y_3 - y_1, \quad c_2 = x_1 - x_3 \quad (43e)$$

$$a_3 = x_1y_2 - x_2y_1, \quad b_3 = y_1 - y_2, \quad c_3 = x_2 - x_1 \quad (43f)$$

and A is the area of the triangle. It is seen from Eqs. (43a) through (43f) that

$$L_1 + L_2 + L_3 = 0 \quad (44)$$

(a)

(b)

Figure 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 .

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

$$C \approx \hat{C} = \sum_{j=1}^N C_j(t) N_j(x, y) \quad (45)$$

where N is the total number of nodes in the region and N_j and C_j are the basis function and the magnitude of C , 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. (45), whereas the latter is based on approximation of derivatives (Hildebrand 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.:

$$W_i(x,y) = N_i(x,y), \quad i = 1, 2, 3, \dots, N \quad (46)$$

to weight all terms except the advection term in Eq. (24), where W_i is the weighting function for nodal point i .

Two options of weighting are provided for the advection term: one is to set $W_i = N_i$ as in Eq. (46) and the other is to use upstream weighting functions. For a quadrilateral element, these functions are given by (Huyakorn 1976):

$$W_1^* = \frac{1}{16} \left[(1+\eta)(3\beta_2\eta - 3\beta_2 - 2) + 4 \right] \left[(1+\xi)(3\alpha_1\xi - 3\alpha_1 - 2) + 4 \right], \quad (47a)$$

$$W_2^* = \frac{1}{16} \left[(1+\eta)(3\beta_1\eta - 3\beta_1 - 2) + 4 \right] \left[(1+\xi)(-3\alpha_1\xi + 3\alpha_1 + 2) \right], \quad (47b)$$

$$W_3^* = \frac{1}{16} \left[(1+\eta)(-3\beta_1\eta + 3\beta_1 + 2) \right] \left[(1+\xi)(-3\alpha_2\xi + 3\alpha_2 + 2) \right], \quad (47c)$$

$$W_4^* = \frac{1}{16} \left[(1+\eta)(-3\beta_2\eta + 3\beta_2 + 2) \right] \left[(1+\xi)(3\alpha_2\xi - 3\alpha_2 - 2) + 4 \right], \quad (47d)$$

$$\alpha = \coth\left(\frac{uL}{2D}\right) - \left(\frac{2D}{uL}\right), \quad (48)$$

where α may denote $\alpha_1, \alpha_2, \beta_1$, or β_2 representing the weighting factor along the side as shown in Fig. 7; and u , L , and D are the corresponding velocity component, length, and dispersion component, respectively, along the corresponding side as shown in Fig. 7. It is seen that as α 's and β 's become equal to zero, the upstream weighting becomes the Galerkin weighting. The weighting functions for $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 1$ are plotted in Fig. 7. Values of α 's and β 's as defined in Eq. (48) are given in Fig. 8.

For a triangular element, upstream weighting functions have been derived elsewhere (Huyakorn 1976) and are written as:

$$W_1^* = L_1 - 3\alpha_3 L_2 L_1 + 3\alpha_2 L_3 L_1 \quad (49a)$$

$$W_2^* = L_2 - 3\alpha_1 L_3 L_2 + 3\alpha_3 L_1 L_2 \quad (49b)$$

$$W_3^* = L_3 + 3\alpha_1 L_2 L_3 - 3\alpha_2 L_1 L_3 \quad (49c)$$

where α_1, α_2 , and α_3 are the weighting factor associated with the element sides opposite to nodes 1, 2, and 3, respectively. For $\alpha_1 = -1$, $\alpha_2 = -1$, and $\alpha_3 = 1$, these weighting functions are plotted in Figure 9.

Step 5--Since $\hat{C}$ is only an approximate solution of Eq. (24), it will not satisfy the equation. Thus, the fifth step of FEM is to define a residual R_r by:

$$\begin{aligned} R_r = n_e R_d h \frac{\partial \hat{C}}{\partial t} + \nabla_H \cdot (V \hat{C}) - \nabla_H \cdot (n_e b \bar{D} \nabla_H \hat{C}) - \hat{M} + (\lambda + \alpha' \frac{\partial h}{\partial t}) b (n_e \hat{C} + \rho_b \hat{S}) \\ + n_e K_w \hat{C} + \rho_b K_b \hat{S} + \hat{Q}_U + \hat{Q}_L \end{aligned} \quad (50)$$

Figure 7. Plots of four upstream weighting functions for a quadrilateral element with $\alpha_1 = \beta_1 = \alpha_2 = \beta_2 = 1.0$.

Figure 8. Optimum weighting factors as functions of Peclet number.

where

$$\hat{S} = K_d \hat{C} \quad \text{for linear isotherm} \quad (51a)$$

$$\hat{S} = K_f \hat{C}^n \quad \text{for Freundlich isotherm} \quad (51b)$$

$$\hat{S} = \frac{K_f s_m \hat{C}}{1 + K_L \hat{C}} \quad \text{for Langmuir isotherm} \quad (51c)$$

and

$$\hat{M} = [(1 + \gamma_M)M + (1 - \gamma_M)QC]/2 \quad (52)$$

in which Q is the rate of water taken out of the region of interest, and $\gamma_M = 1.0$ for source and $\gamma_M = -1.0$ for sink. $\hat{Q}_U$ and $\hat{Q}_L$ are similarly defined as functions of $\hat{C}$.

Step 6—According to the principle of FEM, we choose the coefficients $\hat{C}_s$ in Eq. (45) such that the residual in Eq. (50) weighted by the set of weighting functions is zero. In other words, the sixth step of FEM applied to this problem is

$$\begin{aligned} \int_R W_i R_r dR = \int_R N_i \left\{ n_e R_d b \frac{\partial \hat{C}}{\partial t} - \nabla_H (n_e b \bar{D} \nabla_H \hat{C}) - \hat{M} + (\lambda + \alpha' \frac{\partial h}{\partial t}) b (n_e \hat{C} + \rho_b \hat{S}) \right. \\ \left. + n_e K_w b \hat{C} + \rho_b K_b b \hat{S} + \hat{Q}_U + \hat{Q}_L \right\} dR + \int_R W_i \nabla_H (V \hat{C}) dR \end{aligned} \quad (53)$$

Figure 9. Plots of three upstream weighting functions for a triangular element with $\alpha_1 = -1.0$, $\alpha_2 = -1.0$ and $\alpha_3 = 1.0$.

Step 7—The seventh step of FEM is to derive a system of ordinary differential equations based on Eq. (53). This can be done rather easily. Substituting Eqs. (45), (46), and (47) for quadrilateral elements or (49) for triangular elements into Eq. (53), and interchanging the order of integration and summation, we have

$$[A]\left\{\frac{dC}{dt}\right\} + [B]\{C\} = \{D\} + \{Q\} \quad (54)$$

where $\{C\}$ is the column vector representing the value of C at all nodes, and the matrix $[A]$ is given by

$$A_{ij} = \sum_{e \in M_i} \int_{R_e} N_\alpha^e n_e R_e b N_\beta^e dR \quad (55)$$

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 the column vectors $\{D\}$ and $\{Q\}$ are given by

$$\begin{aligned} B_{ij} = & \sum_{e \in M_i} \int_{R_e} \{(\nabla_H N_\alpha^e) n_e b \bar{D} (\nabla_H N_\beta^e) - (\nabla_H W_\alpha^e) \bar{V} N_\beta^e\} dR + \\ & \sum_{e \in M_i} \int_{R_e} N_\alpha^e \left[(\lambda + \alpha' \frac{\partial h}{\partial t}) b [n_e + \rho_b (\frac{dS}{dC})^*] + n_e K_w b + \rho_b K_s b (\frac{dS}{dC})^* \right. \\ & \left. + (1 + \gamma_U) K_U (h - h_U)/2 + (1 + \gamma_L) K_L (h - h_L)/2 - (1 - \gamma_S) Q/2 \right] N_\beta^e dR \end{aligned} \quad (56)$$

$$\begin{aligned} D_i = & \sum_{e \in M_i} \int_{R_e} N_\alpha^e \left[(\lambda + \alpha' \frac{\partial h}{\partial t} + K_w) b \rho_b [S^* - (\frac{dS}{dC})^* C^*] + (1 + \gamma_S) M/2 \right. \\ & \left. - (1 - \gamma_U) K_U (h - h_U) C_U/2 - (1 - \gamma_L) K_L (h - h_L) C_L/2 \right] N_\beta^e dR \end{aligned} \quad (57)$$

and

$$Q_i = \sum_{e \in N_i} \int_{B_e} \{N_\alpha^e \bar{r} \cdot (n_e b \bar{D} \nabla_H C) - W_\alpha^e \bar{r} \cdot \bar{V} C\} dB \quad (58)$$

where N_i is the set of all global boundary segments; N_{α} is the set of boundary segments that have a local node α coinciding with the global node i . In Eq. (56), $(dS/dC)^*$ is the slope of the sorption curve evaluated with values of dissolved concentration from the previous iteration, and in Eq. (57) S^* and C^* are the values of S and C at the previous iteration. These result from the linearization of the nonlinear isotherm model.

The reduction of the partial differential equation (Eq. (24)) to the system of ordinary differential equations (Eq. (54)) boils down to the evaluation of integrals on the right-hand side of equations Eqs. (55) through (57) for every element e to yield the element matrices $[A^e]$ and $[B^e]$ and the element column vector $\{D^e\}$ as :

$$A_{\alpha\beta}^e = \int_{R_e} N_{\alpha}^e b n_e R_d N_{\beta}^e dR \quad (59)$$

$$\begin{aligned} B_{\alpha\beta}^e = & \int_{R_e} \{ (\nabla_H N_{\alpha}^e) n_e b \bar{D} (\nabla_H N_{\beta}^e) - (\nabla_H W_{\alpha}^e) \bar{V} N_{\beta}^e \} dR + \\ & \int_{R_e} N_{\alpha}^e \left\{ \left(\lambda + \alpha' \frac{\partial h}{\partial t} \right) b [n_e + \rho_b \left(\frac{dS}{dC} \right)^*] + n_e K_w b + \rho_b K_s b \left(\frac{dS}{dC} \right)^* \right. \\ & \left. + (1 + \gamma_U) K_U (h - h_U)/2 + (1 + \gamma_L) K_L (h - h_L)/2 - (1 - \gamma_S) Q/2 \right\} N_{\beta}^e dR \end{aligned} \quad (60)$$

and

$$\begin{aligned} D_{\alpha}^e = & \int_{R_e} N_{\alpha}^e \left\{ \left(\lambda + \alpha' \frac{\partial h}{\partial t} + K_s \right) b \rho_b [S^* - \left(\frac{dS}{dC} \right)^* C^*] + (1 + \gamma_S) M/2 \right. \\ & \left. - (1 - \gamma_U) k_U (h - h_U) C_U/2 - (1 - \gamma_L) K_L (h - h_L) C_L/2 \right\} N_{\beta}^e dR \end{aligned} \quad (61)$$

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. (59) through (61) are computed by Gaussian quadrature (Conte 1965) because it is not easy to solve Eq. (41) 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. (43a) through (43f). This fact and the following integration identity,

$$\int_R L_1^m L_2^n L_3^k dR = 2A \frac{n!m!k!}{(n+m+k+2)!} \quad (62)$$

make the evaluation of Eqs. (59) through (61) 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.

III.2. Incorporation of Boundary Conditions

Step 8—The matrices $[A]$ and $[B]$ in Eq. (54) are singular. To make the problem uniquely defined, boundary conditions [Eqs. (32) and (38)] must be implemented. Hence, the eighth step of FEM is the incorporation of boundary conditions into the system of ordinary differential equations [Eq. (54)]. For the Dirichlet-type boundary condition, an identity equation is generated for each node involved and included in the matrices of Eq. (54). Computationally, this is done as sketched in Figure 10. If k -th variable is prescribed and denoted by C_k^* , then dC_k^*/dt is also known. The k -th column in both matrices $[A]$ and $[B]$ are stored, then the k -th row and k -th column in both $[A]$ and $[B]$ are set to zero and the diagonal entry in $[B]$ set equal to one. Finally, the stored column matrix of $[A]$ is multiplied by C_k^* and both products are subtracted from the right-hand side of Eq. (54). This procedure effectively replace the k -th equation by the prescribed values.

For the variable boundary condition, the flow-in and flow-out conditions must be treated separately. For the flow-in condition, we simply substitute Eq. (38b) into Eq. (58) to yield:

$$\begin{bmatrix} A_{1,1} & \dots & A_{1,k-1} & 0 & A_{1,k+1} & \dots & A_{1,n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ A_{k-1,1} & \dots & A_{k-1,k-1} & 0 & A_{k-1,k+1} & \dots & A_{k-1,n} \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ A_{k+1,1} & \dots & A_{k+1,k-1} & 0 & A_{k+1,k+1} & \dots & A_{k+1,n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ A_{n,1} & \dots & A_{n,k-1} & 0 & A_{n,k+1} & \dots & A_{n,n} \end{bmatrix} \begin{Bmatrix} dC_1/dt \\ \vdots \\ dC_{k-1}/dt \\ dC_k/dt \\ dC_{k+1}/dt \\ \vdots \\ dC_n/dt \end{Bmatrix} + \begin{bmatrix} B_{1,1} & \dots & B_{1,k-1} & 0 & B_{1,k+1} & \dots & B_{1,n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ B_{k-1,1} & \dots & B_{k-1,k-1} & 0 & B_{k-1,k+1} & \dots & B_{k-1,n} \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ B_{k+1,1} & \dots & B_{k+1,k-1} & 0 & B_{k+1,k+1} & \dots & B_{k+1,n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ B_{n,1} & \dots & B_{n,k-1} & 0 & B_{n,k+1} & \dots & B_{n,n} \end{bmatrix} \begin{Bmatrix} C_1 \\ \vdots \\ C_{k-1} \\ C_k \\ C_{k+1} \\ \vdots \\ C_n \end{Bmatrix} =$$

$$\begin{Bmatrix} D_1 \\ \vdots \\ D_{k-1} \\ 0 \\ D_{k+1} \\ \vdots \\ D_n \end{Bmatrix} + \begin{Bmatrix} Q_1 \\ \vdots \\ Q_{k-1} \\ 0 \\ Q_{k+1} \\ \vdots \\ Q_n \end{Bmatrix} - \begin{bmatrix} 0 & \dots & 0 & A_{1,k} & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & A_{k-1,k} & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & A_{k+1,k} & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & A_{n,k} & 0 & \dots & 0 \end{bmatrix} \begin{Bmatrix} 0 \\ \vdots \\ 0 \\ dC_k^*/dt \\ 0 \\ \vdots \\ 0 \end{Bmatrix} - \begin{bmatrix} 0 & \dots & 0 & B_{1,k} & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & B_{k-1,k} & 0 & \dots & 0 \\ 0 & \dots & 0 & -1 & 0 & \dots & 0 \\ 0 & \dots & 0 & B_{k+1,k} & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 0 \end{bmatrix} \begin{Bmatrix} 0 \\ \vdots \\ 0 \\ C_k^* \\ 0 \\ \vdots \\ 0 \end{Bmatrix}$$

Figure 10. Incorporation of Dirichlet boundary conditions.

$$Q_i = \sum_{\epsilon \in N_{in}} \int_{B_\epsilon} W_\alpha^\epsilon(\vec{n} \cdot \vec{V}) C_{in} dB \quad (63)$$

which is independent of C , 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 flow-in boundary-sides, the element column vector $\{Q_\alpha^\epsilon\}$ whose α -th entry is

$$Q_\alpha^\epsilon = \int_{B_\epsilon} W_\alpha^\epsilon(\vec{n} \cdot \vec{V}) C_{in} dB \quad (64)$$

For the flow-out condition, the substitution of Eq. (38a) into Eq. (58) results in

$$Q_i = - \sum_{\epsilon \in N_{out}} \int_{B_\epsilon} W_\alpha^\epsilon(\vec{n} \cdot \vec{V}) C dB \quad (65)$$

which is function of the unknown concentration. Substituting Eq. (45) into Eq. (65), we obtain the boundary condition-produced matrix $[Q]$, whose ij -th entry is

$$Q_{ij} = - \sum_{\epsilon \in N_{out}} \int_{B_\epsilon} W_\alpha^\epsilon(\vec{n} \cdot \vec{V}) N_\beta^\epsilon dB \quad (66)$$

Of course, this boundary matrix is obtained by assembling over all flow-out boundary sides, the boundary element matrix $[Q^\epsilon]$ whose $\alpha\beta$ -th entry is

$$Q_{\alpha\beta}^\epsilon = \int_{B_\epsilon} W_\alpha^\epsilon(\vec{n} \cdot \vec{V}) N_\beta^\epsilon dB \quad (67)$$

The boundary element matrix $[Q]$ is added to the matrix $[B]$ and the implementation of boundary conditions is completed.

After incorporation of the boundary conditions, Eq. (54) becomes:

$$[M] \left\{ \frac{dC}{dt} \right\} + [S][C] = \{Y\} \quad (68)$$

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

III.3. Temporal Integration

Step 9—The final step in finite element modeling is the reduction of Eq. (68) to a matrix equation by time integration and the solution of the resulting matrix equation.

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 transport model. In the first one, the central or Crank-Nicolson formulation may be written as:

$$[M](\{C\}_{t+\Delta t} - \{C\}_t)/\Delta t + \frac{1}{2}[S](\{C\}_{t+\Delta t} + \{C\}_t) = \{Y\} \quad (69)$$

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

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

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:

$$\left\{ \frac{2}{\Delta t} [M] + [S] \right\} \{C\}_{t+\Delta t/2} - \frac{2}{\Delta t} [M] \{C\}_t = \{Y\} \quad (71a)$$

and

$$\{C\}_{t+\Delta t} = 2\{C\}_{t+\Delta t/2} - \{C\}_t \quad (71b)$$

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. (55). 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 inconsistent mass matrix (mass matrix with lumping). Therefore, options are provided for the lumping of matrix $[A]$. More explicitly, $[A]$ will be lumped according to:

$$A_{ij} = \sum_{e \in M_i} \sum_{\beta=1}^4 \int_{R_e} N_{\alpha}^e N_{\beta}^e R_{\alpha\beta} dR \quad \text{if } j = i \quad (72a)$$

and

$$A_{ij} = 0 \quad \text{if } j \neq i \quad (72b)$$

With the mass matrix $[A]$ lumped according to Eqs. (72a) and (72b), 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. (68)] to a set of algebraic equations.

Thus, depending on the weighting function selected for the advection term, the method of time marching, and the treatment of mass matrix, twelve alternative numerical schemes are made available in this report (Table 1). For example, scheme 1 uses the central difference time marching and Galerkin weighting for the reduction term with no lumping on the mass matrix.

Table 1. Optional finite element schemes							
Schemes	Time-marching methods			Weighting function		Mass matrix lumping	
	Crank-Nicolson	Backward	Mid-difference	Galerkin	Upstream	No	Yes
1	X			X		X	
2		X		X		X	
3	X				X	X	
4	X			X			X
5	X				X		X
6		X			X	X	
7		X		X			X
8		X			X		X
9			X	X		X	
10			X		X	X	
11			X	X			X
12			X		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:

$$[T]\{C\} = \{R\} \quad (73)$$

where $[T]$ is the coefficient matrix and $\{R\}$ is the known load vector. The matrix equation, Eq. (73), 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 this problem.

For nonlinear sorption, Eq. (73) is a nonlinear matrix equation since both the coefficient matrix $[T]$ and the load vector $\{R\}$ are functions of the unknown $\{C\}$. Its solution requires some type of iterative procedure. The approach is to make an initial estimate of the unknown $\{C\}$, calculate the coefficient matrix $[T]$ and load vector $\{R\}$ using the estimate, and solve the equation with either the direct elimination or pointwise iteration for the unknown $\{C\}$. The solution or the combination of the solution and its previous estimate becomes a new estimate. This iterative procedure is stopped when the computed value of the unknown differs from the estimate within a tolerable allowance.

Three options are provided here to obtain the estimate: exact relaxation, under-relaxation, and over-relaxation. Normally, the exact relaxation, in which the solution becomes the new estimate, is chosen. However, when the sequence of solutions exhibit oscillatory behavior, the under-relaxation, in which the average of the solution and its previous estimate becomes the new estimate, is used to dampen the fluctuations. When the successive solutions behave monotonically but converge too slowly, over-relaxation may be applied to speed up the rate of convergence.

III.4. Mass Balance Computation

One of the most important aspects in numerical modeling of contaminant transport in aquifers is to check the mass balance over the whole region for every time step. 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. (24):

$$F_V = \int_R \left\{ n_e R_d b \frac{\partial c}{\partial t} + (\lambda + \alpha' \frac{\partial h}{\partial t}) b (n_e C + \rho_b S) + n_e K_w b C + \rho_b K_s b S + Q_U + Q_L - M \right\} dR \quad (74a)$$

and

$$F_B = - \int_B F_n dB \quad (74b)$$

where F_V represents the net rate of mass increase in the region, F_B is the net mass flux rate through the entire boundary into the region, and F_n is the outward normal flux. In fac., F_n is defined by

$$F_n = \vec{n} \cdot [\vec{V}C - n_e b \vec{D} \cdot \nabla_H C] \quad (75)$$

Having obtained the concentration distribution C by solving Eq. (24), one could integrate Eqs. (74a) and (74b) independently. If the solution for C is free of error, one would expect the equality of two integrals. In the present report, the integrals in Eq. (74a) is broken into nine parts:

$$F_W = \int_R n_e b \frac{\partial C}{\partial t} dR \quad (76a)$$

$$F_A = \int_R \rho_b b \frac{\partial S}{\partial t} dR \quad (76b)$$

$$F_L = \int_R \lambda b (n_e C + \rho_b S) dR \quad (76c)$$

$$F_D = \int_R \alpha' \frac{\partial h}{\partial t} (n_e C + \rho_b S) dR \quad (76d)$$

$$F_{KW} = \int_R n_e K_w b C dR \quad (76e)$$

$$F_{KS} = \int_R \rho_b K_s b S dR \quad (76f)$$

$$F_{UA} = \int_R Q_U dR \quad (76g)$$

$$F_{LA} = \int_R Q_L dR \quad (76h)$$

and

$$F_S = \int_R M dR \quad (76i)$$

where F_W , F_A , F_L , F_D , F_{KW} , F_{KS} , F_{UA} , F_{LA} and F_S represent the mass rate of changes due to water-dissolved concentration, adsorbed concentration, radioactive decay, skeleton compression, decomposition through dissolved phase, decomposition through adsorbed phase, leakage through upper aquifer, leakage through lower aquifer, and source/sinks, respectively. Similarly, the integral in Eq. (74b) is broken into four parts:

$$F_D = - \int_{B_D} F_n dB \quad (77a)$$

$$F_{FI} = - \int_{B_{FI}} F_n dB \quad (77b)$$

$$F_{FO} = - \int_{B_{FO}} F_n dB \quad (77c)$$

and

$$F_U = - \int_{B_U} F_n dB \quad (77d)$$

in which F_D , F_{FI} , F_{FO} and F_U represent fluxes into the region through a Dirichlet boundary B_D , a flow-in boundary B_{FI} , a flow-out boundary B_{FO} , and the unspecified boundary $B_U = B - B_D - B_{FI} - B_{FO}$.

For an exact solution, the net flux across the whole boundary, F_B , should equal to the total mass increase rate, F_V . In addition, F_U should theoretically be equal to zero. However, in any practical simulation, F_V will not be equal to F_B , and F_U 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 FEMA Program

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

Governing Equation:

$$\begin{aligned}
 n_e R_d b \frac{\partial C}{\partial t} + \nabla_H \cdot (\vec{V} C) = \nabla_H \cdot (n_e b \bar{D} \nabla_H C) - (\lambda + \alpha' \frac{\partial h}{\partial t}) b (n_e C + \rho_b S) - n_e K_w b C \\
 - \rho_b K_s b S - Q_U - Q_L + M_d + \sum_{p=1} M_p \delta(x - x_p) \delta(y - y_p)
 \end{aligned} \tag{78}$$

where M_d is the distributed source/sink of mass rate per unit horizontal area ($M/T/L^2$), and M_p is the strength of a point source/sink (M/T) located at point (x_p, y_p) .

Initial Condition:

$$C(x, y, 0) = C_i(x, y) \text{ in } R \tag{79}$$

where (x, y) is the prescribed initial value function, which may also be obtained by solving the steady-state version of Eq. (78) under time-invariant boundary conditions.

Boundary Conditions:

$$C = C_D(x_b, y_b, t) \text{ on } B_D \quad (80a)$$

and

$$-\vec{n} \cdot (n_e b \vec{D} \nabla_H C) = 0 \text{ on } B_F \text{ if } \vec{n} \cdot \vec{V} \geq 0 \quad (80b)$$

and

$$\vec{n} \cdot (\vec{V} C - n_e b \vec{D} \nabla_H C) = (\vec{n} \cdot \vec{V}) C_{in}(x_b, y_b, t) \text{ on } B_F \text{ if } \vec{n} \cdot \vec{V} < 0 \quad (80c)$$

where $C_D(x_b, y_b, t)$ is a prescribed concentration as a function of time t and spatial coordinates (x_b, y_b) on the Dirichlet boundary B_D , and $C_{in}(x_b, y_b, t)$ is the given concentration of incoming fluid through the flow-through boundary B_F , also as function of time t and spatial coordinates (x_b, y_b) on B_F .

IV.2 Structure and Function of FEMA Program

To numerically solve the mathematical model defined by Eqs. (78) through (80), the FEMA code consists of a MAIN program and 25 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 GM2D. Figure 11 shows the structure of the program. The functions of these subroutines are described below.

Subroutine GM2D

The subroutine GM2D 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 dynamic 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.

Figure 11. Program structure of FEMA.

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 $NTI > 0$). GM2D calls subroutine DATAIN to input data; subroutine INTERP to compute sources/sinks and boundary conditions at current time step by interpolating the tabular sources/sinks, Dirichlet head,

and fluxes; subroutine AFABTA to calculate the optimum weighting factor based on the Darcy's velocity, dispersivities, and element geometry; subroutine FLUX 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 SOLVE to solve the resulting matrix equations with direct elimination methods or SUBROUTINE PISS to solve the matrix equation with pointwise iteration solution strategies; subroutine SFLOW to calculate flux through all types of boundaries and waste accumulated in the media and waste loss through decay and decomposition; subroutine PRINT to print out the results; subroutine STORE to store the concentration for plotting; and subroutine LNDGEN to automatically generate the correspondence between variable or equation numbers and node numbers for the pointwise iteration solution option. These calling sequences are shown in Fig. 11.

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 number 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 stored in array variable ISB. The information is returned to the subroutine DATAIN for subsequent use elsewhere.

Subroutine READR

This subroutine is also called by the subroutine DATAIN to generate integer variables Data Sets 9, 12(c), 12(f), 13(d), (15d), and (16d). As with subroutine READR, an automatic generation mechanism for regularly patterned data is built into this subroutine.

Subroutine READN

This subroutine is also called by the subroutine DATAIN to generate integer variables Data Sets 9, 12(c), 12(f), 13(d), (15d), and (16d). As with subroutine READR, an automatic generation mechanism for regularly patterned data is built into this subroutine.

Subroutine LNDGEN

This subroutine is called by GM2D to establish the correspondence between node numbers and equation numbers. For the n -th equation, the concentration at the n -th node will be considered as the n -th unknown. There will be JBAND-1 nodes surrounding this n -th node (if it is an interior node). JRE of these nodes will have the node number smaller than n and the other greater than n . The JRE smaller nodes will be designated as the first, second, third, . . . , and JRE-th, respectively, of the n -th equation and the n -th node will be designated as the JDIAG-th node of the n -th equation. Hence, for each equation, an array is used to store these nodes for later use with iteration solution strategies.

Subroutine INTERP

This subroutine is called by the controlling subroutine GM2D to compute the concentrations and hydraulic head in both the upper and lower leaking aquifers, and the sources/sinks, Dirichlet concentration, and prescribed concentration of incoming water if required. The computation is based on the linear interpolation of the input values of these quantities.

Subroutine AFABTA

This routine is called by GM2D to compute weighting factors on four sides of all quadrilateral elements and on three sides of all triangular elements. The basis of computation is Eq. (48). However, if optimum weighting factors are not needed, they will be set to 1 or -1 according to the flow directions along the element sides.

Subroutine FLUX

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 SOLVE to yield the solution if direct elimination methods are employed. The computed field of the vertically integrated flux is then returned to GM2D through the argument.

Subroutine Q4D and Q3D

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

$$QQ(I,J) = \int_{R_e} N_i^* N_j^* dR \quad (81a)$$

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

$$RQ(I) = + \int_{R_e} \vec{e}_x \cdot [W_i^* \vec{\nabla} N_j^* - N_i^* b n_e \vec{D} \cdot (\nabla_H N_j^*)] C_j dR \quad (81b)$$

or

$$RQ(I) = + \int_R \vec{e}_y \cdot [W_i^* \vec{\nabla} N_j^* - N_i^* b n_e \vec{D} \cdot (\nabla_H N_j^*)] C_j dR \quad (81c)$$

where C_j is the concentration at nodal point j , and $\vec{e}_x$ and $\vec{e}_y$ are the unit vectors along the x - and y - coordinates, respectively. Q4D calculates the integrations in Eqs. (81a) and (81b) or (81c) for quadrilateral elements and Q3D evaluates those integrations for triangular elements.

Subroutine SHAPE

This subroutine is called by Q4D, Q4R, Q4SP, and Q4 to evaluate the value of the base and weighting functions and their derivatives with respect to the local coordinates at a Gaussian point.

Subroutine SCLVE

This subroutine is called by FLUX and GM2D to solve for the matrix equation of the type

$$[C]\{x\} = \{y\} \quad (82)$$

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 uses a standard Gaussian elimination procedure.

Subroutine ASEMBL

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

Subroutines Q4 and Q3

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

$$QA(I,J) = \int_R N_i^* n_e b R_d N_j^* dR \quad (83a)$$

and

$$\begin{aligned} QB(I,J) = & \int_R \{ (\nabla_H N_i^*) \cdot n_e b R_d \bar{D} \cdot (\nabla_H N_j^*) - (\nabla_H W_i^*) \cdot \bar{V} N_j^* \\ & + N_i^* [(\lambda + \alpha' \frac{\partial h}{\partial t}) b [n_e + \rho_b (\frac{dS}{dC})^*] + n_e K_w b + \rho_b K_s b (\frac{dS}{dC})^* \\ & + (1 + \gamma_U) K_U (h - h_U)/2 + (1 + \gamma_L) K_L (h - H_L)/2 - (1 - \gamma_S) Q/2] N_j^* \} dR \end{aligned} \quad (83b)$$

Q4 computes Eqs. (83a) and (83b) for quadrilateral elements, while Q3 computes those equations for triangular elements. They also calculate the load vector given by:

$$\begin{aligned} RQ(I) = & \int_R N_i^* [(\lambda + \alpha' \frac{\partial h}{\partial t} + K_s) b \rho_b [S^* - (\frac{dS}{dC})^*] C^* + (1 + \gamma_S) M/2 \\ & - (1 - \gamma_U) K_U (h - h_U) C_U/2 - (1 - \gamma_L) K_L (h - h_L) C_L/2] N_j^* dR \end{aligned} \quad (84)$$

In Eqs. (83) and (84), Q is the distributed sink of water and M is the distributed source of material.

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 $\{Y\}$ in Eq. (68) if the flow is directed into the region. If the flow is directed out from the region, the boundary-condition-produced matrix is added to the coefficient matrix. This subroutine calls Q4SP and Q3SP to evaluate either the load vector or the matrix of the boundary segments.

Subroutines Q4SP and Q3SP

Subroutines Q4SP and Q3SP are called by BC to compute boundary matrix

$$BQ(I,J) = \int_{R_j} W_i^*(\vec{n} \cdot \vec{V}) N_j^* d_B \quad (85)$$

or the boundary load vector

$$RQ(I) = - \int_{B_j} W_i^*(\vec{n} \cdot \vec{V}) C_n dB \quad (86)$$

Q4SP computes Eqs. (85) and (86) for quadrilateral elements and Q3SP evaluates those for triangular elements.

Subroutine PISS

This subroutine is called by GM2D to solve the matrix equation with pointwise iteration solution strategies.

Subroutine PRINT

This subroutine is used to line-print the variables. These include the fluxes through variable boundary segments, the concentration and vertically integrated flux components.

Subroutine STORE

This subroutine is used to store the variables on logical unit 2. It is intended for possible plotting purposes. The information stored include problem title, geometry, and concentration and vertically integrated material flux for each desired time step.

Subroutine SFLOW

This subroutine is used to compute the material flux rates through various types of boundaries and the increasing rate of materials in the region of interest. FRATE(5) stores the flux rate through the whole boundary enclosing the region of interest. It is given by:

$$\text{FRATE}(5) = \int_B (F_x n_x + F_y n_y) dB \quad (87)$$

in which B is the global boundary of the region of interest, F_x and F_y are the vertically integrated flux components on the boundary in the x - and y -direction, respectively, and n_x and n_y are the directional cosines of the outward unit vector normal to the boundary B . FRATE(1), FRATE(2) and FRATE(3) store the flux rates through Dirichlet boundary B_D , flow-in boundary B_{FI} , and flow-out boundary B_{FO} , respectively, given by:

$$\text{FRATE}(1) = \int_{B_D} (F_x n_x + F_y n_y) dB \quad (88)$$

$$\text{FRATE}(2) = \int_{B_n} (F_x n_x + F_y n_y) dB \quad (89)$$

and

$$\text{FRATE}(3) = \int_{B_n} (F_x n_x + F_y n_y) dB \quad (90)$$

FRATE(4), which is related to the numerical loss, is given by

$$\text{FRATE}(4) = \text{FRATE}(5) - [\text{FRATE}(1) + \text{FRATE}(2) + \text{FRATE}(3)] \quad (91)$$

FRATE(6) is used to store an increased rate of material dissolved in water and FRATE(7) to store material absorbed in solid matrix, respectively:

$$\text{FRATE}(6) = \int_R n_e b \frac{\partial C}{\partial t} dR \quad (92)$$

and

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

FRATE(8) and FRATE(9) are used to store material losses through radioactive decay and skeleton compression, respectively:

$$\text{FRATE}(8) = \int_R \lambda (n_e C + \rho_b S) b dR \quad (94)$$

and

$$\text{FRATE}(9) = \int_R \alpha' \frac{\partial h}{\partial t} (n_e C + \rho_b S) b dR \quad (95)$$

FRATE(10) and FRATE(11) store the material decomposition rate through dissolved and absorbed phases, respectively:

$$\text{FRATE}(10) = \int_R b n_e K_w C dR \quad (96)$$

and

$$\text{FRATE}(11) = \int_R b \rho_b K_r S dR \quad (97)$$

FRATE(12) and FRATE(13) store the material flux rate going out of the region through the upper and lower confining leaky beds, respectively:

$$\text{FRATE}(12) = - \int_R K_U (h_U - h) [(1 + \gamma_U) C_U + (1 - \gamma_U) C] / 2 dR \quad (98)$$

and

$$\text{FRATE}(13) = - \int_R K_L (H_L - h) [(1 + \gamma_L) C_L + (1 - \gamma_L) C] / 2 dR \quad (99)$$

Finally, FRATE(14) is used to store the source/sink rate

$$\begin{aligned} \text{FRATE}(14) = & - \int_p [(1 + \gamma_s) M + (1 - \gamma_s) Q C] / 2 dR \\ & - \sum_p [(1 + \gamma_s) M_p + (1 - \gamma_s) Q_p C_p] / 2 \end{aligned} \quad (100)$$

where C_p is the concentration of the point sink at (x_p, y_p) .

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

$$\{\text{FRATE}(6) + \text{FRATE}(7)\} = -\text{FRATE}(5) - \sum_{I=1}^{14} \text{FRATE}(I) \quad (101)$$

and $\text{FRATE}(4)$ should be equal to zero.

Equation (101) simply states that the negative rate of mass flux going out of the region through the entire boundary ($\text{FRATE}(5)$), due to decay ($\text{FRATE}(8)$), due to skeleton compression ($\text{FRATE}(9)$), because of bio-decomposition ($\text{FRATE}(10)$ and $\text{FRATE}(11)$), because of leakage ($\text{FRATE}(12)$ and $\text{FRATE}(13)$), and due to source/sinks ($\text{FRATE}(14)$), is equal to the rate of mass accumulated in the region ($\text{FRATE}(6)$ and $\text{FRATE}(7)$).

Subroutines Q4R and Q3R

These subroutines are used to compute the contribution to $\text{FRATE}(6)$ through $\text{FRATE}(14)$ by quadrilateral and triangular elements, respectively:

$$QRM = \int_R n_e b C dR \quad (102)$$

$$QDM = \int_R \rho_b b S dR \quad (103)$$

$$QDHM = \int_R \frac{\partial h}{\partial t} (n_e b C + \rho_b b S) dR \quad (104)$$

$$UPLEKM = - \int_R K_U (h_U - h) [(1 + \gamma_U) C_U + (1 - \gamma_U) C] / 2 dR \quad (105)$$

$$DNLEKM = - \int_{R_e} K_L(h_L - h)[(1 + \gamma_L)C_L + (1 - \gamma_L)C]2dR \quad (106)$$

and

$$SOSM = - \int_{R_e} [(1 + \gamma_S)M + (1 - \gamma_S)QC]2dR \quad (107)$$

where QRM is the amount of dissolved material in element e at present time, QDM is the amount of adsorbed material in element e at present time, $QDHM$ is the rate of material stored in the skeleton of element e due to compression, $UPLEKM$ is the rate of material lost to the upper aquifer through element e , $DNLEKM$ is the rate of material lost to the lower aquifer through element e , and $SOSM$ is the rate of material added to or withdrawn from element e artificially.

The computation of the above integration is straightforward.

V. VERIFICATIONS AND DEMONSTRATION APPLICATIONS

Five examples are presented in this section, two of which are used to verify the accuracy of the computer code FEMA. One is used to show that for larger field problems the alternative successive iteration methods of solving the matrix equation save both CPU memory and CPU time. The final two are used to demonstrate how FEMA can be applied to heterogeneous and anisotropic media under transient boundary conditions and with time-dependent source/sinks for both confined and unconfined aquifers.

V.1 Example 1: One-dimensional Problem

Consider a confined aquifer with a thickness of 30 m. Assume that the aquifer is filled with a porous medium which has a compressibility of the skeleton, $\alpha' = 0.0$ and longitudinal dispersivity values, $a_L = 10.0$ or 0.5 or 0.01 m. The effective porosity of the aquifer is $n_e = 0.25$, its bulk density is $\rho_b = 1,300 \text{ Kg/m}^3$, and its tortuosity is $\tau = 1.0$. It is further assumed that a chemical is transported through this aquifer by the carrier water having a velocity $V = 53 \text{ m/year}$. We will also assume that the molecular diffusion is negligible and the chemical has a distribution coefficient $K_d = 10^{-2} \text{ m}^3/\text{Kg}$. Initially, the aquifer is clean, that is the initial concentration is zero everywhere in the aquifer. If a constant concentration C_0 is maintained at $x = 0$, what will be the concentration distribution along the x-axis at any time?

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

Governing Equation

$$n_e R_d b \frac{\partial C}{\partial t} + \frac{\partial (V n_e b C)}{\partial x} = \frac{\partial}{\partial x} (n_e b D \frac{\partial C}{\partial x}) \quad (108)$$

Initial Conditions

$$C=0 \text{ at } t=0 \text{ and } 0 \leq x < \infty \quad (109)$$

Boundary conditions

$$C = C_o \text{ at } t > 0 \text{ and } x = 0 \quad (110)$$

and

$$-n_e b D \frac{\partial C}{\partial x} = 0 \text{ at } t > 0 \text{ and } x = \infty \quad (111)$$

Equations (109) through (111) are in fact the special case of Eqs. (77) through (80), respectively. It can be verified that the solution of Eq. (108) subject to initial and boundary conditions, Eqs. (109) through (111), is given by

$$c/c_o = \frac{1}{2} \left\{ \operatorname{erfc} \left(\frac{x - V_o t}{\sqrt{4D_o t}} \right) + \exp \left(\frac{V_o x}{D} \right) \operatorname{erfc} \left(\frac{x + V_o t}{\sqrt{4D_o t}} \right) \right\} \quad (112)$$

where $V_o = V/R_d$ and $D_o = D/R_d$ are the retarded velocity and dispersion coefficient, respectively.

To compare the FEMA calculations with the result obtained from Eq. (112), 4 year real-time simulation was made. For this time span, a length of 50 m can be considered infinite. To start the simulation, the region is divided into 100 elements of size 0.5 m $\times$ 0.5 m and 202 nodes. To identify three different cases associated with three different longitudinal dispersivities ($a_L = 10.0$,

0.5, and 0.01 m, respectively), we define the mesh Peclet number and Courant number as $P_e = V \cdot \Delta x / D$ and $C_n = V \cdot \Delta t / \Delta x$. A constant $C_n = 0.4$ (i.e. a constant time step of 0.2 year) is used throughout the simulation. Three cases with $P_e = 0.05$, 1.0, and 50.0, respectively are solved. They represent cases of highly dispersion dominant, dispersion-advection equally dominant, and highly advection dominant transport, respectively. Both direct elimination solution and three successive iteration methods are used to solve the resulting matrix equation to demonstrate the numerical performance and computational efficiency of various solution strategies. The three iteration methods are the successive under-relaxation (SUR), Gauss-Seidel (G-S), and successive over-relaxation (SOR) techniques, respectively.

Figure 12 (a-d) show the concentration profiles at time $t = 4.0$ years. For $P_e = 0.05$, all three iteration methods produce convergent solutions that are almost identical to the direct elimination solution. Both the direct elimination and three iteration solutions are very accurate compared to the analytical result given by Eq. (112). For $P_e = 1.0$ and 50.0, the SOR leads to divergent simulation, but the G-S and SUR still generate convergent computations that are nearly identical to the direct elimination solution. While for $P_e = 1.0$, all solutions are very close to the analytical result, for $P_e = 50.0$, the solutions are not as accurate as desired. To obtain more accurate results, one has to reduce both the mesh size and time step (Huyakorn and Nijkuha 1979). Fig. 12(d) shows the concentration profile resulting from an element size that is reduced to one fifth the size used in obtaining solutions for Figures 12(a-c). With this element size, the mesh Peclet number becomes $P_e = 10.0$. To keep the same Courant number, the time step becomes 0.04 years. It is seen that all solutions are reasonably accurate. To test if further increase in the Peclet number will render G-S and SUR with divergent simulations, the case of pure advection, i.e. $P_e = \infty$, is simulated. It is found that the G-S and SUR still give convergent computations. Thus, we can reasonably conclude that the G-S and SUR iterations result in convergent calculations for all Peclet numbers. If pointwise iteration methods have to be used to solve the matrix equation to reduce CPU memory, one should use the G-S or SUR rather than the more popular SOR iteration scheme.

Next we examine the efficiency of direct elimination solution and pointwise solutions in terms of CPU time. Table 2 shows the comparison in terms of CPU time for this simple one-dimensional problem. It is seen that for all Peclet numbers, any of the three iteration schemes takes more CPU time than the direct elimination method. This can be expected because the band width (7) of the coefficient matrix for this one-dimensional problem is very small. Among the three pointwise iteration schemes, the Gauss-Seidel iteration is the most efficient one. This is in contrast to the finite difference approximations of the aquifer flows for which the SOR has been shown superior to the other two iterations (Smith 1965).

**Table 2. Comparison of CPU time in sec/time step
for example 1**

Peclet number	Δx	Δt	Direct solution	Pointwise iteration		
				SUR	G-S	SOR
0.05	0.5	2.20	0.420	2.069	1.186	1.128
1.00	0.5	0.20	0.433	0.921	0.470	*
50.0	0.5	0.20	0.624	0.899	0.700	*
∞	0.5	0.20	0.466	1.226	0.624	*
10.0	0.1	0.04	0.437	0.552	0.531	*

*Divergent computations.

V.2 Example 2: Two-dimensional Problem

As a second verification, we consider a confined aquifer that is 30 m thick, i.e. $b = 30$ m. It extends from $x = 0$ to $x = L$ and from $y = 0$ to $y = B$ (Fig. 13). The aquifer has the following

Figure 12. Concentration profiles at time = 4.0 for (a) $P_e = 0.05$, (b) $P_e = 1.0$, (c) $P_e = 50.0$, and (d) $P_e = 10.0$, respectively.

Figure 13. Sketch of the region of interest for example 2.

properties, compressibility of the skeleton $\alpha' = 0.0$; effective porosity $n_e = 0.25$; bulk density $\rho_b = 1,300.0 \text{ kg/m}^3$; tortuosity $\tau = 1.0$; molecular diffusion coefficient $a_m = 0.0$; longitudinal dispersivity $a_L = 10 \text{ or } 1.0 \text{ or } 0.1 \text{ m}$; and transverse dispersivity $a_T = 0.1 a_L$. The velocity field in the region is $V_x = 53 \text{ m/year}$ and $V_y = 0.0 \text{ m/year}$, respectively. A chemical with the following properties is transported through the aquifer: distribution coefficient $K_d = 0.002 \text{ m}^3/\text{Kg}$ and decay constant $\lambda = 2.476 \cdot 10^{-2}/\text{year}$, i.e., a half life of 28 years. Suppose the aquifer is initially clean and a constant concentration is maintained at $x = 0$ and $0 \leq y \leq B_s$, starting from $t = 0$, the question is what would be the concentration distribution of the chemical as functions of x , y , and t ?

This problem can be posed as a special case of Eqs. (77) through (80)

Governing Equation

$$n_e R_d b \frac{\partial C}{\partial t} + \nabla_H \cdot (\vec{V} n_e b C) = \nabla_H \cdot (n_e b \vec{D} \cdot \nabla_H C) - \lambda b n_e C \quad (113)$$

Initial Condition

$$C = 0 \text{ at } t = 0 \text{ } 0 \leq x \leq L \text{ and } 0 \leq y \leq B \quad (114)$$

Boundary conditions

$$C = C_0 \text{ at } t > 0 \text{ and } 0 \leq y \leq B, \quad (115a)$$

$$C = 0 \text{ at } t > 0 \text{ and } B < y \quad (115b)$$

$$D_{xx} \frac{\partial C}{\partial x} + D_{yy} \frac{\partial C}{\partial y} = 0 \text{ at } t < 0 \text{ and } x = L \quad (116)$$

$$V_y C - D_{yx} \frac{\partial C}{\partial x} - D_{yy} \frac{\partial C}{\partial y} = 0 \text{ at } t > 0 \text{ and } y = 0 \quad (117)$$

$$V_y C - D_{yx} \frac{\partial C}{\partial x} - D_{yy} \frac{\partial C}{\partial y} = 0 \text{ at } t > 0 \text{ and } y = B \quad (118)$$

Analytical solution for the system given by Eqs. (113) through (118) has been obtained elsewhere (Cleary 1978) when L tends to infinity.

For numerical computations, let us consider a four year real time simulation. With this real time, a length of $L = 50$ m can be considered infinite since the maximum dispersion coefficient $D_{xx} = a_L \cdot V_x = 10$ m/year. We will further restrict ourself to the region $B = 10$ m and the strip source $B_s = 5.0$ m. For FEMA computations, the region is divided into 10×50 square elements each having the size 1×1 m. This would result in 561 nodes. A constant time step size of $t = 0.4$ year is used. Figures 14(a-c) illustrate the concentration profile at time = 4 years along $y = 2.0$. The main conclusions drawn from the one-dimensional analysis carry over to this two-dimensional

Figure 14. Concentration profiles at time = 4.0 along $y = 2.5$ for (a) $Pe = 0.05$, (b) $Pe = 1.0$, and (c) $Pe = 50.0$, respectively.

example. These are: (1) for dispersion dominant transport ($Pe = 10.0$), all three iteration schemes produce convergent solutions (2) for dispersion-advection equally dominant transport ($Pe = 1.0$) or advection dominant transport ($Pe = 0.1$), the SOR leads to divergent computations while the SUR and G-S still generate convergent calculations, and (3) iteration solutions, if convergent, are almost identical to the direct solution.

For this small two-dimensional problem, the bandwidth of the coefficient matrix is 25 if the band matrix solver is used to solve the system of algebraic equations. On the other hand, the maximum number of nonzero elements in any row of the coefficient matrix is 9 for all two-dimensional problems when pointwise iteration methods are used. Hence, iteration methods, if convergent, will need only nine twenty-fifths of the CPU memory of that required by the direct elimination method to store the coefficient matrix. This represents more than a 100% saving in CPU memory. Table 3 compares the CPU time as required by the direct elimination and iteration solution methods, respectively. It is seen that the CPU time with iteration methods is comparable to that with the direct elimination method even for this small two dimensional problem where the direct elimination method is supposed to be working at its fullest advantage.

Table 3. Comparison of CPU time in sec/time step
for example 2

Peclet number	Δx (m)	Δt (yr)	Direct solution	Pointwise iterations		
				SUR	G-S	SOR
0.10	1.0	0.4	3.014	4.341	3.203	7.945
1.0	1.0	0.4	3.020	3.720	2.967	*
10.0	1.0	0.4	3.069	5.420	3.361	*

*Divergent computation.

V.3 Example 3: a two-dimensional problem

To demonstrate the comparison of the efficiency of the pointwise iteration and direct elimination solution methods presented in this report, we consider a moderate size two-dimensional problem. This example is assumed to be identical to example 2 except that the region of interest is 40×50 m rather than 10×50 m and the strip source $B_s = 20$ m rather than 5 m. All other medium parameters and chemical characteristics remain the same as those given in example 2. To maintain the same degree of accuracy, we used the same element size and time step size. This will require 40×50 square elements resulting in $41 \times 51 = 2091$ nodes for numerical simulations. This is by no means a very large problem, at most it can be considered a moderate size problem that one would encounter often in dealing with real-world problems. Numerical computations with FEMA code indicate that all major conclusions drawn from previous two examples are valid for this case.

With the 41×51 node discretization, the bandwidth of the coefficient matrix will be 85, thus it will require a CPU memory of $2091 \times 85 \times 8 = 1.422$ megabytes just to store the banded coefficient matrix. Had a full matrix solver rather than the band matrix solver been used, it would require a CPU memory of $2091 \times 2091 \times 8 = 34.978$ megabytes to store the full matrix which is beyond the capacity of any of today's mainframe computers. On the other hand, using the pointwise iteration solution algorithm contained in this report, we only need a CPU memory of $2091 \times 9 \times 8 = 0.151$ megabytes to store the compressed matrix. Thus, the CPU memory requirement with pointwise iteration methods represents approximately a nine-fold saving over that with the popular and widely used direct band matrix elimination method. Table 4 compares the CPU time required by the iteration and direct elimination methods for this moderate size problem. It is seen that the saving in CPU time with pointwise iterations ranges from more than 30 times for the case of $P_r = 0.1$ to more than 50 times for the case of $P_r = 1.0$. It would be very useful to employ other more efficient direct elimination methods such as sparse matrix solvers (Tewarson 1973) or the active column profile solution methods (Zienkiewicz 1977) to see if the saving with such solvers over the

**Table 4. Comparison of CPU time in sec/time step
for example 3**

Peclet number	Δx (m)	Δt (yr)	Direct solution	Pointwise iterations		
				SUR	G-S	SOR
0.1	1.0	0.4	317.790	30.890	25.885	41.220
1.0	1.0	0.4	318.223	27.569	22.825	*
10.0	1.0	0.4	232.850	32.291	23.092	*

*Divergent computations.

band matrix solver can be as substantial as the pointwise iteration techniques. Table 4 also indicates that while the G-S iteration is more efficient for the cases of dispersion dominant ($P_e = 0.1$) and dispersion-advection equally dominant ($P_e = 1.0$) transport, the SUR scheme is more efficient for the case of advection dominant ($P_e = 10$) transport. In all three examples, the SOR leads to convergent solutions only for the case of dispersion dominant transport. Hence the SOR scheme is not suitable for simulating the finite element approximations of groundwater contaminant transport. An orthogonal upstream finite element approximation has been developed that will result a matrix with convergent SOR calculations for all Peclet numbers though (Yeh 1984).

V.4 Example 4: A Confined Aquifer

To demonstrate the versatility of FEMA, consider a confined aquifer bounded below and above by two aquitards (Fig. 15a). Other stratigraphic formations are also present (Fig. 15a). The aquifer is bounded on the right side by a lake and on the left side by a groundwater divide (Fig. 15b). It is further assumed that the aquifer is bounded 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 curve 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 a pumping station is located at point P (Fig. 15b). Let's assume that beginning at a certain time, the lake is contaminated with a chemical of concentration C_0 along the upper half of the lake shore line CB (Fig. 15b). Our task is analyze the extent of chemical migration into the aquifer as function of time following this contamination.

Before the simulation of chemical migration, we must know the velocity and hydraulic head distribution in the aquifer. This can be obtained with the companion computer code FEWA (Yeh and Huff 1983). To employ FEWA for investigating the flow in aquifers, 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 5. The upper and lower confining beds will be given the following leakage coefficients: $K_U = 0.0 \text{ d}^{-1}$ to the left of the stream and $K_U = 4.0 \times 10^{-2} \text{ d}^{-1}$ to the right of the stream, and $K_L = 10^{-3} \text{ d}^{-1}$. We will assume that the hydraulic head in the aquifer below the lower aquitard is a constant $h_L = 52 \text{ m}$ in reference to the bottom of the lake, and the hydraulic head in the aquifer above the upper aquitard has a constant head of $h_U = 51 \text{ m}$. The water levels at the lake and in the stream are the same with a value of 52 m measured from the bottom of the lake. A vertically integrated flux $0.2 \text{ m}^2/\text{d}$ of water is flowing through AB into the aquifer from the adjacent aquifer (Fig. 15b). The amount of water withdrawn by pumping at pint P is assumed to be $100 \text{ m}^3/\text{d}$.

With the above problem description, we are now ready to encode the input data according to Appendix A of FEWA (Yeh and Huff 1983) for hydrodynamic simulations to obtain flow variables. 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. 16). Elements 38, 48, and 58 are composed of silt loam (material type 2). Elements

Figure 15. Problem sketch of example 4 for flow in a confined aquifer: (a) three-dimensional view and (b) plane view.

Table 5. Typical material properties for sand stone, silt loam, and clay

Properties	α'	β'	α_c	K_{xx}	K_{yy}	K_{zz}	S_y
Medium	(m^{-1})	(m^{-1})	(dimensionless)	(m/α)	(m/α)	(m/α)	(dimensionless)
ABCD-							
(EFGH + IJKL)	10^{-5}	3×10^{-6}	0.10	1.0	0.5	1.0	0.2
Sand stone							
EFGH	2×10^{-5}	3×10^{-6}	0.27	5.0×10^{-2}	2.5×10^{-2}	5.0×10^{-2}	0.27
Silt loam							
IJKL	4×10^{-5}	3×10^{-6}	0.446	8.0×10^{-4}	4.0×10^{-4}	8.0×10^{-4}	0.446
Clay

Figure 16. Finite element discretization for examples 4 and 5.

93, 103, and 113 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. Figures 17(a-b) show the head distribution and vertically integrated Darcy's flux when they have reached steady state conditions. These flow variables provide the advection and dispersion transport mechanism in FEMA.

For FEMA computations, in addition to the medium properties given in Table 5, the dispersivities, distribution coefficients, decay constant, and biodegradation rate constants must be specified. Table 6 lists some typical hypothetical values used for this demonstration example.

We will further assume that the aquifers below the lower aquitard and above the upper aquitard are not contaminated, i.e. $C_L = 0.0$ and $C_U = 0.0$. It is also assumed that the adjacent aquifer (Fig. 15b) has been contaminated with a concentration of $2C_0$.

For this demonstration, a total of 3,746.24 d (≈ 10 years) simulation time is made. The time step initially is assumed to be 1.0 d. It is increased by 20% in each subsequent time step with the restriction that no time step size is greater than 60 d. This results in a total number of time steps equal to 80.

With the above problem description, the boundary conditions for FEMA simulation can be specified as: (1) no fluxes across the boundaries AD and CD , (2) nodes 226, 227, 228, 229, 230, and 231 are specified as Dirichlet nodes with concentration C_0 , while nodes 221, 222, 223, 224, and 225 are specified as Dirichlet nodes with concentration equal to zero, and (3) boundary segment AB is specified having a variable condition (Fig. 16). Input data can now be prepared according to Appendix A. Figure 18 illustrates the distribution of chemical concentration in the aquifer as function of time.

V.5 Example 5: an Unconfined Aquifer Problem

This example is identical to example 4 except that: the aquifer is unconfined, the lake is not contaminated, and no pumping is to be made. 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

Figure 17. (a) Head distribution and (b) vertically integrated Darcy's fluxes at steady state.

Figure 18. Concentration distribution at time = (a) 386, (b) 1,830, and (c) 3,750 days, respectively.

Table 6. Dispersion and media-chemical interaction properties

Properties	a_L	a_T	a_m	τ	K_d	K_w	K_r	λ	ρ_b
Media	(m)	(m)	m ² /d		m ³ /Kg	1/d	1/d	1/d	kg/m ³
ABCD-(EFGH									
+IJKL)	20	5	10^{-4}	1.0	0.001	0.0	0.0	$6.78 \cdot 10^{-5}$	1,200
Sandstone									
EFGH									
Silt loam	10	2	10^{-4}	2.0	0.002	10^{-5}	0.0	$6.78 \cdot 10^{-5}$	1,400
IJKL									
Clay	5	2	10^{-4}	5.0	0.003	10^{-5}	0.0	$6.78 \cdot 10^{-5}$	1,600

lower aquifers, and changes in water levels of adjacent surface water bodies. The computation of hydrodynamics becomes nonlinear because the transmissivity tensor and the generalized storage coefficient both depend on the piezometric head. The piezometric head and vertically integrated Darcy's flux can be obtained by using the FEWA code (Yeh and Huff 1983). 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. 19). It is further assumed that all aquifer properties, source/sink values, and initial and boundary conditions for hydrodynamic computations remain the same as those in example 4. The absence of the upper confining aquitard can be specified by setting the leaking coefficient $K_U = 0.0$. The inclusion of infiltration or evapotranspiration is done by specifying a source/sink distributed through the area. Infiltration rates from the three geologic formations are assumed to be 2.0×10^{-3} , 1.0×10^{-3} , and 0.5×10^{-3} m/d, respectively. It is further assumed that the infiltrating precipitation has been contaminated with a concentration of C_0 . Our task is to

Figure 19. Problem sketch of example 5 for flow in an unconfined aquifer: (a) three-dimensional view and (b) plane view.

predict the extent of contamination of the aquifer resulting from the polluted precipitation and adjacent aquifer discussed earlier in example 4.

Using the same finite element discretization as that for example 4, we can prepare input data according to Appendix A of FEWA (Yeh and Huff 1984) for computing the flow field required by the FEMA model. Figure 20 illustrates the distribution of free water surface and the vertically integrated Darcy's flux when they have reached the steady state condition.

For FEMA simulations, the boundary conditions are specified as follows: (1) no fluxes across the boundaries AD and CD (Fig. 19). (2) node 11 is a Dirichlet node with zero concentration (Fig. 16), and (3) boundary segments AB and BC are treated as variable boundaries on which the total flux is equal to the incoming concentration times the velocity when the flow is directed into the region or the gradient of the concentration is specified as zero when the flow is directed out of the region. Input data can be prepared according to Appendix A. Figure 21 shows the concentration distributions of the chemicals at three time instants.

V.6 Summary and Conclusion

Three simple examples are used to verify the accuracy and code consistency of the model FEMA. Results show that the model is very accurate provided proper discretization is made of the region of interest. The relative merits of two solution methods for finite element matrix equations are compared in terms of CPU memory requirements and CPU time needs. For all practical two-dimensional problems, pointwise iteration solution strategies such as successive underrelaxation (SUR), Gauss-Seidel (D-S), and successive overrelaxation (SOR) iterations would require much less CPU storage than the direct elimination solution algorithm. For small problems, the direct elimination solution algorithm needs less CPU time than pointwise iteration solution strategies. However, for large problems, pointwise iteration solution strategies, if convergent, require much less CPU time than the direct elimination algorithm. Among three pointwise solution strategies, the SUR and G-S iteration methods produce convergent solutions under all Peclet numbers while the

Figure 20. (a) head distribution and (b) vertically integrated Darcy's fluxes at steady state.

Figure 21. Concentration distributions at time = (a) 386, (b) 1,830, and (c) 3,750 days, respectively.

popular SOR iteration generates divergent computations when the Peclet number is not small. Hence, the SOR method is not an appropriate iteration scheme for solving the matrix equations resulting from finite element approximations to aquifer contaminant transport equations. Instead, the under rated SUR and G-S iterations should be employed for modeling large problems in aquifer contaminant transport.

The flexibility and versatility of FEMA are demonstrated by two realistic examples. One is the investigation of contaminant transport in a heterogeneous, anisotropic confined aquifer resulting from transient contamination of a hydraulically connected lake. The other is the study of waste transport in a heterogeneous, anisotropic unconfined aquifer resulting from polluted precipitation and contaminated adjacent aquifers. In both cases, the migration of plumes is depicted and reasonable results are shown.

VI. REFERENCES

1. Bachmat, Y. J. Bredehoeft, B. Andrews, D. Holtz, and S. Sebastia, 1980. Groundwater Management: The use of Numerical Models. America Geophysical Union, Water Resources Monograph 5, AGU, Washington, D.C., 127 pp.
2. Browman, M. G., M. R. Patterson, and T. J. Sworski, 1982. Physicochemical Processes in the Environment: Background Information for the ORNL Unified Transport Model for Toxicants (UTM-TDX), ORNL-3854, Oak Ridge National Lab., Oak Ridge, TN.
3. Cleary, R. W. 1978. Final Report 208 Project, Nassan-Suffolk Regional Planning Board, Hauppauge, New York, N.Y.
4. Conte, S. D. 1965. Elementary Numerical Analysis. McGraw-Hill Book Company, New York, 278 pp.
5. Duguid, J. D. and M. Reeves, 1976. Material Transport in Porous Media: A Finite-Element Galerkin Model, ORNL-4928, Oak Ridge National Laboratory, Oak Ridge, TN 37830.
6. Forsythe, G. E. and W. R. Wasow, 1960. Finite-Difference Methods for Partial Differential Equations, John Wiley & Sons, Inc., New York, pp. 444.
7. Hildebrand, F. B., 1968. Finite-Difference Equations and Simulations, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, pp. 338.
8. Huyakorn, P. S. 1976. An Upwind Finite Element Scheme for Improved Solution of the Convection-diffusion Equation, Research Report No. 76-WR-2, Dept. of Civil Engineering, Princeton University, Princeton, New Jersey.
9. Kuo, E. Y. T., 1976. Analytical solution for 3-D diffusion model, J. Env. Eng. Div., ASCE 102: 805-820.

10. Owczarek, J. A. 1964. Fundamentals of Gas Dynamics, International Textbook Company, Scranton, PA 648 pp.
11. Pinder, G. F. and W. G. Gray, 1977. Finite Element Simulation in Surface and Subsurface Hydrology, Academic Press, New York.
12. Scheidegger, A. E. 1961. General theory of dispersion in porous media, J. Geophys. Res. 66 (10), 3273-78.
13. Smith, G. D. 1965, Numerical Solution of Partial Differential Equations, Oxford University Press, Cambridge, England.
14. Tewarson, R. P. 1973. Sparse Matrices, Academic Press, New York.
15. Travis, C. C. 1978. Mathematical Description of Adsorption and Transport of Reactive Solutes in Soil: A Review of Selected Literature, ORNL-5403, Oak Ridge National Laboratory, Oak Ridge, TN.
16. van Genuchten, M. Th., 1978. Simulation Models and their Application to Landfill Disposal Sites: A Review of Current Technology, Proceeding Fourth Annual Hazardous Waste Management Symposium, Southwest Research Institute and U.S. EPA, San Antonio, TX, March 6-8, 1978.
17. van Genuchten, M. Th. 1980. A closed form equation for predicting the hydraulic conductivity of unsaturated soils, Soil Sci. Am. J. 44:892-898.
18. van Genuchten, M. Th., 1982. One-dimensional Analytical Transport Modelling, Symposium on Unsaturated Flow and Transport Modelling, Pacific Northwest Laboratory, March 23-24, 1982.
19. Yeh, G. T. 1981. AT123D: Analytical transient One-, Two-, and Three-Dimensional Simulation of Waste Transport in the Aquifer System, ORNL-5602, Oak Ridge National Laboratory, Oak Ridge, TN.
20. 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, TN.

21. Yeh, G. T., 1984. Solution of contaminant transport equations using an orthogonal upstream weighting finite element schemes, *Finite Elements in Water Resources*, 1984 (in press)
22. Zienkiewicz, O. C., 1977. *The Finite Element Method*, Third Edition, McGraw-Hill Book Company, London, England.

APPENDIX A: INPUT DATA GUIDE

The input format for each data card is specified in the following. The number under the drawing is the last column number of each field. In general, an integer has a field of 5 and should be right-justified. On the other hand, a real number has a field length of 10 and can be placed anywhere within the field. However, when a D- and E-Format is used to describe the input of real numbers, the exponential must be right-justified.

1. TITLE: FORMAT(I5,9A8) - One card per problem.

NPROB	TITLE	IBUG
5	77	80

NPROB = Problem number.

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

IBUG = An integer indicating if the diagnostic messages to be printed? 0 = no, 1 = yes.

2. BASIC INTEGER PARAMETERS: Three cards are required per problem.

Card 1 - FORMAT(20A4)

DATNAM
80

DATNAM = Data name to describe this data set.

Cards 2 & 3 - FORMAT(16I5)

NNP	NEL	NNAT	NCM	NTI	KVI	KSTR	KSS
5	10	15	20	25	30	35	40

IWET	ILUMP	IMID	ICONFI	IOPT	KSORP	NITER	NDTCHG
45	50	55	60	65	70	75	80

NNP = Number of nodal points.

NEL = Number of elements.

NNAT = Number of material types.

NCM = Number of elements with IE(l,5) correction.

NTI = Number of time steps or time increments.

KVI = Velocity input control,

- 1 = card input for velocity, and pressure field is assumed zero.
- 0 = card input for both pressure and velocity fields.
- 1 = steady state velocity and pressure input via logical unit 1.
- 2 = transient velocity and pressure head input via auxiliary storage logical unit 1.

KSTR = Auxiliary storage output control;

- 0 = no storage,
- 1 = output stored in logical unit 2.

KSS = steady state simulation control,

- 0 = steady state solution desired,
- 1 = no steady state solution is wanted.

IWET = Weighting function control,

- 0 = Galerkin weighting,
- 1 = Upstream weighting.

ILUMP = Mass lumping control;

- 0 = no lumping,
- 1 = lumping.

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

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

IOPT = Optimization factor computing indicator,

- 1 = optimization factor is to be computed,
- 0 = optimization factor is set to 1 or - 1.

KSORP = Sorption model control,
 1 = linear isotherm,
 2 = Freundlich isotherm,
 3 = Langmuir isotherm.

NITER = No. of iterations allowed for solving
 non-linear equation. If KSORP .EQ. 1,
 NITER should be set .EQ. 1.

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

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

IPNTS	NSTR	NPITER	IVML	
5	10	15	20	80

IPNTS = Pointwise solution iteration method used?
 0 = No, 1 = Yes.

NSTR = No. of records stored on Logical Unit 3
 for restarting the computations.

NPITER = No. of iterations used to solve the
 linearized matrix equation with
 pointwise methods.

IVML = Is the flux solving matrix to be lumped?
 0 = no, 1 = yes.

3. BASIC REAL PARAMETERS

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

Card 1 - FORMAT(20A4)

DATNAM
80

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

Cards 2 & 3 - FORMAT(8D10.3)

DELT	CHNG	DELMAX	TMAX	W	ONE	CHI	TOLA
10	20	30	40	50	60	70	80

TOLE	
10	80

DELT = Initial time step size, (T).

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

DELMAX = Maximum value of DELT, (T).

TMAX = Maximum simulation time, (T).

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

OME = Iteration parameter for solving nonlinear equation. 0 ~ 1 = under-relaxation, 1.0 = exact relaxation, 1 ~ 2 = over-relaxation.

OMI = Relaxation parameter for solving the matrix equation. 0 ~ 1 = under-relaxation, 1.0 = exact relaxation, 1 ~ 2 = over-relaxation.

TOLA = Error tolerance for solving the steady state version of the nonlinear equation.

TOLB = Error tolerance for solving the transient version of the nonlinear equation.

4. **PRINTER OUTPUT 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 control.

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(80A1)

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 TPLCW, 2 = print above (1) plus concentration, 3 = print above (2) plus material 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(80A1)

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 2.

KDSK(I) = Auxiliary storage control for I-th time step similar to KDSK0.

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

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

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

5. MATERIAL PROPERTIES

A total of (2*NMAT+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 [2*NMAT+1] - FORMAT(8D10.3)

PROP(1,1)	PROP(1,2)	----	PROP(1,8)
10	20		80
PROP(1,9)	PROP(1,10)	PROP(1,11)	
10	20	30	

⋮

PROP (I, 1)	PROP (I, 2)	----	PROP (I, 8)
10	20		80
PROP (I, 9)	PROP (I, 10)	PROP (I, 11)	
10	20	30	80
.			
.			
PROP (NMAT, 1)	PROP (NMAT, 2)	----	PROP (NMAT, 8)
10	20		80
PROP (NMAT, 9)	PROP (NMAT, 10)	PROP (NMAT, 11)	
10	20	30	

PROP (I, 1) = Distribution coefficient of
medium I, (L^3/M) .

PROP (I, 2) = Bulk density, (M/L^3) .

PROP (I, 3) = Longitudinal dispersivity, (L) .

PROP (I, 4) = Lateral dispersivity, (L) .

PROP (I, 5) = Radioactive decay constant, $(1/L)$.

PROP (I, 6) = Effective porosity of medium I,
(Dimensionless in decimal point)

PROP (I, 7) = Modified compressibility of
medium I, $(1/L)$.

PROP (I, 8) = Molecular diffusion coefficient,
 (L^2/T) .

PROP (I, 9) = Tortuosity of medium I,
(dimensionless).

PROP (I, 10) = Degradation rate constant through
dissolved phase in medium I,
 $(1/L)$.

PROP (I, 11) = Degradation rate constant through
adsorbed phase in medium I,
 $(1/L)$.

PROP (I, 12) = Freundlich K in $s = K \cdot c^N$

PROP (I, 13) = Freundlich N.

PROP (I, 14) = Langmuir K in $s = K \cdot S_{MAX} \cdot c / (1 + K \cdot c)$.

PROP (I, 15) = Langmuir S_{MAX} .

NOTE: DATA SETS 6, 7, 8, and 9 TO BE DESCRIBED BELOW ARE NEEDED IF AND ONLY IF KVI IS LESS THAN OR EQUAL TO ZERO.

6. BOTTOM ELEVATION AND THICKNESS OF AQUIFERS

A set of cards are needed per problem. The first card is used to describe this data set.

Card 1 - FORMAT(20A4)

DATNAM

80

DATNAM = 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	X	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, (L).

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

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

**** A group of cards similar to Group Card 2 is needed for inputting aquifer thickness.

7. NODAL POINT COORDINATE

Usually a total of $(2*NNP+1)$ cards are required. One card for the description of this data set, NNP cards for x-coordinate, and NNP cards for y-coordinate. However, if a group of cards appear in a regular pattern, automatic generation may be made.

Card 1 - FORMAT(20A4)

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

Card 2 to Card $(NNP+1)$ - FORMAT(3I5,5X,4D10.3).

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

NSEQ = NSEQ subsequent nodes will be automatically generated.

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

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

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

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

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

Card $(NNP+2)$ to Card $(2*NNP+1)$ are input similarly but for the y-coordinate.

8. ELEMENT INCIDENCES

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(20A~)

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

Card 2 on - FORMAT(16I5)

NI = Global element number.

IE(NI,1) = Global node number of the first node of element NI.

IE(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.

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. Element numbering proceeds most rapidly along MODL dimension and least rapidly along the NLAY dimension. The following figure provides an example. The object is considered

ORNL-DWG 81-3924 ESD

to be rectangular, since it has width MODL = 3 on two opposite sides and length NLY = 5 on the other two opposite sides. To generate definitions of elements 2 through 15 automatically, including the both the incidence and material type, only one card is necessary:

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

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

**** NOTE: A 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 being to describe the data set and the rest for the material type correction, one for each correction.

Card 1 - FORMAT(20A4)

DATNAM

80

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

Card 2 on - FORMAT(5I5)

MI	NSEQ	NAD	MITYP	MTYPAD	
5	10	15	20		80

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

NSEQ = NSEQ subsequent elements will be assigned the material type MITYP.

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

MITYP = Type of material made up of element MI.

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

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

10. CARD INPUT FOR INITIAL OR PRE-INITIAL CONDITIONS

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

Card 1 - FORNAT(20A4)

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

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

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

NI	NSEQ	NAD		RNI	RAD	
5	10	15	20	30	40	80

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

NSEQ = NSEQ subsequent nodes will be generated automatically.

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

RNI = Initial or pre-initial concentration node NI, (L).

RAD = Increment of initial or pre-initial conc. for each of the NSEQ nodes, (L).

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

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

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

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

Card 1 - FORMAT(20A4)

DATNAM

80

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

Card 2 - FORMAT(16I5)

NSPR	NSDP	NRSN	NRSEL	NRSPR	NRSDP	NDN	NDPR
5	10	15	20	25	30	35	40
NDDP	NHUPR	NHUDP	NHLPR	NHLDP	NWNP	NWPR	NWDP
45	50	55	60	65	70	75	80

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

NSDP = No. of data points in each source profile, should be .GE. 2.

NRSN = No. of run-out seep-in nodes.

NRSEL = No. of RS boundary element sides.

NRSPR = No. of incoming fluid concentration profiles to be applied to RS boundary element sides.

NRSDP = No. of data points in each of the NRSPR profiles.

NDN = No. of Dirichlet nodes, should be .GE. 1.

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

NDDP = No. of data points in each of the NDPR profiles, should be .GE. 2.

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

NHUDP = No. of data points in each of the NHUPR

profiles, should be .GE. 2.

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

NHLDP = No. of data points in each of the NHLPR profiles, should be .GE. 2.

NWNP = No. of well source/sink nodes.

NWPR = No. of well source/sink profiles.

NWDP = No. of data points in each well source/sink profile.

12. SOURCE/SINKS: Subdata sets (d) through (f) are needed if and only if NWNP > 0.

(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 NWPR 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

Second profile

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

:

---	TSOS (2,NSDP)	SOS (2,NSDP)	
-----	---------------	--------------	--

80

I-th profile

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

---	TSOS (I, NSDP)	SOS (I, NSDP)	
			80

NSPR-th profile

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

---	TSOS (NSPR, NSDP)	SOS (NSPR, NSDP)	
			80

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

SOS (I, J) = Source/sink value of J-th data point in I-th profile, (M/T/L**2) if value .gt. 0; (L**3/T/L**2) if value .lt. 0.

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

MI	NSEQ	MAD	MITYP	MTYPAD	
5	10	15	20		80

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

NSEQ = NSEQ elements will contain the source type MITYP.

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

MITYP = Source type in element MI.

MTYPAD = Increment of source type in each of

the WSEQ subsequent elements.

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

- (d) Well Source/Sink Profile: FORMAT(8D10.3).
Number of cards depends on NWPR and NWDP. 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,NWDP)	WSSP(1,NWDP)		
				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
.				
---	TWSSP(I,NWDP)	WSSP(I,NWDP)		
				80

NWPR-th Profile

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

---	TWSSP (NWPR, NWDP)	WSSP (NWPR, 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, (M/T) if WSSP .GE. 0, (L**3/T) if WSSP .LT. 0.

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

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

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

- (f) Type of Well Source/Sink Nodes: FORMAT(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	NIAD	NITYP	NITYPA	
5	10	15	20	25	80

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

NSEQ = NSEQ subsequent well nodes will be generated automatically.

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

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

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

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

13. DIRICHLET BOUNDARY CONDITIONS: This data set is required if $N_{DN} > 0$.

(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

TCO _N (1,1)	CO _N (1,1)	TCO _N (1,2)	CO _N (1,2)	---
10	20	30	40	80

⋮

---	TCO _N (1,NDDP)	CO _N (1,NDDP)	
-----	---------------------------	--------------------------	--

80

Second profile

TCO _N (2,1)	CO _N (2,1)	TCO _N (2,2)	CO _N (2,2)	---
10	20	30	40	80

⋮

---	TCO _N (2,NDDP)	CO _N (2,NDDP)	
-----	---------------------------	--------------------------	--

80

⋮

I-th profile

TCO _N (I,1)	CO _N (I,1)	TCO _N (I,2)	CO _N (I,2)	---
10	20	30	40	80

⋮

---	TCON(I,NDPR)	CON(I,NDPR)	
-----	--------------	-------------	--

80

⋮

NDPR-th profile

TCON(NDPR,1)	CON(NDPR,1)	TCON(NDPR,2)	CON(NDPR,2)	---
--------------	-------------	--------------	-------------	-----

10

20

30

40

80

⋮

---	TCON(NDPR,NDPR)	CON(NDPR,NDPR)	
-----	-----------------	----------------	--

80

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

CON(I,J) = Conc. value fo J-th data point in I-th
Dirichlet-concentration profile, (I).

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

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

5

10

15

80

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

- (d) Type of Dirichlet Node: FORMAT(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	NAD	NITYP	NTYPAD	
----	------	-----	-------	--------	--

5

10

15

20

80

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

NSEQ = NSEQ subsequent Dirichlet nodes will be generated automatically.

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

NTYP = Type of Dirichlet-con. profile for node NI and the NSEQ subsequent nodes.

NTYPAD = Increment of the type of Dirichlet-con. profile for each of the NSEQ subsequent Dirichlet nodes.

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

14. **Run-out Seep-in Conditions:** This data set is required if and only if NRSEL > 0.

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

DATNAM

80

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

- (b) **Run-out Seep-in Nodes:** FORMAT(16I5) - The number of cards required in this sub-data set depends on NRSN. Each card contains 16 nodes.

NRSN (1)	NRSN (2)	---	NRSN (I)	---	NRSN (NRSN)	
5	10					80

NRSN (I) = Global node number of I-th RS node.

- (c) Prescribed Seep-in Concentration Profiles: FORMAT (8D10.3)
This is read in similar to that in sub-data set 12 (b).

First profile

TRSTAB (1, 1)	RSTAB (1, 1)	TRSTAB (1, 2)	RSTAB (1, 2)	---
10	20	30	40	80

.

---	TRSTAB (1, NRSDP)	RSTAB (1, NRSDP)		
				80

Second profile

TRSTAB (2, 1)	RSTAB (2, 1)	TRSTAB (2, 2)	RSTAB (2, 2)	---
10	20	30	40	80

.

---	TRSTAB (2, NRSDP)	RSTAB (2, NRSDP)		
				80

.

I-th profile

TRSTAB (I, 1)	RSTAB (I, 1)	TRSTAB (I, 2)	RSTAB (I, 2)	---
10	20	30	40	

.

---	TRSTAB (I, NRSDP)	RSTAB (I, NRSDP)		
				80

.

NRSEL-th profile

TRSTAB (NRSPR, 1)	RSTAB (NRSPR, 1)	TRSTAB (NRSPR, 2)	---	
10	20	30	80	

.

```
--- TRSTAB(NRSPR,NRSDP) RSTAB(NRSPR,NRSDP)
```

80

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

RSTAB(I,J) = Value of seep-in concentration of J-th data point in I-th seep-in concentration profile, (M/L**3)

(d) Run-out Seep-in Boundary Element-sides: FORMAT(16I5)

MI	NSEQ	N1	N2	NMI	N1AD	N2AD	NAD	MITYP	
5	10	15	20	25	30	35	40	45	80

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

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

N1 = Global node number of the first node in the RS element-side MI.

N2 = Global node number of the second node in the RS element side MI.

NMI = Global element number of the MI-th RS element side.

N1AD = Increment of the first global node number of the MI-th RS element-side.

N2AD = Increment of the second global node number of the MI-th RS element-side.

NAD = Increment of the global element number in each of the NSEQ RS element sides.

MITYP = Type of incoming fluid concentration profile to be applied to the MI-th RS element-side and its NSEQ subsequent RS element-sides.

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

NOTE: DATA SETS 15 AND 16 TO BE DESCRIBED BELOW ARE NEEDED ONLY IF KVI IS LESS OR EQUAL TO ZERO.

15. UPPER LEAKING AQUIFERS

(a) Data Description: FORMAT(20A4)

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

I-th profile

-----	THUTAB (I, NHUPD)	HUTAB (I, NHODP)	
			80

.

.

.

NHUPR-th profile

THUTAB (NHUPR, 1)	HUTAB (NHUPR, 1)	THUTAB (NHUPR, 2)	-----
10	20	30	80

.

.

-----	THUTAB (NHUPR, NHODP)	HUTAB (NHUPR, NHODP)	
			80

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

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

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

RKU (1)	RKU (2)	--- RKU (I)	---	RKU (NHUPR)	
10	20				80

RKU (I) = Leaking coefficient of I-th upper leaking aquifer type, (1/T).

- (d) Type of Upper Leaking Aquifers: FORMAT(4I5)

MI	NSEQ	MAD	MITYP	
5	10	15	20	80

MI = Global element number of the first element in the sequence that has the leaking aquifer of type MITYP lying above.

NSEQ = NSEQ subsequent elements will be given the leaking aquifer type MITYP overlaying them.

MAD = Increment of element number for each of the **MSEQ** elements.

HITYP = Upper leaking aquifer type for element **HI**.

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

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

(a) Data Description: **FORMAT(20A4)**

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

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

First profile

Second profile

I-th profile

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

NHLPR-th profile

THLTAB (NHLPR,1)	HLTAB (NHLPR,1)	THLTAB (NHLPR,2)	-----
10	20	30	80
:			
-----	THLTAB (NHLPR,NHLDP)	HLTAB (NHLPR,NHIDP)	
			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.

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

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

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

(d) Type of Lower Leaking Aquifer: FORMAT(4I5)

HI	NSEQ	NAD	NITYP	
5	10	15	20	80

HI = Global element number of the first element in the sequence that has the leaking aquifer of type NITYP lying below.

NSEQ = NSEQ subsequent elements will be given the leaking aquifer of type NITYP laying below them.

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

NITYP = Lower leaking aquifer type for element HI.

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

17. CCNCENTRATION PROFILE IN UPPER LEAKY AQUIFER

This data set is read in similar to that in sub-data set 12 (a) and (b). The number of cards depends on NHUPR and NHUDP.

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

DATANAM
80

DATANAM = 80 columns of characters to describe the nature of this data set.

(b) Upper Leaking Aquifer Concentration Profiles: FORMAT(8D10.3)
This data set is read NHUPR-wisely. Each card contains four data points.

First profile

TCUTAB(1,1)	CUTAB(1,1)	TCUTAB(1,2)	----
10	20	30	80

:

Second profile

:

:

I-th profile

:

:

NHOPR-th profile

:

TCUTAB (I, J) = Time of J-th data point in I-th upper-aquifer concentration vs time profile, (T).

CUTAB (I, J) = Concentration in the upper leaking aquifer of J-th data point in I-th profile, (N/L**3).

18. CONCENTRATION PROFILE IN THE LOWER LEAKY AQUIFER

This data set is read in similar to that in data set 17.

The number of cards depends on FLIPR and NHLDP.

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

DATANAM

80

DATANAM = 80 columns of characters to describe
the nature of this data set.

- (b) Lower leaking Aquifer Concentration Profiles: FORNAT(8D10.3).
This data set is read NHL-wisely. Each card contains four
data points.

First profile

TCLTAB (1, 1)	CLTAB (1, 1)	TCLTAB (1, 2)	-----
10	20	30	80

:

TCLTAB (1, NHLDP)	CLTAB (1, NHLDP)
80	

Second profile

TCLTAB (2, 1)	CLTAB (2, 1)	TCLTAB (2, 2)	-----
10	20	30	80

:

-----	TCLTAB (2, NHLDP)	CLTAB (2, NHLDP)	
80			

:

I-th profile

TCLTAB (I, 1)	CLTAB (I, 1)	TCLTAB (I, 2)	-----
10	20	30	80

:

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

CLTAB (I,J) = Concentration in the lower leaky aquifer of J-th data point in I-th profile, (N/L**3).

19. VELOCITY AND PRESSURE FIELD

If KVI .GT. 0, this data set is not needed since these will be read via Logical Unit 1. If KVI .LT. 0, only data set in (b) is needed. If KVI .EQ. 0, both (a) and (b) will be needed.

(a) Pressure Field: FORMAT(3I5,5I,4D10.3)

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

NSEQ = NSEQ subsequent nodes will be automatically generated.

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

HNI = Pressure head of node NI, (L).

HAD = Increment of pressure head for each of the NSEQ subsequent nodes, (L).

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

(b) Velocity Data: `FORMAT(3I5,5I,4D10.3)`

NI	NSEQ	NAD		VXNI	VZNI	VXAD	VZAD	
5	10	15	20	30	40	50	60	80

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

NSEQ = NSEQ subsequent nodes will automatically generated.

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

VXNI = x-velocity component of node NI, (L/T).

VZNI = z-velocity component of node NI, (L/T).

VXAD = Increment of x-velocity for each of the NSEQ subsequent nodes, (L/T).

VZAD = Increment of z-velocity for each of the NSEQ subsequent nodes, (L/T).

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

20. 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.

B.1: FEHA INPUT DATA FOR EXAMPLE 1 - ONE DIMENSIONAL PROBLEM

END OF JOB

B.2: PENA INPUT DATA FOR EXAMPLE 2 - SMALL TWO-DIMENSIONAL PROBLEM

```

1
DATA SET 2: BASIC INTEGERS
561 500 1 0 10 -1 1 1 0 0 0 1 1 1 1 1
1 0 300
DATA SET 3: BASIC REAL NUMBERS
0.4D0 0.0D0 C.4D0 1G.0DG 1.0D0 1.0D0 1.85D0 1.0D-3
1.0D-4
DATA SET 4: PRINT AND DISK STORE CONTROL AND TIME RESETING CONTROL
11111111113111111113
1 1
1.0D50
DATA SET 5: MATERIAL PROPERTIES
0.02D00 1.3D3 0.1D0 0.01D0 2.476D-2 0.25D0 0.0D0 0.0D0
0.0D0 0.0D0 0.0D0 0.0D0 0.0D0 0.0D0 0.0D0
DATA SET 6: BOTTOM ELEVATIONS AND AQUIFER THICKNESS.
1 560 1 0.0DC 0.0D0
1 560 1 3.0D1 0.0D0
DATA SET 7: NODAL COORDINATES
1 50 11 0.0DC 1.0D0 0.0D0
2 50 11 0.0D0 1.0D0 0.0D0
3 50 11 0.0D0 1.0DC 0.0D0
4 50 11 0.0D0 1.0D0 0.0D0
5 50 11 0.0DC 1.0D0 0.0D0
6 50 11 0.0D0 1.0D0 0.0D0
7 50 11 0.0D0 1.0D0 0.0D0
8 50 11 0.0D0 1.0D0 0.0D0
9 50 11 0.0D0 1.0DC 0.0D0
10 50 11 0.0D0 1.0D0 0.0D0
11 50 11 0.0DC 1.0D0 0.0D0
1 50 11 0.0DC 0.0D0 0.0D0
2 50 11 1.0D0 0.0D0 0.0D0
3 50 11 2.0D0 0.0D0 0.0D0
4 50 11 3.0D0 0.0D0 0.0D0
5 50 11 4.0DC 0.0D0 0.0D0
6 50 11 5.0D0 0.0D0 0.0D0
7 50 11 6.0D0 0.0D0 0.0D0
8 50 11 7.0D0 0.0D0 0.0D0
9 50 11 8.0D0 0.0D0 0.0D0
10 50 11 9.0D0 0.0D0 0.0D0
11 50 11 10.0D0 0.0D0 0.0D0
DATA SET 8: ELEMENT INCIDENCES
1 1 12 13 2 1 10 50
DATA SET 10: INITIAL CONDITIONS
1 560 1 0.0D0 0.0D0
DATA SET 11: TRANSIENT INTEGER CONTROL
1 2 11 10 1 2 11 2 2 1 2 1 2 0 1 2
DATA SET 12: SOURCES/SINKS
0.0D0 0.0D0 1.0D6C 0.0D0
1 499 1 1 0
DATA SET 13: DIRICHLET BOUNDARY CONDITIONS
0.0D0 1.0D0 1.0D50 1.0D0
0.0D0 0.0D0 1.0D5C 0.0D0
1 2 3 4 5 6 7 8 9 10 11
1 5 1 1 0
7 4 1 2 0

```

1

1

1

1

1

1

1

1

1

1

1

1

1

1

19	50	41	0.0D0	1.0D0	0.0D0
20	50	41	0.0D0	1.0D0	0.0D0
21	50	41	0.0D0	1.0D0	0.0D0
22	50	41	0.0D0	1.0D0	0.0D0
23	50	41	0.0D0	1.0D0	0.0D0
24	50	41	0.0D0	1.0D0	0.0D0
25	50	41	0.0D0	1.0D0	0.0D0
26	50	41	0.0D0	1.0D0	0.0D0
27	50	41	0.0D0	1.0D0	0.0D0
28	50	41	0.0D0	1.0D0	0.0D0
29	50	41	0.0D0	1.0D0	0.0D0
30	50	41	0.0D0	1.0D0	0.0D0
31	50	41	0.0D0	1.0D0	0.0D0
32	50	41	0.0D0	1.0D0	0.0D0
33	50	41	0.0D0	1.0D0	0.0D0
34	50	41	0.0D0	1.0D0	0.0D0
35	50	41	0.0D0	1.0D0	0.0D0
36	50	41	0.0D0	1.0D0	0.0D0
37	50	41	0.0D0	1.0D0	0.0D0
38	50	41	0.0D0	1.0D0	0.0D0
39	50	41	0.0D0	1.0D0	0.0D0
40	50	41	0.0D0	1.0D0	0.0D0
41	50	41	0.0D0	1.0D0	0.0D0
1	50	41	0.0D0	0.0D0	0.0D0
2	50	41	1.0D0	0.0D0	0.0D0
3	50	41	2.0D0	0.0D0	0.0D0
4	50	41	3.0D0	0.0D0	0.0D0
5	50	41	4.0D0	0.0D0	0.0D0
6	50	41	5.0D0	0.0D0	0.0D0
7	50	41	6.0D0	0.0D0	0.0D0
8	50	41	7.0D0	0.0D0	0.0D0
9	50	41	8.0D0	0.0D0	0.0D0
10	50	41	9.0D0	0.0D0	0.0D0
11	50	41	10.0D0	0.0D0	0.0D0
12	50	41	11.0D0	0.0D0	0.0D0
13	50	41	12.0D0		
14	50	41	13.0D0		
15	50	41	14.0D0		
16	50	41	15.0D0		
17	50	41	16.0D0		
18	50	41	17.0D0		
19	50	41	18.0D0		
20	50	41	19.0D0		
21	50	41	20.0D0		
22	50	41	21.0D0		
23	50	41	22.0D0		
24	50	41	23.0D0		
25	50	41	24.0D0		
26	50	41	25.0D0		
27	50	41	26.0D0		
28	50	41	27.0D0		
29	50	41	28.0D0		
30	50	41	29.0D0		
31	50	41	30.0D0		
32	50	41	31.0D0		
33	50	41	32.0D0		
34	50	41	33.0D0		
35	50	41	34.0D0		
36	50	41	35.0D0		
37	50	41	36.0D0		
38	50	41	37.0D0		
39	50	41	38.0D0		
40	50	41	39.0D0		
41	50	41	40.0D0		

DATA SET 8: ELEMENT INCIDENCES

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
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	18.9000	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
220	0	1	19.0000	0.0	0.0
221	1	1	18.0000	0.1000	0.0
223	1	1	18.2000	0.1000	0.0
225	1	1	18.4000	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: 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	3.4000	-0.4000	0.0
51	1	1	37.6000	-0.4000	0.0
53	1	1	36.3000	-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
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
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	33.9000	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
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	10	20
---	---	----	----	---	---	----	----

DATA SET 9: MATERIAL CORRECTIONS FOR REGIONS IN REGK AND ICKL

38	2	10	2	0
93	2	10	3	0

DATA SET 10: PRE-INITIAL CONDITIONS

1	230	1	5.0001	0.000	0.000
---	-----	---	--------	-------	-------

DATA SET 11: TRANSIENT INTEGER PARAMETERS
 20 21 1 2 11 1 2 1 2 2 2 1 2 1 1 2
 DATA SET 12: SOURCES/SINKS
 0.0DC0 0.7DD0 1.0D50 0.0D00
 1 199 1 1 0

0.0D00 -1.0D02 1.0C0D50 -1.0D02
 83
 1 0 1 1 0

DATA SET 13: DIRICHLET BOUNDARY CONDITIONS
 0.0DC0 5.2D01 1.000D50 5.2D01
 221 222 223 224 225 226 227 228 229 230 231
 1 10 1 1 0

DATA SET 14: FLUX BOUNDARY CONDITIONS
 0.0D00 -0.2D0 1.000D50 -0.2D0
 1 12 23 34 45 56 67 78 89 100 111 122 133 144 155 166
 177 188 199 210 221
 1 20 1 1 0
 1 19 1 2 1 1

DATA SET 15: UPPER LEAKING AQUIFER
 0.0DC0 5.1D01 1.000D50 5.1D01
 0.3D00 5.1D01 1.000D50 5.1D01
 4.0D-2 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
 0.3DC0 5.2D01 1.000D50 5.2D01
 1.0D-3
 1 159 1 1 0

END OF JOB

B.4.2: FEMA INPUT DATA FOR EXAMPLE 4 - A CONFINED AQUIFER PROBLEM

4 FEMA INPUT DATA OF EXAMPLE 4
 DATA SET 2: BASIC INTEGER PARAMETERS
 231 200 3 6 80 1 1 1 0 0 0 0 1 1 1 1
 0 0 300
 DATA SET 3: BASIC REAL PARAMETERS
 1.0E0 0.2D0 60.0D0 1.0D50 1.3D0 1.0D0 1.0C0D0 1.0D-4
 1.0D-5
 DATA SET 4: PRINTER, DISK OUTPUT, AND TIME STEP CHANGE CONTROL
 131 131 131 131 131
 1 1 1 1 1
 1.0D50
 DATA SET 5: MATERIAL PROPERTIES
 1.0D-3 1.2D3 2.0D1 5.0D0 6.78D-5 1.0D-1 1.0D-5 1.0D-4
 1.0E0 0.0D0 0.0D0 0.0D0 0.0D0 0.0D0 0.0D0

2.0D-3	1.4D3	1.0D1	2.0D0	6.78D-5	2.7D-1	2.0D-5	1.0D-4
2.0D0	1.0D-5	0.0D0	0.0D0	0.0D0	0.0D0	0.0D0	
3.0D-3	1.6D3	5.0D0	2.0D0	6.78D-5	4.5D-1	4.0D-5	1.0D-4
5.0D0	1.0D-5	0.0D0	0.0D0	0.0D0	0.0D0	0.0D0	

DATA SET 10: INITIAL CONDITIONS

1	230	1	0.0D0	0.0D0
---	-----	---	-------	-------

DATA SET 11: TRANSIENT INTEGERS

1	2	21	20	1	2	11	2	2	1	2	1	2	1	1	2
---	---	----	----	---	---	----	---	---	---	---	---	---	---	---	---

DATA SET 12: SOURCE/SINKS

G.GEC	0.0D0	1.0D50	0.0D0
-------	-------	--------	-------

1	199	1	1	0
---	-----	---	---	---

0.0D0	-1.0D2	1.0D50	-1.0D2
-------	--------	--------	--------

83	1	0	1	1	0
----	---	---	---	---	---

DATA SET 13: DIRICHLET BOUNDARY CONDITIONS

0.0D0	1.0D0	1.0D50	1.0D0							
0.0D0	0.0D0	1.0D50	0.0D0							
221	222	223	224	225	226	227	228	229	230	231
1	4	1	2	0						
6	5	1	1	0						

DATA SET 14: VARIABLE BOUNDARY CONDITIONS

1	12	23	34	45	56	67	78	89	100	111	122	133	144	155	166
177	1E8	199	210	221											
0.0D0	2.0D0	1.0D50	2.0D0												
1	19	1	12	1	11	11	10	1							

DATA SET 17: CONCENTRATION IN UPPER LEAKING AQUIFER

0.0D0	0.0D0	1.0D50	0.0D0
-------	-------	--------	-------

DATA SET 18: CONCENTRATION IN LOWER LEAKING AQUIFER

0.0D0	0.0D0	1.0D50	0.0D0
-------	-------	--------	-------

DATA SET 20: END OF JOB

B.5.1: FEWA INPUT DATA FOR EXAMPLE 5 - AN UNCONFINED AQUIFER PROBLEM

5A FEWA INPUT DATA FOR DEMONSTRATION EXAMPLE 2

231	200	3	6	0	0	1	0	0	50	0	0	0	1	0	300
-----	-----	---	---	---	---	---	---	---	----	---	---	---	---	---	-----

DATA SET 3: BASIC REAL PARAMETERS

0.5D00	0.0D00	0.5D00	1.000D00	1.0D-4	1.0D-5	1.0D03	7.3D10
1.12D2	1.0D00	1.0D00	1.0D00				

DATA SET 4: LINEPRINTER, DISKOUTPUT, AND TIME STEP SETTING CONTROL

33
11

1.0D50

DATA SET 5: MATERIAL PROPERTIES

1.0D-5	3.0D-6	0.10D00	1.00D00	1.00D00	0.50D00	0.10D00
2.0D-5	3.0D-6	0.27D00	0.50D-1	0.50D-1	0.25D-1	0.27D00
4.0D-5	3.0D-6	0.45D00	0.80D-3	0.80D-3	0.40D-3	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	1.0000	-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	1.6000	-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
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
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

1 230 1 50.0 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	9.2000	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
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
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	33.9000	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
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	10	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	5.0001	0.000	0.000
---	-----	---	--------	-------	-------

DATA SET 11: TRANSIENT INTEGER PARAMETERS

20	21	1	2	11	1	2	3	2	1	2	1	2	0	1	2
----	----	---	---	----	---	---	---	---	---	---	---	---	---	---	---

DATA SET 12: SOURCES/SINKS

0.0000	2.00-3	1.000050	2.00-3
0.0000	1.00-3	1.000050	1.00-3

0.0DC0	0.5D-8	1.000D50	0.5D-3
1 36	1 1	0	
38 2	10 2	0	
53 8	1 1	0	
49 8	1 1	0	
59 33	1 1	0	
93 2	10 3	0	
94 8	1 1	0	
104 8	1 1	0	
114 86	1 1	0	

DATA SET 13: DIRICHLET BOUNDARY CONDITIONS

0.0DC0	5.2D01	1.000D50	5.2D01
221 222	223 224	225 226	227 228 229 230 231
1 10	1 1	0	

DATA SET 14: FLUX BOUNDARY CONDITIONS

0.0D00	-2.0D-1	1.000D50	-2.0D-1
1 12	23 34	45 56	67 78 89 100 111 122 133 144 155 166
177 188	199 210	221	
1 20	1 1	0	
1 19	1 2	1 1	

DATA SET 15: UPPER LEAKING AQUIFER

0.0D00	5.1D01	1.000D50	5.1D01
0.0D00			
1 199	1 1	0	

DATA SET 16: LOWER LEAKING AQUIFERS

0.0D00	5.2D01	1.000D50	5.2D01
0.1D-2			
1 199	1 1	0	

END OF JOB

B.5.2: FEMA INPUT DATA FOR EXAMPLE 5 - AN UNCONFINED AQUIFER PROBLEM

5 FEMA INPUT DATA OF EXAMPLE 5

DATA SET 2: BASIC INTEGER PARAMETERS

231 200	3 6	80 1	1 1	0 0	0 1	1 1	1 1	1 1
0 0	300							

DATA SET 3: BASIC REAL PARAMETERS

1.0E0	0.2D0	60.0D0	1.0D50	1.0D0	1.0D0	1.0D00	1.0D-4
1.0D-5							

DATA SET 4: PRINTER, DISK OUTPUT, AND TIME STEP CHANGE CONTROL

3	131	131	131	131	131
3					
1	1	1	1	1	1
1					

1.0D50

DATA SET 5: MATERIAL PROPERTIES

1.0D-3	1.2D3	2.0D1	5.0D0	6.78D-5	1.0D-1	1.0D-5	1.0D-4
1.0E0	0.0D0	0.0D0	0.3D3	0.0D0	0.0D0	0.0D0	
2.0D-3	1.4D3	1.0D1	2.0D0	6.78D-5	2.7D-1	2.0D-5	1.0D-4
2.0E0	1.0D-5	0.0D0	0.0D0	0.0D0	0.0D0	0.0D0	
3.0D-3	1.6D3	5.0D0	2.0D0	6.78D-5	4.5D-1	4.0D-5	1.0D-4
5.0D0	1.0D-5	0.0D0	0.3D0	0.0D0	0.0D0	0.0D0	

DATA SET 10: INITIAL CONDITIONS

1 230	1	0.0D0	0.0D0
-------	---	-------	-------

DATA SET 11: TRANSIENT INTEGERS

1 2	31 30	2 2	1 1	2 1	2 1	2 0	1 2
-----	-------	-----	-----	-----	-----	-----	-----

DATA SET 12: SOURCE/SINKS

0.0D0	1.0D-3	1.0D50	1.0D-3
1 199	1 1	0	

DATA SET 13: DIRICHLET BOUNDARY CONDITIONS

0.0E0 0.0D0 1.0D50 0.0D0

11

1 0 1 1 0

DATA SET 14: VARIABLE BOUNDARY CONDITIONS

1	12	23	34	45	56	67	78	89	100	111	122	133	144	155	166
177	188	199	210	221	222	223	224	225	226	227	228	229	230	231	

0.0E0 2.0D0 1.0D50 2.0D0

0.0D0 0.0D0 1.0D50 0.0D0

1 19 1 12 1 11 11 10 1

21 9 221 222 191 1 1 1 2

DATA SET 17: CONCENTRATION IN UPPER LEAKING AQUIFER

0.0E0 0.0D0 1.0D50 0.0D0

DATA SET 18: CONCENTRATION IN LOWER LEAKING AQUIFER

0.0D0 0.0D0 1.0D50 0.0D0

DATA SET 20: END OF JOB

APPENDIX C: FORTRAN SOURCE PROGRAM

C		MAIN 005
CC		MAIN 010
C	*****	MAIN 015
C	*	MAIN 020
C		MAIN 025
C	*	MAIN 030
C	*****	MAIN 035
CC		MAIN 040
C		MAIN 045
CC		MAIN 050
C	YEH, G. T. AND D. D. HOFF, 1983. "FEMA: A FINITE ELEMENT MODEL OF	MAIN 055
C	MATERIAL TRANSPORT THROUGH AQUIPERS," ORNL-6063, OAK RIDGE	MAIN 060
C	NATIONAL LABORATORY, OAK RIDGE, TENNESSEE 37830	MAIN 065
CC		MAIN 070
C		MAIN 075
CC		MAIN 080
C	*****	MAIN 085
C	*	MAIN 090
C	* DEFINITIONS OF VARIABLES	MAIN 095
C	*	MAIN 100
C	* MAXNP = MAXIMUM NO. OF NODAL POINTS	MAIN 105
C	* MAXEL = MAXIMUM NO. OF ELEMENTS	MAIN 110
C	* MAXBNP = MAXIMUM NO. OF BOUNDARY NODAL POINTS	MAIN 115
C	* MAXBEL = MAXIMUM NO. OF BOUNDARY ELEMENT SIDES	MAIN 120
C	* MAXBW = MAXIMUM BAND WIDTH OF THE GLOBAL MATRIX	MAIN 125
C	* MAXNTI = MAXIMUM NO. OF TIME INCREMENTS	MAIN 130
C	* MAXMAX = MAXIMUM NO. OF MATERIALS	MAIN 135
C	* MAXDTC = MAXIMUM NO. OF DELT CHANGES	MAIN 140
C	*	MAIN 145
C	*	MAIN 150
C	* NNP = NO. OF NODAL POINTS	MAIN 155
C	* NEL = NO. OF ELEMENTS	MAIN 160
C	* MBN = NO. OF BOUNDARY NODES	MAIN 165
C	* MBEL = NO. OF BOUNDARY ELEMENT SIDES	MAIN 170
C	* IBAND = BAND-WIDTH OF THE GLOBAL MATRIX	MAIN 175
C	* INBP = HALF BAND-WIDTH PLUS ONE	MAIN 180
C	* IHALPB = HALF BAND-WIDTH OF THE GLOBAL MATRIX	MAIN 185
C	* MNAT = NO. OF MATERIALS	MAIN 190
C	* MCH = NO. OF MATERIAL CORRECTIONS	MAIN 195
C	* MDTCHG = NO. OF DELT CHANGES	MAIN 200
C	*	MAIN 205
C	*	MAIN 210
C	* NTI = NO. OF TIME INCREMENTS OR NO. OF TIME STEPS	MAIN 215
C	* NSTRT = NO. OF RESTART RECORDS TO BE READ VIA LOGICAL UNIT 2	MAIN 220
C	* KSTR = CONTROL STORAGE ON LOGICAL UNIT 1, 0=NO/, 1=YES	MAIN 225
C	*	MAIN 230
C	* KPRO = LINE PRINTER CONTROL FOR THE STEADY STATE OR PRE-INITIAL	MAIN 235
C	CONDITIONS	MAIN 240
C	0= PRINT NOTHING, 1=PRINT FLOYES THROUGH ALL BOUNDARIES	MAIN 245
C	2=PRINT CONCENTRATION,	MAIN 250
C	3=ALSO PRINT MATERIAL FLUXES	MAIN 255
C	*	MAIN 260
C	* KDSK0 = DISK STORAGE CONTROL FOR THE STEADY STATE OR PRE-INITIAL	MAIN 265
C	CONDITIONS, 0=NO STORAGE, 1=YES	MAIN 270
C	*	MAIN 275
C	* KSS = STEADY STATE SOLUTION DESIRED? 0=YES, 1=NO	MAIN 280
C	* KVI = VELOCITY FIELD INPUT CONTROL,	MAIN 285
C	-1 = CARD INPUT VELOCITY AND PRESSURE FIELD,	MAIN 290
C	0 = CARD INPUT PRESSURE HEAD AND VELOCITY IS SET ZERO	MAIN 295
C	*	MAIN 300

C * NITER = NO. OF ITERATIONS ALLOWED FOR SOLVING NONLINEAR EQUATIONS	*	MAIN 305
C *	*	MAIN 310
C * DELT = TIME STEP SIZE, (T)	*	MAIN 315
C * CHNG = PERCENTAGE CHANGE IN TIME STEP SIZE, (DECIMAL VALUES)	*	MAIN 320
C * DELHAX = MAXIMUM VALUE OF TIME STEP SIZE ALLOWED, (T)	*	MAIN 325
C * THAX = MAXIMUM SIMULATION TIME, (T)	*	MAIN 330
C * DELT0 = INITIAL TIME STEP SIZE	*	MAIN 335
C * TOLA = TOLERANCE FOR ITERATING STEADY STATE SOLUTION	*	MAIN 340
C * TOLB = TOLERANCE FOR ITERATING TRANSIENT SOLUTION	*	MAIN 345
C * ONE = ITERATION PARAMETER	*	MAIN 350
C * 0.0 TO 1.0 = UNDER RELAXATION	*	MAIN 355
C * 1.0 TO 2.0 = OVER-RELAXATION	*	MAIN 360
C * RW(N) = CONCENTRATION ITERATE	*	MAIN 365
C *	*	MAIN 370
C *	*	MAIN 375
C * NXSPE = MAXIMUM NO. OF SOURCE PROFILES	*	MAIN 380
C * NXSDF = MAXIMUM NO. OF DATA POINTS IN EACH SOURCE PROFILE	*	MAIN 385
C * NSPR = NO. OF SOURCE PROFILES	*	MAIN 390
C * NSDP = NO. OF DATA POINTS IN EACH SOURCE PROFILE	*	MAIN 395
C * NWNPF = MAXIMUM NO. OF WELL NODAL POINTS	*	MAIN 400
C * NWNP = NO. OF WELL NODAL POINTS	*	MAIN 405
C * NWSPE = MAXIMUM NO. OF WELL SOURCE PROFILES	*	MAIN 410
C * NWSDF = MAXIMUM NO. OF DATA POINT IN EACH WELL SOURCE PROFILE	*	MAIN 415
C * NWPR = NO. OF WELL SOURCE/SINK PROFILES	*	MAIN 420
C * NWDP = NO. OF DATA POINT IN EACH OF THE WELL SOURCE/SINK PROFILE	*	MAIN 425
C *	*	MAIN 430
C * NXRSP = MAXIMUM NO. OF RUN-OUT SEEP-IN NODAL POINTS	*	MAIN 435
C * NXRSEL = MAXIMUM NO. OF RUN-OUT SEEP-IN BOUNDARY ELEMENT SIDES	*	MAIN 440
C * NXRSEF = MAXIMUM NO. OF RUN-OUT AND SEEP-IN PROFILES	*	MAIN 445
C * NXRSDP = MAXIMUM NO. OF DATA POINTS IN EACH RS PROFILE	*	MAIN 450
C * NRSN = NO. OF RUN-OUT SEEP-IN MODES	*	MAIN 455
C * NRSSEL = NO. OF RUN-OUT SEEP-IN BOUNDARY ELEMENT-SIDES	*	MAIN 460
C * NRSPE = NO. OF RUN-OUT SEEP-IN PROFILES	*	MAIN 465
C * NRSDF = NO. OF DATA POINTS IN EACH RS PROFILE	*	MAIN 470
C *	*	MAIN 475
C * NXDNF = MAXIMUM NO. OF DIRICHLET BOUNDARY NODAL POINTS	*	MAIN 480
C * NXDPE = MAXIMUM NO. OF DIRICHLET BOUNDARY PROFILES	*	MAIN 485
C * NXDDE = MAXIMUM NO. OF DATA POINTS IN EACH DIRICHLET PROFILE	*	MAIN 490
C * NDN = NO. OF DIRICHLET BOUNDARY NODAL POINTS	*	MAIN 495
C * NDPE = NO. OF DIRICHLET BOUNDARY PROFILES	*	MAIN 500
C * NDDP = NO. OF DATA POINTS IN EACH DIRICHLET PROFILE	*	MAIN 505
C *	*	MAIN 510
C * NXEUPF = MAXIMUM NO. OF HEAD-IN-UPPER-AQUIFER (HU) PROFILES	*	MAIN 515
C * NXHUCP = MAXIMUM NO. OF DATA POINTS IN EACH HU-TIME(T) PROFILE	*	MAIN 520
C * NHUPF = NO. OF HU-T PROFILES	*	MAIN 525
C * NHUPE = NO. OF DATA POINTS IN EACH HU-T PROFILE	*	MAIN 530
C *	*	MAIN 535
C * NXHLPF = MAXIMUM NO. OF HEAD-IN-LOWER-AQUIFER (HL) PROFILES	*	MAIN 540
C * NXHLDP = MAXIMUM NO. OF DATA POINTS IN EACH HL-TIME(T) PROFILE	*	MAIN 545
C * NHLPF = NO. OF HL-T PROFILES	*	MAIN 550
C * NHLDE = NO. OF DATA POINTS IN EACH HL-T PROFILE	*	MAIN 555
C *	*	MAIN 560
C *	*	MAIN 565
C * ALPHA2 = WEIGHTING FACTOR ALONG THE SIDE 4-3	*	MAIN 570
C * ALPHA1 = WEIGHTING FACTOR ALONG THE SIDE 1-2	*	MAIN 575
C * BETA1 = WEIGHTING FACTOR ALONG THE SIDE 2-3	*	MAIN 580
C * BETA2 = WEIGHTING FACTOR ALONG THE SIDE 1-4	*	MAIN 585
C * IWET = WEIGHTING FUNCTION INDICATOR	*	MAIN 590
C * 0 = GALERKIN WEIGHTING	*	MAIN 595
C * 1 = UPSTREAM WEIGHTING	*	MAIN 600
C * ILUMF = MASS MATRIX LUMPING OPTIONS, 0=NO LUMPING, 1=YES	*	MAIN 605
C * IMID = MID-DIFFERENCE OPTION, 0=NO MID-DIFFERENCE, 1=YES	*	MAIN 610
C * ICONFI = IS CONFINED AQUIFER? 0=NO, 1=YES	*	MAIN 615
C * IOPT = WEIGHTING FACTOR OPTIMIZATION INDICATOR	*	MAIN 620

```

C * KSORF = SORPTION CONTROL * MAIN 625
C * 1 = LINEAR ISOTHERM * MAIN 630
C * 2 = FREUNDLICH ISOTHERM * MAIN 635
C * 3 = LANGMUIR ISOTHERM * MAIN 640
C * IPNTS = INDICATING IF POINTWISE ITERATION SOLUTION IS TO BE USED * MAIN 645
C * * * MAIN 650
C * * * MAIN 655
C * X(N) = X-COORDINATE OF N-TH NODE, (L) * MAIN 660
C * Z(N) = Z-COORDINATE OF N-TH NODE, (L) * MAIN 665
C * B(N) = BED ROCK ELEVATION OF N-TH NODE, (L) * MAIN 670
C * T(N) = THICKNESS OF THE AQUIFER AT NODE N, (L) * MAIN 675
C * IE(N,1) = GLOBAL NODE NUMBER OF THE FIRST NODE OF N-TH ELEMENT * MAIN 680
C * IE(N,2) = GLOBAL NODE NUMBER OF THE SECOND NODE OF N-TH ELEMENT * MAIN 685
C * IE(N,3) = GLOBAL NODE NUMBER OF THE THIRD NODE OF N-TH ELEMENT * MAIN 690
C * IE(N,4) = GLOBAL NODE NUMBER OF THE FOURTH NODE OF N-TH ELEMENT * MAIN 695
C * IF N-TH ELEMENT IS A TRIANGULAR ELEMENT, IE(N,4) = 0 * MAIN 700
C * IE(N,5) = INTEGER TO INDICATE THE MATERIAL TYPE OF N-TH ELEMENT * MAIN 705
C * IF N-TH ELEMENT IS UNCONFINED, IE(N,5) SHOULD BE (-) * MAIN 710
C * * * MAIN 715
C * C(N,I) = AN ARRAY TO STORE THE ASSEMBLED GLOBAL MATRIX * MAIN 720
C * ILNODE(I,N) = GLOBAL NODE NUMBER OF THE I-TH COMPRESSED LOWER-LEFT * MAIN 725
C * NODE TO THE N-TH GLOBAL NODE * MAIN 730
C * R(N) = CONCENTRATION AT THE PRESENT TIME STEP, (M/L**3) * MAIN 735
C * RP(N) = CONCENTRATION AT THE PREVIOUS TIME, (M/L**3) * MAIN 740
C * WEIAB(N,J) = WEIGHTING FACTOR FOR J-TH SIDE OF N-TH ELEMENT * MAIN 745
C * J = 1 = THE SIDE 1-2 * MAIN 750
C * J = 2 = THE SIDE 2-3 * MAIN 755
C * J = 3 = THE SIDE 4-3 * MAIN 760
C * J = 4 = THE SIDE 1-4 * MAIN 765
C * * * MAIN 770
C * * * MAIN 775
C * FX(N) = X-DIRECTION MATERIAL FLUX AT N-TH NODE, (M/T/L**2) * MAIN 780
C * FZ(N) = Z-DIRECTION MATERIAL FLUX AT N-TH NODE, (M/T/L**2) * MAIN 785
C * * * MAIN 790
C * * * MAIN 795
C * VX(N) = X-COMPONENT VELOCITY AT N-TH NODE, (L/T) * MAIN 800
C * VXP(N) = X-COMPONENT VELOCITY OF PREVIOUS TIME AT N-TH NODE, (L/T) * MAIN 805
C * VZ(N) = Z-COMPONENT VELOCITY AT N-TH NODE, (L/T) * MAIN 810
C * VZP(N) = Z-COMPONENT VELOCITY OF PREVIOUS TIME AT N-TH NODE, (L/T) * MAIN 815
C * H(N) = PRESSURE HEAD AT N-TH NODE, (L) * MAIN 820
C * HP(N) = PRESSURE HEAD OF N-TH NODE AT PREVIOUS TIME STEP, (L) * MAIN 825
C * DH(N) = (H(N) - HP(N))/DELT, (L/T) * MAIN 830
C * * * MAIN 835
C * * * MAIN 840
C * DLB(I) = LENGTH OF I-TH BOUNDARY ELEMENT SIDE, (L) * MAIN 845
C * DCOSXB(I) = X-DIRECTIONAL COSINE OF I-TH BOUNDARY ELEMENT SIDE * MAIN 850
C * DCOSZB(I) = Z-DIRECTIONAL COSINE OF I-TH BOUNDARY ELEMENT SIDE * MAIN 855
C * BFLX(I) = BOUNDARY FLUX AT I-TH BOUNDARY NODE, (M/T/L) * MAIN 860
C * BFLXF(I) = BOUNDARY FLUX AT PREVIOUS TIME OF I-TH BOUNDARY NODE * MAIN 865
C * NBE(I) = ELEMENT NUMBER OF I-TH BOUNDARY ELEMENT SIDE * MAIN 870
C * ISB(I,1) = GLOBAL NODE NUMBER OF THE FIRST NODE OF I-TH BOUNDARY * MAIN 875
C * ELEMENT SIDE * MAIN 880
C * ISB(I,2) = GLOBAL NODE NUMBER OF THE SECOND NODE OF I-TH BOUNDARY * MAIN 885
C * ELEMENT SIDE * MAIN 890
C * ISB(I,3) = LOCAL NODE NUMBER OF THE FIRST NODE OF I-TH BOUNDARY * MAIN 895
C * ELEMENT SIDE * MAIN 900
C * ISB(I,4) = LOCAL NODE NUMBER OF THE SECOND NODE OF I-TH BOUNDARY * MAIN 905
C * ELEMENT SIDE * MAIN 910
C * NPB(I) = GLOBAL NODE NUMBER OF I-TH BOUNDARY NODE * MAIN 915
C * * * MAIN 920
C * * * MAIN 925
C * PROP(I,J) = J-TH MATERIAL PROPERTY OF I-TH MATERIAL * MAIN 930
C * PROP(I,1) = DISTRIBUTION COEFFICIENT, (L**3/M) * MAIN 935
C * PROP(I,2) = BULK DENSITY, (M/L**3) * MAIN 940

```

```

C *      PROP(I,3) = LOGITUDINAL DISPERSIVITY, (L) *      HAIN 945
C *      PROP(I,4) = LATERAL DISPERSIVITY, (L) *      HAIN 950
C *      PROP(I,5) = DECAY CONSTANT, (1/T) *      HAIN 955
C *      PROP(I,6) = POROSITY, (DIMENSIONLESS IN DECIMAL POINT) *      HAIN 960
C *      PROP(I,7) = MODIFIED COMPRESSIBILITY OF MEDIA, (1/L) *      HAIN 965
C *      PROP(I,8) = MOLECULAR DIFFUSION COEFFICIENT, (L**2/T) *      HAIN 970
C *      PROP(I,9) = TORTUOSITY, (DIMENSIONLESS) *      HAIN 975
C *      PROP(I,10) = RATE CONSTANT IN DISSOLVED PHASE, (1/T) *      HAIN 980
C *      PROP(I,11) = RATE CONSTANT IN SORBED PHASE, (1/T) *      HAIN 985
C *      PROP(I,12) = FREUNDLICH K *      HAIN 990
C *      PROP(I,13) = FREUNDLICH N *      HAIN 995
C *      PROP(I,14) = LANGMUIR K *      HAIN1000
C *      PROP(I,15) = LANGMUIR SHAX *      HAIN1005
C *      *      HAIN1010
C *      *      HAIN1015
C * KPR(I) = LINE PRINTING FOR I-TH TIME STEP? *      HAIN1020
C *      0 = PRINT NOTHING *      HAIN1025
C *      1 = PRINT FLUXES THROUGH ALL TYPES OF BOUNDARIES *      HAIN1030
C *      2 = PRINT CONCENTRATION ALSO, *      HAIN1035
C *      3 = PRINT MATERIAL FLUX ALSO *      HAIN1040
C *      *      HAIN1045
C * KDSK(I) = STORE RESULTS ON LOGICAL UNI 2? 0=NO, 1=YES *      HAIN1050
C * TDTCH(I) = TIME OF I-TH TIME TO RESET TIME STEP SIZE = DELTO, (T) *      HAIN1055
C *      SHOULD BE DIMENSIONED AS TDCH(MXNDTC) *      HAIN1060
C *      *      HAIN1065
C *      *      HAIN1070
C * TSTAB(I,J) = TIME OF J-TH DATA POINT IN I-TH SOURCE VS TIME *      HAIN1075
C *      PROFILE, (T) *      HAIN1080
C * STAB(I,J) = SOURCE STRENGTH OF J-TH DATA POINT IN I-TH SOURCE VS *      HAIN1085
C *      TIME PROFILE, (M/T/L**2) IF STAB .GE.0.0; *      HAIN1090
C *      (L**3/T/L**2) IF STAB .LT.0.0. *      HAIN1095
C * SOS(I) = SOURCE STRENGTH AT THE PRESENT TIME OF I-TH PROFILE *      HAIN1100
C * ISTYP(M) = TYPE OF SOURCE PROFILE ASSIGNED TO M-TH ELEMENT *      HAIN1105
C *      *      HAIN1110
C * NPW(I) = GLOBAL NODE NUMBER OF I-TH WELL MODE *      HAIN1115
C * INTYP(I) = TYPE OF WELL SOURCE/SINK PROFILE ASSIGNED TO -TH WELL *      HAIN1120
C * WSS(I) = WELL SOURCE/SINK OF I-TH PROFILE, (M/T) OR L**3/T) *      HAIN1125
C * WSSF(I,J) = WELL SOURCE/SINK FUNCTION OF J-TH DATA POINT OF I-TH *      HAIN1130
C *      PROFILE, (M/T) IF +, (L**3/T) IF -. *      HAIN1135
C * TWSSF(I,J) = TIME OF J-TH DATA POINT OF I-TH WELL SOURCE/SINK *      HAIN1140
C *      PROFILE, (T) *      HAIN1145
C *      *      HAIN1150
C * DL(MP) = LENGTH OF MP-TH RS BOUNDARY ELEMENT SIDE, (L) *      HAIN1155
C * DCOSX(MP) = DIRECTIONAL COSINE WITH X-AXIS OF MP-TH RS SIDE *      HAIN1160
C * DCOSZ(MP) = DIRECTIONAL COSINE WITH Z-AXIS OF MP-TH RS SIDE *      HAIN1165
C * NRSE(MP) = ELEMENT NUMBER TO WHICH THE MP-TH SIDE BELONG *      HAIN1170
C * IS(MP,1) = GLOBAL NODE NUMBER OF THE FIRST NODE OF MP-TH RS SIDE *      HAIN1175
C * IS(MP,2) = GLOBAL NODE NUMBER OF THE SECOND NODE OF MP-TH RS SIDE *      HAIN1180
C * IS(MP,3) = LOCAL NODE NUMBER OF THE FIRST NODE OF MP-TH RS SIDE *      HAIN1185
C * IS(MP,4) = LOCAL NODE NUMBER OF THE SECOND NODE OF MP-TH RS SIDE *      HAIN1190
C * NPRS(MP) = GLOBAL MODAL POINT NUMBER OF MP-TH RS-MODE *      HAIN1195
C * TRSTAB(I,J) = TIME OF J-TH DATA POINT IN I-TH INCOMING CONCENTRA- *      HAIN1200
C *      TION VS TIME PROFILE, (T) *      HAIN1205
C * RSTAB(I,J) = CONCENTRATION OF INCOMING FLUID OF J-TH DATA POINT *      HAIN1210
C *      I-TH CONCENTRATION VS TIME PROFILE, (M/L**3) *      HAIN1215
C * CIN(I) = CONCENTRATION OF INCOMING FLUID AT THE PRESENT TIME OF I- *      HAIN1220
C *      CONCENTRATION VS TIME PROFILE, (M/L**3) *      HAIN1225
C *IRSTYP(MP) = TYPE OF INCOMING-FLUID-CONCENTRATION VS TIME PROFILE *      HAIN1230
C *      ASSIGNED TO MP-TH RS BOUNDARY ELEMENT SIDE *      HAIN1235
C *      *      HAIN1240
C * ***** NOTE: ON THE RS BOUNDARY, IT IS A NEUMANN CONDITION WHEN THE *      HAIN1245
C * ***** FLUID IS OUT FROM THE REGION AND DISPERSIVE FLUX IS *      HAIN1250
C * ***** ASSUMED TO BE ZERO. IF DISPERSIVE FLUX IS NOT ZERO *      HAIN1255
C * ***** FOR NEUMANN CONDITION, THE PROGRAM MUST BE MODIFIED. *      HAIN1260

```


[illegible]

```

COMMON /CSIZE/ MAXWP,MAXEL,MAXBNP,MAXBEL,MAXBW,MAXNTI,MAXMAT,MAXDTC
COMMON /CPIS/ JLE,JDIAG,JRB,JBAND
COMMON /CINTG/ WNP,NEL,NBN,NBEL,IBAND,INBP,IBALPB,SNAT,WCH,NDTCHG
COMMON /IDNGE/ WTI,WSTR,KSTL,KPRO,KDSK,KSS,KVI,WITER,WPITER
COMMON /RDNGE/ DELT,CHNG,DELMAX,THAX,DELTO,W,ONE,OMI,TOLA,TOLB
COMMON /OPTN/ IWET,ILUHP,INID,ICONFI,IOPF,KSORP,IPNTS,IVHL
COMMON /WET/ APHA1,APHA2,BETA1,BETA2
C
COMMON /CS/ HXSPE,HXSDE,HSPR,NSDP,HXWNP,HXWPR,HXWDP,HWNP,HWPR,HWDP
COMMON /RSEND/ HXSNP,HXSEL,HXSPR,HXSDP,HRSN,HXSEL,HXSPR,HXSDP
COMMON /DIRICH/ HXDNP,HXDPR,HXDDP,HDN,HDP,HDDP
COMMON /KUH/ HXHUP,HXHUDP,HXUP,HXUDP
COMMON /KLHL/ HXHLPR,HXHLDP,HHLPE,HHLDP
C
DIMENSION I(671),Z(671),B(671),T(671),IE(600,5)
C
DIMENSION C(671,25),R(671),RP(671),RW(671),WETAB(600,4)
DIMENSION RL(671),RI(671),LLNODE(9,671)
DIMENSION FX(671),FZ(671)
C
DIMENSION VX(671),VXP(671),VZ(671),VZP(671),H(671),HP(671),
> DH(671)
C
DIMENSION DLB(202),DCOSXB(202),DCOSZB(202),BPLX(203),BPLXP(203),
> NBE(202),ISB(202,4),NPB(203)
C
DIMENSION PROP(3,15)
C
DIMENSION KPR(500),KDSK(500),TITCH(20)
C
DIMENSION TSTAB(5,4),STAB(5,4),SOS(5),ISTYP(600)
DIMENSION TWSSF(6,4),WSSF(6,4),WSS(6),INTYP(10),NPN(10)
C
DIMENSION DL(50),DCOSX(50),DCOSZ(50),NRSE(50),IS(50,4),NPRS(51)
DIMENSION TRSTAB(3,16),RSTAB(3,16),CIN(3),IRSTYP(50)
C
DIMENSION IDTYP(51),NPD(51)
DIMENSION TDTAB(11,4),DTAB(11,4),CON(11)
C
DIMENSION THUTAB(2,4),HUTAB(2,4),HU(2),RKU(2)
DIMENSION TCUTAB(2,4),CUTAB(2,4),CU(2)
DIMENSION THTAB(2,4),HLTAB(2,4),HL(2),RKL(2)
DIMENSION TCLTAB(2,4),CLTAB(2,4),CL(2)
DIMENSION IHUTYP(600),IHLTYP(600)
C
DIMENSION PRATE(14),FLOW(14),TFLOW(14)
CC
C
C ----- PASS THE PROGRAM TO GM2DXZ
C
CALL GM2D(X,Z,IE,WETAB,C,R,RP,RW,VX,VXP,VZ,VZP,H,HP,DH,B,T,
1 FX,FZ,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,BPLX,BPLXP,
2 TSTAB,STAB,SOS,ISTYP,DL,DCOSX,DCOSZ,NRSE,IS,NPRS,TRSTAB,RSTAB,
2 CIN,IRSTYP,TDTAB,DTAB,CON,IDTYP,NPD,KPR,KDSK,TDICH,PROP,
3 PRATE,FLOW,TFLOW,THUTAB,HUTAB,HU,RKU,TCUTAB,CUTAB,CU,
4 THTAB,HLTAB,HL,RKL,TCLTAB,CLTAB,CL,IHUTYP,IHLTYP,
5 NPN,INTYP,WSS,WSSF,TWSSF,RL,RI,LLNODE)
C
STOP
END

BLOCK DATA
IMPLICIT REAL*8(A-H,O-Z)
COMMON /CSIZE/ MAXWP,MAXEL,MAXBNP,MAXBEL,MAXBW,MAXNTI,MAXMAT,MAXDTC
COMMON /CPIS/ JLE,JDIAG,JRB,JBAND

```

```

MAIN 1585
MAIN 1590
MAIN 1595
MAIN 1600
MAIN 1605
MAIN 1610
MAIN 1615
MAIN 1620
MAIN 1625
MAIN 1630
MAIN 1635
MAIN 1640
MAIN 1645
MAIN 1650
MAIN 1655
MAIN 1660
MAIN 1665
MAIN 1670
MAIN 1675
MAIN 1680
MAIN 1685
MAIN 1690
MAIN 1695
MAIN 1700
MAIN 1705
MAIN 1710
MAIN 1715
MAIN 1720
MAIN 1725
MAIN 1730
MAIN 1735
MAIN 1740
MAIN 1745
MAIN 1750
MAIN 1755
MAIN 1760
MAIN 1765
MAIN 1770
MAIN 1775
MAIN 1780
MAIN 1785
MAIN 1790
MAIN 1795
MAIN 1800
MAIN 1805
MAIN 1810
MAIN 1815
MAIN 1820
MAIN 1825
MAIN 1830
MAIN 1835
MAIN 1840
MAIN 1845
MAIN 1850
MAIN 1855
MAIN 1860
MAIN 1865
MAIN 1870
MAIN 1875
MAIN 1880
BLOC 005
BLOC 010
BLOC 015
BLOC 020

```

```

COMMON /CS/ HXSPR, HXSDF, NSPR, NSDP, HXWNP, HXWPR, HXWDP, HXWNP, HXWPR, HXWDP
COMMON /RSND/ HXRSP, HXRSEL, HXRSPR, HXRSDP, HRSN, HRSSEL, HRSRSP, HRSRDP
COMMON /DIRICH/ HXDNP, HXDPR, HXDOP, HDN, HDP, HDDP
COMMON /KUBU/ HXHUPR, HXHUDP, HXUPR, HXUDP
COMMON /KLHL/ HXHLPR, HXHLDP, HHLPR, HHLDP
DATA HAXNP, HAXEL, HAXBWP, HAXBEL, HAXBW/61, 600, 203, 202, 25/
DATA JLE, JDIAG, JRB, JBAND/4, 5, 6, 9/
DATA HAXMTI, HAXHAT, HXDTC/500, 3, 20/
DATA HXSPR, HXSDF/5, 4/
DATA HXWNP, HXWDP, HXWNP/6, 4, 10/
DATA HXRSP, HXRSEL, HXRSPR, HXRSDP/51, 50, 3, 16/
DATA HXDNP, HXDPR, HXDOP/51, 11, 4/
DATA HXHUPR, HXHUDP, HXHLPR, HXHLDP/2, 4, 2, 4/
END
BLOC 025
BLOC 030
BLOC 035
BLOC 040
BLOC 045
BLOC 050
BLOC 055
BLOC 060
BLOC 065
BLOC 070
BLOC 075
BLOC 080
BLOC 085
BLOC 090

SUBROUTINE GM2D(X, Z, IE, WETAB, C, R, RP, RW, VX, VXP, VZ, VZP, H, HP, DH, B, T,
1 FX, FZ, DLB, DCOSXB, DCOSZB, NBE, ISB, NPB, BPLX, BFLXP,
2 TSTAB, STAB, SOS, ISTYP, DL, DCOSX, DCOSZ, NRSE, IS, NPS, TRSTAB, RSTAB,
3 CIN, IRSTYP, TDTAB, DTAB, CON, IDTYP, NPD, KPR, KDSK, TDICH, PROP,
4 PRATE, FLOW, TFLOW, THUTAB, HUTAB, HU, RKU, TCUTAB, CUTAB, CU,
5 THLTAB, HLTAB, HL, RKL, TCLTAB, CLTAB, CL, IHUTYP, IHLTYP,
MFW, INTYP, WSS, WSSF, TWSSF, RL, RI, LLNODE)
GM2D 005
GM2D 010
GM2D 015
GM2D 020
GM2D 025
GM2D 030
GM2D 035
GM2D 040
GM2D 045
GM2D 050
GM2D 055
GM2D 060
GM2D 065
GM2D 070
GM2D 075
GM2D 080
GM2D 085
GM2D 090

IMPLICIT REAL*8 (A-H, O-Z)
GM2D 095
GM2D 100
GM2D 105
GM2D 110
GM2D 115
GM2D 120
GM2D 125
GM2D 130
GM2D 135
GM2D 140
GM2D 145
GM2D 150
GM2D 155
GM2D 160
GM2D 165
GM2D 170
GM2D 175
GM2D 180
GM2D 185
GM2D 190
GM2D 195
GM2D 200
GM2D 205
GM2D 210
GM2D 215
GM2D 220
GM2D 225
GM2D 230
GM2D 235
GM2D 240
GM2D 245

COMMON /CSIZE/ HAXNP, HAXEL, HAXBWP, HAXBEL, HAXBW, HAXMTI, HAXHAT, HXDTC
COMMON /CPIS/ JLE, JDIAG, JRB, JBAND
COMMON /CINTG/ NWP, NEL, NBN, NBEL, IBAND, IHBP, IHALPB, NHAT, NCH, NDTCHG
COMMON /IDHGH/ MTI, MSTR, KSTR, KPRO, KDSKJ, KSS, KVI, NITER, NPITER
COMMON /RDHGH/ DELT, CHNG, DELHAX, THAX, DELTO, W, ONE, ONI, TOLA, TOLB
COMMON /OPTN/ IWET, ILUMP, INID, ICONFI, IOPT, KSOPR, IPNTS, IVHL
COMMON /WET/ ALPHA1, ALPHA2, BETA1, BETA2

COMMON /CS/ HXSPR, HXSDF, NSPR, NSDP, HXWNP, HXWPR, HXWDP, HXWNP, HXWPR, HXWDP
COMMON /RSND/ HXRSP, HXRSEL, HXRSPR, HXRSDP, HRSN, HRSSEL, HRSRSP, HRSRDP
COMMON /DIRICH/ HXDNP, HXDPR, HXDOP, HDN, HDP, HDDP
COMMON /KUBU/ HXHUPR, HXHUDP, HXUPR, HXUDP
COMMON /KLHL/ HXHLPR, HXHLDP, HHLPR, HHLDP

DIMENSION TITLE(9)
DIMENSION X(HAXNP), Z(HAXNP), B(HAXNP), IE(HAXEL, 5), WETAB(HAXEL, 4)
DIMENSION C(HAXNP, HAXBW), R(HAXNP), RP(HAXNP), RW(HAXNP)
DIMENSION RL(HAXNP), RI(HAXNP), LLNODE(JBAND, HAXNP)
DIMENSION FX(HAXNP), FZ(HAXNP)
DIMENSION VX(HAXNP), VXP(HAXNP), VZ(HAXNP), VZP(HAXNP), H(HAXNP),
> HF(HAXNP), DH(HAXNP), T(HAXNP)
DIMENSION DLB(HAXBEL), DCOSXB(HAXBEL), DCOSZB(HAXBEL), BPLX(HAXBWP),
> BFLXP(HAXBWP), NBE(HAXBEL), ISB(HAXBEL, 4), NPB(HAXBWP)
DIMENSION TSTAB(HXSPR, HXSDF), STAB(HXSPR, HXSDF)
DIMENSION SOS(HXSPR), ISTYP(HAXEL)
DIMENSION WSS(HXWPR), WSSF(HXWPR, HXWDP), TWSSF(HXWPR, HXWDP)
DIMENSION NPN(HXWNP), INTYP(HXWNP)
DIMENSION DL(HXRSEL), DCOSX(HXRSEL), DCOSZ(HXRSEL)
DIMENSION NRSE(HXRSEL), IS(HXRSEL, 4), NPS(HXRSP)
DIMENSION TRSTAB(HXRSPR, HXRSDP), RSTAB(HXRSPR, HXRSDP)
DIMENSION CIN(HXRSPR), IRSTYP(HXRSEL)
DIMENSION TDTAB(HXDPR, HXDOP), DTAB(HXDPR, HXDOP), CON(HXDPR)
DIMENSION IDTYP(HXDNP), NPD(HXDNP)

```

C	DIMENSION THOTAB(MXHUPR,MXHUDP),HUTAB(MXHUPR,MXHUDP)	GN2D 250
	DIMENSION HU(MXHUPR),RKU(MXHUPR)	GN2D 255
	DIMENSION TCUTAB(MXHUPR,MXHUDP),CUTAB(MXHUPR,MXHUDP),CU(MXHUPR)	GN2D 260
	DIMENSION THLTAB(MXHLPB,MXHLDP),HLTAB(MXHLPB,MXHLDP)	GN2D 265
	DIMENSION HL(MXHLPB),RKL(MXHLPB)	GN2D 270
	DIMENSION TCLTAB(MXHLPB,MXHLDP),CLTAB(MXHLPB,MXHLDP),CL(MXHLPB)	GN2D 275
	DIMENSION IHUTYP(MAXEL),IFUTYP(MAXEL)	GN2D 280
C		GN2D 285
	DIMENSION PEOP(MAXHAT,15)	GN2D 290
	DIMENSION KPR(MAXHTI),KDSK(MAXHTI),TDTCH(MXDTC)	GN2D 295
	DIMENSION PRATE(14),FLOW(14),TFLOW(14)	GN2D 300
C		GN2D 305
C	----- INPUT PROBLEM IDENTIFICATION AND DESCRIPTION	GN2D 310
C		GN2D 315
	100 READ 10, NPROB, (TITLE(I), I=1,9), IBUG	GN2D 320
	IF (NPROB.LE.0) GO TO 990	GN2D 325
	PRINT 1000, NPROB, (TITLE(I), I=1,9), IBUG	GN2D 330
C		GN2D 335
C	----- READ AND PRINT INPUT BY CALLING DATAIN	GN2D 340
C		GN2D 345
	KOUT=0	GN2D 350
C		GN2D 355
	CALL DATAIN(X,Z,B,T,IE,DLB,DCOSIB,DCOSZB,NBE,ISB,NPB,	GN2D 360
	1 TSTAB,STAB,ISTYP, DL,DCOSI,DCOSZ,NRSE,IS,NPRS, TRSTAB,RSTAB,	GN2D 365
	2 IESTYP, TDTAB,DTAB,IDTYP,NPD, RP,PROP, KPR,KDSK,MAXDIF, ISTOP,	GN2D 370
	3 THUTAB,HUTAB,RKU,TCUTAB,CUTAB, THLTAB,HLTAB,RKL,TCLTAB,CLTAB,	GN2D 375
	4 IHUTYP,IHLTYP,TDTCH, NPW,IWTP,WSSP,TWSSP)	GN2D 380
C		GN2D 385
	IF (IPNTS.EQ.1) CALL LDGEN(LLNODE,IE,MAXNP,MAXEL,NWP,NEL)	GN2D 390
C		GN2D 395
	KDIG=NSTE	GN2D 400
C		GN2D 405
	IF (ISTOP.GT.0) GO TO 950	GN2D 410
C		GN2D 415
C	----- COMPUTE BAND-WIDTH VARIABLES	GN2D 420
C		GN2D 425
	IHALPB=MAXDIF	GN2D 430
	IBAND=2*IHALPB+1	GN2D 435
	IHBP=IHALPB+1	GN2D 440
	IF (IBAND.GT.MAXBW .AND. IPNTS.EQ.C) GO TO 900	GN2D 445
C		GN2D 450
C	----- PREPARE INITIAL OR PRE-INITIAL VARIABLES	GN2D 455
C		GN2D 460
C	----- CALL INTERP TO OBTAIN SOS, CIN AND CON AT THE PRESENT TIME	GN2D 465
C		GN2D 470
	TIME=0.	GN2D 475
	CALL INTERP(SOS,TSTAB,STAB,TIME,MXSPR,MXSDP,NSPR,NSDP)	GN2D 480
	IF (NWP.NE.0) CALL INTERP(WSS,TWSSP,WSSP,TIME,MXWPR,MXWDP,	GN2D 485
	1 NWPR,NWDP)	GN2D 490
	CALL INTERP(CON,TDTAB,DTAB,TIME,MXDPR,MXDDP,N DPR,NDDP)	GN2D 495
	IF (NRSEL.GT.0) CALL INTERP(CIN,TRSTAB,RSTAB,TIME,	GN2D 500
	1 MXRSPR,MXRSDP,NRSPR,NRSDP)	GN2D 505
	CALL INTERP(CU,TCUTAB,CUTAB,TIME,MXHUPR,MXHUDP,NHUPR,NHUDP)	GN2D 510
	CALL INTERP(CL,TCLTAB,CLTAB,TIME,MXHLPR,MXHLDP,NHLPR,NHLDP)	GN2D 515
C		GN2D 520
C	----- APPLY DIRICHLET BOUNDARY SPECIFICATIONS TO INITIAL CONDITIONS	GN2D 525
C		GN2D 530
	DO 105 NPP=1,NBN	GN2D 535
	NP=NFD(NPP)	GN2D 540
	ITYP=IDTYP(NPP)	GN2D 545
	RP(NP)=CON(ITYP)	GN2D 550
	105 CONTINUE	GN2D 555
C		GN2D 560
		GN2D 565

C	----- INITIATE VI, VZ, H, AND R	GH2D 570
C	DO 110 NP=1, NNP	GH2D 575
	VX(NP)=0.0	GH2D 580
	VZ(NP)=0.0	GH2D 585
	H(NP)=0.0	GH2D 590
	R(NP)=RP(NP)	GH2D 595
		GH2D 600
	110 CONTINUE	GH2D 605
C	IF (KVI.LT.0) GO TO 200	GH2D 610
	IF (KVI.GT.0) GO TO 300	GH2D 615
C		GH2D 620
C	----- READ VELOCITY AND PRESSURE VIA CARDS IF KVI .EQ. 0	GH2D 625
C		GH2D 630
	CALL READR(H,MAXPP,NNP)	GH2D 635
C		GH2D 640
C	----- READ VELOCITY VIA CARDS IF KVI .LT. 0	GH2D 645
C		GH2D 650
	200 CONTINUE	GH2D 655
	210 READ 20, NI, NSEQ, NAD, VXNI, VZNI, VIAD, VZAD	GH2D 660
	IF (NI.EQ.0) GO TO 250	GH2D 665
	NJ=NI+NSEQ	GH2D 670
	DO 220 NP=NI, NJ	GH2D 675
	I=NI+(NP-NI)*NAD	GH2D 680
	VX(I)=VXNI+VIAD*DFLOAT(NP-NI)	GH2D 685
	VZ(I)=VZNI+VZAD*DFLOAT(NP-NI)	GH2D 690
	220 CONTINUE	GH2D 695
	GO TO 210	GH2D 700
	250 CONTINUE	GH2D 705
C		GH2D 710
	CALL INTERP(HU,THUTAB,HUTAB,TIME,MXHOPR,MXHUDP,MHOPR,MHUDP)	GH2D 715
	CALL INTERP(HL,THLTAB,HLTAB,TIME,MXHLPR,MXHLDP,MHLPR,MHLDP)	GH2D 720
C		GH2D 725
	GO TO 400	GH2D 730
C		GH2D 735
C	----- READ INITIAL VELOCITY AND PRESSURE VIA LOGICAL UNIT 1	GH2D 740
C		GH2D 745
	300 READ (1) TIMEN, (H(N), N=1, NNP), (VX(N), N=1, NNP), (VZ(N), N=1, NNP),	GH2D 750
	1 (HU(I), I=1, MHOPR), (HL(I), I=1, MHLPR)	GH2D 755
C		GH2D 760
	400 DO 410 NP=1, NNP	GH2D 765
	VXF(NP)=VX(NP)	GH2D 770
	VZF(NP)=VZ(NP)	GH2D 775
	410 CONTINUE	GH2D 780
C		GH2D 785
	CALL AFABTA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,	GH2D 790
	> MAXMAT, NEL, W, KSS, IOPT)	GH2D 795
C		GH2D 800
	KLINE=0	GH2D 805
	DO 420 NP=1, NEL, 2	GH2D 810
	KLINE=KLINE+1	GH2D 815
	IF (MOD(KLINE-1,50).EQ.0) PRINT 2000	GH2D 820
	NJEN=NP	GH2D 825
	NJEX=MIND(NP+1, NEL)	GH2D 830
	PRINT 2100, (NJ, (WETAB(NJ, I), I=1, 4), NJ=NJEN, NJEX)	GH2D 835
	420 CONTINUE	GH2D 840
C		GH2D 845
	CALL FLOW(X,Z,IE,WETAB,C,FX,FZ,RP,VX,VZ,PROP,MAXNP,MAXEL,MAXBW,	GH2D 850
	> MAXMAT)	GH2D 855
C		GH2D 860
	KFLOW=-1	GH2D 865
C		GH2D 870
	CALL SFLOW(X,Z,IE,WETAB,RP,FX,FZ, PROP,DLB,DCOSXB,DCOSZB,	GH2D 875
	1 NEE,ISB,NPB,BFLX,BFLXP, NPRS,RPD, PRATE,FLOW,TFLOW, MAXNP,MAXEL,	GH2D 880
		GH2D 885

```

2 MAXNP,MAXREL,MAXMAT,DELT,DR,KFLOW,B,T,H,SOS,ISTYP,WSS,INTYP,NPW, GH2D 890
3 HU,RKU,CU,INTYP,HL,RKL,CL,IHLTYP,MXHUPR,MXHLPR) GH2D 895
C DO 430 I=1,14 GH2D 900
IF (I.EQ.6 .OR. I.EQ.7) GO TO 430 GH2D 910
PLCW(I)=0.0 GH2D 915
TFLOW(I)=0.0 GH2D 920
430 CONTINUE GH2D 925
PLCW(6)=0.0 GH2D 930
PLCW(7)=0.0 GH2D 935
C GH2D 940
C ----- PRINT INITIAL OR PRE-INITIAL VARIABLES GH2D 945
C GH2D 950
KDIAG=-2 GH2D 955
C GH2D 960
CALL PRINTT(RP,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPRO,KOUT,KDIAG, GH2D 965
> MAXNP,NWP,IBAND, -1) GH2D 970
C GH2D 975
IF (KSTR.EQ.1 .AND. KDSKO.EQ.1 .AND. KSS.EQ.1 .AND. NSTR.EQ.0) CALL GH2D 980
1 STORE (X,Z,IE,RP,FX,FZ,TITLE,NPROB,NWP,NEL,NTI,MAXNP,MAXEL,TIME) GH2D 985
C GH2D 990
IF (KSS.EQ.1) GO TO 600 GH2D 995
C GH2D 1000
C ----- PERFORM STEADY STATE COMPUTATION GH2D 1005
C GH2D 1010
DO 440 NP=1,NWP GH2D 1015
EP(NP)=H(NP) GH2D 1020
DH(NP)=0.0 GH2D 1025
RW(NP)=OML*R(NP)+(1.0D0-OML)*RP(NP) GH2D 1030
IF (IENTS.EQ.1) RI(NP)=RW(NP) GH2D 1035
440 CONTINUE GH2D 1040
C GH2D 1045
KDIG=KDIG+1 GH2D 1050
IF (IBUG.NE.0) PRINT 111C, KDIG,TIME,DELT GH2D 1055
IF (IBUG.NE.0) PRINT 1120 GH2D 1060
C GH2D 1065
DO 540 ITER=1,NITER GH2D 1070
C GH2D 1075
CALL ASSEMBL(X,Z,IE,WETAB,C,R,RP,RW,VX,VXP,VZ,VZP,B,T,H,DR, GH2D 1080
1 PROP,W,MAXNP,MAXEL,MAXBW,MAXMAT, SOS,ISTYP, NPW,INTYP,WSS, GH2D 1085
2 HU,RKU,CU,INTYP,HL,RKL,CL,IHLTYP, MXHUPR,MXHLPR, KSS,DELT, GH2D 1090
3 LINCDE) GH2D 1095
C GH2D 1100
CALL BC(IE,WETAB,C,R,RP, VX,VXP,VZ,VZP, CON,IDTYP,NPD, GH2D 1105
> DL,DCOSX,DCOSZ,NRSE,IS,CIN,IRSTYP, W, MAXNP,MAXEL,MAXBW, KSS, GH2D 1110
> LINCDE) GH2D 1115
C GH2D 1120
IF (IENTS.EQ.1) GO TO 500 GH2D 1125
C GH2D 1130
CALL SOLVE(1,C,R,NWP,IHALPB,MAXNP,MAXBW) GH2D 1135
CALL SOLVE(2,C,R,NWP,IHALPB,MAXNP,MAXBW) GH2D 1140
GO TO 510 GH2D 1145
C GH2D 1150
500 CALL PISS(R,RL,RI,C,OMI,TOIA,LLNODE,NWP,NPITER,MAXNP,MAXBW,IBUG, GH2D 1155
1 1) GH2D 1160
C GH2D 1165
510 CONTINUE GH2D 1170
DIFMAX=0.0 GH2D 1175
NOCCUR=1 GH2D 1180
DO 520 NP=1,NWP GH2D 1185
IF (RW(NP).EQ.0.0) GO TO 520 GH2D 1190
DIF=(R(NP)-RW(NP))/RW(NP) GH2D 1195
DIF=DABS(DIF) GH2D 1200

```

IF (DIF.LE.DIFMAX) GO TO 520	GM2D 1205
DIFMAX=DIF	GM2D 1210
NOCCUR=NP	GM2D 1215
520 CONTINUE	GM2D 1220
C	GM2D 1225
C ----- UPDATE NONLINEAR ITERATE	GM2D 1230
C	GM2D 1235
DO 530 NP=1,NNP	GM2D 1240
RW(NP)=ONE*R(NP)+(1.0D0-ONE)*RW(NP)	GM2D 1245
530 IF (IPNTS.EQ.1) RI(NP)=RW(NP)	GM2D 1250
C	GM2D 1255
IF (NITER.EQ.1) GO TO 550	GM2D 1260
IF (IBUG.NE.0) PRINT 1200, ITER,DIFMAX,TOLA,NOCCUR	GM2D 1265
IF (ITER.EQ.1) GO TO 540	GM2D 1270
IF (DIFMAX.LE.TOLA) GO TO 550	GM2D 1275
C	GM2D 1280
540 CONTINUE	GM2D 1285
C	GM2D 1290
WRITE(6,1210) ITER,NITER,DIFMAX,TOLA,NOCCUR	GM2D 1295
C	GM2D 1300
550 DO 555 NP=1,NNP	GM2D 1305
555 RP(NP)=R(NP)	GM2D 1310
C	GM2D 1315
CALL FLUX(X,Z,IE,WETAB,C,FX,FZ,RP,VX,VZ,PROP,MAXNP,MAXEL,MAXBW, > MAXMAT)	GM2D 1320
C	GM2D 1325
KFLOW=0	GM2D 1330
C	GM2D 1335
CALL SFLOW(X,Z,IE,WETAB,RP,FX,FZ, PROP,DLB,DCOSXB,DCOSZB, 1 NBE,ISB,NPB,BFLX,BFLXP, NPRS,NPD, FRATE,FLOW,TFLOW, MAXNP,MAXEL, 2 MAXBNP,MAXBEL,MAXMAT,DELT,DH,KFLOW,B,T,H,SOS,ISTYP,WSS,IWTYP,NPW, 3 HU,RKU,CU,IHUTYP, HL,BKL,CL,IHLTYP, MXHUPR,MXHLPR)	GM2D 1340
C	GM2D 1345
DO 560 I=1,14	GM2D 1350
IF (I.EQ.6 .OR. I.EQ.7) GO TO 560	GM2D 1355
FLCW(I)=0.0	GM2D 1360
TFLOW(I)=0.0	GM2D 1365
560 CONTINUE	GM2D 1370
FLCW(6)=0.0	GM2D 1375
FLCW(7)=0.0	GM2D 1380
C	GM2D 1385
C ----- PRINT STEADY STATE VARIABLES	GM2D 1390
C	GM2D 1395
KDIAG=-1	GM2D 1400
CALL PRINTT(RP,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPRO,KOUT,KDIAG, > MAXNP,NNP,IBAND, 0)	GM2D 1405
C	GM2D 1410
IF (KSTR.EQ.1 .AND. KDSKO.EQ.1) CALL STORE(X,Z,IE, > RF,FX,FZ,TITLE, NPROB,NNP,WEL,NTI,MAXNP,MAXEL,TIME)	GM2D 1415
C	GM2D 1420
IF (NTI.EQ.0) GO TO 100	GM2D 1425
C	GM2D 1430
KS=1	GM2D 1435
C	GM2D 1440
C ----- PERFORM TRANSIENT-STATE OR TRANSIENT COMPUTATION	GM2D 1445
C	GM2D 1450
600 TIME=DELT	GM2D 1455
W1=W	GM2D 1460
W2=1.-W	GM2D 1465
KFLOW=1	GM2D 1470
TFLOW(6)=0.0	GM2D 1475
TFLOW(7)=0.0	GM2D 1480
KDIAG=0	GM2D 1485
IDELT=0	GM2D 1490
C	GM2D 1495
	GM2D 1500
	GM2D 1505
	GM2D 1510
	GM2D 1515
	GM2D 1520
	GM2D 1525

```

C ***** BEGIN TIME LOOP *****
C
C      DO 890 ITH=1,NTI
C
C      ----- COMPUTE SOURCE AND B. C. VALUE AT THE PRESENT TIME
C
C      ----- CALL INTERP TO OBTAIN SOS, CIN, AND CON AT THE PRESENT TIME
C
C      CALL INTERP(SOS,TSTAB,STAB,TIME,MXSPR,MXSDP,MSPR,MSDP)
C      IF (MWNP.NE.0) CALL INTERP(WSS,TWSSF,WSSF,TIME,MXWPR,MXWDP,
C      1 MWPR,MWDP)
C      CALL INTERP(CON,TDTAB,DTAB,TIME,MXDPE,MXDDP,M DPR,MDDP)
C      IF (MRSEL.GT.0) CALL INTERP(CIN,TRSTAB,RSTAB,TIME,
C      1 MXRSPR,MXRSDP,MRSPE,MRSDP)
C      CALL INTERP(CU,TCUTAB,CUTAB,TIME,MXHUPR,MXHUDP,MHUPR,MHUDP)
C      CALL INTERP(CL,TCLTAB,CLTAB,TIME,MXHLPR,MXHLDP,MHLPR,MHLDP)
C
C      ----- READ TIME-DEPENDENT VELOCITIES IF REQUIRED
C
C      DO 610 NP=1,NNP
C      VXF(NP)=VX(NP)
C      VZF(NP)=VZ(NP)
C      610 HP(NP)=H(NP)
C
C      KDIG=KDIG+1
C      IF (IBUG.NE.0 .AND. KPR(ITH).NE.0) PRINT 1110, KDIG, TIME, DELT
C      IF (IBUG.NE.0 .AND. KPR(ITH).NE.0) PRINT 1120
C
C      IF (KVI.NE.2) GO TO 620
C
C      READ (1) TIMEN, (H(NP), NP=1, NNP), (VX(NP), NP=1, NNP),
C      >      (VZ(NP), NP=1, NNP), (HU(I), I=1, MHUPR), (HL(I), I=1, MHLPR)
C
C      CALL AFABTA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,
C      >      MAXHAT, NEL, W, KSS, IOPT)
C
C      620 DO 640 NP=1,NNP
C      DH(NP)=(H(NP)-HP(NP))/DELT
C      RP(NP)=R(NP)
C      RW(NP)=OMR*R(NP)+(1.0DO-OMR)*RP(NP)
C      IF (IPNTS.EQ.1) RI(NP)=RW(NP)
C      640 CONTINUE
C
C      DO 740 ITER=1, NITER
C
C      ----- ASSEMBLE COEFFICIENT MATRICES, A, B, AND C, AND CONSTRUCT
C      ----- LOAD VECTOR R
C
C      CALL ASEMBL(X,Z,IE,WETAB, C,R,RP,RW, VX,VXP,VZ,VZP,B,T,H,DH,
C      1 PROP,W,MAXNP,MAXEL,MAXBW,MAXHAT, SOS,ISTYP, NPW,INTYP,WSS,
C      2 HU,RKU,CU,IHUTYP, HL,RKL,CL,IHLTYP, MXHUPR,MXHLPR, KSS,DELT,
C      3 LLNODE)
C
C      ----- APPLY BOUNDARY CONDITIONS
C
C      CALL BC(IE,WETAB, C,R,RP, VX,VXP,VZ,VZP, CON,IDTYP,NPD,
C      >      DL,DCOSX,DCOSZ,MRSE,IS, CIN,IRSTYP, W, MAXNP,MAXEL,MAXBW,KSS,
C      >      LLNODE)
C
C      IF (IPNTS.EQ.1) GO TO 700
C
C      ----- SOLVE THE MATRIX EQUATION WITH DIRECT ELIMINATION.
C
C      CALL SOLVE(I,C,R,NNP,IALPB,MAXNP,MAXBW)

```

GH2D 1530
 GH2D 1535
 GH2D 1540
 GH2D 1545
 GH2D 1550
 GH2D 1555
 GH2D 1560
 GH2D 1565
 GH2D 1570
 GH2D 1575
 GH2D 1580
 GH2D 1585
 GH2D 1590
 GH2D 1595
 GH2D 1600
 GH2D 1605
 GH2D 1610
 GH2D 1615
 GH2D 1620
 GH2D 1625
 GH2D 1630
 GH2D 1635
 GH2D 1640
 GH2D 1645
 GH2D 1650
 GH2D 1655
 GH2D 1660
 GH2D 1665
 GH2D 1670
 GH2D 1675
 GH2D 1680
 GH2D 1685
 GH2D 1690
 GH2D 1695
 GH2D 1700
 GH2D 1705
 GH2D 1710
 GH2D 1715
 GH2D 1720
 GH2D 1725
 GH2D 1730
 GH2D 1735
 GH2D 1740
 GH2D 1745
 GH2D 1750
 GH2D 1755
 GH2D 1760
 GH2D 1765
 GH2D 1770
 GH2D 1775
 GH2D 1780
 GH2D 1785
 GH2D 1790
 GH2D 1795
 GH2D 1800
 GH2D 1805
 GH2D 1810
 GH2D 1815
 GH2D 1820
 GH2D 1825
 GH2D 1830
 GH2D 1835
 GH2D 1840
 GH2D 1845

CALL SOLVE (2,C,R,NWP, IHALPB, MAXNP, MAXBW)	GH2D1850
GO TO 710	GH2D1855
C ----- SOLVE THE MATRIX EQUATION WITH POINTWISE ITERATION STRATEGIES.	GH2D1860
C	GH2D1865
700 CALL PISS (R,RL,RI,C,OMI,TOLB,LLNODE,NWP,NPITER,MAXNP,MAXBW,IBUG,	GH2D1870
1 KER (ITM))	GH2D1875
C	GH2D1880
710 CONTINUE	GH2D1885
DIFMAX=0.0	GH2D1890
NOCCUR=1	GH2D1895
DO 720 NP=1,NWP	GH2D1900
IF (RW(NP).EQ.0.0) GO TO 720	GH2D1905
DIF= (R(NP)-RW(NP))/RW(NP)	GH2D1910
DIF=DABS(DIF)	GH2D1915
IF (DIF.LE.DIFMAX) GO TO 720	GH2D1920
DIFMAX=DIF	GH2D1925
NOCCUR=NP	GH2D1930
720 CONTINUE	GH2D1935
C	GH2D1940
C ----- UPDATE THE NONLINEAR ITERATE.	GH2D1945
C	GH2D1950
DO 730 NP=1,NWP	GH2D1955
RW(NP)=ONE*R(NP)+(1.0D0-ONE)*RW(NP)	GH2D1960
730 IF (IPNTS.EQ.1) RI(NP)=RW(NP)	GH2D1965
C	GH2D1970
IF (NITER.EQ.1) GO TO 750	GH2D1975
IF (IBUG.NE.0.AND.KPR(ITM).NE.0) PRINT 1200,ITER,DIFMAX,TOLB,NOCCUR	GH2D1980
IF (ITER.EQ.1) GO TO 740	GH2D1985
IF (DIFMAX.LT.TOLB) GO TO 750	GH2D1990
C	GH2D1995
740 CONTINUE	GH2D2000
C	GH2D2005
WRITE(6,9100) ITM,ITER,NITER,DIFMAX,TOLB,NOCCUR	GH2D2010
C	GH2D2015
750 CONTINUE	GH2D2020
C	GH2D2025
IF (IMID.EQ.0) GO TO 850	GH2D2030
C	GH2D2035
DO 810 NP=1,NWP	GH2D2040
810 R(NP)=2.0D0*R(NP)-RP(NP)	GH2D2045
DO 820 NPP=1,NDN	GH2D2050
NP=NED(NPP)	GH2D2055
ITYP=IDTYP(NPP)	GH2D2060
R(NP)=CON(ITYP)	GH2D2065
820 CONTINUE	GH2D2070
C	GH2D2075
850 CONTINUE	GH2D2080
C	GH2D2085
C ----- CALCULATE MATERIAL FLUX FX AND FZ	GH2D2090
C	GH2D2095
CALL FLOX(X,Z,IE,WETAB,C,FX,FZ,R,VX,VZ,PROP,MAXNP,MAXEL,	GH2D2100
> MAXBW,MAXMAT)	GH2D2105
C	GH2D2110
C ----- DETERMINE FLUX THROUGH ALL BOUNDARIES	GH2D2115
C	GH2D2120
CALL SPLOW(X,Z,IE,WETAB,R,FX,FZ,PROP,DLB,DCOSXB,DCOSZB,	GH2D2125
1 NEE,ISB,NPB,BFLX,BFLXP,NPRS,NPD,FRATE,FLOW,TFLOW,MAXNP,MAXEL,	GH2D2130
2 MAXBNP,MAXBFL,MAXMAT,DELT,DH,KFLOW,B,T,H,SOS,ISTYP,WSS,IWTYP,NPW,	GH2D2135
3 HU,RKU,CU,IHUTYP,HL,RKL,CL,IHLTYP,MXHUPR,MXHLPR)	GH2D2140
C	GH2D2145
C ----- PRINT VARIABLES AT CURRENT TIME STEP	GH2D2150
C	GH2D2155
CALL PRINTT(R,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPR(ITM),KOUT,	GH2D2160
> KDIAG,MAXNP,NWP,IEAND,ITM)	GH2D2165
	GH2D2170

```

C      IF (KSTR.EQ.1 .AND. KDSK(ITH).EQ.1) CALL STORE(X,Z,IE,B,PX,PZ,
1      TITLE,NPROB,NWP,NEL,NTI,MAXNP,MAXEL,TIME)
C
C      ----- PREPARE FOR NEXT TIME STEP
C
      IF (TIME.GE.TMAX) GO TO 160
      DELT=DELT*(1.000+CHNG)
      DELT=DMIN1(DELT,DELHAX)
      IF (TIME.EQ.TDTCH(IDELT+1)) DELT=DELTO
      TIME=TIME+DELT
      IF (TIME.LE.TDTCH(IDELT+1)) GO TO 860
      IDELT=IDELT+1
      TIME=TIME-DELT
      DELT=TDTCH(IDELT)-TIME
      IF (DELT.LE.0.0) DELT=DELTO
      TIME=TIME+DELT
C
C      860 CONTINUE
C
C      ----- UPDATE RP
C
      DO 870 NP=1,NWP
      870 RP(NP)=R(NP)
C
C      890 CONTINUE
C
C      ----- END OF TIME LOOP, START A NEW JOB
C
      GO TO 100
      900 PRINT 3000, IBAND,MAXBW
      950 PRINT 3100, ISTOP
      990 RETURN
C
      10 FORMAT (I5,9A8,I3)
      20 FORMAT (3I5,5X,4F10.0)
C
      1000 FORMAT (/8H1PROBLEM,I5,3H..,9A8,I3/)
      1110 FORMAT (IH1,52H*****
> 62H*****
> 5H****//17H DIAGNOSTIC TABLE,I4,12H.. AT TIME =,1PD12.4,
> 9H ,(DELT = 1PD12.4,1H))
      1120 FORMAT (///30H TABLE OF ITERATIVE PARAMETERS// 6X,
> 9HITERATION,7X,8H MAX DIP,6X,9HTOLERANCE,6X,
> 19H MAX OCCUPANCE NODE)
      1200 FORMAT (5X,11C,3X,E12.4,3X,E12.4,15X,I10)
      1210 FORMAT (IH0/5X,'** WARNING: NO CONVERGENCE AFTER',I4,' ITERATIONS',
1/8X,'NITER =' ,I4,' DIFMAX =' ,D12.4,' TOLA =' ,D12.4,' NOCCUR =' ,I4)
      2060 FORMAT (IH1,'TABLE OF WEIGHTING FACTORS OF EVERY ELEMENTS'//)
      2100 FORMAT (IH ,I5,4E12.4,5X,I5,4E12.4)
      3000 FORMAT (///12H BANDWIDTH =,I4,25H EXCEEDS MAX. ALLOWABLE =,I4//)
      3100 FORMAT (IH0,4X,'ISTOP = ',I5)
C
      9100 FORMAT (IH0/5X,'** WARNING: NO CONVERGENCE AT',I4,'-TH TIME STEP AF
ITEF',I4,' ITERATIONS'/8X,'NITER =' ,I4,' DIFMAX =' ,D12.4,' TOLB =' ,
2 D12.4,' NOCCUR =' ,I4)
      ENF
C
      SUBROUTINE DATAIN(X,Z,B,T,IE,DLB,DCOSXB,DCOSZB,NBE,ISB,NPE,
1 1STAB,STAB,ISTYP, DL,DCOSX,DCOSZ,NRSE,IS,NPRS, TRSTAB,RSTAB,
2 1ESTYP, TDTAB,DTAB,IDTYP,NPD, RP,PROP, KPR,KDSK,MAXDIP, ISTOP,
3 THDTAB,HUTAB,RKU,TCUTAB,CUTAB, THITAB,HITAB,RKL,TCLTAB,CLTAB,
4 IHUTYP,IHLTYP,TDTCH, NPW,IWTYP,WSSF,TWSSF)
C
C

```

```

GH2D2175
GH2D2180
GH2D2185
GH2D2190
GH2D2195
GH2D2200
GH2D2205
GH2D2210
GH2D2215
GH2D2220
GH2D2225
GH2D2230
GH2D2235
GH2D2240
GH2D2245
GH2D2250
GH2D2255
GH2D2260
GH2D2265
GH2D2270
GH2D2275
GH2D2280
GH2D2285
GH2D2290
GH2D2295
GH2D2300
GH2D2305
GH2D2310
GH2D2315
GH2D2320
GH2D2325
GH2D2330
GH2D2335
GH2D2340
GH2D2345
GH2D2350
GH2D2355
GH2D2360
GH2D2365
GH2D2370
GH2D2375
GH2D2380
GH2D2385
GH2D2390
GH2D2395
GH2D2400
GH2D2405
GH2D2410
GH2D2415
GH2D2420
GH2D2425
GH2D2430
GH2D2435
GH2D2440
GH2D2445
GH2D2450
GH2D2455
DATA 005
DATA 010
DATA 015
DATA 020
DATA 025
DATA 030
DATA 035

```

```

C  FUNCTION OF SUBROUTINE--TO READ, PRINT, AND CHECK VARIABLES          DATA 040
C  PERTAINING TO SIMULATION TIME, GEOMETRY OF THE SYSTEM, BOUNDARY-    DATA 045
C  INITIAL CONDITIONS, AND PROPERTIES OF BOTH THE MATERIAL BEING       DATA 050
C  TRANSPORTED AND THE POROUS MEDIA.                                   DATA 055
C                                                                      DATA 060
C                                                                      DATA 065
C  IMPLICIT REAL*8 (A-H,O-Z)                                           DATA 070
C  REAL*4 PHAT1                                                         DATA 075
C  REAL*4 DATNAM(20)                                                    DATA 080
C                                                                      DATA 085
C  COMMON /CSIZE/ MAXNP,MAXEL,MAXBNP,MAXBEL,MAXBW,MAXNTI,MAXNMT,MAXDTC DATA 090
C  COMMON /CINTG/ NNP,NEL,NBN,NBEL,IBAND,IHBP,IHALPB,NMAT,NCH,NDTCHG DATA 095
C  COMMON /IDNGH/ NTI,NSTR,KSTR,KPRJ,KDSK,KSS,KVI,NITER,NPITER DATA 100
C  COMMON /RDNGH/ DELT,CHNG,DELMAX,THAX,DELTO,N,ONE,ONI,TOLA,TOLB DATA 105
C  COMMON /OPTN/ IWET,ILUMP,IMID,ICONFI,IOPF,KSORP,IPNTS,IVHL DATA 110
C  COMMON /WET/ APHA1,APHA2,BETA1,BETA2                                DATA 115
C                                                                      DATA 120
C  COMMON /CS/ MXSPR,MXSDP,NSPR,NSDP,MXWNP,MXWPR,MXWDP,MWNP,MWPR,MWDP DATA 125
C  COMMON /RSEND/ MXRNP,MXRSEL,MXRSPR,MXRSDP,MRSN,MRSEL,MRSR,MRSRDP DATA 130
C  COMMON /DIRICH/ MXDNP,MXDPR,MXDDP,NDN,NDR,NDDP DATA 135
C  COMMON /KUH/ MXHUPR,MXHUDP,MHUPR,MHUDP DATA 140
C  COMMON /KLHL/ MXHLPR,MXHLDP,MHLPR,MHLDP DATA 145
C                                                                      DATA 150
C  DIMENSION X(MAXNP),Z(MAXNP),B(MAXNP),T(MAXNP),IE(MAXEL,5)          DATA 155
C                                                                      DATA 160
C  DIMENSION DLE(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),NBE(MAXBEL), DATA 165
C  > ISB(MAXBEL,4),NPB(MAXBNP)                                          DATA 170
C                                                                      DATA 175
C  DIMENSION TSTAB(MXSPR,MXSDP),STAB(MXSPR,MXSDP),ISTYP(MAXEL) DATA 180
C  DIMENSION NPW(MXWNP),INTYP(MXWNP) DATA 185
C  DIMENSION WSSF(MXWPR,MXWDP),TWSSF(MXWPR,MXWDP) DATA 190
C                                                                      DATA 195
C  DIMENSION DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL) DATA 200
C  DIMENSION MRSE(MXRSEL),IS(MXRSEL,4),NPRS(MXRNP) DATA 205
C  DIMENSION TRSTAB(MXRSPR,MXRSDP),RSTAB(MXRSPR,MXRSDP) DATA 210
C  DIMENSION IRSTYP(MXRSEL) DATA 215
C                                                                      DATA 220
C  DIMENSION TDTAB(MXDPR,MXDDP),DTAB(MXDPR,MXDDP) DATA 225
C  DIMENSION IDTYP(MXDNP),NPD(MXDNP) DATA 230
C                                                                      DATA 235
C  DIMENSION THUTAB(MXHUPR,MXHUDP),HUTAB(MXHUPR,MXHUDP),RKU(MXHUPR) DATA 240
C  DIMENSION TCUTAB(MXHUPR,MXHUDP),CUTAB(MXHUPR,MXHUDP) DATA 245
C  DIMENSION THLTAB(MXHLPR,MXHLDP),HLTAB(MXHLPR,MXHLDP),RKL(MXHLPR) DATA 250
C  DIMENSION TCLTAB(MXHLPR,MXHLDP),CLTAB(MXHLPR,MXHLDP) DATA 255
C  DIMENSION IHUTYP(MAXEL),IHLYTYP(MAXEL) DATA 260
C                                                                      DATA 265
C  DIMENSION RP(MAXNP),PROP(MAXMAT,15) DATA 270
C  DIMENSION KPR(MAXNTI),KDSK(MAXNTI),TDTCH(MXDTC) DATA 275
C  DIMENSION TITEL(9) DATA 280
C  DIMENSION PHAT(2,11) DATA 285
C                                                                      DATA 290
C  DATA PHAT/4H K,4HD ,4H RH,4HOB ,4H A,4HL , DATA 295
C  > 4H A,4HT ,4H LAM,4HBD ,4H P,4HOE ,4H A,4HLP DATA 300
C  > 4H A,4HM ,4H T,4HAU ,4H K,4HW ,4H K,4HS DATA 305
C                                                                      DATA 310
C  ISTOP=0 DATA 315
C                                                                      DATA 320
C ***** DATA SET 2: BASIC INTEGERS DATA 325
C                                                                      DATA 330
C  READ 9, (DATNAM(I),I=1,20) DATA 335
C  READ 10, NNP,NEL,NMAT,NCH,NTI,KVI,KSTR,KSS,IWET,ILUMP,IMID,ICONFI, DATA 340
C  > IOPF,KSORP,NITER,NDTCHG,IPNTS,NSTR,NPITER,IVHL DATA 345
C  IF (NDTCHG.LE.0) NDTCHG=1 DATA 350
C                                                                      DATA 355
C ***** DATA SET 3: BASIC REAL PARAMETERS DATA 360

```

```

C      READ 9, (DATNAM(I),I=1,20)
      READ 20, DELT, CHNG, DELMAX, THAX, W, ONE, ONI, TOLA, TOLB
      DEIT0=DELT
C
C ***** DATA SET 4: PRINTER AND AUXILIARY STORAGE CONTROL
C
      READ 9, (DATNAM(I),I=1,20)
      READ 30, KPRO, (KPR(I),I=1,NTI)
      READ 30, KDSK0, (KDSK(I),I=1,NTI)
      READ 20, (IDTCH(I),I=1,NDTCHG)
C
      IF (THAX.LE.0.0) THAX=1.0E50
C
      PRINT 1000, NNP, NEL, NHAT, NCH, NTI, KVI, KSTR, KSS, IWET, ILUMP, INID,
1 ICONFI, IOPT, KSORP, WITER, WUTCHG, IPNTS, MSTR, NPITER, IVHL
      PRINT 1100, DEIT, CHNG, DELMAX, THAX, W, ONE, ONI, TOLA, TOLB
      PRINT 1210, KPRO, (KPR(I),I=1,NTI)
      PRINT 1220, KDSK0, (KDSK(I),I=1,NTI)
C
      PRINT 1230, (IDTCH(I),I=1,NDTCHG)
C
C ***** DATA SET 5: MATERIAL PROPERTIES
C
C ----- READ AND PRINT MATERIAL PROPERTIES
C
      READ 9, (DATNAM(I),I=1,20)
      PRINT 2000, ((PHAT(I,J),I=1,2),J=1,11)
      DO 110 I=1,NHAT
      READ 20, (PROP(I,J),J=1,15)
110 PRINT 2100, I, (PROP(I,J),J=1,15)
C
120 IF (KVI.LE.0) GO TO 200
C
C ----- READ NODAL AND ELEMENT DATA FROM AUXILIARY STORAGE
C
      REWIND 1
      READ(1) (TITLEM(I),I=1,9), NPROBH, NNP, NEL, NBN, NBEL, NTIM, NHUPR, NHLPR
      READ(1) (X(NP),NP=1,NNP), (Z(NP),NP=1,NNP), (B(NP),NP=1,NNP),
> (T(NP),NP=1,NNP), ((IE(M,IQ),M=1,NEL),IQ=1,5), (DLB(M),M=1,NBEL),
> (DCOSXB(M),M=1,NBEL), (DCOSZB(M),M=1,NBEL), (NBE(M),M=1,NBEL),
> ((ISB(M,IQ),M=1,NBEL),IQ=1,4), (NFB(NP),NP=1,NBN),
4 (FKU(I),I=1,NHUPR), (IHUTYP(M),M=1,NEL), (RKL(I),I=1,NHLPR),
5 (IHLTYP(M),M=1,NEL)
C
      GO TO 310
C
C ***** DATA SET 6: BED ELEVATION OR AQUIFER THICKNESS
C
C ----- READ BED ROCK ELEVATION (UNCONFINED AQUIFER) AND THE
C ----- THICKNESS OF THE AQUIFER (CONFINED AQUIFER) VIA CARDS
C
200 CONTINUE
      READ 9, (DATNAM(I),I=1,20)
      CALL READR(B,MAXNP,NNP)
      CALL READR(T,MAXNP,NNP)
C
C ***** DATA SET 7: NODE COORDINATE
C
C ----- READ NODAL POINT DATA VIA CARD IF NECESSARY
C
      READ 9, (DATNAM(I),I=1,20)
      CALL READR(X,MAXNP,NNP)
      CALL READR(Z,MAXNP,NNP)

```

```

DATA 365
DATA 370
DATA 375
DATA 380
DATA 385
DATA 390
DATA 395
DATA 400
DATA 405
DATA 410
DATA 415
DATA 420
DATA 425
DATA 430
DATA 435
DATA 440
DATA 445
DATA 450
DATA 455
DATA 460
DATA 465
DATA 470
DATA 475
DATA 480
DATA 485
DATA 490
DATA 495
DATA 500
DATA 505
DATA 510
DATA 515
DATA 520
DATA 525
DATA 530
DATA 535
DATA 540
DATA 545
DATA 550
DATA 555
DATA 560
DATA 565
DATA 570
DATA 575
DATA 580
DATA 585
DATA 590
DATA 595
DATA 600
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

```

```

C
C ***** DATA SET 8: ELEMENT INCIDENCES AND MATERIAL TYPES
C
C ----- READ ELEMENT DATA VIA CARDS IF NECESSARY
C
    READ 9, (DATNAM(I),I=1,20)
    NJ = 0
230 READ 10, NI, (IE(NI,I),I=1,5),MODL,NLAY
    NTYP=IE(NI,5)
240 NJ = NJ + 1
    IF (NI-NJ) 245,260,250
245 PRINT 2700, NI
    ISTOF = ISTOP + 1
250 DO 255 IQ=1,4
255 IE(NJ,IQ)=IE(NJ-1,IQ) + 1
    IE(NJ,5) = IE(NJ-1,5)
260 IF (NJ.LT.NI) GO TO 240
C
    IF (NJ.EQ.NEL) GO TO 280
    IF (MODL.LE.0) GO TO 230
C
    DO 275 I=1,NLAY
    LL=2
    DO 275 J=1,MODL
    IF (NJ.EQ.NI) GO TO 270
    DO 265 KQ=1,4
265 IE(NJ,KQ)=IE(NJ-1,KQ) + LL
    IE(NJ,5)=IE(NJ-1,5)
270 LL=1
275 NJ=NJ+1
C
    NJ = NJ - 1
    IF (NJ.LT.NEL) GO TO 230
280 READ 9, (DATNAM(I),I=1,20)
C
C ----- IDENTIFY BOUNDARY ELEMENTS AND COMPUTE DIRECTIONAL COSINES
C ----- OF BOUNDARY SIDES IF GEOMETRICAL ARE READ VIA CARDS
C
    IF (KVI.LE.0) CALL SURF(X,Z,IE,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,
    > MAXNP,MAXEL,MAXBEL,MAXBNP)
C
C ----- PRINT NODE COORDINATE AND BED ELEVATION OR AQUIFER THICKNESS
C
310 KLINE=-1
    DO 315 NP=1,NNP,2
    NJMN=NP
    NJMX=MINC(NP+1,NNP)
    KLINE=KLINE+1
    IF (MOD(KLINE,50).EQ.0) PRINT 3000
    PRINT 3100, (NJ,X(NJ),Z(NJ),B(NJ),T(NJ),NJ=NJMN,NJMX)
315 CONTINUE
C
C ----- COMPUTE MAXDIF
C
    MAXDIF = 0
    DO 325 M=1,NEL
    NQ=4
    NQ1=3
    IEN4=IE(M,4)
    IF (IEN4.EQ.0) NQ=3
    IF (IEN4.EQ.-1) NQ1=2
C
    MND=0
    DO 320 IQ=1,NQ1
    IQ1=IQ+1

```

```

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
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
DATA 1000
DATA 1005
DATA 1010

```

```

DO 320 JQ=IQ1,NQ
ND=IABS(IE(N,IQ)-IE(N,JQ))
NND=MAX0(ND,NND)
MAXDIF=MAX0(NND,MAXDIF)
320 CONTINUE
C
325 CONTINUE
C
C ***** DATA SET 9: MATERIAL CORRECTIONS
C
C ----- MODIFY MATERIAL TYPES FOR SELECTED ELEMENTS, IF NECESSARY
C
IF (KVI.GT.0) GO TO 330
IF (MCH.LE.0) GO TO 330
READ 9, (DATNAM(I),I=1,20)
CALL READN(IE(1,5),MAXEL,MCH)
C
C ----- WRITE ELEMENT INFORMATION
C
330 KLINE=-1
DO 340 NI=1,NEL,3
NJNN=NI
NJFX=NINO(NI+2,NEL)
KLINE=KLINE+1
IF (MOD(KLINE,50).EQ.0) PRINT 3500
PRINT 3600, (NJ, (IE(NJ,IQ),IQ=1,5), NJ=NJNN,NJNX)
340 CONTINUE
IHALFB=MAXDIF
IBAND=2*IHALFB+1
IHBP=IHALFB+1
PRINT 3300, IHALFB,IBAND,IHBP
C
C
C ----- CHECK MATERIAL TYPES FOR EACH ELEMENT
C
DO 380 M=1,NEL
MTYP=IE(M,5)
NTYP=IABS(MTYP)
IF (MTYP.GT.0 .AND. MTYP.LE.NMAT) GO TO 380
PRINT 4200, M
ISTOP=ISTOP+1
380 CONTINUE
IF (ISTOP.EQ.0) GO TO 390
PRINT 4300, ISTOP
C
C ***** DATA SET 10: INITIAL OR PRE-INITIAL CONDITIONS
C
390 TIME=0.0
C
C ----- READ INITIAL OR PREINITIAL CONDITIONS VIA LOGICAL UNIT 3.
C
IF (NSTR.EQ.0) GO TO 430
REWIND 2
REWIND 3
READ (3) (TITLEM(I),I=1,9),NPROBM,NNPM,NELM,NTIM
IF (KSIR.EQ.1) WRITE(2) (TITLEM(I),I=1,9),NPROBM,NNPM,NELM,NTIM
READ (3) (X(N),N=1,NNPM), (Z(N),N=1,NNPM), ((IE(M,I),M=1,NELM),I=1,5)
IF (KSIR.EQ.1) WRITE(2) (X(N),N=1,NNPM), (Z(N),N=1,NNPM),
1 ( (IE(M,I),M=1,NELM),I=1,5)
DO 410 I=1,NSTR
READ (3) TIME, (RP(N),N=1,NNP), (FXN,N=1,NNP), (PZN,N=1,NNP)
IF (KSTR.EQ.0) GO TO 410
WRITE(2) TIME, (RP(N),N=1,NNP), (FXN,N=1,NNP), (PZN,N=1,NNP)
410 CONTINUE
GO TO 450

```

```

DATA 10 15
DATA 10 20
DATA 10 25
DATA 10 30
DATA 10 35
DATA 10 40
DATA 10 45
DATA 10 50
DATA 10 55
DATA 10 60
DATA 10 65
DATA 10 70
DATA 10 75
DATA 10 80
DATA 10 85
DATA 10 90
DATA 10 95
DATA 11 00
DATA 11 05
DATA 11 10
DATA 11 15
DATA 11 20
DATA 11 25
DATA 11 30
DATA 11 35
DATA 11 40
DATA 11 45
DATA 11 50
DATA 11 55
DATA 11 60
DATA 11 65
DATA 11 70
DATA 11 75
DATA 11 80
DATA 11 85
DATA 11 90
DATA 11 95
DATA 12 00
DATA 12 05
DATA 12 10
DATA 12 15
DATA 12 20
DATA 12 25
DATA 12 30
DATA 12 35
DATA 12 40
DATA 12 45
DATA 12 50
DATA 12 55
DATA 12 60
DATA 12 65
DATA 12 70
DATA 12 75
DATA 12 80
DATA 12 85
DATA 12 90
DATA 12 95
DATA 13 00
DATA 13 05
DATA 13 10
DATA 13 15
DATA 13 20
DATA 13 25
DATA 13 30
DATA 13 35

```

C		DATA 1340
C	----- READ INITIAL OR PRE-INITIAL CONDITIONS VIA CARDS	DATA 1345
C		DATA 1350
	430 READ 9, (DATNAM(I),I=1,20)	DATA 1355
	CALL READR(RP,MAXRP,NRP)	DATA 1360
C		DATA 1365
C	***** DATA SET 11: CONTROL INTEGERS FOR TRANSIENT INPUT	DATA 1370
C		DATA 1375
C	----- READ TRANSIENT INTEGERS	DATA 1380
C		DATA 1385
	450 READ 9, (DATNAM(I),I=1,20)	DATA 1390
	READ 10, NSPR,NSDP, NRSN,NRSEL,NRSPR,NRSDP, NDN,NDPR,NDDP,	DATA 1395
	1 NHUPR,NHUDP,NHLPR,NHLDP,NWNP,NWPR,NWDP	DATA 1400
	PRINT 5100, NSPR,NSDP, NRSN,NRSEL,NRSPR,NRSDP, NDN,NDPR,NDDP,	DATA 1405
	1 NHUPR,NHUDP,NHLPR,NHLDP,NWNP,NWPR,NWDP	DATA 1410
C		DATA 1415
C	***** DATA SET 12: SOURCE/SINK	DATA 1420
C		DATA 1425
C	----- READ AND WRITE SOURCE/SINK PROFILES	DATA 1430
C		DATA 1435
	READ 9, (DATNAM(I),I=1,20)	DATA 1440
	PRINT 5300	DATA 1445
	DO 460 I=1,NSPR	DATA 1450
	READ 20, (TSTAB(I,J),STAB(I,J),J=1,NSDP)	DATA 1455
	PRINT 5500,I	DATA 1460
	PRINT 5510, (TSTAB(I,J),STAB(I,J),J=1,NSDP)	DATA 1465
	460 CONTINUE	DATA 1470
C		DATA 1475
C	----- READ SOURCE TYPE ASSIGNED TO EACH ELEMENT	DATA 1480
C		DATA 1485
	CALL READN(ISTYP,MAXEL,NEL)	DATA 1490
	PRINT 5600, (ISTYP(I),I=1,NEL)	DATA 1495
C		DATA 1500
	IF (NWNP.EQ.0) GO TO 500	DATA 1505
C		DATA 1510
C	----- READ AND WRITE WELL SOURCE/SINK PROFILES	DATA 1515
C		DATA 1520
	PRINT 5700	DATA 1525
	DO 480 I=1,NWPR	DATA 1530
	READ 20, (TWSSF(I,J),WSSF(I,J),J=1,NWDP)	DATA 1535
	PRINT 5710, I	DATA 1540
	PRINT 5510, (TWSSF(I,J),WSSF(I,J),J=1,NWDP)	DATA 1545
	480 CONTINUE	DATA 1550
C		DATA 1555
C	----- READ WELL SOURCE/SINK NODES AND TYPE OF PROFILES ASSIGNED TO	DATA 1560
C	----- EACH NWNP NODE.	DATA 1565
C		DATA 1570
	READ 10, (NPW(I),I=1,NWNP)	DATA 1575
	CALL READN(IWTYP,MXWNP,NWNP)	DATA 1580
C		DATA 1585
C	----- PRINT GLOBAL WELL NODE NUMBERS AND PROFILE TYPE OF WELL NODE	DATA 1590
C		DATA 1595
	PRINT 5800	DATA 1600
	PRINT 5850, (I,NPW(I),IWTYP(I),I=1,NWNP)	DATA 1605
C		DATA 1610
C	***** DATA 13: DIRICHLET BOUNDARY CONDITIONS	DATA 1615
C		DATA 1620
C	----- READ AND WRITE DIRICHLET CONCENTRATION PROFILES	DATA 1625
C		DATA 1630
	500 IF (NDN.EQ.0) GO TO 600	DATA 1635
	PRINT 6000	DATA 1640
	READ 9, (DATNAM(I),I=1,20)	DATA 1645
	DO 520 I=1,NDPR	DATA 1650
	READ 20, (DTAB(I,J),DTAB(I,J),J=1,NDDP)	DATA 1655
	PRINT 6100,I	DATA 1660

```

      PRINT 5510, (DTAB(I,J),DTAB(I,J),J=1,NDDP)
520 CONTINUE
C
C ----- READ DIRICHLET NODES AND TYPE OF DIRICHLET CONCENTRATION
C ----- ASSIGNED TO EACH DIRICHLET NODE
C
      READ 10, (NPD(I),I=1,NDN)
      PRINT 6150, (NPD(I),I=1,NDN)
      CALL READN(IDTYP,NIDNP,NDN)
      PRINT 6200
      PRINT 6210, (I,NPD(I),IDTYP(I),I=1,NDN)
C
C ***** DATA SET 14: VARIABLE BOUNDARY CONDITIONS
C
C ----- READ VARIABLE BOUNDARY CONDITION, I.E. THE CONDITIONS WHERE
C ----- EITHER NEUMANN CONDITIONS OR DIRICHLET CONDITIONS MAY BE
C ----- APPLIED TO. THE COMPUTER CODE WILL MAKE THE DECISION.
C
C ***** NORMALLY WHEN THE FLOW IS DIRECTED INTO THE REGION,
C ***** THE CAUCHY CONDITIONS ARE USED. ON THE OTHER HAND, WHEN THE
C ***** FLOW IS OUT FROM THE REGION, NEUMANN CONDITIONS ARE USED.
C ***** FOR THE FORMER CASE, CIN STORE THE CONCENTRATION OF THE
C ***** INCOMING FLUID. FOR THE LATER CASE, IT BECOMES CONCENTRATION
C ***** DEPENDENT BOUNDARY CONDITIONS AND THE DISPERSIVE FLUX IS
C ***** ASSUMED TO BE ZERO. IF DISPERSIVE FLUX IS NOT ZERO, AN
C ***** ARRAY TO STORE SUCH FLUXES IS REQUIRED AND THE PROGRAM MUST
C ***** BE MODIFIED.
C
660 IF(NRSEL.LE.0) GO TO 820
C
      READ 9, (DATNAM(I),I=1,20)
C
C ----- READ AND WRITE GLOBAL NODE NUMBER OF RUNOUT-SEEPIN NODES
C
      PRINT 6500
C ***** NPRS IS NEEDED ONLY IN THE SUBROUTINE SFLOW.
      READ 10, (NPRS(I),I=1,NRSH)
      PRINT 6600, (NPRS(I),I=1,NRSH)
C
C ----- READ AND WRITE SEEP-IN CONCENTRATION PROFILES
C
      PRINT 7000
      DO 660 I=1,NRSEPR
      READ 20, (TRSTAB(I,J),RSTAB(I,J),J=1,NRSDP)
      PRINT 7100, I
      PRINT 5510, (TRSTAB(I,J),RSTAB(I,J),J=1,NRSDP)
660 CONTINUE
C
C ----- READ RUNOUT-SEEPIN ELEMENT SIDES
C
      NODES=C
670 READ 10, MI,NSEQ,N1,N2,MNI,N1AD,N2AD,MAD,MITYP
      IF (MI.EQ.0) GO TO 690
      NJ=MI+NSEQ
      DO 680 M=M1,NJ
      NRSE(M)=MNI+(M-M1)*MAD
      IS(M,1)=N1+(M-M1)*N1AD
      IS(M,2)=N2+(M-M1)*N2AD
     IRSTYP(M)=MITYP
      NODES=NODES+1
680 CONTINUE
      GO TO 670
690 IF (MCDES.EQ.NRSEL) GO TO 695
      PRINT 7200
      STCP

```

DATA 1665
 DATA 1670
 DATA 1675
 DATA 1680
 DATA 1685
 DATA 1690
 DATA 1695
 DATA 1700
 DATA 1705
 DATA 1710
 DATA 1715
 DATA 1720
 DATA 1725
 DATA 1730
 DATA 1735
 DATA 1740
 DATA 1745
 DATA 1750
 DATA 1755
 DATA 1760
 DATA 1765
 DATA 1770
 DATA 1775
 DATA 1780
 DATA 1785
 DATA 1790
 DATA 1795
 DATA 1800
 DATA 1805
 DATA 1810
 DATA 1815
 DATA 1820
 DATA 1825
 DATA 1830
 DATA 1835
 DATA 1840
 DATA 1845
 DATA 1850
 DATA 1855
 DATA 1860
 DATA 1865
 DATA 1870
 DATA 1875
 DATA 1880
 DATA 1885
 DATA 1890
 DATA 1895
 DATA 1900
 DATA 1905
 DATA 1910
 DATA 1915
 DATA 1920
 DATA 1925
 DATA 1930
 DATA 1935
 DATA 1940
 DATA 1945
 DATA 1950
 DATA 1955
 DATA 1960
 DATA 1965
 DATA 1970
 DATA 1975
 DATA 1980
 DATA 1985

695 CONTINUE	DATA1990
C ----- WRITE RUNOUT-SEEPIN ELEMENT SIDES	DATA1995
C	DATA2000
PRINT 7300	DATA2005
DO 720 MP=1,NRSEL	DATA2010
N=NRSE(MP)	DATA2015
PRINT 7400, MP,N,IS(MP,1),IS(MP,2),IRSTYP(MP)	DATA2020
720 CONTINUE	DATA2025
C	DATA2030
C ----- DETERMINE DIRECTION COSINES DCOSX(MP) AND DCOSZ(MP) FOR THE	DATA2035
C ----- RS BOUNDARY SIDES	DATA2040
C	DATA2045
DO 810 MPI=1,NRSEL	DATA2050
MI=NRSE(MPI)	DATA2055
DO 800 MPJ=1,NBEL	DATA2060
MJ=NRSE(MPJ)	DATA2065
IF (MJ.NE.MI) GO TO 800	DATA2070
MI=IS(MPI,1)	DATA2075
MJ=IS(MPJ,2)	DATA2080
MIE=ISB(MPJ,1)	DATA2085
MJE=ISB(MPJ,2)	DATA2090
IF (MIB.EQ.MI .AND. MJB.EQ.MJ) GO TO 780	DATA2095
IF (MIB.EQ.MJ .AND. MJB.EQ.MI) GO TO 780	DATA2100
GO TO 800	DATA2105
C	DATA2110
780 DO 790 J=1,4	DATA2115
790 IS(MPI,J)=ISB(MPJ,J)	DATA2120
DL(MFI)=DLB(MPJ)	DATA2125
DCOSX(MPI)=DCOSXB(MPJ)	DATA2130
DCOSZ(MPI)=DCOSZB(MPJ)	DATA2135
GO TO 810	DATA2140
800 CONTINUE	DATA2145
PRINT 7500, MPI, MI,MI,MJ	DATA2150
ISTOP=ISTOP+1	DATA2155
810 CONTINUE	DATA2160
C	DATA2165
C	DATA2170
C ***** DATA SET 15: UPPER LEAKING AQUIFERS	DATA2175
C	DATA2180
820 IF (KVI.GT.0) GO TO 900	DATA2185
READ 9, (DATNAM(I),I=1,20)	DATA2190
PRINT 8000	DATA2195
DO 840 I=1,NHUPR	DATA2200
READ 20, (IHUTAB(I,J),HUTAB(I,J),J=1,NHJDP)	DATA2205
PRINT 8100, I	DATA2210
PRINT 5510, (IHUTAB(I,J),HUTAB(I,J),J=1,NHJDP)	DATA2215
840 CONTINUE	DATA2220
C	DATA2225
C ----- READ AND PRINT UPPER AQUIFER LEAKING COEFFICIENT	DATA2230
C	DATA2235
READ 20, (RKU(I),I=1,NHUPR)	DATA2240
PRINT 8200, (RKU(I),I=1,NHUPR)	DATA2245
C	DATA2250
C ----- READ TYPE OF UPPER LEAKING AQUIFER ASSIGNED TO EACH ELEMENT	DATA2255
C	DATA2260
CALL READN(IHUTYP,MAXEL,NEL)	DATA2265
PRINT 8300, (IHUTYP(I),I=1,NEL)	DATA2270
C	DATA2275
C ***** DATA SET 16: LOWER LEAKING AQUIFERS	DATA2280
C	DATA2285
C ----- READ AND PRINT HEAD PROFILE IN THE LOWER LEAKING AQUIFER	DATA2290
C	DATA2295
READ 9, (DATNAM(I),I=1,20)	DATA2300
PRINT 8500	DATA2305
	DATA2310

DO 860 I=1,NHLPR	DATA2315
READ 20, (THLTAB(I,J),HLTAB(I,J),J=1,NHLDP)	DATA2320
PRINT 8600, I	DATA2325
PRINT 5510, (THLTAB(I,J),HLTAB(I,J),J=1,NHLDP)	DATA2330
860 CONTINUE	DATA2335
C ----- READ AND PRINT LOWER AQUIFER LEAKING COEFFICIENT	DATA2340
C	DATA2345
READ 20, (RKL(I),I=1,NHLPR)	DATA2350
PRINT 8700, (RKL(I),I=1,NHLPR)	DATA2355
C	DATA2360
C ----- READ TYPE OF LOWER LEAKING AQUIFER ASSIGNED TO EACH ELEMENT	DATA2365
	DATA2370
900 CONTINUE	DATA2405
READ 9, (DATNAB(I),I=1,20)	DATA2410
PRINT 9000	DATA2415
DO 920 I=1,NHUPR	DATA2420
READ 20, (ICUTAB(I,J),CUTAB(2,J),J=1,NHODP)	DATA2425
PRINT 9100, I	DATA2430
PRINT 5510, (TCUTAB(I,J),CUTAB(I,J),J=1,NHODP)	DATA2435
920 CONTINUE	DATA2440
C ***** DATA SET 18: CONCENTRATION PROFILES IN LOWER LEAKING AQUIFERS	DATA2445
C	DATA2450
PRINT 9400	DATA2455
READ 9, (DATNAB(I),I=1,20)	DATA2460
DO 940 I=1,NHLPR	DATA2465
READ 20, (ICLTAB(I,J),CLTAB(I,J),J=1,NHLDP)	DATA2470
PRINT 9500, I	DATA2475
PRINT 5510, (TCLTAB(I,J),CLTAB(I,J),J=1,NHLDP)	DATA2480
940 CONTINUE	DATA2485
C	DATA2490
IF (ISTOP.EQ.0) GO TO 990	DATA2495
PRINT 4300, ISTOP	DATA2500
C	DATA2505
990 CONTINUE	DATA2510
RETURN	DATA2515
C	DATA2520
9 FORMAT(20A4)	DATA2525
10 FORMAT(16I5)	DATA2530
20 FORMAT(8D10.0)	DATA2535
30 FORMAT(80I1)	DATA2540
40 FORMAT(3I5,5X,4F10.0)	DATA2545
50 FORMAT(15,5X,F10.0)	DATA2550
C	DATA2555
1000 FORMAT(4H0/5X,35H **** BASIC INTEGER PARAMETERS ****//5X,	DATA2560
> 4CH NUMBER OF MODAL POINTS.,15/ 5X,	DATA2565
> 4CH NUMBER OF ELEMENTS.,15/ 5X,	DATA2570
> 4CH NUMBER OF DIFFERENT MATERIALS.,15/ 5X,	DATA2575
> 4CH NUMBER OF CORRECTION MATERIALS.,15/ 5X,	DATA2580
> 4CH NUMBER OF TIME INCREMENTS.,15/5X,	DATA2585
> 4CH VELOCITY INPUT CONTROL.,15/ 5X,	DATA2590
> 4CH AUXILIARY STORAGE CONTROL.,15/ 5X,	DATA2595
> 4CH STEADY-STATE CONTROL.,15/5X,	DATA2600
> 4CH UPSTREAM WEIGHTING INDICATOR, IWET. . .,15/ 5X,	DATA2605
> 4CH LUMPING INDICATOR, ILUMP.,15/ 5X,	DATA2610
> 4CH TIME-DIFFERENCE INDICATOR, IMID. . .,15/ 5X,	DATA2615
> 4CH IS CONFINED AQUIFER?,15/ 5X,	DATA2620
> 4CH WEIGHTING FACTOR OPTIMIZATION INDICATOR,15/5X,	DATA2625
	DATA2630

```

> 4CH SORPTION MODEL CONTROL . . . . .,I5/ 5X,
> 4CH NO. OF ITERATIONS FOR SOLVING NONLINEAR,I5/ 5X,
> 4CH NO. OF TIMES TO RESET TIME STEP SIZE .,I5/5X,
> 4CH IS POINTWISE SOLUTION TO BE USED? . .,I5/5X,
> 4CH NO. OF RECORDS FOR RESTART . . . . .,I5/ 5X,
> 4CH NO. OF POINTWISE ITERATIONS . . . . .,I5/ 5X,
> 4CH IS FLUX-SOLVING MATRIX LUMPED? . . . ,I5/)
1100 FORMAT (IHO//5X,
1 4CH TIME INCREMENT. . . . .,E15.6/ 5X,
2 4CH MULTIPLIER FOR INCREASING DELT. . . .,E15.6/ 5X,
3 4CH MAXIMUM VALUE OF DELT . . . . .,E15.6/ 5X,
4 4CH MAXIMUM VALUE OF TIME . . . . .,E15.6/ 5X,
5 4CH TIME-INTEGRATION PARAMETER. . . . .,E15.6/5X,
6 4CH ITERATION PARAMETER FOR NONLINEAR EQ. ,E15.6/5X,
7 4CH RELAXATION PARAMETER FOR POINTWISE SOL.,E15.6/5X,
8 4CH ERROR TOLERANCE FOR STEADY STATE SOL. ,E15.6/5X,
9 4CH ERROR TOLERANCE FOR TRANSIENT SOL . .,E15.6/)
1210 FORMAT (IHO/5X,29H **** LINE PRINT CONTROL ****/(10X,50I2))
1220 FORMAT (IHO/5X,26H **** STORAGE CONTROL ****/(10X,50I2))
1230 FORMAT (IHO/5X,34H **** TIME TO RESET STEP SIZE ****/(10X,8D12.4))
2000 FORMAT (IH1/5X,30H **** MATERIAL PROPERTIES ****/9H MAT. NO.,11(
> 2A4,2X))
2100 FORMAT (I8,11D10.3/8X,11D10.3)
2700 FORMAT (IHO,5X,16HERROR IN READING,I5,18H ELEMENT CARD STOP/)
3000 FORMAT (IH1/5X,27H **** NODAL POINT DATA ****/
1 2 (2X,5H NODE,11X,2H X,11X,2H Z,11X,2H B,11X,2H T) / 2 (2X,5H ---,
2 11X,2H -,11X,2H -,11X,2H -,11X,2H -))
3100 FORMAT ( 2 (2X,I5,4D13.5))
3300 FORMAT (IHO/5X, 9H IHALPB =,I2,9H IBAND =,I2,8H IHBP =,I2)
3500 FORMAT (IH1/5X,55H **** ELEMENT DATA: GLOBAL INDICES OF ELEMENT NOD
1E ****/3 (5X,5H ELN,5H MOD1,5H MOD2,5H MOD3,5H MOD4,5H NTYP) /
2 3 (5X,5H ---,5H ---,5H ---,5H ---,5H ---))
3600 FORMAT (3 (5X,6I5))
4200 FORMAT (////4H ERROR IN MATERIAL TYPE CODE FOR ELEMENT ,I5//)
4300 FORMAT (////45H ASSEMBLY AND SOLUTION WILL NOT BE PERFORMED,,I5,
> 19H FATAL CARD ERRORS//)
5100 FORMAT (IH1/5X,39H **** TRANSIENT INTEGER PARAMETERS ****//5X,
1 5CH NO. OF SOURCE PROFILE . . . . .,I5/5X,
2 5CH NO. OF SOURCE DATA POINT IN EACH PROFILE . . . . .,I5/5X,
3 5CH NO. OF RUN-IN AND SEEP-OUT NODAL POINTS . . . . .,I5/5X,
4 5CH NO. OF RUN-IN AND SEEP-OUT ELEMENT SIDES . . . . .,I5/5X,
5 5CH NO. OF RUN-IN AND SEEP-OUT PROFILES . . . . .,I5/5X,
6 5CH NO. OF RS DATA POINTS IN EACH PROFILE . . . . .,I5/5X,
7 5CH NO. OF DIRICHLET NODAL POINTS . . . . .,I5/5X,
8 5CH NO. OF DIRICHLET BOUNDARY CONDITION PROFILES . .,I5/5X,
9 5CH NO. OF DIRICHLET DATA POINTS IN EACH PROFILE . .,I5/5X,
A 5CH NO. OF UPPER LEAKING AQUIFER PROFILES . . . . .,I5/5X,
B 5CH NO. OF DATA POINTS IN UPPER LEAKING AQUIFER . .,I5/5X,
C 5CH NO. OF LOWER LEAKING AQUIFER PROFILES . . . . .,I5/5X,
D 5CH NO. OF DATA POINTS IN LOWER LEAKING AQUIFERS . .,I5/5X,
E 5CH NO. OF WELL SOURCE/SINK NODES . . . . .,I5/5X,
F 5CH NO. OF WELL SOURCE/SINK PROFILES . . . . .,I5/5X,
G 5CH NO. OF DATA POINTS IN WELL SOURCE/SINK PROFILE .,I5/)
5300 FORMAT (IH1/5X,27H *** SOURCE INFORMATION ***)
5500 FORMAT (IHO/5X,12H PROFILE NO.,I2,//4 (4X,4HTIME,6X,6HSOURCE,2X) /
> 4 (4X,4H----,6X,6H----,2X))
5510 FORMAT (IH ,4(2D11.3))
5600 FORMAT (IHO//5X,60H LIST OF SOURCE PROFILE TYPE ASSIGNED TO ELEMENT
1 1,2,....,NEL// (4X,30I3))
5700 FORMAT (IHO//5X,37H *** WELL SOURCE/SINK INFORMATION ***)
5710 FORMAT (IHO/5X,12H PROFILE NO.,I2/ 4 (4X,4HTIME,6X,6HSOURCE,2X) /
1 4 (4X,4H----,6X,6H----,2X))
5800 FORMAT (IHO//10X,65H GLOBAL NODAL NUMBER AND PROFILE TYPE OF WELL S
OURCE/SINK NODES // 5 (5X,15H I NPW TYP)/5 (5X,15H - ---
2---))

```

DATA2635
DATA2640
DATA2645
DATA2650
DATA2655
DATA2660
DATA2665
DATA2670
DATA2675
DATA2680
DATA2685
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

```

5850 FORMAT(5(5X,3I5))
6000 FORMAT(1H1/5X,40H **** DIRICHLET BOUNDARY CONDITIONS ****)
6100 FORMAT(1H0/5X,12H PROFILE NO.,I2//4(4X,4HTIME,6X,6H DCON ,2X) /
> 4(4X,4H----,6X,6H ---- ,2X))
6150 FORMAT(1H0/10X,39H --- LIST OF DIRICHLET MODAL POINTS ---//
1 (10X,2G15))
6200 FORMAT(1H0/// 5(5X,15H I NPD TYP)/5(5X,15H - --- ---))
6210 FORMAT(5(5X,3I5))
6500 FORMAT(1H1/5X,39H **** VARIABLE BOUNDARY CONDITIONS ****)
6600 FORMAT(1H0/10X,32H --- LIST OF RS-MODAL POINTS ---// (10X,20I5))
7000 FORMAT(1H0///10X,
1 45H --- INCOMING FLUID CONCENTRATION PROFILE ---)
7100 FORMAT(1H0/5X,12H PROFILE NO.,I2//4(4X,4HTIME,6X,6H RCIN ,2X) /
> 4(4X,4H----,6X,6H ---- ,2X))
7200 FORMAT(1H1/5X,34H ERROR IN READING RS-ELEMENT-SIDES)
7300 FORMAT(1H0///10X,37H --- RS ELEMENT-SIDES INFORMATION ---//15X,
> 45H SIDE ELEMENT MODE1 MODE2 TYPE OF CIN)
7400 FORMAT(1H ,15X,15,5X,15,3X,15,3X,15,9X,15)
7500 FORMAT(1H0/5X,15,61H-TH RS SIDE IS NOT ONE OF THE BOUNDARY SIDES;
1 M1, N1, N2 =,5I5)
8000 FORMAT(1H1/5X,43H **** UPPER AQUIFER LEAKING PARAMETERS ****)
8100 FORMAT(1H0/10X,43H --- UPPER LEAKY AQUIFER HEAD PROFILE NO.,I2,
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 PCB PROFILES 1, 2, ..., ---/ (4D20.3))
8300 FORMAT(1H0//10X,75H --- LIST OF UPPER LEAKY AQUIFER TYPE ASSIGNED
1 TC ELEMENT 1,2,....,NEL ---// (5X,30I3))
8500 FORMAT(1H1/5X,43H **** LOWER AQUIFER LEAKING PARAMETERS ****)
8600 FORMAT(1H0/10X,43H --- LOWER LEAKY AQUIFER HEAD PROFILE NO.,I2,
1 4H ---//4(4X,4HTIME,6X,6H HEAD ,2X) / 4(4X,4H----,6X,6H ---- ,2X))
8700 FORMAT(1H0/10X,77H --- LIST OF LOWER AQUIFER LEAKING COEFFICIENT
1 PCB PROFILES 1, 2, ..., ---/ (4D20.3))
8800 FORMAT(1H0//10X,75H --- LIST OF LOWER LEAKY AQUIFER TYPE ASSIGNED
1 TO ELEMENT 1,2,....,NEL ---// (5X,30I3))
9000 FORMAT(1H1/5X,43H **** UPPER AQUIFER CONCENTRATION **** )
9100 FORMAT(1H0/10X,43H --- UPPER LEAKY AQUIFER CON. PROFILE NO.,I2,
1 4H ---//4(4X,4HTIME,6X,6H CONC ,2X) / 4(4X,4H----,6X,6H ---- ,2X))
9400 FORMAT(1H1/5X,43H **** LOWER AQUIFER CONCENTRATION **** )
9500 FORMAT(1H0/10X,43H --- LOWER LEAKY AQUIFER CONC PROFILE NO.,I2,
1 4H ---//4(4X,4HTIME,6X,6H CONC ,2X) / 4(4X,4H----,6X,6H ---- ,2X))

C
END

SUBROUTINE SURF(X,Z,IE, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,
> MAXNP,MAXEL, MAXBEL,MAXBNP)

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 IMPLICIT REAL*8(A-H,O-Z)
C
C FIND SURFACE SIDES BY LOCATING NONDUPLICATED SIDES
C
COMMON /CINTG/ NNP,NEL,NBN,NBEL,IBAND,IHBP,IHALFB,NMAT,NCH,NDTCHG
C
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)
C DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),NBE(MAXBEL),
C > ISB(MAXBEL,4),NPB(MAXBNP)
C
C ----- FIND SURFACE SIDES BY LOCATING NONDUPLICATED SIDES
C
NBEL=0
NBN=0

```

```

DATA2960
DATA2965
DATA2970
DATA2975
DATA2980
DATA2985
DATA2990
DATA2995
DATA3000
DATA3005
DATA3010
DATA3015
DATA3020
DATA3025
DATA3030
DATA3035
DATA3040
DATA3045
DATA3050
DATA3055
DATA3060
DATA3065
DATA3070
DATA3075
DATA3080
DATA3085
DATA3090
DATA3095
DATA3100
DATA3105
DATA3110
DATA3115
DATA3120
DATA3125
DATA3130
DATA3135
DATA3140
DATA3145
DATA3150
DATA3155
DATA3160
DATA3165

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

```

DO 390 MI=1,NBI	SURF 115
NQ=4	SURF 120
IEH4=IE(MI,4)	SURF 125
IF (IEH4.EQ.0) NQ=3	SURF 130
DO 380 IQ=1,NQ	SURF 135
IQ=IQ+1	SURF 140
IF (IQ.EQ.NQ) IQI=1	SURF 145
C	SURF 150
DO 190 MJ=1,NBI	SURF 155
IF (MJ.EQ.MI) GO TO 190	SURF 160
NQ=4	SURF 165
IEH4=IE(MJ,4)	SURF 170
IF (IEH4.EQ.0) NQ=3	SURF 175
DO 180 JQ=1,NQ	SURF 180
JQI=JQ+1	SURF 185
IF (JQ.EQ.NQ) JQI=1	SURF 190
IF (IE(MI,IQ).EQ.IE(MJ,JQ).AND.IE(MI,IQI).EQ.IE(MJ,JQI)) GO TO 380	SURF 195
IF (IE(MI,IQ).EQ.IE(MJ,JQI).AND.IE(MI,IQI).EQ.IE(MJ,JQ)) GO TO 380	SURF 200
180 CONTINUE	SURF 205
190 CONTINUE	SURF 210
C	SURF 215
C ***** A BOUNDARY SIDE HAS BEEN FOUND.	SURF 220
C	SURF 225
NI=IE(MI,IQ)	SURF 230
NJ=IE(MI,IQI)	SURF 235
NBEL=NBEL+1	SURF 240
NBF(NBEL)=MI	SURF 245
ISF(NBEL,1)=NI	SURF 250
ISF(NBEL,2)=NJ	SURF 255
ISF(NBEL,3)=IQ	SURF 260
ISF(NBEL,4)=IQI	SURF 265
C	SURF 270
IF (NBEL.GT.1) GO TO 200	SURF 275
C	SURF 280
NBN=NBEN+1	SURF 285
NPE(NBN)=NI	SURF 290
NBN=NBEN+1	SURF 295
NPE(NBN)=NJ	SURF 300
C	SURF 305
200 DO 210 I=1,NBN	SURF 310
IJ=NPE(I)	SURF 315
IF (IJ.EQ.NI) GO TO 250	SURF 320
210 CONTINUE	SURF 325
NBN=NBEN+1	SURF 330
NPE(NBN)=NI	SURF 335
C	SURF 340
250 DO 260 J=1,NBN	SURF 345
IJ=NPE(J)	SURF 350
IF (IJ.EQ.NJ) GO TO 290	SURF 355
260 CONTINUE	SURF 360
NBN=NBEN+1	SURF 365
NPE(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	SURF 405
C ----- CALCULATE SIDE LENGTH AND DIRECTION COSINES	SURF 410
C	SURF 415
DO 490 MP=1,NBEL	SURF 420
NI=ISB(MP,1)	SURF 425
NJ=ISB(MP,2)	SURF 430
DX=X(NJ)-X(NI)	SURF 435

```

      DZ=Z(NJ)-Z(NI)
      DLE(NP)=DSQRT(DX*DX+DZ*DZ)
C
      BETA=DATAN2(DZ,DX)
      DCCSXB(NP)=DSIN(BETA)
      DCOSXB(NP)=-DCOS(BETA)
490 CONTINUE
C
C
C ----- PRINT BOUNDARY SIDE INFORMATION
C
      LINE=0
      DO 695 NP=1,NBEL,2
      LINE=LINE+1
      IF (MOD(LINE-1,50).EQ.0) PRINT 6900
      NJHN=NP
      NJHX=NINO(NP+1,NBEL)
      PRINT 6950, (J, NBE(J), (ISB(J,I), I=1,4), DLB(J), DCOSXB(J), DCOSZB(J)
      ,J=NJHN,NJHX)
695 CONTINUE
C
690C FORMAT(1H1//10X,51H *** COMPUTED BOUNDARY ELEMENT SIDE INFORMATION
      1 ***//5X,2(5H NP,5H N,5H NP1,5H NP2,3H L1,3H L2,11H DLB(N
      2P), 11H DCOSXB(NP), 11H DCOSZB(NP), 1X))
6950 FORMAT(1H ,4X,2(4I5,2I3,3D11.3, 1X))
C
      RETURN
      ENCL

      SUBROUTINE LNDGEN(LLMODE,IE,MAXNP,MAXEL,NNP,NEL)
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /CPIS/ JLE,JDIAG,JRB,JBAND
C
      DIMENSION LLNOEE(JBAND,MAXNP),IE(MAXEL,5)
C
      DO 490 NP=1,NNP
C
      DO 110 I=1,JLE
110 LLMODE(I,NP)=0
      DO 115 I=JRB,JBAND
115 LLMODE(I,NP)=0
C
      NLMOD=0
      KONT=0
      NRMOD=JDIAG
      KONT=JDIAG
C
      DO 390 M=1,NEL
      IEM1=IE(M,1)
      IEM2=IE(M,2)
      IEM3=IE(M,3)
      IEM4=IE(M,4)
      NQ=4
      IF (IEM4.EQ.0) NQ=3
C
      IF (IEM4.EQ.0) GO TO 210
C
C ----- FOR THE CASE OF QUADRILATERAL ELEMENT
C
      IF (IEM1.NE.NP .AND. IEM2.NE.NP .AND. IEM3.NE.NP .AND. IEM4.NE.NP)
      1 GO TO 390
      GO TO 220
C
C ----- FOR THE CASE OF TRIANGULAR ELEMENT

```

```

SURF 440
SURF 445
SURF 450
SURF 455
SURF 460
SURF 465
SURF 470
SURF 475
SURF 480
SURF 485
SURF 490
SURF 495
SURF 500
SURF 505
SURF 510
SURF 515
SURF 520
SURF 525
SURF 530
SURF 535
SURF 540
SURF 545
SURF 550
SURF 555
SURF 560
SURF 565
SURF 570
SURF 575
LNDG C05
LNDG C10
LNDG 015
LNDG 020
LNDG 025
LNDG 030
LNDG 035
LNDG 040
LNDG 045
LNDG 050
LNDG 055
LNDG 060
LNDG 065
LNDG 070
LNDG 075
LNDG 080
LNDG 085
LNDG 090
LNDG 095
LNDG 100
LNDG 105
LNDG 110
LNDG 115
LNDG 120
LNDG 125
LNDG 130
LNDG 135
LNDG 140
LNDG 145
LNDG 150
LNDG 155
LNDG 160
LNDG 165
LNDG 170
LNDG 175
LNDG 180

```

C	210 IF (IEN1.NE.NP .AND. IEN2.NE.NP .AND. IEN3.NE.NP) GO TO 390	LNDG 185
C	----- MERGE THE QUADRILATERAL AND TRIANGULAR ELEMENT BRANCH	LNDG 190
C	220 DO 290 IQ=1,NQ	LNDG 195
C	NI=IE(M,IQ)	LNDG 200
C	----- WHEN NI IS TO THE RIGHT OF OR THE SAME AS NP, GO TO 250.	LNDG 205
C	IF (NI.GE.NP) GC TO 250	LNDG 210
C	***** WHEN NI IS TO THE LEFT OF NP, COMPRESS THE NODES TO 1 TO JLE.	LNDG 215
C	NLNOD=NLNOD+1	LNDG 220
C	IF (NLNOD.GT. 1) GO TO 230	LNDG 225
C	----- THE FIRST LOWER-LEFT NODE IS ENCOUNTERED	LNDG 230
C	KOUNT=KOUNT+1	LNDG 235
C	LLNODE (KOUNT,NP)=NI	LNDG 240
C	GO TO 290	LNDG 245
C	----- IF NLNOD IS GREATER THAN 1, WE HAVE TO CHECK IF NI IS THE	LNDG 250
C	----- NODE ALREADY COMPRESSED? IF YES, SKIP. IF NOT INCREASE THE	LNDG 255
C	----- KOUNT.	LNDG 260
C	230 DO 240 J=1,KOUNT	LNDG 265
C	NJ=LLNODE (J,NP)	LNDG 270
C	IF (NI.EQ.NJ) GC TO 290	LNDG 275
C	240 CONTINUE	LNDG 280
C	----- THE NODE NI HAS NOT BEEN COMPRESSED YET. HENCE COMPRESS IT.	LNDG 285
C	KOUNT=KOUNT+1	LNDG 290
C	LLNODE (KOUNT,NP)=NI	LNDG 295
C	GO TO 290	LNDG 300
C	----- WHEN NI IS THE SAME AS NP, SKIP THE COUNT.	LNDG 305
C	250 IF (NI.EQ.NP) GO TO 290	LNDG 310
C	***** WHEN NI IS TO THE RIGHT OF NP, COMPRESS NODES TO JRB TO JBAND.	LNDG 315
C	NRNOD=NRNOD+1	LNDG 320
C	IF (NRNOD .GT. JRB) GO TO 260	LNDG 325
C	----- THE FIRST UPPER-RIGHT NODE IS ENCOUNTERED.	LNDG 330
C	KONT=KONT+1	LNDG 335
C	LLNODE (KONT,NP)=NI	LNDG 340
C	GO TO 290	LNDG 345
C	----- IF NRNOD IS GREATER THAN JRB, WE HAVE TO CHECK IF NI IS THE	LNDG 350
C	----- NODE ALREADY COMPRESSED? YES, SKIP. IF NOT INCREASE THE	LNDG 355
C	----- KONT.	LNDG 360
C	260 DO 270 J=JRB,KCNT	LNDG 365
C	NJ=LLNODE (J,NP)	LNDG 370
C	IF (NI.EQ.NJ) GO TO 290	LNDG 375
C	270 CONTINUE	LNDG 380
C	----- THE NODE NI HAS NOT BEEN COMPRESSED YET. HENCE COMPRESS IT.	LNDG 385
C		LNDG 390
C		LNDG 395
C		LNDG 400
C		LNDG 405
C		LNDG 410
C		LNDG 415
C		LNDG 420
C		LNDG 425
C		LNDG 430
C		LNDG 435
C		LNDG 440
C		LNDG 445
C		LNDG 450
C		LNDG 455
C		LNDG 460
C		LNDG 465
C		LNDG 470
C		LNDG 475
C		LNDG 480
C		LNDG 485
C		LNDG 490
C		LNDG 495
C		LNDG 500

KONT=KONT+1	LNDG 505
LLNODE (KONT, NP) =NI	LNDG 510
C 290 CONTINUE	LNDG 515
C 390 CONTINUE	LNDG 520
C IF (KOUNT.LE.JLE) GO TO 420	LNDG 525
WRITE(6,1000) NP,KCUNT,JLE	LNDG 530
STCP	LNDG 535
C 420 IF (KONT.LE.JBAND) GO TO 470	LNDG 540
JREND=JBAND-JDIAG	LNDG 545
KRCNT=KONT-JDIAG	LNDG 550
WRITE(6,2000) NP,KRCNT,JREND	LNDG 555
STCP	LNDG 560
C 470 LLNODE (JDIAG, NP) =NP	LNDG 565
C 490 CONTINUE	LNDG 570
C LINE=0	LNDG 575
DO 590 NP=1,NNP,2	LNDG 580
NJEN=NP	LNDG 585
NJMX=MIND (NP+1,NNP)	LNDG 590
LINE=LINE+1	LNDG 595
IF (MCD (LINE-1,50).EQ.0) PRINT 5000	LNDG 600
PRINT 5100, (NJ, (LLNODE (I, NJ), I=1,JBAND), NJ=NJEN,NJMX)	LNDG 605
590 CONTINUE	LNDG 610
C 1000 FORMAT (1H0//5X,4H ***,14,12H-TH NODE HAS,14,54H NODES TO ITS LOWER	LNDG 615
1-LEFT SIDE WHICH IS MORE THAN JLE =,15,10H STOP ***)	LNDG 620
2000 FORMAT (1H0//5X,4H ***,14,12H-TH NODE HAS,14,57H NODES TO ITS UPPER	LNDG 625
1-RIGHT SIDE WHICH IS MORE THAN JREND =,15,10H STOP ***)	LNDG 630
5000 FORMAT (1H1/5X,47H ** GENERATED SURROUNDING NODES OF ALL NODES **,/	LNDG 635
1 2(2X,5H NE,5H 1,5H 2,5H 3,5H 4,5H 5,5H 6,5H -	LNDG 640
2 7,5H 8,5H 9,5H 10,5H 11) / 2(2X,5H --,5H --,5H -	LNDG 645
3,5H --,5H --,5H --,5H --,5H --,5H --,5H --,	LNDG 650
4 5E --) //)	LNDG 655
5100 FORMAT (1H ,2(2X,12I5))	LNDG 660
C RETURN	LNDG 665
END	LNDG 670
SUBROUTINE APAETA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,	LNDG 675
> MAXMAT, NEL, W, KSS, IOPT)	LNDG 680
C FUNCTION OF THE SUBROUTINE-TO CALCULATE THE WEIGHTING FACTORS	LNDG 685
C ON FOUR SIDES OF EACH OF THE ELEMENTS	LNDG 690
C IMPLICIT REAL*8 (A-H,O-Z)	LNDG 695
C DIMENSION X(MAXNP),Z (MAXNP),IE (MAXEL,5),WETAB (MAXEL,4)	LNDG 700
DIMENSION VX (MAXNP),VXP (MAXNP),VZ (MAXNP),VZP (MAXNP)	LNDG 705
DIMENSION PROP (MAXMAT,15)	LNDG 710
C DIMENSION ALPHA (4),BETA (4),NDIN (4)	LNDG 715
C W1=W	AFAB 005
W2=1.0-W	AFAB 010
IF (KSS.EQ.0) GO TO 100	AFAB 015
W1=1.0	AFAB 020
W2=0.0	AFAB 025
100 CONTINUE	AFAB 030
DO 500 M=1,NEL	AFAB 035
	AFAB 040
	AFAB 045
	AFAB 050
	AFAB 055
	AFAB 060
	AFAB 065
	AFAB 070
	AFAB 075
	AFAB 080
	AFAB 085
	AFAB 090
	AFAB 095
	AFAB 100
	AFAB 105

NQ=4	AFAB 110
IEF4=IE (M, 4)	AFAB 115
IF (IEM4.EQ.0) NQ=3	AFAB 120
MTYP=IE (M, 5)	AFAB 125
MTYP=IABS (MTYP)	AFAB 130
AL=PROP (MTYP, 3)	AFAB 135
AT=PROP (MTYP, 4)	AFAB 140
AM=PROP (MTYP, 8)	AFAB 145
TAU=PROP (MTYP, 9)	AFAB 150
DD=TAU*AM	AFAB 155
C	AFAB 160
M1=IE (M, 1)	AFAB 165
M2=IE (M, 2)	AFAB 170
M3=IE (M, 3)	AFAB 175
M4=IE (M, 4)	AFAB 180
C	AFAB 185
A=Z (M2) -Z (M1)	AFAB 190
B=X (M2) -X (M1)	AFAB 195
BETA (1)=DATAN2 (A, B)	AFAB 200
A=Z (M3) -Z (M2)	AFAB 205
B=X (M3) -X (M2)	AFAB 210
BETA (2)=DATAN2 (A, B)	AFAB 215
C	AFAB 220
IF (NQ.EQ.3) GO TO 120	AFAB 225
C	AFAB 230
A=Z (M3) -Z (M4)	AFAB 235
B=X (M3) -X (M4)	AFAB 240
BETA (3)=DATAN2 (A, B)	AFAB 245
A=Z (M4) -Z (M1)	AFAB 250
B=X (M4) -X (M1)	AFAB 255
BETA (4)=DATAN2 (A, B)	AFAB 260
GO TO 160	AFAB 265
C	AFAB 270
120 CONTINUE	AFAB 275
A=Z (M1) -Z (M3)	AFAB 280
B=X (M1) -X (M3)	AFAB 285
BETA (3)=DATAN2 (A, B)	AFAB 290
C	AFAB 295
160 CONTINUE	AFAB 300
MDIM (1)=M1	AFAB 305
MDIM (2)=M2	AFAB 310
MDIM (3)=M3	AFAB 315
MDIM (4)=M4	AFAB 320
C	AFAB 325
DO 400 J=1, NQ	AFAB 330
JP I=J+1	AFAB 335
IF (J.EQ.NQ) JP I=1	AFAB 340
M1=MDIM (J)	AFAB 345
M2=MDIM (JP I)	AFAB 350
ALENG=DSQRT ((X (M2) -X (M1)) **2 + (Z (M2) -Z (M1)) **2)	AFAB 355
CBETA=DCOS (BETA (J))	AFAB 360
IF (DABS (CBETA) .LE. 1.0D-15) CBETA=0.0D0	AFAB 365
SBETA=DSIN (BETA (J))	AFAB 370
IF (DABS (SBETA) .LE. 1.0D-15) SBETA=0.0D0	AFAB 375
VXX=0.5 * ((VX (M1) +VX (M2)) *W1 + (VXP (M1) +VXP (M2)) *W2)	AFAB 380
VZZ=0.5 * ((VZ (M1) +VZ (M2)) *W1 + (VZP (M1) +VZP (M2)) *W2)	AFAB 385
VAI=VXX*CBETA+VZZ*SBETA	AFAB 390
VV=DSQRT (VXX*VXX+VZZ*VZZ)	AFAB 395
IF (VV.EQ.0.0) GO TO 330	AFAB 400
VVI=1.0/VV	AFAB 405
DLI=(AL*VXX*VXX + AT*VZZ*VZZ) *VVI + DD	AFAB 410
DLT=(AL-AT) *VXX*VZZ*VVI	AFAB 415
DTT=(AL*VZZ*VZZ + AT*VXX*VXX) *VVI + DD	AFAB 420
DAL=DABS (DLI*CBETA*CBETA+2.0D0*CBETA*SBETA*DLT+DTT*SBETA*SBETA)	AFAB 425
VEI=VAL*ALENG	AFAB 430

IF (IOPT.EQ.0) GO TO 350	AFAB 435
DISP=2.0*DAL	AFAB 440
IF (DISP.EQ.0.0 .OR. VEL.EQ.0.0) GO TO 320	AFAB 445
APHA (J)=1.0/DTANH(VEL/DISP)- DISP/VEL	AFAB 450
GO TO 400	AFAB 455
320 IF (VEL.EQ.0.0) APHA (J)=0.0	AFAB 460
IF (VEL.GT.0.0) APHA (J)=1.0D0	AFAB 465
IF (VEL.LT.0.0) APHA (J)=-1.0D0	AFAB 470
GO TO 400	AFAB 475
330 APHA (J)=0.0	AFAB 480
GO TO 400	AFAB 485
350 APHA (J)=1.0D0	AFAB 490
IF (VEL.LT.0.0) APHA (J)=-1.0D0	AFAB 495
400 CONTINUE	AFAB 500
DO 410 J=1,NQ	AFAB 505
410 WETAB(M,J)=APHA (J)	AFAB 510
500 CONTINUE	AFAB 515
RETURN	AFAB 520
END	AFAB 525
SUBROUTINE FLUX(X,Z,IE,WETAB, C,FX,PZ, E,VI,VZ,PROP,	FLUX 005
> MAXNP,MAXEL,MAXBW,MAXMAT)	FLUX 010
C	FLUX 015
C FUNCTION OF SUBROUTINE TO COMPUTE DARCY MATERIAL FLUX PX AND PZ,	FLUX 020
C RESPECTIVELY	FLUX 025
C	FLUX 030
IMPLICIT REAL*8(A-H,O-Z)	FLUX 035
C	FLUX 040
COMMON /CINTG/ NNP,NEL,NBW,NBEL,IBAND,IHBP,IHALPB,NMAT,NCH	FLUX 045
COMMON /OPTN/ IWET,ILUMP,IMID,ICONFI,ILOPT,KSORP,IPNTS,IVEL	FLUX 050
COMMON /WET/ APHA1,APHA2,BETA1,BETA2	FLUX 055
C	FLUX 060
DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)	FLUX 065
DIMENSION C(MAXNP,MAXBW),FX(MAXNP),PZ(MAXNP)	FLUX 070
DIMENSION R(MAXNP),VI(MAXNP),VZ(MAXNP),PROP(MAXMAT,15)	FLUX 075
DIMENSION QB(4,4),QRX(4),QRZ(4),IQ(4),ZQ(4),RQ(4),VIQ(4),VZQ(4)	FLUX 080
C	FLUX 085
C ----- INITIALIZE THE FLUX PX(NP) AND PZ(NP)	FLUX 090
C	FLUX 095
DO 100 NP=1,NNP	FLUX 100
FX(NP)=0.0	FLUX 105
100 PZ(NP)=0.0	FLUX 110
C	FLUX 115
C ----- COMPUTE VERTICALLY INTEGRATED FLUX BY APPLYING FEM TO	FLUX 120
C ----- THE VERTICAL INTEGRATED FLUX EQUATIONS.	FLUX 125
C	FLUX 130
IF (IPNTS.EQ.1 .OR. IVML.EQ.1) GO TO 150	FLUX 135
C	FLUX 140
DO 110 NP=1,NNP	FLUX 145
DO 110 IB=1,IBAND	FLUX 150
110 C(NP,IB)=0.0	FLUX 155
GO TO 190	FLUX 160
C	FLUX 165
150 DO 155 NP=1,NNP	FLUX 170
155 C(NP,1)=0.0	FLUX 175
190 CONTINUE	FLUX 180
C	FLUX 185
C ----- COMPUTE ELEMENT MATRICES QB(IQ,JQ) AND QRX(IQ) & QRZ(IQ)	FLUX 190
C	FLUX 195
DO 290 M=1,NEL	FLUX 200
NQ=4	FLUX 205
ITEM4=1E(M,4)	FLUX 210
IF (ITEM4.EQ.0) NQ=3	FLUX 215
C	FLUX 220
APHA1=WETAB(M,1)	FLUX 225

ALPHA2=WETAB(N,3)	FLOX 230
BETA1=WETAB(N,2)	FLOX 235
IF(NQ.EQ.3) GO TO 205	FLOX 240
BETA2=WETAB(N,4)	FLOX 245
205 CONTINUE	FLOX 250
NTYP=IE(N,5)	FLOX 255
NTYP=IABS(NTYP)	FLOX 260
AL=PROP(NTYP,3)	FLOX 265
AT=PROP(NTYP,4)	FLOX 270
POE=PROP(NTYP,6)	FLOX 275
AM=PROP(NTYP,8)	FLOX 280
TAU=PROP(NTYP,9)	FLOX 285
DD=AM*TAU*FOR	FLOX 290
C	FLOX 295
DO 210 IQ=1,NQ	FLOX 300
NP=IE(N,IQ)	FLOX 305
XQ(IQ)=X(NP)	FLOX 310
ZQ(IQ)=Z(NP)	FLOX 315
RQ(IQ)=R(NP)	FLOX 320
VXQ(IQ)=VX(NP)	FLOX 325
VZQ(IQ)=VZ(NP)	FLOX 330
210 CONTINUE	FLOX 335
C	FLOX 340
IF(NQ.EQ.3) CALL Q3D(QB,QRX,QRZ,XQ,ZQ,VXQ,VZQ,RQ,DD,AL,AT)	FLOX 345
IF(NQ.EQ.4) CALL Q4D(QB,QRX,QRZ,XQ,ZQ,VXQ,VZQ,RQ,DD,AL,AT)	FLOX 350
C	FLOX 355
C ----- ASSEMBLE QB(IQ,JQ) INTO THE GLOBAL MATRIX C(NP,IB) AND	FLOX 360
C ----- FORM THE LOAD VECTOR PX(NP) OR PZ(NP).	FLOX 365
C	FLOX 370
DO 280 IQ=1,NQ	FLOX 375
NI=IE(N,IQ)	FLOX 380
C	FLOX 385
DO 240 JQ=1,NQ	FLOX 390
NJ=IE(N,JQ)	FLOX 395
C	FLOX 400
IF(IFNTS.EQ.1 .OR. IVHL.EQ.1) GO TO 230	FLOX 405
C	FLOX 410
IB=NJ-NI+INBP	FLOX 415
C(NI,IB)=C(NI,IB)+QB(IQ,JQ)	FLOX 420
GO TO 240	FLOX 425
C	FLOX 430
230 C(NI,1)=C(NI,1)+QB(IQ,JQ)	FLOX 435
240 CONTINUE	FLOX 440
C	FLOX 445
PX(NI)=PX(NI)+QRX(IQ)	FLOX 450
PZ(NI)=PZ(NI)+QRZ(IQ)	FLOX 455
280 CONTINUE	FLOX 460
C	FLOX 465
290 CONTINUE	FLOX 470
C	FLOX 475
C ***** END OF FORMING MATRIX EQUATION	FLOX 480
C	FLOX 485
IF(IFNTS.EQ.1 .OR. IVHL.EQ.1) GO TO 360	FLOX 490
C	FLOX 495
C ----- SOLVE THE MATRIX EQUATION CX=B	FLOX 500
C	FLOX 505
CALL SOLVE(1,C,PX,NNP,IHALPB,MAXNP,MAXBW)	FLOX 510
CALL SOLVE(2,C,PX,NNP,IHALPB,MAXNP,MAXBW)	FLOX 515
CALL SOLVE(2,C,PZ,NNP,IHALPB,MAXNP,MAXBW)	FLOX 520
GO TO 390	FLOX 525
C	FLOX 530
360 DO 370 NP=1,NNP	FLOX 535
PX(NP)=PX(NP)/C(NP,1)	FLOX 540
370 PZ(NP)=PZ(NP)/C(NP,1)	FLOX 545
C	FLOX 550

390 CONTINUE	FLUX 555
C RETURN	FLUX 560
END	FLUX 565
	FLUX 570
SUBROUTINE Q4D(CB,QRX,QRZ,IQ,ZQ,VXQ,VZQ,RQ,DD,AL,AT)	Q4D 005
C	Q4D 010
C FUNCTION OF SUBROUTINE-TO EVALUATE THE MATRIX QUADRATURE OVER THE	Q4D 015
C AREA OF ONE ELEMENT. THESE INTEGRALS ARISE THROUGH THE	Q4D 020
C APPLICATION OF THE GALERKIN INTEGRATION SCHEME	Q4D 025
C	Q4D 030
IMPLICIT REAL*8 (A-H,O-Z)	Q4D 035
REAL*8 N(4)	Q4D 040
DIMENSION QB(4,4),QRX(4),QRZ(4),IQ(4),ZQ(4),RQ(4),VXQ(4),VZQ(4)	Q4D 045
DIMENSION S(4),T(4),W(4),DNX(4),DNZ(4),DNX(4),DNZ(4)	Q4D 050
DIMENSION PJAB(2,2),DNSS(4),DNMT(4),DNSS(4),DNMT(4)	Q4D 055
C	Q4D 060
DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-	Q4D 065
> 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /	Q4D 070
C	Q4D 075
C ----- INITIALIZE CB(IQ,JQ) AND QRX(IQ) & QRZ(IQ)	Q4D 080
C	Q4D 085
DO 100 IQ=1,4	Q4D 090
QRX(IQ)=0.0	Q4D 095
QRZ(IQ)=0.0	Q4D 100
DO 100 JQ=1,4	Q4D 105
100 QB(IQ,JQ)=0.0	Q4D 110
C	Q4D 115
C ----- EVALUATE INTEGRATIONS BY SUMMING OVER ALL GAUSSIAN POINTS	Q4D 120
C	Q4D 125
DO 590 KG=1,4	Q4D 130
C	Q4D 135
C ----- DETERMINE LOCAL COORDINATES (SS,TT) OF THE GAUSSIAN POINT KG	Q4D 140
C	Q4D 145
SS=P*S(KG)	Q4D 150
TT=P*T(KG)	Q4D 155
C	Q4D 160
C ----- COMPUTE VALUES OF BASIS AND WEIGHTING FUNCTIONS N(IQ) AND	Q4D 165
C ----- W(IQ) AND THEIR DERIVATIVES DNX DNX AND DNZ DNZ WITH RESPECT	Q4D 170
C ----- TO X AND Z, RESPECTIVELY, AT THE GAUSSIAN POINT KG.	Q4D 175
C	Q4D 180
CALL SHAPE(N,W,DNSS,DNMT,DWSS,DNMT,SS,TT)	Q4D 185
C	Q4D 190
C ----- COMPUTE THE INVERSE TRANSFORMATION MATRIX I.E. THE INVERSE	Q4D 195
C ----- JACOBIAN: PJAB(1,1)=(DZ/DT)/DJAC, PJAB(1,2)=-(DZ/DS)/DJAC,	Q4D 200
C ----- PJAB(2,1)=-(DX/DT)/DJAC, PJAB(2,2)=(DX/DS)/DJAC,	Q4D 205
C ----- WHERE DJAC IS THE DETERMINANT OF THE JACOBIAN.	Q4D 210
C	Q4D 215
C	Q4D 220
C	Q4D 225
DO 210 I=1,2	Q4D 230
DO 210 J=1,2	Q4D 235
PJAB(I,J)=0.0	Q4D 240
210 CONTINUE	Q4D 245
C	Q4D 250
DO 220 I=1,4	Q4D 255
PJAB(1,1)=PJAB(1,1) + ZQ(I)*DNMT(I)	Q4D 260
PJAB(1,2)=PJAB(1,2) - ZQ(I)*DNSS(I)	Q4D 265
PJAB(2,1)=PJAB(2,1) - XQ(I)*DNMT(I)	Q4D 270
PJAB(2,2)=PJAB(2,2) + XQ(I)*DNSS(I)	Q4D 275
220 CONTINUE	Q4D 280
C	Q4D 285
C ----- COMPUTE THE DETERMINANT OF THE JACOBIAN.	Q4D 290
C	Q4D 295
DJAC=PJAB(2,2)*PJAB(1,1) - PJAB(1,2)*PJAB(2,1)	Q4D 300

C	DJACI=1.0D0/DJAC	Q4D 305
C	DO 230 I=1,2	Q4D 310
	DO 230 J=1,2	Q4D 315
	PJAB(I,J)=PJAB(I,J)*DJACI	Q4D 320
C	230 CONTINUE	Q4D 325
		Q4D 330
C	DO 310 I=1,4	Q4D 335
	DNX(I)=PJAB(I,1)*DNSS(I) + PJAB(I,2)*DNIT(I)	Q4D 340
	DNZ(I)=PJAB(I,1)*DNSS(I) + PJAB(I,2)*DNIT(I)	Q4D 345
C	310 CONTINUE	Q4D 350
		Q4D 355
C	----- INTERPOLATE WITH THE BASIS FUNCTION TO OBTAIN THE VALUES OF	Q4D 360
C	----- INTEGRANDS AT EACH GAUSSIAN POINT.	Q4D 365
C		Q4D 370
	VXK=0.0	Q4D 375
	VZK=0.0	Q4D 380
	DO 410 IQ=1,4	Q4D 385
	VXK=VXK+N(IQ)*VXQ(IQ)	Q4D 390
	VZK=VZK+N(IQ)*VZQ(IQ)	Q4D 395
C	410 CONTINUE	Q4D 400
		Q4D 405
C	VK=DSQRT(VXK*VXK+VZK*VZK)	Q4D 410
	VKI=1.0D0/VK	Q4D 415
	AKXK=(AL*VXK*VXK+AT*VZK*VZK)*VKI + DD	Q4D 420
	AKZK=(AL*VZK*VZK+AT*VXK*VXK)*VKI + DD	Q4D 425
	AKXZK=(AL-AT)*VXK*VZK*VKI	Q4D 430
C		Q4D 435
C	----- ACCUMULATE THE SUMS TO OBTAIN THE MATRIX INTEGRALS QB(IQ,JQ)	Q4D 440
C	----- AND QRX(IQ) & QRZ(IQ)	Q4D 445
C		Q4D 450
	DO 520 IQ=1,4	Q4D 455
	DO 520 JQ=1,4	Q4D 460
	QB(IQ,JQ)=QB(IQ,JQ)+N(IQ)*N(JQ)*DJAC	Q4D 465
	QRX(IQ)=QRX(IQ)+N(IQ)*RQ(JQ)*(AKXK*DNX(JQ)+AKXZK*DNZ(JQ))*DJAC +	Q4D 470
	1 N(IQ)*RQ(JQ)*VXK*N(JQ)*DJAC	Q4D 475
	QRZ(IQ)=QRZ(IQ)+N(IQ)*RQ(JQ)*(AKXZK*DNX(JQ)+AKZK*DNZ(JQ))*DJAC +	Q4D 480
	1 N(IQ)*RQ(JQ)*VZK*N(JQ)*DJAC	Q4D 485
C		Q4D 490
	520 CONTINUE	Q4D 495
C		Q4D 500
C	590 CONTINUE	Q4D 505
		Q4D 510
C	RETURN	Q4D 515
	END	Q4D 520
		Q4D 525
	SUBROUTINE Q3D(QB,QRX,QRZ,XQ,ZQ,VXQ,VZQ,RQ,DD,AL,AT)	Q3D 005
C		Q3D 010
C	FUNCTION OF SUBROUTINE-TO EVALUATE THE MATRIX QUADRATURE OVER THE	Q3D 015
C	AREA OF ONE ELEMENT. THESE INTEGRALS ARISE THROUGH THE	Q3D 020
C	APPLICATION OF THE GALERKIN INTEGRATION SCHEME	Q3D 025
C		Q3D 030
	IMPLICIT REAL*8 (A-H,C-Z)	Q3D 035
C		Q3D 040
	DIMENSION QB(4,4),QRX(4),QRZ(4),XQ(4),ZQ(4),RQ(4),VXQ(4),VZQ(4)	Q3D 045
	DIMENSION A(3),E(3),C(3)	Q3D 050
	DIMENSION AKXG(3),AKZG(3),AKXZG(3)	Q3D 055
C		Q3D 060
	DATA OSIX/0.1666666666666667D0/	Q3D 065
	DATA OTWLV/0.8333333333333333D-1/	Q3D 070
	DATA OTHRTY/0.3333333333333333D-1/	Q3D 075
	DATA OSIXTY/0.1666666666666667D-1/	Q3D 080
C		Q3D 085
	DO 100 IQ=1,4	Q3D 090
	QX(IQ)=0.0	Q3D 095

QRZ(IQ)=0.0	Q3D 100
DO 100 JQ=1,4	Q3D 105
100 QB(IQ,JQ)=0.0	Q3D 110
C	Q3D 115
X1=XQ(1)	Q3D 120
X2=XQ(2)	Q3D 125
X3=XQ(3)	Q3D 130
Z1=ZQ(1)	Q3D 135
Z2=ZQ(2)	Q3D 140
Z3=ZQ(3)	Q3D 145
C	Q3D 150
A(1)=X2*Z3 - X3*Z2	Q3D 155
A(2)=X3*Z1 - X1*Z3	Q3D 160
A(3)=X1*Z2 - X2*Z1	Q3D 165
C	Q3D 170
AREA2=DABS(A(1)+A(2)+A(3))	Q3D 175
AREA=AREA2/2.000	Q3D 180
C	Q3D 185
B(1)=(Z2-Z3)/AREA2	Q3D 190
B(2)=(Z3-Z1)/AREA2	Q3D 195
B(3)=(Z1-Z2)/AREA2	Q3D 200
C(1)=(X3-X2)/AREA2	Q3D 205
C(2)=(X1-X3)/AREA2	Q3D 210
C(3)=(X2-X1)/AREA2	Q3D 215
C	Q3D 220
DRDX=0.0	Q3D 225
DRDZ=0.0	Q3D 230
DO 210 IQ=1,3	Q3D 235
DRDX=DRDX+BQ(IQ)*B(IQ)	Q3D 240
210 DRDZ=DRDZ+BQ(IQ)*C(IQ)	Q3D 245
C	Q3D 250
DO 250 IQ=1,3	Q3D 255
VIX=VXQ(IQ)	Q3D 260
VIZ=VZQ(IQ)	Q3D 265
VK=DSQRT(VIX*VIX+VIZ*VIZ)	Q3D 270
IF(VK.EQ.0.0) GO TO 230	Q3D 275
VKI=1.000/VK	Q3D 280
AKXG(IQ)=(AL*VIX*VIX+AT*VIZ*VIZ)*VKI+DD	Q3D 285
AKZG(IQ)=(AL*VIZ*VIZ+AT*VIX*VIX)*VKI+DD	Q3D 290
AKXZG(IQ)=(AL-AT)*VIX*VIZ*VKI	Q3D 295
GO TO 250	Q3D 300
230 AKXG(IQ)=DD	Q3D 305
AKZG(IQ)=DD	Q3D 310
AKXZG(IQ)=0.0	Q3D 315
250 CONTINUE	Q3D 320
C	Q3D 325
C	Q3D 330
C ***** COMPUTE QB(I,J) = INTEGRATION OF W(I)*N(J).	Q3D 335
C	Q3D 340
C \$\$\$\$\$\$ GALERKIN WEIGHTING FUNCTION	Q3D 345
C	Q3D 350
AOSIX=AREA*OSIX	Q3D 355
AOTWLV=AREA*OTWLV	Q3D 360
DO 310 IQ=1,3	Q3D 365
DO 310 JQ=1,3	Q3D 370
IF(IQ.EQ.JQ) QB(IQ,JQ)=AOSIX	Q3D 375
IF(IQ.NE.JQ) QB(IQ,JQ)=AOTWLV	Q3D 380
310 CONTINUE	Q3D 385
C	Q3D 390
C ***** COMPUTE QRX(I) = INTEGRATION OF W(I)*(VX*C - EX.D.GRAD(C)) OR	Q3D 395
C ***** COMPUTE QRZ(I) = INTEGRATION OF W(I)*(VY*C - EY.D.GRAD(C))	Q3D 400
C ***** WHERE W(I) = I-TH WEIGHTING FUNCTION, VX AND VY ARE THE	Q3D 405
C ***** VELOCITY COMPONENTS ALONG X- AND Y-AXIS, RESPECTIVELY,	Q3D 410
C ***** EX AND EY ARE THE UNIT VECTOR ALONG THE X- AND Y-AXIS,	Q3D 415
C ***** RESPECTIVELY, D IS THE DISPERSION TENSOR, AND C IS THE	Q3D 420

C ***** CONCENTRATION. C IS STORED IN RQ(IQ) AND D IS STORED IN
C ***** AKIG(IQ), AKZG(IQ), AND AKXZG(IQ).

C
C \$\$\$\$\$\$\$\$\$\$ GALERKIN WEIGHTING FUNCTION

QEX(1)=AREA*((0.1D0*VXQ(1)+OTHRTY*(VXQ(2)+VXQ(3)))*RQ(1) +
1 (OTHRTY*(VXQ(1)+VXQ(2))+OSIXTY*VXQ(3))*RQ(2) +
2 (OTHRTY*(VXQ(1)+VXQ(3))+OSIXTY*VXQ(2))*RQ(3) -
3 OSIX*(DRDX*AKIG(1)+DRDZ*AKXZG(1)) -
4 OTWLVE*(DRDX*AKIG(2)+DRDZ*AKXZG(2)) -
5 OTWLVE*(DRDX*AKIG(3)+DRDZ*AKXZG(3)))
QRX(2)=AREA*((0.1D0*VXQ(2)+OTHRTY*(VXQ(3)+VXQ(1)))*RQ(2) +
1 (OTHRTY*(VXQ(2)+VXQ(3))+OSIXTY*VXQ(1))*RQ(3) +
2 (OTHRTY*(VXQ(2)+VXQ(1))+OSIXTY*VXQ(3))*RQ(1) -
3 OSIX*(DRDX*AKIG(2)+DRDZ*AKXZG(2)) -
4 OTWLVE*(DRDX*AKIG(3)+DRDZ*AKXZG(3)) -
5 OTWLVE*(DRDX*AKIG(1)+DRDZ*AKXZG(1)))
QRX(3)=AREA*((0.1D0*VXQ(3)+OTHRTY*(VXQ(1)+VXQ(2)))*RQ(3) +
1 (OTHRTY*(VXQ(3)+VXQ(1))+OSIXTY*VXQ(2))*RQ(1) +
2 (OTHRTY*(VXQ(3)+VXQ(2))+OSIXTY*VXQ(1))*RQ(2) -
3 OSIX*(DRDX*AKIG(3)+DRDZ*AKXZG(3)) -
4 OTWLVE*(DRDX*AKIG(1)+DRDZ*AKXZG(1)) -
5 OTWLVE*(DRDX*AKIG(2)+DRDZ*AKXZG(2)))

C
QRZ(1)=AREA*((0.1D0*VZQ(1)+OTHRTY*(VZQ(2)+VZQ(3)))*RQ(1) +
1 (OTHRTY*(VZQ(1)+VZQ(2))+OSIXTY*VZQ(3))*RQ(2) +
2 (OTHRTY*(VZQ(1)+VZQ(3))+OSIXTY*VZQ(2))*RQ(3) -
3 OSIX*(DRDZ*AKZG(1)+DRDX*AKXZG(1)) -
4 OTWLVE*(DRDZ*AKZG(2)+DRDX*AKXZG(2)) -
5 OTWLVE*(DRDZ*AKZG(3)+DRDX*AKXZG(3)))
QRZ(2)=AREA*((0.1D0*VZQ(2)+OTHRTY*(VZQ(3)+VZQ(1)))*RQ(2) +
1 (OTHRTY*(VZQ(2)+VZQ(3))+OSIXTY*VZQ(1))*RQ(3) +
2 (OTHRTY*(VZQ(2)+VZQ(1))+OSIXTY*VZQ(3))*RQ(1) -
3 OSIX*(DRDZ*AKZG(2)+DRDX*AKXZG(2)) -
4 OTWLVE*(DRDZ*AKZG(3)+DRDX*AKXZG(3)) -
5 OTWLVE*(DRDZ*AKZG(1)+DRDX*AKXZG(1)))
QRZ(3)=AREA*((0.1D0*VZQ(3)+OTHRTY*(VZQ(1)+VZQ(2)))*RQ(3) +
1 (OTHRTY*(VZQ(3)+VZQ(1))+OSIXTY*VZQ(2))*RQ(1) +
2 (OTHRTY*(VZQ(3)+VZQ(2))+OSIXTY*VZQ(1))*RQ(2) -
3 OSIX*(DRDZ*AKZG(3)+DRDX*AKXZG(3)) -
4 OTWLVE*(DRDZ*AKZG(1)+DRDX*AKXZG(1)) -
5 OTWLVE*(DRDZ*AKZG(2)+DRDX*AKXZG(2)))

C
RETURN
END

SUBROUTINE ASEMBL(X,Z,IE,WETAB,C,R,RP,RW,VX,VXP,VZ,VZP,B,1,H,DH,
1 PROP,W,MAXNP,MAXEL,MAXBW,MAXMAT, SOS,ISTYP, NPW,IWTYP,WSS,
2 HU,BKU,CU,IHUTYP, HL,BKL,CL,IHLTYE, MXHUPR,MXHLPR,KSS,DELT,
3 ILNODE)

C
C FUNCTION OF SUBROUTINE--TO ASSEMBLE THE GLOBAL COEFFICIENT MATRIX
C C(NP,IB) AND LOAD VECTOR R(NP) FROM THE ELPMENT MATRICES QA(IQ,JQ)
C AND QF(IQ,JQ).

C
C
C
C
C IMPLICIT REAL*8 (A-H,C-Z)
C REAL*8 KD,LAMEDA

C
COMMON /CPIS/ JLE,JDIAG,JRE,JBAND
COMMON /CINTG/ NNP,NEL,NBN,NBEL,IBAND,IHBP,IHALPB,NMAT,NCH,NDTCHG
COMMON /CS/ MXSPR,MXSDP,NSPR,NSDP,MXWNP,MXWPR,MXWDP,NWNP,NWPR,NWDP
COMMON /OPTN/ IWET,ILUMP,IMID,ICONFI,IOPT,KSORP,IPNTS,IVNL
COMMON /WZT/ APHA1,APHA2,BETA1,BETA2

Q3D 425
Q3D 430
Q3D 435
Q3D 440
Q3D 445
Q3D 450
Q3D 455
Q3D 460
Q3D 465
Q3D 470
Q3D 475
Q3D 480
Q3D 485
Q3D 490
Q3D 495
Q3D 500
Q3D 505
Q3D 510
Q3D 515
Q3D 520
Q3D 525
Q3D 530
Q3D 535
Q3D 540
Q3D 545
Q3D 550
Q3D 555
Q3D 560
Q3D 565
Q3D 570
Q3D 575
Q3D 580
Q3D 585
Q3D 590
Q3D 595
Q3D 600
Q3D 605
Q3D 610
Q3D 615
Q3D 620
Q3D 625
Q3D 630
Q3D 635
Q3D 640
Q3D 645

ASEM 005
ASEM 010
ASEM 015
ASEM 020
ASEM 025
ASEM 030
ASEM 035
ASEM 040
ASEM 045
ASEM 050
ASEM 055
ASEM 060
ASEM 065
ASEM 070
ASEM 075
ASEM 080
ASEM 085
ASEM 090

C	DIMENSION I(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)	ASEM 095
	DIMENSION C(MAXIF,MAXBW),R(MAXNP),RP(MAXNP),RW(MAXNP)	ASEM 100
	DIMENSION ILMODE(JFAND,MAXNP)	ASEM 105
	DIMENSION V1(MAXNP),VIP(MAXNP),VZ(MAXNP),VZP(MAXNP),B(MAXNP),	ASEM 110
	I H(MAXNP),DH(MAXNP),PROP(MAXHAT,15),T(MAXNP)	ASEM 115
	DIMENSION SOS(MISPR),ISTYP(MAXEL)	ASEM 120
	DIMENSION WSS(FWPR),NPR(MXWNP),INTYP(MXWNP)	ASEM 125
	DIMENSION HU(MXHUPR),RKU(MXHUPR),CU(MXHUPR),IHUTYP(MAXEL)	ASEM 130
	DIMENSION HL(MXHLPR),EKL(MXHLPR),CL(MXHLPR),IHLYTP(MAXEL)	ASEM 135
C		ASEM 140
	DIMENSION QA(4,4),QB(4,4),VXQ(4),VZQ(4),XQ(4),ZQ(4),DSQ(4),	ASEM 145
	> RC(4),HNBQ(4),HONHQ(4),HLHQ(4)	ASEM 150
	DIMENSION DSDCQ(4),SPQ(4),CPQ(4)	ASEM 155
C		ASEM 160
	DEITI=1./DELT	ASEM 165
	W1=W	ASEM 170
	W2=1.-W	ASEM 175
	IF (KSS.NE.0) GO TO 100	ASEM 180
	DEITI=0.	ASEM 185
	W1=1.	ASEM 190
	W2=0.	ASEM 195
C		ASEM 200
C	----- INITIALIZE MATRICES C(NP,IB) AND R(NP)	ASEM 205
C		ASEM 210
	100 IF (IFNTS.EQ.1) GO TO 140	ASEM 215
C		ASEM 220
	DO 120 NP=1,NMP	ASEM 225
	R(NP)=0.0	ASEM 230
	DO 120 IB=1,IBAND	ASEM 235
	120 C(NP,IB)=0.0	ASEM 240
	GO TO 190	ASEM 245
C		ASEM 250
	140 DO 145 NP=1,NMP	ASEM 255
	R(NP)=0.0	ASEM 260
	DO 145 I=1,JFAND	ASEM 265
	C(NP,I)=0.0	ASEM 270
	145 CONTINUE	ASEM 275
	190 CONTINUE	ASEM 280
C		ASEM 285
C	----- COMPUTE MATRICES QA(IQ,JQ) AND QB(IQ,JQ) FOR EACH ELEMENT M.	ASEM 290
C		ASEM 295
	DO 690 M=1,NEL	ASEM 300
	NQ=4	ASEM 305
	IM4=IE(M,4)	ASEM 310
	IF (IM4.EQ.0) NQ=3	ASEM 315
C		ASEM 320
	I=ISTYP(M)	ASEM 325
	SOSH=SOS(I)	ASEM 330
C		ASEM 335
	APEA1=WETAB(M,1)	ASEM 340
	APHA2=WETAB(M,3)	ASEM 345
	BETA1=WETAB(M,2)	ASEM 350
	IF (NQ.EQ.3) GO TO 205	ASEM 355
	BETA2=WETAB(M,4)	ASEM 360
	205 CONTINUE	ASEM 365
C		ASEM 370
	MCCNFI=IE(M,5)	ASEM 375
	MTYP=IABS(MCCNFI)	ASEM 380
	KD=PROP(MTYP,1)	ASEM 385
	RHCB=PROP(MTYP,2)	ASEM 390
	AL=PROP(MTYP,3)	ASEM 395
	AT=PROP(MTYP,4)	ASEM 400
	LAMBDA=PROP(MTYP,5)	ASEM 405
	POE=PROP(MTYP,6)	ASEM 410
		ASEM 415

ALF=PROP (MTYP,7)	ASEM 420
AM=PROP (MTYP,8)	ASEM 425
TAU=PROP (MTYP,9)	ASEM 430
DECPKW=PROP (MTYP,10)	ASEM 435
DECPKS=PROP (MTYP,11)	ASEM 440
FRENDK=PROP (MTYP,12)	ASEM 445
FRENDN=PROP (MTYP,13)	ASEM 450
RLANGK=PROP (MTYP,14)	ASEM 455
RLANGX=PROP (MTYP,15)	ASEM 460
C	ASEM 465
I=IHUTYP (H)	ASEM 470
HUM=HU (I)	ASEM 475
RKUM=RKU (I)	ASEM 480
CUP=CU (I)	ASEM 485
I=IHUTYP (H)	ASEM 490
HLE=EL (I)	ASEM 495
RKLN=RKL (I)	ASEM 500
CLF=CL (I)	ASEM 505
C	ASEM 510
DO 250 IQ=1,NQ	ASEM 515
NP=IE (H,IQ)	ASEM 520
XQ (IQ)=X (NP)	ASEM 525
ZQ (IQ)=Z (NP)	ASEM 530
DHQ (IQ)=DH (NP)	ASEM 535
IF (MCONF1.GT.0) DHQ (IQ)=0.0	ASEM 540
HMEQ (IQ)=H (NP) -B (NP)	ASEM 545
IF (MCONF1.GT.0) HMBQ (IQ)=T (NP)	ASEM 550
HUMHQ (IQ)=RKUM*(HUM-H (NP))	ASEM 555
HLMHQ (IQ)=RKLN*(HLE-H (NP))	ASEM 560
VXQ (IQ)=VX (NP) *W1+VXP (NP) *W2	ASEM 565
VZQ (IQ)=VZ (NP) *W1+VZP (NP) *W2	ASEM 570
CP=W1*RW (NP) +W2*RP (NP)	ASEM 575
CPC (IQ)=CP	ASEM 580
C	ASEM 585
GO TC (210,220,230), KSOEP	ASEM 590
C	ASEM 595
210 DSDCQ (IQ)=FD	ASEM 600
SPC (IQ)=KD*CF	ASEM 605
GO TC 250	ASEM 610
C	ASEM 615
220 DSDCQ (IQ)=FRENDN*FRIENDK*CP** (FRENDN-1.0D0)	ASEM 620
SPC (IQ)=FRIENDK*CP**FRENDN	ASEM 625
GO TO 250	ASEM 630
C	ASEM 635
230 DSDCQ (IQ)=RLANGK*RLANGX/((1.0D0+RLANGK*CP)*(1.0D0+RLANGK*CP))	ASEM 640
SPC (IQ)=RLANGK*RLANGX*CP/(1.0D0+RLANGK*CP)	ASEM 645
250 CONTINUE	ASEM 650
C	ASEM 655
IF (NQ.EQ.3) CALL Q3 (QA,QB,XQ,ZQ,VXC,VZQ,DHQ,HMBQ, KD,RHOB,AL,AT,	ASEM 660
1 LAMBDA,POR,ALP,AM,TAU, RQ,SOSH, DECPKW,DECPKS,	ASEM 665
2 HUMHQ,CUM,HLMHQ,CLM, DSDCQ,SPQ,CPQ)	ASEM 670
IF (NQ.EQ.4) CALL Q4 (QA,QB,XQ,ZQ,VXC,VZQ,DHQ,HMBQ, KD,RHOB,AL,AT,	ASEM 675
1 LAMBDA,POR,ALP,AM,TAU, RQ,SOSH, DECPKW,DECPKS,	ASEM 680
2 HUMHQ,CUM,HLMHQ,CLM, DSDCQ,SPQ,CPQ)	ASEM 685
C	ASEM 690
----- ASSEMBLE QA (IQ,JQ) AND QB (IQ,JQ) INTO THE GLOBAL MATRIX	ASEM 695
C ----- C (NP, IQ) = W1*B + A/DELT AND FORM THE LOAD VECTOR	ASEM 700
C ----- R (NP) = (A/DELT -W2*B) *RP. MATRIX C IS ASYMMETRIC DUE TO	ASEM 705
C ----- ADVECTION TERM.	ASEM 710
C	ASEM 715
IF (IMID.EQ.1) GO TO 400	ASEM 720
C	ASEM 725
DO 390 IQ=1,NQ	ASEM 730
NI=IE (H,IQ)	ASEM 735
R (NI)=R (NI)+RQ (IQ)	ASEM 740

C	DO 340 JQ=1,NQ	ASEM 745
	NJ=IE(N,JQ)	ASEM 750
	QA(IQ,JQ)=QA(IQ,JQ)*DELT I	ASEM 755
	R(NI)=R(NI)+(QA(IQ,JQ)-N2*QB(IQ,JQ))*RP(NJ)	ASEM 760
C		ASEM 765
	IF(IPNTS.EQ.1) GO TO 320	ASEM 770
C		ASEM 775
	IB=NJ-NI+IHBP	ASEM 780
	C(NI,IB)=C(NI,IB)+QA(IQ,JQ)+N1*QB(IQ,JQ)	ASEM 785
	GO TC 340	ASEM 790
C		ASEM 795
320	DO 325 I=1,JBAND	ASEM 800
	LNODE=LLNODE(I,NI)	ASEM 805
	IF(NJ.EQ.1*LNODE) GO TO 330	ASEM 810
325	CONTINUE	ASEM 815
C		ASEM 820
	WRITE(6,1000) NI,N,JQ	ASEM 825
	STCP	ASEM 830
C		ASEM 835
330	C(NI,I)=C(NI,I)+QA(IQ,JQ)+N1*QB(IQ,JQ)	ASEM 840
C		ASEM 845
340	CONTINUE	ASEM 850
C		ASEM 855
390	CONTINUE	ASEM 860
C		ASEM 865
	GO TO 690	ASEM 870
C		ASEM 875
400	DO 490 IQ=1,NQ	ASEM 880
	NI=IE(N,IQ)	ASEM 885
	R(NI)=R(NI)+RQ(IQ)	ASEM 890
C		ASEM 895
	DO 440 JQ=1,NQ	ASEM 900
	NJ=IE(N,JQ)	ASEM 905
	QA(IQ,JQ)=2.0D0*QA(IQ,JQ)*DELT I	ASEM 910
	R(NI)=R(NI) + QA(IQ,JQ)*RP(NJ)	ASEM 915
C		ASEM 920
	IF(IPNTS.EQ.1) GO TO 420	ASEM 925
C		ASEM 930
	IB=NJ-NI+IHBP	ASEM 935
	C(NI,IB)=C(NI,IB) + QA(IQ,JQ) + QB(IQ,JQ)	ASEM 940
	GO TO 440	ASEM 945
C		ASEM 950
420	DO 425 I=1,JBAND	ASEM 955
	LNODE=LLNODE(I,NI)	ASEM 960
	IF(NJ.EQ.1*LNODE) GO TO 430	ASEM 965
425	CONTINUE	ASEM 970
C		ASEM 975
	WRITE(6,1000) NI,N,JQ	ASEM 980
	STCP	ASEM 985
C		ASEM 990
430	C(NI,I)=C(NI,I)+QA(IQ,JQ)+QB(IQ,JQ)	ASEM 995
C		ASEM 1000
440	CONTINUE	ASEM 1005
C		ASEM 1010
490	CONTINUE	ASEM 1015
C		ASEM 1020
690	CONTINUE	ASEM 1025
C		ASEM 1030
-----	INCORPORATE WELL SOURCE/SINK CONDITIONS	ASEM 1035
C		ASEM 1040
	IF(NWNP.EQ.0) GO TO 900	ASEM 1045
	DO 790 I=1,NWNP	ASEM 1050
	NI=NFW(I)	ASEM 1055
	ITYP=IWTYP(I)	ASEM 1060
		ASEM 1065

WSSI=WSS(IITY)	ASEM1076
IF (WSSI.LT.0.0) GO TO 750	ASEM1075
R(NI)=R(NI)+WSSI	ASEM1080
GO TO 790	ASEM1085
750 IF (IPNTS.EQ.1) GO TO 770	ASEM1090
C(NI,IHBP)=C(NI,IHBP)-WSSI	ASEM1095
GO TO 790	ASEM1100
770 C(NI,JDIAG)=C(NI,JDIAG)-WSSI	ASEM1105
790 CONTINUE	ASEM1110
990 CONTINUE	ASEM1115
C	ASEM1120
1000 FORMAT(1H1/5X,'*** WARNING: NONE OF THE LOWER-LEFT NODE IN EQUATIO	ASEM1125
1N',I3,/5X,'*** IS CORRESPONDING TO ',I5,'-TH ELEMENT-S',I2,	ASEM1130
2'-TH NODE; STOP *****)	ASEM1135
C	ASEM1140
RETURN	ASEM1145
END	ASEM1150
SUBROUTINE Q4(CA,QE,XQ,ZQ,VXQ,VZQ,DHQ, HMBQ, KD,RHOB,AL,AT,	Q4 C05
1 LAMBDA,POR,ALP,AM,TAU, RQ,SOSH, DECPKW,DECPKS,	Q4 C10
2 HUMHQ,CUM,ELMHQ,CLM, DSDCQ,SPQ,CPQ)	Q4 C15
C	Q4 C20
C	Q4 C25
C FUNCTION OF SUBROUTINE--TO EVALUATE THE MATRIX QUADRATURES QA(IQ,JQ)	Q4 C30
C AND QE(IQ,JQ) OVER THE AREA OF ONE ELEMENT.	Q4 C35
C	Q4 C40
C	Q4 C45
IMPLICIT REAL*8 (A-H,O-Z)	Q4 C50
REAL*8 N(4),NN,KD,LAMBDA	Q4 C55
COMMON /WET/ ALPHA1,ALPHA2,BETA1,BETA2	Q4 C60
COMMON /OPTN/ IWET,ILUMP,IMID,ICONFI,IOPT,KSORP,IPNTS,IWML	Q4 C65
C	Q4 C70
DIMENSION W(4),DNX(4),DNZ(4),DWX(4),DWZ(4)	Q4 C75
DIMENSION PJAB(2,2),DNSS(4),DNIT(4),DNSS(4),DNIT(4)	Q4 C80
DIMENSION QA(4,4),QB(4,4),VXQ(4),VZQ(4),XQ(4),ZQ(4),DHQ(4),	Q4 C85
> RQ(4),HMBQ(4),HUMHQ(4),ELMHQ(4)	Q4 C90
DIMENSION DSDCQ(4),SPQ(4),CPQ(4)	Q4 C95
DIMENSION S(4),T(4)	Q4 C100
C	Q4 C105
DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-	Q4 C110
> 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /	Q4 C115
C	Q4 C120
DD=AM*TAU	Q4 C125
C	Q4 C130
C ----- INITIATE MATRICES QA(IQ,JQ), QB(IQ,JQ), AND RQ(IQ)	Q4 C135
C	Q4 C140
DO 120 IQ=1,4	Q4 C145
RQ(IQ)=0.0	Q4 C150
DO 110 JO=1,4	Q4 C155
QA(IQ,JO)=0.0	Q4 C160
110 QB(IQ,JO)=0.0	Q4 C165
120 CONTINUE	Q4 C170
C	Q4 C175
C ----- EVALUATE INTEGRATIONS BY SUMMING OVER ALL GAUSSIAN POINTS	Q4 C180
C	Q4 C185
DO 590 KG=1,4	Q4 C190
C	Q4 C195
C ----- DETERMINE LOCAL COORDINATES (SS,TT) OF THE GAUSSIAN POINT KG	Q4 C200
C	Q4 C205
SS=P*S(KG)	Q4 C210
TT=P*T(KG)	Q4 C215
C	Q4 C220
C ----- COMPUTE VALUES OF BASIS AND WEIGHTING FUNCTIONS N(IQ) AND	Q4 C225
C ----- W(IQ) AND THEIR DERIVATIVES DWX DNX AND DWZ DNZ WITH RESPECT	Q4 C230

C	----- TO X AND Z, RESPECTIVELY, AT THE GAUSSIAN POINT KG.	Q4	235
C		Q4	240
	CALL SHAPE(N,W,DNSS,DNTT,DWSS,DWTT,SS,TT)	Q4	245
C		Q4	250
C	----- COMPUTE THE INVERSE TRANSFORMATION MATRIX I.E. THE INVERSE	Q4	255
C	----- JACOBIAN: PJAB(1,1)=(DZ/DT)/DJAC, PJAB(1,2)=-(DZ/DS)/DJAC,	Q4	260
C	----- PJAB(2,1)=-(DX/DT)/DJAC, PJAB(2,2)=(DX/DS)/DJAC,	Q4	265
C	----- WHERE DJAC IS THE DETERMINANT OF THE JACOBIAN.	Q4	270
C		Q4	275
C		Q4	280
C		Q4	285
	DO 210 I=1,2	Q4	290
	DO 210 J=1,2	Q4	295
	PJAB(I,J)=0.0	Q4	300
	210 CONTINUE	Q4	305
C		Q4	310
	DO 220 I=1,4	Q4	315
	PJAB(1,1)=PJAB(1,1) + ZQ(I)*DNTT(I)	Q4	320
	PJAB(1,2)=PJAB(1,2) - ZQ(I)*DNSS(I)	Q4	325
	PJAB(2,1)=PJAB(2,1) - IQ(I)*DNTT(I)	Q4	330
	PJAB(2,2)=PJAB(2,2) + IQ(I)*DNSS(I)	Q4	335
	220 CONTINUE	Q4	340
C		Q4	345
C	----- COMPUTE THE DETERMINANT OF THE JACOBIAN.	Q4	350
C		Q4	355
	DJAC=PJAB(2,2)*PJAB(1,1) - PJAB(1,2)*PJAB(2,1)	Q4	360
	DJACI=1.0D0/DJAC	Q4	365
C		Q4	370
	DO 230 I=1,2	Q4	375
	DO 230 J=1,2	Q4	380
	PJAB(I,J)=PJAB(I,J)*DJACI	Q4	385
	230 CONTINUE	Q4	390
C		Q4	395
	DO 310 I=1,4	Q4	400
	DNX(I)=PJAB(1,1)*DNSS(I) + PJAB(1,2)*DNTT(I)	Q4	405
	DNZ(I)=PJAB(2,1)*DNSS(I) + PJAB(2,2)*DNTT(I)	Q4	410
	DWX(I)=PJAB(1,1)*DWSS(I) + PJAB(1,2)*DWTT(I)	Q4	415
	DWZ(I)=PJAB(2,1)*DWSS(I) + PJAB(2,2)*DWTT(I)	Q4	420
	310 CONTINUE	Q4	425
C		Q4	430
C	----- INTERPOLATE WITH THE BASIS FUNCTION TO OBTAIN THE VALUES OF	Q4	435
C	----- INTEGRANDS AT EACH GAUSSIAN POINT.	Q4	440
C		Q4	445
	HMEK=0.0	Q4	450
	HUMK=0.0	Q4	455
	HLMHK=0.0	Q4	460
	DHK=0.0	Q4	465
	VXK=0.0	Q4	470
	VZK=0.0	Q4	475
	DSECK=0.0	Q4	480
	SORPSK=0.0	Q4	485
	DO 410 IQ=1,4	Q4	490
	HMEK=HMEK+N(IQ)*HMEQ(IQ)	Q4	495
	HUMK=HUMK+N(IQ)*HUMHQ(IQ)	Q4	500
	HLMHK=HLMHK+N(IQ)*HLMHQ(IQ)	Q4	505
	DHK=DHK+N(IQ)*DHQ(IQ)	Q4	510
	VXK=VXK+N(IQ)*VXQ(IQ)	Q4	515
	VZK=VZK+N(IQ)*VZQ(IQ)	Q4	520
	DSDCK=DSDCK+N(IQ)*DSDCQ(IQ)	Q4	525
	SORPSK=SORPSK+N(IQ)*(SPQ(IQ)-DSDCQ(IQ)*CPQ(IQ))	Q4	530
	410 CONTINUE	Q4	535
C		Q4	540
	VK=DSQRT(VXK*VXK+VZK*VZK)	Q4	545
	VKI=1.0D0/VK	Q4	550
C		Q4	555

C	----- ACCUMULATE SUMS TO EVALUATE THE MATRICES AQ(IQ,JQ), QB(IQ,JQ)	Q4	560
C	----- AND RQ(IQ).	Q4	565
C		Q4	570
C		Q4	575
	DXX=DJAC*((AL*VXK*VXK+AT*VZK*VZK)*VKI+DD*POR)	Q4	580
	DZZ=DJAC*((AL*VZK*VZK+AT*VXK*VXK)*VKI+DD*POR)	Q4	585
	DXZ=DJAC*(AL-AT)*VXK*VZK*VKI	Q4	590
C		Q4	595
	VXK=VXK*DJAC	Q4	600
	VZK=VZK*DJAC	Q4	605
C		Q4	610
	SOSK=SOSM*DJAC	Q4	615
	GAMAS=1.0D0	Q4	620
	IF (SOSK.LT.0.0) GAMAS=-1.0D0	Q4	625
	SORPSK=SORPSK*DJAC*HMBK	Q4	630
C		Q4	635
	HUMHK=HUMHK*DJAC	Q4	640
	GAMAU=1.0D0	Q4	645
	IF (HUMHK.LT.0.0) GAMAU=-1.0D0	Q4	650
	HLMHK=HLMHK*DJAC	Q4	655
	GAMAL=1.0D0	Q4	660
	IF (HLMHK.LT.0.0) GAMAL=-1.0D0	Q4	665
C		Q4	670
	A=LJAC*(POR+RHOB*DSDCK)*HMBK	Q4	675
	B=0.5D0*(1.0D0+GAMAS)*SOSK - RHOB*(ALP*DHK+LAMBDA+DECEKS)*SORPSK+	Q4	680
	1 0.5D0*(1.0D0+GAMAU)*HUMHK*CUH + 0.5D0*(1.0D0+GAMAL)*HLMHK*CLH	Q4	685
	C=DJAC*(ALP*(POR+RHOB*DSDCK)*DHK+LAMBDA*(POR+RHOB*DSDCK) +	Q4	690
	1 (DECPKN*POR)+(DECPKS*RHOB*DSDCK))*HMBK-0.5D0*(1.0D0-GAMAS)*SOSK-	Q4	695
	2 0.5D0*(1.0D0-GAMAU)*HUMHK-0.5D0*(1.0D0-GAMAL)*HLMHK	Q4	700
C		Q4	705
	DO 520 IQ=1,4	Q4	710
	RQ(IQ)=RQ(IQ) + N(IQ)*B	Q4	715
	DO 510 JQ=1,4	Q4	720
	WN=N(JQ)*N(JQ)	Q4	725
	DW1DNX=DNX(IQ)*DNX(JQ)	Q4	730
	DW2DNZ=DNZ(IQ)*DNZ(JQ)	Q4	735
	DWYN=DWY(IQ)*N(JQ)	Q4	740
	DWZN=DWZ(IQ)*N(JQ)	Q4	745
	DW2DNX=DNZ(IQ)*DNX(JQ)	Q4	750
	DW1DNZ=DNX(IQ)*DNZ(JQ)	Q4	755
	QA(IQ,JQ)=QA(IQ,JQ) + A*WN	Q4	760
	QB(IQ,JQ)=QB(IQ,JQ)+DW1DNX*DXX+DXZ*(DW1DNZ+DW2DNX)+DZZ*DW2DNZ -	Q4	765
	1 (VXK*DWYN+VZK*DWZN)+C*WN	Q4	770
	510 CONTINUE	Q4	775
	520 CONTINUE	Q4	780
C		Q4	785
	590 CONTINUE	Q4	790
C		Q4	795
C	***** CHECK IF MASS MATRIX IS TO BE LUMPED?	Q4	800
C		Q4	805
	IF (ILUMP.NE.0) GO TO 600	Q4	810
C		Q4	815
C	----- MASS LUMPING OPTION IS TO BE USED.	Q4	820
C		Q4	825
	RETURN	Q4	830
C		Q4	835
C	----- MID-DIFFERENCE OPTION IS USED	Q4	840
C		Q4	845
	600 DO 620 I=1,4	Q4	850
	SUM=0.0	Q4	855
	DO 610 J=1,4	Q4	860
	SUM=SUM+QA(I,J)	Q4	865
	QA(I,J)=0.0	Q4	870
	610 CONTINUE	Q4	875
	QA(I,I)=SUM	Q4	880

E20 CONTINUE	Q4 885
C	Q4 890
C	Q4 895
RETURN	Q4 900
END	Q4 905
SUBROUTINE SHAPE(M,N,DN1SS,DN1TT,DN2SS,DN2TT,SS,TT)	SHAP 005
C	SHAP 010
C FUNCTION OF THE SUBROUTINE-TO COMPUTE THE VALUES OF BASIS FUNCTIONS	SHAP 015
C AND WEIGHTING FUNCTIONS IN LOCAL COORDINATES	SHAP 020
C	SHAP 025
IMPLICIT REAL*8 (A-H,O-Z)	SHAP 030
REAL*8 N(4)	SHAP 035
C	SHAP 040
COMMON /OPTN/ IWET,ILUMP,IMID,ICONFI,IOP,KSORP,IPNTS,IVHL	SHAP 045
COMMON /NET/ ALPHA1,ALPHA2,BETA1,BETA2	SHAP 050
DIMENSION DN1SS(4),DN1TT(4),DN2SS(4),DN2TT(4),N(4)	SHAP 055
C	SHAP 060
C ----- DEFINITION OF WEIGHTING FUNCTIONS AND THEIR DERIVATIVES.	SHAP 065
C	SHAP 070
WF1(X,A)=0.25D0*((2.0D0-3.0D0*A)-2.0D0*X+3.0D0*A*X*X)	SHAP 075
WF2(X,A)=0.25D0*((2.0D0+3.0D0*A)+2.0D0*X-3.0D0*A*X*X)	SHAP 080
DWF1(X,A)=0.5D0*(-1.0D0+3.0D0*A*X)	SHAP 085
DWF2(X,A)=0.5D0*(1.0D0-3.0D0*A*X)	SHAP 090
C	SHAP 095
C ----- COMPUTE BASIS FUNCTIONS AND THEIR DERIVATIVES.	SHAP 100
C	SHAP 105
SH=1.0D0-SS	SHAP 110
SP=1.0D0+SS	SHAP 115
TH=1.0D0-TT	SHAP 120
TP=1.0D0+TT	SHAP 125
N(1)=0.25D0*SH*TH	SHAP 130
N(2)=0.25D0*SP*TH	SHAP 135
N(3)=0.25D0*SP*TP	SHAP 140
N(4)=0.25D0*SH*TP	SHAP 145
DN1SS(1)=-0.25D0*TH	SHAP 150
DN1SS(2)=0.25D0*TH	SHAP 155
DN1SS(3)=0.25D0*TP	SHAP 160
DN1SS(4)=-0.25D0*TP	SHAP 165
DN1TT(1)=-0.25D0*SH	SHAP 170
DN1TT(2)=-0.25D0*SP	SHAP 175
DN1TT(3)=0.25D0*SP	SHAP 180
DN1TT(4)=0.25D0*SH	SHAP 185
C	SHAP 190
C ----- COMPUTE WEIGHTING FUNCTIONS AND THEIR DERIVATIVES.	SHAP 195
C	SHAP 200
IF (IWET.EQ.0) GO TO 100	SHAP 205
C	SHAP 210
W(1)=WF1(SS,ALPHA1)*WF1(TT,BETA2)	SHAP 215
W(2)=WF2(SS,ALPHA1)*WF1(TT,BETA1)	SHAP 220
W(3)=WF2(SS,ALPHA2)*WF2(TT,BETA1)	SHAP 225
W(4)=WF1(SS,ALPHA2)*WF2(TT,BETA2)	SHAP 230
DN2SS(1)=DWF1(SS,ALPHA1)*WF1(TT,BETA2)	SHAP 235
DN2SS(2)=DWF2(SS,ALPHA1)*WF1(TT,BETA1)	SHAP 240
DN2SS(3)=DWF2(SS,ALPHA2)*WF2(TT,BETA1)	SHAP 245
DN2SS(4)=DWF1(SS,ALPHA2)*WF2(TT,BETA2)	SHAP 250
DN2TT(1)=WF1(SS,ALPHA1)*DWF1(TT,BETA2)	SHAP 255
DN2TT(2)=WF2(SS,ALPHA1)*DWF1(TT,BETA1)	SHAP 260
DN2TT(3)=WF2(SS,ALPHA2)*DWF2(TT,BETA1)	SHAP 265
DN2TT(4)=WF1(SS,ALPHA2)*DWF2(TT,BETA2)	SHAP 270
RETURN	SHAP 275
C	SHAP 280
100 DO 200 I=1,4	SHAP 285
W(I)=N(I)	SHAP 290

DNSS(I)=DNSS(I)	SHAP 295
200 DWT(I)=DWT(I)	SHAP 300
RETURN	SHAP 305
END	SHAP 310
SUBROUTINE Q3(QA,QB,XQ,ZQ,VXQ,VZQ,DHQ, HMBQ, KD,RHOB,AL,AT,	Q3 005
1 LAMBDA,POR,ALP,AM,TAU, RQ,SOSH, DECPKW,DECPKS,	Q3 010
2 HOMHQ,CUM,HLHHQ,CLM, DSDCQ,SPQ,CPQ)	Q3 015
C	Q3 020
C FUNCTION OF SUBROUTINE--TO EVALUATE THE MATRIX QUADRATURES QA(IQ,JQ)	Q3 025
C AND QE(IQ,JQ) GVEE THE AREA OF ONE ELEMENT.	Q3 030
C	Q3 035
IMPLICIT REAL*8 (A-H,O-Z)	Q3 040
COMMON /WET/ APHA1,APHA2,BETA1,BETA2	Q3 045
COMMON /OPTN/ IWET,ILUMP,INID,ICONFI,IOPT,KSORP,IPNTS,IVHL	Q3 050
C	Q3 055
DIMENSION QA(4,4),QB(4,4),VXQ(4),VZQ(4),XQ(4),ZQ(4),DHQ(4),	Q3 060
> RQ(4),HMBQ(4),HOMHQ(4),HLHHQ(4)	Q3 065
DIMENSION DSDCQ(4),SPQ(4),CPQ(4)	Q3 070
DIMENSION A(3),B(3),C(3)	Q3 075
DIMENSION AKXG(3),AKZG(3),AKXZG(3)	Q3 080
DIMENSION APA(3),FPQA(3),FPQB(3),FPRQ(3)	Q3 085
C	Q3 090
DATA OSIX/0.16666666666666667D0/	Q3 095
DATA OTWLVE/0.8333333333333333D-1/	Q3 100
DATA OTHRTY/0.3333333333333333D-1/	Q3 105
DATA OSIXTY/0.16666666666666667D-1/	Q3 110
C	Q3 115
DD=AM*TAU*POR	Q3 120
C	Q3 125
C ----- INITIATE MATRICES QA(IQ,JQ), QB(IQ,JQ), AND RQ(IQ)	Q3 130
C	Q3 135
DO 120 IQ=1,4	Q3 140
RQ(IQ)=0.0	Q3 145
DO 110 JQ=1,4	Q3 150
QA(IQ,JQ)=0.0	Q3 155
110 QB(IQ,JQ)=0.0	Q3 160
120 CONTINUE	Q3 165
C	Q3 170
X1=XQ(1)	Q3 175
X2=XQ(2)	Q3 180
X3=XQ(3)	Q3 185
Z1=ZQ(1)	Q3 190
Z2=ZQ(2)	Q3 195
Z3=ZQ(3)	Q3 200
C	Q3 205
A(1)=X2*Z3 - X3*Z2	Q3 210
A(2)=X3*Z1 - X1*Z3	Q3 215
A(3)=X1*Z2 - X2*Z1	Q3 220
C	Q3 225
AREA2=DABS(A(1)+A(2)+A(3))	Q3 230
AREA=AREA2/2.0D0	Q3 235
C	Q3 240
B(1)=(Z2-Z3)/AREA2	Q3 245
B(2)=(Z3-Z1)/AREA2	Q3 250
B(3)=(Z1-Z2)/AREA2	Q3 255
C(1)=(X3-X2)/AREA2	Q3 260
C(2)=(X1-X3)/AREA2	Q3 265
C(3)=(X2-X1)/AREA2	Q3 270
C	Q3 275
DO 150 IQ=1,3	Q3 280
VXK=VXQ(IQ)	Q3 285
VZK=VZQ(IQ)	Q3 290
VK=DSQRT(VXK*VXK+VZK*VZK)	Q3 295
IF(VK.EQ.0.0) GO TO 140	Q3 300

VKI=1.0D0/VK	Q3	305
AKYG(IQ)=(AL*VK*VK*AT*VZK*VZK)*VKI+DD	Q3	310
AKZG(IQ)=(AL*VZK*VZK*AT*VK*VK)*VKI+DD	Q3	315
AKXZG(IQ)=(AL-AT)*VK*VZK*VKI	Q3	320
GO TO 150	Q3	325
140 AKYG(IQ)=DD	Q3	330
AKZG(IQ)=DD	Q3	335
AKXZG(IQ)=0.0	Q3	340
150 CONTINUE	Q3	345
C IF(IWET.EQ.0) GO TO 170	Q3	350
C	Q3	355
AFA(1)=APHA1	Q3	360
APA(2)=BETA1	Q3	365
APA(3)=APHA2	Q3	370
170 CONTINUE	Q3	375
C	Q3	380
C ***** COMPUTE QA(I,J) = THE INTEGRATION OF	Q2	385
C ***** W(I)*(POR+RHOB*DSDC)*HMB*W(J).	Q3	390
C	Q3	395
DO 210 IQ=1,3	Q3	400
210 PFQA(IQ)=(POR+RHOB*DSDCQ(IQ))*HMBQ(IQ)	Q3	405
C	Q3	410
DO 290 IQ=1,3	Q3	415
GO TO (220,240,260),IQ	Q3	420
220 I=1	Q3	425
J=2	Q3	430
K=2	Q3	435
GO TO 280	Q3	440
240 I=2	Q3	445
J=3	Q3	450
K=1	Q3	455
GO TO 280	Q3	460
260 I=3	Q3	465
J=1	Q3	470
K=2	Q3	475
C	Q3	480
280 QA(I,I)=AREA*(PFQA(I)*0.1D0 + PFQA(J)*OTHRTY + PFQA(K)*OTHRTY)	Q3	485
QA(I,J)=AREA*(PFQA(I)*OTHRTY + PFQA(J)*OTHRTY + PFQA(K)*OSIXTY)	Q3	490
QA(I,K)=AREA*(PFQA(I)*OTHRTY + PFQA(J)*OSIXTY + PFQA(K)*OTHRTY)	Q3	495
C	Q3	500
290 CONTINUE	Q3	505
C	Q3	510
C ***** COMPUTE QB(I,J) = THE INTEGRATION OF	Q3	515
C ***** GRAD(W(I)).D.GRAD(W(J)) - GRAD(W(I)).V*W(J) +	Q3	520
C ***** W(I)*LANADA*(POR+RHOB*DSDC)*HMB*W(J) +	Q3	525
C ***** W(I)*ALP*(POR+RHOB*DSDC)*HMB*W(J) +	Q3	530
C ***** W(I)*(POR*KW+RHOB*KS*DSDC)*HMB*W(J) +	Q3	535
C ***** W(I)*KU*(H-HU)*W(J) + W(I)*KL*(H-HL)*W(J) - W(I)*M.	Q3	540
C ***** WHERE W(I) = I-TH WEIGHTING FUNCTION, W(J) = J-TH BASE	Q3	545
C ***** FUNCTION; D = DISPERSION TENSOR X THICKNESS; V = VELOCITY X	Q3	550
C ***** THICKNESS = VERTICALLY INTEGRATED FLUX; LANADA = DECAY	Q3	555
C ***** CONSTANT, POR = EFFECTIVE POROSITY, RHOB = BULK DENSITY,	Q3	560
C ***** DSDC = DERIVATIVE OF SORBED CONCENTRATION WITH RESPECT TO	Q3	565
C ***** DISSOLVED CONCENTRATION, HMB = THICKNESS;	Q3	570
C ***** ALP = MODIFIED COMPRESSIBILITY OF THE MEDIA;	Q3	575
C ***** KW = DEGRADATION RATE CONSTANT THROUGH DISSOLVED PHASE,	Q3	580
C ***** KS = DEGRADATION RATE CONSTANT THROUGH SORBED PHASE;	Q3	585
C ***** KU = EXCHANGE COEFFICIENT IN THE UPPER CONFINING AQUITARD,	Q3	590
C ***** HU = HYDRAULIC HEAD IN THE UPPER CONFINING AQUITARD,	Q3	595
C ***** KL = EXCHANGE COEFFICIENT IN THE LOWER CONFINING AQUITARD,	Q3	600
C ***** HL = HYDRAULIC HEAD IN THE LOWER CONFINING AQUITARD,	Q3	605
C ***** M = THE SINK.	Q3	610
C ***** NOTE: M IS INCLUDED IN QB IF IT IS NEGATIVE; OTHERWISE IN RQ.	Q3	615
C ***** NOTE: KU*(H-HU) IS INCLUDED IN QB IF IT IS POSITIVE;	Q3	620
	Q3	625


```

C ***** OTHERWISE IT WILL BE INCLUDED IN RQ. Q3 630
C ***** NOTE: KL*(H-HL) IS INCLUDED IN QB IF IT IS POSITIVE; Q3 635
C ***** OTHERWISE IT WILL BE INCLUDED IN RQ. Q3 640
C Q3 645
      AREAD=AREA/3.0D0 Q3 650
      AKXQS=AKIG(1)+AKIG(2)+AKIG(3) Q3 655
      AKZQS=AKZG(1)+AKZG(2)+AKZG(3) Q3 660
      AKXZQS=AKXZG(1)+AKXZG(2)+AKXZG(3) Q3 665
C Q3 670
      DO 310 IQ=1,3 Q3 675
      SOFPSK=(SPQ(IQ)-DSDCQ(IQ)*CPQ(IQ))*HMBQ(IQ) Q3 680
      HUNHK=HUMHC(IQ) Q3 685
      HLMHK=HLMHC(IQ) Q3 690
      GANAS=1.0D0 Q3 695
      IF(SCSH.LT.0.0) GANAS=-1.0D0 Q3 700
      GANAU=1.0D0 Q3 705
      IF(HUNHK.LT.0.0) GANAU=-1.0D0 Q3 710
      GANAL=1.0D0 Q3 715
      IF(HLMHK.LT.0.0) GANAL=-1.0D0 Q3 720
      PFQB(IQ)=PFQA(IQ)*(LANBDA+ALP*DHQ(IQ)) + HMBQ(IQ)*(POR*DECPK + Q3 725
1 DEC PK*RHOB*DSDCQ(IQ)) - 0.5D0*((1.0D0-GANAU)*HUNHK+(1.0D0-GANAL) Q3 730
2*HLMHK+(1.0D0-GANAS)*SOSH) Q3 735
      PFQ(IQ)=-RHOB*(LANBDA+ALP*DHQ(IQ)+DECPK)*SORPSK + 0.5D0*( Q3 740
1 (1.0D0+GANAU)*HUNHK*CUH+(1.0D0+GANAL)*HLMHK*CLH+(1.0D0+GANAS)*SOSH) Q3 745
310 CONTINUE Q3 750
C Q3 755
      DO 390 IQ=1,3 Q3 760
      GO TO (320,340,360),IQ Q3 765
320 I=1 Q3 770
      J=2 Q3 775
      K=3 Q3 780
      GO TO 380 Q3 785
340 I=2 Q3 790
      J=3 Q3 795
      K=1 Q3 800
      GO TO 380 Q3 805
360 I=3 Q3 810
      J=1 Q3 815
      K=2 Q3 820
C Q3 825
C ----- CONTRIBUTION FROM DISPERSION TERM. Q3 830
C Q3 835
380 QB(I,I)=AREAD*(B(I)*B(I)*AKXQS+C(I)*C(I)*AKZQS + Q3 840
1 (E(I)*C(I)+E(I)*C(I))*AKXZQS) Q3 845
      QB(I,J)=AREAD*(B(I)*B(J)*AKXQS+C(I)*C(J)*AKZQS + Q3 850
1 (E(I)*C(J)+E(J)*C(I))*AKXZQS) Q3 855
      QB(I,K)=AREAD*(B(I)*B(K)*AKXQS+C(I)*C(K)*AKZQS + Q3 860
1 (E(I)*C(K)+E(K)*C(I))*AKXZQS) Q3 865
C Q3 870
C ----- CONTRIBUTION FROM ADVECTION TERM. Q3 875
C Q3 880
      QB(I,I)=QB(I,I) - (B(I)*(OSIX*VXQ(I)+OTWLVE*(VXQ(J)+VXQ(K))) + Q3 885
1 C(I)*(OSIX*VZQ(I)+OTWLVE*(VZQ(J)+VZQ(K))))*AREA Q3 890
      QB(I,J)=QB(I,J) - (B(I)*(OSIX*VXQ(J)+OTWLVE*(VXQ(K)+VXQ(I))) + Q3 895
1 C(I)*(OSIX*VZQ(J)+OTWLVE*(VZQ(K)+VZQ(I))))*AREA Q3 900
      QB(I,K)=QB(I,K) - (B(I)*(OSIX*VXQ(K)+OTWLVE*(VXQ(I)+VXQ(J))) + Q3 905
1 C(I)*(OSIX*VZQ(K)+OTWLVE*(VZQ(I)+VZQ(J))))*AREA Q3 910
C Q3 915
C ----- CONTRIBUTION FROM DECAY, CONSOLIDATION, DEGRADATION, Q3 920
C ----- LEAKING OUT, AND SINK. Q3 925
C Q3 930
      QB(I,I)=QB(I,I) + AREA*(PFQB(I)*0.1D0+(PFQB(J)+PFQB(K))*OTHRTY) Q3 935
      QB(I,J)=QB(I,J) + AREA*(OTHRTY*(PFQB(I)+PFQB(J))+OSIXTY*PFQB(K)) Q3 940
      QB(I,K)=QB(I,K) + AREA*(OTHRTY*(PFQB(I)+PFQB(K))+OSIXTY*PFQB(J)) Q3 945

```

C	IF (INET.EQ.0) GO TO 390	Q3	950
C		Q3	955
C	\$\$\$\$\$ ADD ADDITIONAL TERMS DUE TO UPSTREAM WEIGHTING.	Q3	960
C		Q3	965
C		Q3	970
C	----- CONTRIBUTION FROM ADVECTION TERM.	Q3	975
C		Q3	980
C		Q3	985
	VXIL=0.3D0*VXQ(I) + 0.1D0*(VXQ(J)+VXQ(K))	Q3	990
	VXJL=0.3D0*VXQ(J) + 0.1D0*(VXQ(K)+VXQ(I))	Q3	995
	VXKL=0.3D0*VXQ(K) + 0.1D0*(VXQ(I)+VXQ(J))	Q3	1000
	VXIS=0.05D0*VXQ(I) + 0.1D0*(VXQ(J)+VXQ(K))	Q3	1005
	VXJS=0.05D0*VXQ(J) + 0.1D0*(VXQ(K)+VXQ(I))	Q3	1010
	VXKS=0.05D0*VXQ(K) + 0.1D0*(VXQ(I)+VXQ(J))	Q3	1015
	VZIL=0.3D0*VZQ(I) + 0.1D0*(VZQ(J)+VZQ(K))	Q3	1020
	VZJL=0.3D0*VZQ(J) + 0.1D0*(VZQ(K)+VZQ(I))	Q3	1025
	VZKL=0.3D0*VZQ(K) + 0.1D0*(VZQ(I)+VZQ(J))	Q3	1030
	VZIS=0.05D0*VZQ(I) + 0.1D0*(VZQ(J)+VZQ(K))	Q3	1035
	VZJS=0.05D0*VZQ(J) + 0.1D0*(VZQ(K)+VZQ(I))	Q3	1040
	VZKS=0.05D0*VZQ(K) + 0.1D0*(VZQ(I)+VZQ(J))	Q3	1045
	QB(I,I)=QB(I,I)-(AFA(J)*(B(K)*VXIL+B(I)*VXJS+C(K)*VZIL+C(I)*VZJS)	Q3	1050
	1 -AFA(K)*(B(J)*VXIL+B(I)*VXKS+C(J)*VZIL+C(I)*VZKS)*AREA	Q3	1055
	QB(I,J)=QB(I,J)-(AFA(J)*(B(K)*VXKS+B(I)*VXIS+C(K)*VZKS+C(I)*VZIS)	Q3	1060
	1 -AFA(K)*(B(J)*VXKS+B(I)*VXJL+C(J)*VZKS+C(I)*VZJL)*AREA	Q3	1065
	QB(I,K)=QB(I,K)-(AFA(J)*(B(K)*VXJS+B(I)*VXKL+C(K)*VZJS+C(I)*VZKL)	Q3	1070
	1 -AFA(K)*(B(J)*VXJS+B(I)*VXIS+C(J)*VZJS+C(I)*VZIS)*AREA	Q3	1075
C		Q3	1080
	390 CONTINUE	Q3	1085
C		Q3	1090
C	***** COMPUTE BQ(I) = THE INTEGRATION OF	Q3	1095
C	***** W(I)*(H - ALP*RHOB*(S-DSDC)*HMB - LAMADA*RHOB*(S-DSDC)*HMB -	Q3	1100
C	***** RHOB*KS*(S-DSDC)*HMB - KU*(H-HU)*CU - KL*(H-HL)*CL).	Q3	1105
C	***** WHERE H IS THE SOURCE, CU IS THE CONCENTRATION IN THE UPPER	Q3	1110
C	***** CONFINING AQUITARD, AND CL IS THE CONCENTRATION IN THE	Q3	1115
C	***** LOWER CONFINING AQUITARD.	Q3	1120
C		Q3	1125
	BQ(1)=(OSIX*PFRQ(1)+OTWLV*(PFRQ(2)+PFRQ(3)))*AREA	Q3	1130
	BQ(2)=(OSIX*PFRQ(2)+OTWLV*(PFRQ(3)+PFRQ(1)))*AREA	Q3	1135
	BQ(3)=(OSIX*PFRQ(3)+OTWLV*(PFRQ(1)+PFRQ(2)))*AREA	Q3	1140
C		Q3	1145
C	----- IF MASS MATRIX IS NOT TO BE LUMPED , RETURN	Q3	1150
C		Q3	1155
	900 IF (ILUMP.EQ.0) RETURN	Q3	1160
C		Q3	1165
C	----- FOR THE CASES WHEN THE MASS MATRIX IS TO BE LUMPED	Q3	1170
C		Q3	1175
	DO 990 I=1,3	Q3	1180
	SUM=0.0	Q3	1185
	DO 940 J=1,3	Q3	1190
	SUM=SUM+QA(I,J)	Q3	1195
	940 QA(I,J)=0.0	Q3	1200
	QA(I,I)=SUM	Q3	1205
	990 CONTINUE	Q3	1210
C		Q3	1215
C		Q3	1220
	RETURN	Q3	1225
	END	Q3	1230
	SUBROUTINE BC(IE,WETAB,C,F,RP, VX,VXP,VZ,VZP, CON,IDTYP,NPD,	BC	005
	> DI,DCOSX,DCOSZ,NRSE,IS,CIN,IRSTYP, W, MAXNP,MAXEL,MAXBW,KSS,	BC	010
	> LINODE)	BC	015
C		BC	020
C	FUNCTION OF SUBROUTINE--TO APPLY CONSTANT-CONCENTRATION DIRICHLET	BC	025
C	CONDITIONS AND BOTH CONSTANT AND TRANSIENT NEUMANN SURFACE	BC	030
C	BOUNDARY CONDITIONS.	BC	035

C		BC	040
C		BC	045
	IMPLICIT REAL*8 (A-H,O-Z)	BC	050
	REAL*8 KD	BC	055
C		BC	060
	COMMON /CPIS/ JLE,JDIAG,JRB,JBAND	BC	065
	COMMON /CINTG/ WNP,NEL,NBN,NBEL,IBAND,IHBP,IHALPB,NMAT,NCH,NDTCHG	BC	070
	COMMON /KSEND/ NXRSP,NXRSEL,NXRSPB,NXRSDP,NRSH,NRSEL,NRSPB,NRSDP	BC	075
	COMMON /DIRICH/ NXDNP,NXDPR,NXDPP,NDM,NDPR,NDDP	BC	080
	COMMON /OPTM/ IWET,ILUMP,IYID,ICONFI,IOPT,KSORP,IPNTS,IYML	BC	085
	COMMON /WET/ ALPHA1,ALPHA2,BETA1,BETA2	BC	090
C		BC	095
	DIMENSION IE(MAXEL,5),WETAB(MAXEL,4)	BC	100
	DIMENSION C(MAXNP,MAXBW),R(MAXNP),RP(MAXNP)	BC	105
	DIMENSION LLNODE(JBAND,MAXNP)	BC	110
	DIMENSION VI(MAXNP),VIP(MAXNP),VZ(MAXNP),VZP(MAXNP)	BC	115
	DIMENSION DL(NXRSEL),DCOSX(NXRSEL),DCOSZ(NXRSEL),NRSE(NXRSEL),	BC	120
	> IS(NXRSEL,4)	BC	125
	DIMENSION CIN(NXRSP),IRSTYP(NXRSEL)	BC	130
	DIMENSION CON(NXDPR),IDTYP(NXDNP),NPD(NXDNP)	BC	135
C		BC	140
	DIMENSION KQ(2),VXQ(4),VZQ(4),BQ(2,2),EQ(2)	BC	145
C		BC	150
	IF(NRSEL.LE.0) GO TO 500	BC	155
C		BC	160
C	----- APPLY RUN IN AND SEEP OUT BOUNDARY CONDITIONS	BC	165
C		BC	170
	W1=W	BC	175
	W2=1.-W	BC	180
	IF (KSS.NE.0) GO TO 410	BC	185
	W1=1.	BC	190
	W2=0.	BC	195
C		BC	200
410	DO 490 MP=1,NRSEL	BC	205
	ITYP=IRSTYP(MP)	BC	210
	CINMP=CIN(ITYP)	BC	215
	M=NRSE(MP)	BC	220
	NQ=4	BC	225
	IEH4=IE(M,4)	BC	230
	IF (IEH4.EQ.0) NQ=3	BC	235
	ALPHA1=WETAB(M,1)	BC	240
	ALPHA2=WETAB(M,3)	BC	245
	BETA1=WETAB(M,2)	BC	250
	IF (NQ.EQ.3) GO TO 415	BC	255
	BETA2=WETAB(M,4)	BC	260
415	CONTINUE	BC	265
	NI=IS(MP,1)	BC	270
	NJ=IS(MP,2)	BC	275
	KQ(1)=IS(MP,3)	BC	280
	KQ(2)=IS(MP,4)	BC	285
C		BC	290
	DO 420 IQ=1,NQ	BC	295
	MP=IE(M,IQ)	BC	300
	VXQ(IQ)=VX(MP)*W1+VIP(MP)*W2	BC	305
	VZQ(IQ)=VZ(MP)*W1+VZP(MP)*W2	BC	310
420	CONTINUE	BC	315
C		BC	320
	IF (NQ.EQ.3) CALL Q3VB(BQ,RQ,KQ,DL(MP),DCOSX(MP),DCOSZ(MP),CINMP,	BC	325
	1 VXQ,VZQ)	BC	330
	IF (NQ.EQ.4) CALL Q4VB(BQ,RQ,KQ,DL(MP),DCOSX(MP),DCOSZ(MP),CINMP,	BC	335
	1 VXQ,VZQ)	BC	340
C		BC	345
	LB=IHBP+(NJ-NI)	BC	350
	JB=IHBP+(NI-NJ)	BC	355

C	IF (IMID.NE.0) GO TO 460	BC	360
C	R (NI)=R (NI)+BQ (1)	BC	365
	R (NI)=R (NI)-W2*(BQ (1, 1)*RP (NI)+BQ (1, 2)*RP (NJ))	BC	370
	R (NJ)=R (NJ)+BQ (2)	BC	375
	R (NJ)=R (NJ)-W2*(BQ (2, 1)*RP (NI)+BQ (2, 2)*RP (NJ))	BC	380
C	IF (IFNTS.EQ. 1) GO TO 430	BC	385
C	C (NI, IHBP)=C (NI, IHBP)+W1*BQ (1, 1)	BC	390
	C (NI, IB)=C (NI, IB)+W1*BQ (1, 2)	BC	395
	C (NJ, IHBP)=C (NJ, IHBP)+W1*BQ (2, 2)	BC	400
	C (NJ, JB)=C (NJ, JB)+W1*BQ (2, 1)	BC	405
	GO TO 490	BC	410
C	430 C (NI, JDIAG)=C (NI, JDIAG)+W1*BQ (1, 1)	BC	415
	C (NJ, JDIAG)=C (NJ, JDIAG)+W1*BQ (2, 2)	BC	420
	DO 435 J=1, JBAND	BC	425
	LNODE=LLNODE (J, NJ)	BC	430
	IF (NI.EQ. LNODE) GO TO 440	BC	435
	435 CONTINUE	BC	440
C	WRITE (6, 1000) NJ, MP	BC	445
	1000 FORMAT (1H1/5X, '*** WARNING: NONE OF THE LOWER-LEFT NODE IN EQ.',	BC	450
	1 I3/5X, '*** IS CORRESPONDING TO ', I5, '-TH R-S SIDES OTHER NODE *')	BC	455
	STCP	BC	460
C	440 C (NJ, J)=C (NJ, J)+W1*BQ (2, 1)	BC	465
C	DO 445 I=1, JBAND	BC	470
	LNODE=LLNODE (I, NI)	BC	475
	IF (NJ.EQ. LNODE) GO TO 450	BC	480
	445 CONTINUE	BC	485
C	WRITE (6, 1000) NI, MP	BC	490
	STCP	BC	495
C	450 C (NI, I)=C (NI, I)+W1*BQ (1, 2)	BC	500
	GO TO 490	BC	505
C	460 R (NI)=R (NI)+BQ (1)	BC	510
	R (NJ)=R (NJ)+BQ (2)	BC	515
C	IF (IFNTS.EQ. 1) GO TO 470	BC	520
C	C (NI, IHBP)=C (NI, IHBP)+BQ (1, 1)	BC	525
	C (NI, IB)=C (NI, IB)+BQ (1, 2)	BC	530
	C (NJ, IHBP)=C (NJ, IHBP)+BQ (2, 2)	BC	535
	C (NJ, JB)=C (NJ, JB)+BQ (2, 1)	BC	540
	GO TO 490	BC	545
C	470 C (NI, JDIAG)=C (NI, JDIAG)+BQ (1, 1)	BC	550
	C (NJ, JDIAG)=C (NJ, JDIAG)+BQ (2, 2)	BC	555
	DO 475 J=1, JBAND	BC	560
	LNODE=LLNODE (J, NJ)	BC	565
	IF (NI.EQ. LNODE) GO TO 480	BC	570
	475 CONTINUE	BC	575
C	WRITE (6, 1000) NJ, MP	BC	580
	STCP	BC	585
C	480 C (NJ, J)=C (NJ, J)+BQ (2, 1)	BC	590
C	DO 485 I=1, JBAND	BC	595
		BC	600
		BC	605
		BC	610
		BC	615
		BC	620
		BC	625
		BC	630
		BC	635
		BC	640
		BC	645
		BC	650
		BC	655
		BC	660
		BC	665
		BC	670
		BC	675
		BC	680

LNODE=LLNODE(I,NJ)	BC	685
IF(NJ.EQ.LNODE) GO TO 487	BC	690
485 CONTINUE	BC	695
C WRITE(6,1000) NI,NP	BC	700
STCP	BC	705
C	BC	710
487 C(NI,I)=C(NI,I)+BQ(1,2)	BC	715
C	BC	720
490 CONTINUE	BC	725
C	BC	730
500 IF (NDN.EQ.0) GO TO 900	BC	735
C	BC	740
C ----- APPLY DIRICHLET BOUNDARY CONDITIONS	BC	745
C	BC	750
IF(IPNTS.EQ.1) GO TO 200	BC	755
C	BC	760
DO 190 NPP=1,NDN	BC	765
C	BC	770
C MODIFY ROW NPN(NPP) OF MATRIX C(NP,IB)	BC	775
C	BC	780
NI=NED(NPP)	BC	785
ITYP=IDTYP(NPP)	BC	790
BB=CON(ITYP)	BC	795
DO 110 IB=1,IBAND	BC	800
110 C(NI,IB)=0.0	BC	805
C(NI,IHBP)=1.0DO	BC	810
R(NI)=BB	BC	815
C	BC	820
C MODIFY LOAD VECTOR R(NP) FOR NON-ZERO BB(NPP)	BC	825
C	BC	830
IF(BB.EQ.0.0) GO TO 150	BC	835
DO 120 IB=1,IHALFB	BC	840
NJ=NI-IB	BC	845
IF(NJ.LT.1) GO TO 130	BC	850
JB=IHBP+IB	BC	855
120 R(NJ)=R(NJ)-EB*C(NJ,JB)	BC	860
130 DO 140 IB=1,IHALFB	BC	865
NJ=NI+IB	BC	870
IF(NJ.GT.NNP) GO TO 150	BC	875
JB=IHBP-IB	BC	880
140 R(NJ)=R(NJ)-EB*C(NJ,JB)	BC	885
C	BC	890
C ZERO COLUMN NPN(NPP) OF MATRIX C(NP,IB)	BC	895
C	BC	900
150 DO 160 IB=1,IHALFB	BC	905
NJ=NI-IB	BC	910
IF(NJ.LT.1) GO TO 170	BC	915
JB=IHBP+IB	BC	920
160 C(NJ,JB)=0.0	BC	925
170 DO 180 IB=1,IHALFB	BC	930
NJ=NI+IB	BC	935
IF(NJ.GT.NNP) GO TO 190	BC	940
JB=IHBP-IB	BC	945
180 C(NJ,JB)=0.0	BC	950
190 CONTINUE	BC	955
GO TO 900	BC	960
C	BC	965
200 DO 220 NPP=1,NDN	BC	970
NI=NED(NPP)	BC	975
ITYP=IDTYP(NPP)	BC	980
BB=CCN(ITYP)	BC	985
R(NI)=BB	BC	990
DO 210 I=1,JEAND	BC	995
C(NI,I)=0.0	BC	1000
	BC	1005

210 CONTINUE	BC 1010
C (NI,JDIAG)=1.0D0	BC 1015
220 CONTINUE	BC 1020
C	BC 1025
900 RETURN	BC 1030
END	BC 1035
SUBROUTINE Q4VB(BQ,RQ,KQ,DL,DCOSXQ,DCOSZQ,CINQ,VXQ,VZQ)	Q4VB 005
C	Q4VB 010
C	Q4VB 015
C FUNCTION OF SUBROUTINE--TO EVALUATE THE SEEPAGE-FLUX INTEGRALS	Q4VB 020
C ALONG THE BOUNDARY LINE EXTENDING FROM NODE LQ TO NODE HQ.	Q4VB 025
C	Q4VB 030
C	Q4VB 035
IMPLICIT REAL*8 (A-H,O-Z)	Q4VB 040
REAL*8 N(4)	Q4VB 045
C	Q4VB 050
DIMENSION W(3), DNSS(4),DNIT(4),DNSS(4),DNIT(4)	Q4VB 055
DIMENSION S(2,4),T(2,4)	Q4VB 060
DIMENSION BQ(2,2),RQ(2), KQ(2), VXQ(4),VZQ(4)	Q4VB 065
C	Q4VB 070
DATA S/-0.577350269189626D0, 0.577350269189626D0, 1.0D0, 1.0D0,	Q4VB 075
1 0.577350269189626D0, -0.577350269189626D0, -1.0D0, -1.0D0/	Q4VB 080
DATA T/-1.0D0, -1.0D0, -0.577350269189626D0, 0.577350269189626D0,	Q4VB 085
1 1.0D0, 1.0D0, 0.577350269189626D0, -0.577350269189626D0/	Q4VB 090
C	Q4VB 095
C ----- INITIALIZE MODAL COMPONENTS OF LINE INTEGRAL.	Q4VB 100
C	Q4VB 105
DO 110 JQ=1,2	Q4VB 110
RQ(JQ)=0.0	Q4VB 115
DO 110 IQ=1,2	Q4VB 120
110 BQ(IQ,JQ)=0.0	Q4VB 125
C	Q4VB 130
NQ=KQ(1)	Q4VB 135
C	Q4VB 140
DO 590 KG=1,2	Q4VB 145
SS=S(KG,NQ)	Q4VB 150
TT=T(KG,NQ)	Q4VB 155
C	Q4VB 160
C ----- CALCULATE VALUES OF BASIS AND WEIGHTING FUNCTIONS N(IQ) AND	Q4VB 165
C ----- W(IQ).	Q4VB 170
C	Q4VB 175
CALL SHAPE(N,W,DNSS,DNIT,DNSS,DNIT,SS,TT)	Q4VB 180
C	Q4VB 185
C ----- INTERPOLATE WITH BASIS FUNCTIONS N(IQ) TO OBTAIN VALUES OF	Q4VB 190
C ----- VERTICALLY INTEGRATED VOLUMETRIC FLUX.	Q4VB 195
C	Q4VB 200
VXK=0.0	Q4VB 205
VZK=0.0	Q4VB 210
DO 210 IQ=1,4	Q4VB 215
VXK=VXK+VXQ(IQ)*N(IQ)	Q4VB 220
VZK=VZK+VZQ(IQ)*N(IQ)	Q4VB 225
210 CONTINUE	Q4VB 230
C	Q4VB 235
C ----- EVALUATE THE NORMAL COMPONENT OF THE VERTICALLY INTEGRATED	Q4VB 240
C ----- VOLUMETRIC FLUX.	Q4VB 245
C	Q4VB 250
VNK=VXK*DCOSXQ+VZK*DCOSZQ	Q4VB 255
C	Q4VB 260
IF (VNK.LE.0.0) GO TO 400	Q4VB 265
C	Q4VB 270
DO 310 JQ=1,2	Q4VB 275
HQ=KQ(JQ)	Q4VB 280
DO 310 IQ=1,2	Q4VB 285

LQ=KQ(IQ)	Q4VB 290
BQ(IQ,JQ)=BQ(IQ,JQ)+H(LQ)*VHK*H(HQ)	Q4VB 295
310 CONTINUE	Q4VB 300
GO TO 590	Q4VB 305
C	Q4VB 310
400 DO 410 IQ=1,2	Q4VB 315
HQ=KQ(IQ)	Q4VB 320
RQ(IQ)=RQ(IQ)-H(HQ)*VHK*CINQ	Q4VB 325
410 CONTINUE	Q4VB 330
C	Q4VB 335
590 CONTINUE	Q4VB 340
C	Q4VB 345
DO 690 JQ=1,2	Q4VB 350
RQ(JQ)=0.5D0*DL*RQ(JQ)	Q4VB 355
DO 690 IQ=1,2	Q4VB 360
BQ(IQ,JQ)=0.5D0*DL*BQ(IQ,JQ)	Q4VB 365
690 CONTINUE	Q4VB 370
C	Q4VB 375
RETURN	Q4VB 380
END	Q4VB 385
SUBROUTINE Q3VB(BQ,RQ,KQ,DL,DCOSXQ,DCOSZQ,CINQ,VXQ,VZQ)	Q3VB 005
C	Q3VB 010
C FUNCTION OF SUBROUTINE--TO EVALUATE THE SEEPAGE-FLUX INTEGRALS	Q3VB 015
C ALONG THE BOUNDARY LINE EXTENDING FROM NODE LQ TO NODE HQ.	Q3VB 020
C	Q3VB 025
IMPLICIT REAL*8 (A-H,O-Z)	Q3VB 030
C	Q3VB 035
COMMON /WET/ APHA1,APHA2,BETA1,BETA2	Q3VB 040
COMMON /OPTN/ IWET,ILUMP,IMID,ICONFI,IOPT,KSOEP,IPNTS,IVHL	Q3VB 045
C	Q3VB 050
DIMENSION BQ(2,2),RQ(2), KQ(2), VXQ(4),VZQ(4)	Q3VB 055
DIMENSION AFA(3)	Q3VB 060
C	Q3VB 065
DATA OTHRD/0.3333333333333333D0/	Q3VB 070
DATA OSIX/0.16666666666666667D0/	Q3VB 075
DATA OTWLV/0.8333333333333333D-1/	Q3VB 080
C	Q3VB 085
C ----- INITIALIZE NODAL COMPONENTS OF LINE INTEGRAL.	Q3VB 090
C	Q3VB 095
DO 110 JQ=1,2	Q3VB 100
RQ(JQ)=0.0	Q3VB 105
DO 110 IQ=1,2	Q3VB 110
110 BQ(IQ,JQ)=0.0	Q3VB 115
C	Q3VB 120
AFA(1)=APHA1	Q3VB 125
AFA(2)=BETA1	Q3VB 130
AFA(3)=APHA2	Q3VB 135
C	Q3VB 140
IQ=KQ(1)	Q3VB 145
GO TO (210,220,230), IQ	Q3VB 150
210 I=1	Q3VB 155
J=2	Q3VB 160
K=3	Q3VB 165
GO TO 240	Q3VB 170
220 I=2	Q3VB 175
J=3	Q3VB 180
K=1	Q3VB 185
GO TO 240	Q3VB 190
230 I=3	Q3VB 195
J=1	Q3VB 200
K=2	Q3VB 205
240 VN1=VXQ(I)*DCOSXQ+VZQ(I)*DCOSZQ	Q3VB 210
VN2=VXQ(J)*DCOSXQ+VZQ(J)*DCOSZQ	Q3VB 215
C	Q3VB 220

```

C ***** FOR THE CASE OF GALERKIN WEIGHTING
C
C ----- CONTRIBUTION FROM V1
C
  IF (VN1.LT.0.0) GO TO 330
  BQ(1,1) = BQ(1,1) + DL*VN1*0.25D0
  BQ(1,2) = BQ(1,2) + DL*VN1*0TWLVE
  BQ(2,1) = BQ(2,1) + DL*VN1*0TWLVE
  BQ(2,2) = BQ(2,2) + DL*VN1*0TWLVE
  GO TO 350
C
330 RQ(1) = RQ(1) - DL*VN1*OTHRD*CINQ
  RQ(2) = RQ(2) - DL*VN1*OSIX*CINQ
C
C ----- CONTRIBUTION FROM VN2
C
350 IF (VN2.LT.0.0) GO TO 380
  BQ(1,1) = BQ(1,1) + DL*VN2*0TWLVE
  BQ(1,2) = BQ(1,2) + DL*VN2*0TWLVE
  BQ(2,1) = BQ(2,1) + DL*VN2*0TWLVE
  BQ(2,2) = BQ(2,2) + DL*VN2*0.25D0
  GO TO 400
C
380 RQ(1) = RQ(1) - DL*VN2*OSIX*CINQ
  RQ(2) = RQ(2) - DL*VN2*OTHRD*CINQ
C
400 IF (IWET.EQ.0) RETURN
C
C ***** ADD ADDITIONAL TERMS DUE TO UPSTREAM WEIGHTING.
C
  IF (VN1.LT.0.0) GO TO 430
  BQ(1,1) = BQ(1,1) - AFA(K)*DL*VN1*0.15D0
  BQ(1,2) = BQ(1,2) - AFA(K)*DL*VN1*0.1D0
  BQ(2,1) = BQ(2,1) + AFA(K)*DL*VN1*0.15D0
  BQ(2,2) = BQ(2,2) + AFA(K)*DL*VN1*0.1D0
  GO TO 450
C
C ----- CONTRIBUTION FROM VN1
C
430 RQ(1) = RQ(1) + AFA(K)*DL*VN1*CINQ*0.25D0
  RQ(2) = RQ(2) - AFA(K)*DL*VN1*CINQ*0.25D0
C
C ----- CONTRIBUTION FROM VN2
C
450 IF (VN2.LT.0.0) GO TO 480
  BQ(1,1) = BQ(1,1) - AFA(K)*DL*VN2*0.1D0
  BQ(1,2) = BQ(1,2) - AFA(K)*DL*VN2*0.15D0
  BQ(2,1) = BQ(2,1) + AFA(K)*DL*VN2*0.1D0
  BQ(2,2) = BQ(2,2) + AFA(K)*DL*VN2*0.15D0
  GO TO 500
480 RQ(1) = RQ(1) + AFA(K)*DL*VN2*CINQ*0.25D0
  RQ(2) = RQ(2) - AFA(K)*DL*VN2*CINQ*0.25D0
C
500 RETURN
  ENL

```

Q3VB 225
 Q3VB 230
 Q3VB 235
 Q3VB 240
 Q3VB 245
 Q3VB 250
 Q3VB 255
 Q3VB 260
 Q3VB 265
 Q3VB 270
 Q3VB 275
 Q3VB 280
 Q3VB 285
 Q3VB 290
 Q3VB 295
 Q3VB 300
 Q3VB 305
 Q3VB 310
 Q3VB 315
 Q3VB 320
 Q3VB 325
 Q3VB 330
 Q3VB 335
 Q3VB 340
 Q3VB 345
 Q3VB 350
 Q3VB 355
 Q3VB 360
 Q3VB 365
 Q3VB 370
 Q3VB 375
 Q3VB 380
 Q3VB 385
 Q3VB 390
 Q3VB 395
 Q3VB 400
 Q3VB 405
 Q3VB 410
 Q3VB 415
 Q3VB 420
 Q3VB 425
 Q3VB 430
 Q3VB 435
 Q3VB 440
 Q3VB 445
 Q3VB 450
 Q3VB 455
 Q3VB 460
 Q3VB 465
 Q3VB 470
 Q3VB 475
 Q3VB 480
 Q3VB 485
 Q3VB 490
 Q3VB 495
 Q3VB 500
 Q3VB 505

```

SUBROUTINE SOLVE(KKK,C,R,NMP,HALFB,MAXNP,MAXBW)
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(NP,IB)
C CONTAINS THE FULL BAND OF AN ASYMMETRIC MATRIX.

```

SOLV 005
 SOLV 010
 SOLV 015
 SOLV 020
 SOLV 025
 SOLV 030

C		SOLV 035
C		SOLV 040
	IMPLICIT REAL*8(A-H,O-Z)	SOLV 045
C		SCLV 050
	DIMENSION C(MAXNP,MAXBW),R(MAXNP)	SOLV 055
C		SOLV 060
	IHEP=IHALPB+1	SOLV 065
C		SOLV 070
C	IF KKK = 1, THEN TRIANGULARIZE THE BAND MATRIX C(NP,IB), BUT	SOLV 075
C	IF KKK = 2, THEN SIMPLY SOLVE WITH THE RIGHT-HAND SIDE R(NP)	SOLV 080
C		SCLV 085
	IF (KKK.EQ.2) GO TO 50	SOLV 090
C		SOLV 095
C	TRIANGULARIZE MATRIX C(NP,IB)	SOLV 100
C		SCLV 105
	NU=NP-IHALPB	SOLV 110
	DO 20 NI=1,NU	SOLV 115
	PIVOTI=1.0D0/C(NI,IHBP)	SOLV 120
	NJ=NI+1	SOLV 125
	IE=IHBP	SOLV 130
	NK=NI+IHALPB	SOLV 135
	DO 10 NL=NJ,NK	SOLV 140
	IB=IB-1	SOLV 145
	A=-C(NL,IB)*PIVOTI	SOLV 150
	C(NL,IB)=A	SOLV 155
	JB=IB+1	SOLV 160
	KB=IB+IHALPB	SOLV 165
	LB=IHBP-IB	SOLV 170
	DO 10 MB=JB,KB	SCLV 175
	MB=LB+MB	SOLV 180
10	C(NL,MB)=C(NL,MB)+A*C(NI,MB)	SOLV 185
20	CONTINUE	SOLV 190
	NR=NU+1	SOLV 195
	NU=NR-1	SOLV 200
	NK=NP	SOLV 205
	DO 40 NI=NR,NU	SOLV 210
	PIVOTI=1.0D0/C(NI,IHBP)	SOLV 215
	NJ=NI+1	SOLV 220
	IE=IHBP	SOLV 225
	DC 30 NL=NJ,NK	SOLV 230
	IB=IB-1	SOLV 235
	A=-C(NL,IB)*PIVOTI	SOLV 240
	C(NL,IB)=A	SOLV 245
	JB=IB+1	SOLV 250
	KB=IB+IHALPB	SOLV 255
	LB=IHBP-IB	SOLV 260
	DO 30 MB=JB,KB	SOLV 265
	MB=LB+MB	SCLV 270
30	C(NL,MB)=C(NL,MB)+A*C(NI,MB)	SOLV 275
40	CONTINUE	SOLV 280
	RETURN	SOLV 285
C		SCLV 290
C	MODIFY LOAD VECTOR R(NP)	SOLV 295
C		SOLV 300
	50 NU=NP+1	SOLV 305
	IBAND=2*IHALPB+1	SOLV 310
	DO 70 NI=2,IHBP	SOLV 315
	IE=IHBP-NI+1	SOLV 320
	NJ=1	SOLV 325
	SUM=0.0	SOLV 330
	DO 60 JB=IB,IHALPB	SOLV 335
	SUM=SUM+C(NI,JB)*R(NJ)	SOLV 340
60	NJ=NJ+1	SOLV 345
70	R(NI)=R(NI)+SUM	SOLV 350
	IB=i	SOLV 355

NL=INBP+1	SOLV 360
DO 90 NI=NL,NBP	SOLV 365
NJ=NI-INBP+1	SOLV 370
SUM=0.0	SOLV 375
DO 80 JB=IB,INALPB	SOLV 380
SUM=SUM+C(NI,JB)*R(NJ)	SOLV 385
80 NJ=NJ+1	SCLV 390
90 R(NI)=R(NI)+SUM	SOLV 395
C	SOLV 400
C BACK SOLVE	SOLV 405
C	SOLV 410
R(NBP)=R(NBP)/C(NBP,INBP)	SOLV 415
DO 110 IB=2,INBP	SOLV 420
NI=NU-IB	SOLV 425
NJ=NI	SOLV 430
NR=INALPB+IB	SOLV 435
SUM=0.0	SOLV 440
DO 100 JB=NI,NB	SOLV 445
NJ=NJ+1	SOLV 450
100 SUM=SUM+C(NI,JB)*R(NJ)	SOLV 455
110 R(NI)=(R(NI)-SUM)/C(NI,INBP)	SOLV 460
NB=IBAN	SOLV 465
DO 130 IB=NL,NBP	SOLV 470
NI=NU-IB	SCLV 475
NJ=NI	SOLV 480
SUM=0.0	SOLV 485
DO 120 JB=NI,NB	SOLV 490
NJ=NJ+1	SOLV 495
120 SUM=SUM+C(NI,JB)*R(NJ)	SOLV 500
130 R(NI)=(R(NI)-SUM)/C(NI,INBP)	SOLV 505
RETURN	SOLV 510
END	SOLV 515
SUBROUTINE PISS(R,IL,RI,C,ONI,EPS,LRN,NBP,NITER,MAXNP,MAXBW,IBUG,	PISS 005
I KTIH)	PISS 010
C	PISS 015
C FUNCTION OF THIS SUBROUTINE IS TO SOLVE THE MATRIX EQUATION USING	PISS 020
C POINTWISE INTERSECTION SOLUTION STRATEGIES.	PISS 025
C	PISS 030
IMPLICIT REAL*8(A-H,O-Z)	PISS 035
C	PISS 040
COMMON /CPIS/ JLE,JDIAG,JRB,JBAND	PISS 045
C	PISS 050
DIMENSION R(MAXNP),C(MAXNP,MAXBW)	PISS 055
DIMENSION RL(MAXNP),RI(MAXNP),LRN(JBAND,MAXNP)	PISS 060
C	PISS 065
C ----- STORE THE LOAD VECTOR IN TEMPORARY STORAGE RL(MAXNP).	PISS 070
C	PISS 075
DO 190 NP=1,NBP	PISS 080
190 RL(NP)=R(NP)	PISS 085
C	PISS 090
C ***** NOW R(MAXNP) IS FREE AND CAN BE USED TO STORE THE SOLUTION.	PISS 095
C	PISS 100
C ----- PRINT ITERATION INFORMATION IF DESIRED.	PISS 105
C	PISS 110
IF (IBUG.NE.0 .AND. KTIH.NE.0) PRINT 1000	PISS 115
C	PISS 120
DO 290 IT=1,NITER	PISS 125
C	PISS 130
C ----- FOR EACH ITERATION, PUT THE LOAD VECTOR INTO R(MAXNP).	PISS 135
C	PISS 140
DO 210 NP=1,NBP	PISS 145
210 R(NP)=RL(NP)	PISS 150
C	PISS 155

C ----- THE MATRIX C = L + I + U, WHERE L IS THE LOWER TRIANGULAR	PISS 160
C ----- MATRIX AND U IS THE UPPER TRIANGULAR MATRIX AND I IS THE	PISS 165
C ----- DIAGONAL MATRIX.	PISS 170
C ----- NOW ADD U.RI TO THE RIGHT HAND SIDE, WHERE RI IS THE PREVIOUS	PISS 175
C ----- ITERATE.	PISS 180
C	PISS 185
DO 230 NP=1,NMF	PISS 190
DO 220 I=JRB,JEAND	PISS 195
LNCDE=LRN(I, NP)	PISS 200
IF (LNODE.EQ.0) GO TO 220	PISS 205
R(NP)=R(NP)-C(NP,I)*RI(LNODE)	PISS 210
220 CONTINUE	PISS 215
230 CONTINUE	PISS 220
C	PISS 225
C ----- START TO COMPUTE NEW ITERATE WITH POINTWISE ITERATIONS.	PISS 230
C	PISS 235
DO 250 NP=1,NMF	PISS 240
DO 240 I=1,JLE	PISS 245
LNCDE=LRN(I, NP)	PISS 250
IF (LNODE.EQ.0) GO TO 240	PISS 255
R(NP)=R(NP)-C(NP,I)*R(LNODE)	PISS 260
240 CONTINUE	PISS 265
R(NP)=ONI*(R(NP)/C(NP,JDIAG)) + (1.000-ONI)*RI(NP)	PISS 270
250 CONTINUE	PISS 275
C	PISS 280
C ----- CHECK IF A CONVERGENT SOLUTION IS OBTAINED?	PISS 285
C	PISS 290
DIFMAX=0.0	PISS 295
NOCCUR=1	PISS 300
DO 260 NP=1,NMF	PISS 305
IF (RI(NP).EQ.0.0) GO TO 260	PISS 310
DIF=(R(NP)-RI(NP))/RI(NP)	PISS 315
DIF=ABS(DIF)	PISS 320
IF (DIF.LE.DIFMAX) GO TO 260	PISS 325
DIFMAX=DIF	PISS 330
NOCCUR=NP	PISS 335
260 CONTINUE	PISS 340
C	PISS 345
C ----- UPDATE THE ITERATE.	PISS 350
C	PISS 355
270 DO 280 NF=1,NMF	PISS 360
280 RI(NF)=R(NF)	PISS 365
C	PISS 370
C ----- PRINT ITERATION INFORMATION IF DESIRED.	PISS 375
C	PISS 380
IF (IPUG.NE.C .AND. KTIM.NE.0) PRINT 1100, IT,DIFMAX,EPS,NOCCUR	PISS 385
C	PISS 390
IF (IT.EQ.1) GO TO 290	PISS 395
IF (DIFMAX.LT.EPS) GO TO 990	PISS 400
C	PISS 405
290 CONTINUE	PISS 410
C	PISS 415
PRINT 2000, IT,NITER,DIFMAX,EPS,NOCCUR	PISS 420
C	PISS 425
990 CONTINUE	PISS 430
C	PISS 435
1000 FORMAT(1H0//75X,5H IT,12H DIFMAX ,12H EPSLOW ,10H NOC	PISS 440
1CUE/75X,5H --,12H ----- ,12H ----- ,10H -----)	PISS 445
1100 FORMAT(1H , 75X,15,2D12.4,110)	PISS 450
2000 FORMAT(1H0/60X,43H *** WARNING: NO CONVERGENCE IN PISS AFTER ,14,	PISS 455
1 13H ITERATIONS/60X,9H NITER =,14,10H DIFMAX =,D11.4,	PISS 460
2 1CH EPSLOW =,D11.4,10H NOCCUR =,14)	PISS 465
C	PISS 470
RETURN	PISS 475
END	PISS 480

```

SUBROUTINE SFLOW(I,Z,IE,WETAB,R,PI,PZ,  PROP,DLB,DCOSXB,DCOSZB,  SFLO 005
1 WEX,ISB,NPB,BPLX,BPLIF, KPRS,NPD, FRATE,FLOW,TFLOW, HAXBP,HAXEL,  SFLO 010
2 HAXBNP,HAXBEL,HAXHAT,DELT,DE,KFLOW,B,T,H,SOS,I,TYP,WSS,IWTP,NPW,  SFLO 015
3 H0,RK0,C0,INOTYP, HL,RKL,CL,IHLTY, HXHUPR,HXHLPR)  SFLO 020
C  SFLO 025
C FUNCTION OF SUBROUTINE--TO COMPUTE BOUNDARY FLUXES, FLOW RATES,  SFLO 030
C INCREMENTAL FLOWS OCCURRING DURING TIME DELT, TOTAL FLOWS SINCE  SFLO 035
C TIME ZERO, AND THE CHANGE IN MOISTURE CONTENT FOR THE ENTIRE  SFLO 040
C SYSTEM DURING TIME DELT.  SFLO 045
C  SFLO 050
C  SFLO 055
C  INELICIT REAL*8 (A-H,O-Z)  SFLO 060
C  REAL*8 KD,LANSDA  SFLO 065
C  SFLO 070
C  COMMON /CINTG/ NNP,NEL,NBN,NBEL,IBAND,IHBP,IHALPB,NHAT,NCH,NPTCHG  SFLO 075
C  COMMON /CS/ HXSPR,HXSDP,HXSPR,NPD,HXBNP,HXWPR,HXWDP,HXWPR,HXWDP  SFLO 080
C  COMMON /OPTN/ IWET,ILUMP,INID,ICONFI,IOPR,KSOEP,IPNTS,IWEL  SFLO 085
C  COMMON /WET/ APHA1,APHA2,BETA1,BETA2  SFLO 090
C  COMMON /RSPND/ HXRSNP,HXRSNP,HXRSNP,HXRSNP,HXRSNP,HXRSNP,HXRSNP  SFLO 095
C  COMMON /DIRICH/ HXIDNP,HXIDNP,HXIDNP,HXIDNP,HXIDNP,HXIDNP,HXIDNP  SFLO 100
C  SFLO 105
C  DIMENSION I(HAXNP),Z(HAXNP),IE(HAXEL,5),WETAB(HAXEL,4)  SFLO 110
C  DIMENSION R(HAXNP),PI(HAXNP),PZ(HAXNP),DE(HAXNP)  SFLO 115
C  DIMENSION DLF(HAXBEL),DCOSIB(HAXBEL),DCOSZB(HAXBEL),BPLX(HAXBNP),  SFLO 120
C  > BPLIF(HAXBNP),NBP(HAXBEL),ISB(HAXBEL,4),NPB(HAXBNP)  SFLO 125
C  DIMENSION NPRS(HXRSNP),NPD(HXIDNP)  SFLO 130
C  DIMENSION PROP(HAXHAT,15)  SFLO 135
C  DIMENSION FRATE(14),FLOW(14),TFLOW(14)  SFLO 140
C  DIMENSION B(HAXNP),H(HAXNP),T(HAXNP)  SFLO 145
C  SFLO 150
C  DIMENSION SOS(HXSPR),I,TYP(HAXEL)  SFLO 155
C  DIMENSION WSS(HXWPR),IWTP(HXWNP),NPW(HXWNP)  SFLO 160
C  DIMENSION H0(HXHUPR),RK0(HXHUPR),C0(HXHUPR), INOTYP(HAXEL)  SFLO 165
C  DIMENSION HL(HXHLPR),RKL(HXHLPR),CL(HXHLPR), IHLTY(HAXEL)  SFLO 170
C  SFLO 175
C  DIMENSION IC(4),ZQ(4),RQ(4),DEQ(4),HBBQ(4),HUNHQ(4),HLNHQ(4)  SFLO 180
C  DIMENSION CSC(4)  SFLO 185
C  SFLO 190
C  SFLO 195
C  DATA QR,QD,QL,GDH /0.000,0.000,0.000,0.000/  SFLO 200
C  DATA QLKW,QLKS /0.000,0.000/  SFLO 205
C  DATA H0LEAK,H1LEAK,SOURCE/0.000,0.000,0.000/  SFLO 210
C  SFLO 215
C  SFLO 220
C  SFLO 225
C ***** CALCULATE NODAL FLOW RATES  SFLO 230
C  SFLO 235
C  DO 110 NP=1,NBN  SFLO 240
C  BFIX(NP)=BPLX(NP)  SFLO 245
C  BFIX(NP)=0.0  SFLO 250
C 110 CONTINUE  SFLO 255
C  SFLO 260
C  DO 170 NP=1,NBEL  SFLO 265
C  NI=ISB(NP,1)  SFLO 270
C  NJ=ISB(NP,2)  SFLO 275
C  DO 120 I=1,NBN  SFLO 280
C  IJ=WE8(I)  SFLO 285
C  IF (IJ.NE.NI) GO TO 120  SFLO 290
C  NII=I  SFLO 295
C  GO TO 130  SFLO 300
C 120 CONTINUE  SFLO 305
C  SFLO 310
C 130 DO 140 J=1,NBN  SFLO 315
C  IJ=NPB(J)  SFLO 320
C  IF (IJ.NE.NJ) GO TO 140  SFLO 325

```

NJJ=J	SFLO 330
GO TO 150	SFLO 335
140 CONTINUE	SFLO 340
C	SFLO 345
150 CONTINUE	SFLO 350
C	SFLO 355
PNJ= (PX (NI) *DCOSXB (NP) +PZ (NI) *DCOSZB (NP)) *DLB (NP)	SFLO 360
PNJ= (PX (NJ) *DCOSXB (NP) +PZ (NJ) *DCOSZB (NP)) *DLB (NP)	SFLO 365
BPLX (NII)=BPLX (NII)+PNI/3. 0D0+PNJ/6. 0D0	SFLO 370
BPLX (NJJ)=BPLX (NJJ)+PNJ/3. 0D0+PNI/6. 0D0	SFLO 375
170 CONTINUE	SFLO 380
C	SFLO 385
IF (KFLOW.GT.0) GO TO 200	SFLO 390
DO 180 NP=1,NBN	SFLO 395
180 BPLXF (NP)=BPLX (NP)	SFLO 400
DO 190 I=1,14	SFLO 405
190 TFLOW (I)=0.0	SFLO 410
C	SFLO 415
C ***** DETERMINE FLOWS AND FLOW RATES THROUGH VARIOUS TYPES OF NODES,	SFLO 420
C ***** STARTING WITH THE NET FLOWS THROUGH ALL BOUNDARY NODES	SFLO 425
C	SFLO 430
200 S=C.	SFLO 435
SP=0.	SFLO 440
DO 210 NP=1,NBN	SFLO 445
S=S+BPLX (NP)	SFLO 450
SP=SP+BPLXP (NP)	SFLO 455
210 CONTINUE	SFLO 460
C	S. LO 465
FRATE (5)=S	SFLO 470
PLCW (5)=.5*(S+SP)*DELT	SFLO 475
C	SFLO 480
C ***** THROUGH DIRICHLET BOUNDARY NODES	SFLO 485
C	SFLO 490
FRATE (1)=0.	SFLO 495
PLCW (1)=0.	SFLO 500
IF (NEN.LE.0) GO TO 340	SFLO 505
S=0.	SFLO 510
SP=0.	SFLO 515
DO 330 NPP=1,NEN	SFLO 520
NF=NED (NPP)	SFLO 525
DO 310 I=1,NBN	SFLO 530
IJ=NFB (I)	SFLO 535
IF (IJ.NE.NP) GO TO 310	SFLO 540
NII=I	SFLO 545
GO TO 320	SFLO 550
310 CONTINUE	SFLO 555
C	SFLO 560
320 CONTINUE	SFLO 565
S=S+BPLX (NII)	SFLO 570
SP=SP+BPLXP (NII)	SFLO 575
330 CONTINUE	SFLO 580
C	SFLO 585
FRATE (1)=S	SFLO 590
PLCW (1)=C.5D0*(S+SP)*DELT	SFLO 595
C	SFLO 600
C ***** THROUGH PLCW-THROUGH NODES	SFLO 605
C	SFLO 610
340 FRATE (2)=0.	SFLO 615
PLCW (2)=0.	SFLO 620
FRATE (3)=0.	SFLO 625
PLCW (3)=0.	SFLO 630
IF (NRSEL.LE.0) GO TO 500	SFLO 635
SM=0.0	SFLO 640
SME=0.0	SFLO 645
S=0.	SFLO 650

SP=C.	SFLO 655
DO 490 NPP=1, NRSN	SFLO 660
NP=NRSN(NPP)	SFLO 665
DO 420 I=1, NPN	SFLO 670
IJ=NPB(I)	SFLO 675
IF (IJ.NE.NP) GO TO 420	SFLO 680
NII=I	SFLO 685
GO TO 430	SFLO 690
420 CONTINUE	SFLO 695
C	SFLO 700
430 CONTINUE	SFLO 705
IF (BFLX(NII).LT.0.0) GO TO 460	SFLO 710
S=S+BFLX(NII)	SFLO 715
GO TO 470	SFLO 720
460 SH=SH+BFLX(NII)	SFLO 725
C	SFLO 730
470 IF (BFLXP(NII).LT.0.0) GO TO 480	SFLO 735
SP=SP+BFLXP(NII)	SFLO 740
GO TO 490	SFLO 745
480 SHP=SHP+BFLXP(NII)	SFLO 750
C	SFLO 755
490 CONTINUE	SFLO 760
C	SFLO 765
FRATE(2)=SH	SFLO 770
FLCW(2)=0.5D0*(SH+SHP)*DELT	SFLO 775
FRATE(3)=S	SFLO 780
PLCW(3)=0.5D0*(S+SP)*DELT	SFLO 785
C	SFLO 790
C ***** NUMERICAL FLOW THROUGH UNSPECIFIED BOUNDARY NODES	SFLO 795
C	SFLO 800
500 S=C.	SFLO 805
SP=0.	SFLO 810
DO 510 I=1,3	SFLO 815
S=S+FRATE(I)	SFLO 820
510 SP=SP+FLOW(I)	SFLO 825
FRATE(4)=FRATE(5)-S	SFLO 830
PLCW(4)=PLCW(5)-SP	SFLO 835
C	SFLO 840
C ***** CALCULATE INCREASES OF INTEGRATED MATERIAL CONTENTS IN THE	SFLO 845
C ***** FLUID QF AND IN SOLID QD; DETERMINE LOSSES DUE TO RADIOACTIVE	SFLO 850
C ***** DECAY QL, THROUGH DISSOLVED PHASE QLKW, AND THROUGH SOLID	SFLO 855
C ***** PHASE QLKS; COMPUTE CHANGE DUE TO CONSOLIDATION QDH;	SFLO 860
C ***** INTEGRATE THE SOURCE/SINK SOURCE; AND EVALUATE THE ADDITION	SFLO 865
C ***** TO THE REGION FROM UPPER AND LOWER LEAKY AQUIFERS, HULEAK AND	SFLO 870
C ***** HLEAK, RESPECTIVELY.	SFLO 875
C	SFLO 880
QBF=QB	SFLO 885
QDF=QD	SFLO 890
QLF=QL	SFLO 895
QLKWF=QLKW	SFLO 900
QLKSF=QLKS	SFLO 905
QDHP=QDH	SFLO 910
HULEKP=HULEAK	SFLO 915
HLEKP=HLEAK	SFLO 920
SOURCE=SOURCE	SFLO 925
C	SFLO 930
QA=0.	SFLO 935
QD=0.	SFLO 940
QL=0.	SFLO 945
QLKW=0.0	SFLO 950
QLKS=0.0	SFLO 955
QDE=0.0	SFLO 960
HULEAK=0.0	SFLO 965
HLEAK=0.0	SFLO 970
SOURCE=0.0	SFLO 975

C	DO 69C N=1,NEL	SFLO 980
	NQ=4	SFLO 985
	IEH4=IE(N,4)	SFLO 990
	IF(IEH4.EQ.3) NQ=3	SFLO 995
	APHA1=WETAB(N,1)	SFLO 1000
	APHA2=WETAB(N,3)	SFLO 1005
	BETA1=WETAB(N,2)	SFLO 1010
	IF(NQ.EQ.3) GO TO 605	SFLO 1015
	BETA2=WETAB(N,4)	SFLO 1020
	605 CONTINUE	SFLO 1025
C	MCCNFI=IE(N,5)	SFLO 1030
	NTYP=IABS(MCCNFI)	SFLO 1035
	KD=PROP(NTYP,1)	SFLO 1040
	RHCB=PROP(NTYP,2)	SFLO 1045
	LAEBDA=PROP(NTYP,5)	SFLO 1050
	POB=PROP(NTYP,6)	SFLO 1055
	ALF=PROP(NTYP,7)	SFLO 1060
	RKW=PROP(NTYP,10)	SFLO 1065
	RKS=PROP(NTYP,11)	SFLO 1070
C	PRENDK=PROP(NTYP,12)	SFLO 1075
	PRENDN=PROP(NTYP,13)	SFLO 1080
	RLANGK=PROP(NTYP,14)	SFLO 1085
	RLANGX=PROP(NTYP,15)	SFLO 1090
C	ITYP=ISTYP(N)	SFLO 1095
	SOSQP=SOS(ITYP)	SFLO 1100
C	ITYP=IHUTYP(N)	SFLO 1105
	HUF=HU(ITYP)	SFLO 1110
	RKOE=RKU(ITYP)	SFLO 1115
	CUP=CU(ITYP)	SFLO 1120
C	ITYP=IHLYTP(N)	SFLO 1125
	HLE=HL(ITYP)	SFLO 1130
	RKLE=RKL(ITYP)	SFLO 1135
	CLCP=C1(ITYP)	SFLO 1140
C	DO 65C IQ=1,NQ	SFLO 1145
	NP=IE(N,IQ)	SFLO 1150
	XQ(IQ)=X(NP)	SFLO 1155
	ZQ(IQ)=Z(NP)	SFLO 1160
	DHC(IQ)=DH(NP)	SFLO 1165
	IF(MCONF1.GT.0) DHQ(IQ)=0.0	SFLO 1170
	HMEQ(IQ)=H(NP)-B(NP)	SFLO 1175
	IF(MCONF1.GT.0) HMBQ(IQ)=T(NP)	SFLO 1180
	HUMHQ(IQ)=RKOE*(HUF-H(NP))	SFLO 1185
	HLNHQ(IQ)=RKLE*(HLE-H(NP))	SFLO 1190
	RQ(IQ)=R(NP)	SFLO 1195
	CW=RQ(IQ)	SFLO 1200
C	GO TO (610,620,630), KSORP	SFLO 1205
C	610 CSC(IQ)=KD*CW	SFLO 1210
	GO TO 650	SFLO 1215
C	620 CSC(IQ)=PRENDK*CW**PRENDN	SFLO 1220
	GO TO 650	SFLO 1225
C	630 CSC(IQ)=RLANGK*RLANGX*CW/(1.0D0+RLANGK*CW)	SFLO 1230
C	650 CONTINUE	SFLO 1235
		SFLO 1240
		SFLO 1245
		SFLO 1250
		SFLO 1255
		SFLO 1260
		SFLO 1265
		SFLO 1270
		SFLO 1275
		SFLO 1280
		SFLO 1285
		SFLO 1290
		SFLO 1295
		SFLO 1300

C		SFLO1305
C	IF (MC.EQ.3) CALL Q3R(QRH,QDH,QDHH,XQ,ZQ,RQ,CSQ,DHQ,HNBQ,POR,RHOB, 1 UPLEKH,DNLEKH,SOSH,HUNHQ,HLHHQ,CUQF,CLQP,SOSQP)	SFLO1310
	IF (MC.EQ.4) CALL Q4R(QRH,QDH,QDHH,XQ,ZQ,RQ,CSQ,DHQ,HNBQ,POR,RHOB, 1 UPLEKH,DNLEKH,SOSH,HUNHQ,HLHHQ,CUQF,CLQP,SOSQP)	SFLO1315
C	QR=QB+QRH	SFLO1320
	QD=QD+QDH+RHOB	SFLO1325
	QL=CRH+QDH+RHCB	SFLO1330
	QL=QI+LAMBDA*QIH	SFLO1335
	QLKH=QLKV+BKN*CBH	SFLO1340
	QLKS=QLKS+BKS+BHOB+QDH	SFLO1345
	QDE=QDH+ALP*QDHH	SFLO1350
	HULEAK=HULEAK+UPEKH	SFLO1355
	HLLEAK=HLLEAK+DNLEKH	SFLO1360
	SOURCE=SOURCE+SOSH	SFLO1365
	690 CONTINUE	SFLO1370
C	IF (NWNP.LE.0) GO TO 705	SFLO1375
	DO 700 I=1,NWNP	SFLO1380
	ITYP=INTYP(I)	SFLO1385
	WSSI=WSS(ITYP)	SFLO1390
	NI=NEW(I)	SFLO1395
	IF (WSSI.LT.0.0) SOURCE=SOURCE-WSSI*B(NI)	SFLO1400
	700 IF (WSSI.GE.0.0) SOURCE=SOURCE-WSSI	SFLO1405
C	705 IF (KFLOW.GT.0) GO TO 710	SFLO1410
	QRE=C.0	SFLO1415
	QDE=0.0	SFLO1420
	QLE=QL	SFLO1425
	QDHP=QDH	SFLO1430
	QLKWF=QLKW	SFLO1435
	QLKSF=QLKS	SFLO1440
	HULEKP=HULEAK	SFLO1445
	HLLEKP=HLLEAK	SFLO1450
	SOURSP=SOURCE	SFLO1455
	S=FRATE(5)+QL+QDH+QLKW+QLKS+HULEAK+HLLEAK+SOURCE	SFLO1460
	SUE=CR+QD	SFLO1465
C	710 FLCW(6)=QE-QRP	SFLO1470
	FRATE(6)=FLOW(6)/DELT	SFLO1475
	IF (KFLOW.LE.0 .AND. SUM.WE.0.0 DO) FRATE(6)=-S*QR/SUM	SFLO1480
	FLCW(7)=QD-QDP	SFLO1485
	FRATE(7)=FLOW(7)/DELT	SFLO1490
	IF (KFLOW.LE.0 .AND. SUM.WE.C.0 DO) FRATE(7)=-S*QD/SUM	SFLO1495
C	FRATE(8)=QL	SFLO1500
	FRATE(9)=QDH	SFLO1505
	FRATE(10)=QLKW	SFLO1510
	FRATE(11)=QLKS	SFLO1515
	FRATE(12)=HULEAK	SFLO1520
	FRATE(13)=HLLEAK	SFLO1525
	FRATE(14)=SOURCE	SFLO1530
	FLCW(8)=0.5D0*(QL+QLP)*DELT	SFLO1535
	FLCW(9)=0.5D0*(QDH+QDHP)*DELT	SFLO1540
	FLCW(10)=0.5D0*(QLKW+CLKWP)*DELT	SFLO1545
	FLCW(11)=0.5D0*(QLKS+QLKSP)*DELT	SFLO1550
	FLCW(12)=0.5D0*(HULEAK+HULEKP)*DELT	SFLO1555
	FLCW(13)=0.5D0*(HLLEAK+HLLEKP)*DELT	SFLO1560
	FLCW(14)=0.5D0*(SOURCE+SOURSP)*DELT	SFLO1565
C	DO 890 I=1,14	SFLO1570
	890 TFLOW(I)=TFLOW(I)+FLOW(I)	SFLO1575
		SFLO1580
		SFLO1585
		SFLO1590
		SFLO1595
		SFLO1600
		SFLO1605
		SFLO1610
		SFLO1615
		SFLO1620

C	RETURN	SFLO1625
	END	SFLO1630
		SFLO1635
	SUBROUTINE Q4R (QRH, QDE, QDHH, XQ, ZQ, RQ, CSQ, DHQ, HMBQ, POR, RHOB, 1 UPLEKH, DNLEKH, SOSH, HUMHQ, HLMHQ, CUQP, CLQP, SOSQP)	Q4R 005
C		Q4R 010
C		Q4R 015
C	FUNCTION OF SUBROUTINE--TO EVALUATE THE CONCENTRATION INTEGRAL	Q4R 020
C	OVER THE AREA OF ONE ELEMENT.	Q4R 025
C		Q4R 030
C		Q4R 035
	IMPLICIT REAL*8 (A-H,O-Z)	Q4R 040
	REAL*8 N(4)	Q4R 045
C		Q4R 050
	DIMENSION XQ(4), ZQ(4), RQ(4), DHQ(4), HMBQ(4), HUMHQ(4), HLMHQ(4)	Q4R 055
	DIMENSION CSQ(4)	Q4R 060
	DIMENSION S(4), T(4)	Q4R 065
	DIMENSION W(4)	Q4R 070
	DIMENSION PJAB(2,2), DNSS(4), DNTT(4), DWSS(4), DWTI(4)	Q4R 075
C		Q4R 080
	DATA D / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00, -	Q4R 085
	> 1.0, 00 /, T / -1.0D+00, -1.0D+00, 1.0D+00, 1.0D+00 /	Q4R 090
C		Q4R 095
C	EVALUATE QUANTITIES FOR USE IN THE JACOBIAN DJAC, BELOW, NECESSARY	Q4R 100
C	FOR TRANSFORMATION FROM GLOBAL TO LOCAL COORDINATES	Q4R 105
C		Q4R 110
	QRH=0.	Q4R 115
	QDE=0.0	Q4R 120
	QDHH=0.0	Q4R 125
	UPLKX=0.0	Q4R 130
	DNLEKH=0.0	Q4R 135
	SOSH=0.0	Q4R 140
C		Q4R 145
	DO 49C KG=1,4	Q4R 150
C		Q4R 155
C	***** DETERMINE LOCAL COORDINATES (SS,TT) OF GAUSSIAN POINTS KG	Q4R 160
C		Q4R 165
	SS=P*S(KG)	Q4R 170
	TT=P*T(KG)	Q4R 175
C		Q4R 180
C	***** EVALUATE THE JACOBIAN	Q4R 185
C		Q4R 190
	CALL SHAPE(N,W,DNSS,DNTT,DWSS,DWTI,SS,TT)	Q4R 195
C		Q4R 200
	DO 110 I=1,2	Q4R 205
	DO 110 J=1,2	Q4R 210
	PJAB(I,J)=0.0	Q4R 215
110	CONTINUE	Q4R 220
C		Q4R 225
	DO 120 I=1,4	Q4R 230
	PJAB(1,1)=PJAB(1,1) + ZQ(I)*DNTT(I)	Q4R 235
	PJAB(1,2)=PJAB(1,2) - ZQ(I)*DNSS(I)	Q4R 240
	PJAB(2,1)=PJAB(2,1) - XQ(I)*DNTT(I)	Q4R 245
	PJAB(2,2)=PJAB(2,2) + XQ(I)*DNSS(I)	Q4R 250
120	CONTINUE	Q4R 255
C		Q4R 260
	DJAC=PJAB(2,2)*PJAB(1,1) - PJAB(1,2)*PJAB(2,1)	Q4R 265
C		Q4R 270
C	***** INTERPOLATE TO OBTAIN THE CONCENTRATION ROP AND DHQP AT	Q4R 275
C	***** THE GAUSSIAN POINT KG	Q4R 280
C		Q4R 285
	RQE=0.0	Q4R 290
		Q4R 295
		Q4R 300

```

DHQP=0.0
HNEQP=0.0
HJHHQP=0.0
HLHHQP=0.0
CSQP=0.0
SOSQK=0.0
DO 210 IQ=1,4
RQF=RQP+RQ(IQ)*N(IQ)
HNEQP=HNEQP+HNEQ(IQ)*N(IQ)
HJHHQP=HJHHQP+HJHHQ(IQ)*N(IQ)
HLHHQP=HLHHQP+HLHHQ(IQ)*N(IQ)
DHQP=DHQP+DHQ(IQ)*N(IQ)
CSQP=CSQP+CSQ(IQ)*N(IQ)
IF (SOSQP.LT.0.0) SOSQK=SOSQK+SOSQP*RQ(IQ)*N(IQ)
210 CONTINUE
IF (SCSQP.GE.0.0) SOSQK=SOSQP
C
C ***** ACCUMULATE THE SUM TO EVALUATE THE INTEGRAL QRH,QDH,QDHH,SOSH
C
QRH=QRH+POR*RQP*DJAC+HNEQP
QDE=QDH+CSQP*DJAC+HNEQP
QDHH=QDHH + (POR*RQP+RHOB*CSQP)*HNEQP*DJAC+HNEQP
C
GANAU=1.0D0
IF (HJHHQP.LT.0.0) GANAU=-1.0D0
GAFAI=1.0D0
IF (HLHHQP.LT.0.0) GAHAL=-1.0D0
UPLKHH=UPLKHH-0.5D0*((1.0D0+GANAU)*CUQP+(1.0D0-GANAU)*RQP)*
1 HJHHQP*DJAC
DNLEKH=DNLEKH-0.5D0*((1.0D0+GAHAL)*CLQP+(1.0D0-GAHAL)*RQP)*
1 HLHHQP*DJAC
SOSH=SOSH-SOSQK*DJAC
C
490 CONTINUE
C
CC
RETURN
END

SUBROUTINE C3R (QRH,QDH,QDHH,XQ,ZQ,RQ,CSQ,DHQ,HNEQ,POR,RHOB,
1 UPLEKH,DNLEKH,SOSH,HJHHQ,HLHHQ,CUQP,CLQP,SOSQP)
C
C FUNCTION OF SUBROUTINE--TO EVALUATE THE CONCENTRATION INTEGRAL
C OVER THE AREA OF ONE ELEMENT.
C
IMPLICIT REAL*8 (A-H,O-Z)
C
DIMENSION XQ(4),ZQ(4),RQ(4),DHQ(4),HNEQ(4),HJHHQ(4),HLHHQ(4)
DIMENSION CSQ(4)
DIMENSION A(3)
DIMENSION F1Q(3),F2Q(3),F3Q(3),F4Q(3),F5Q(3),F6Q(3)
DIMENSION F7Q(3),F8Q(3),F9Q(3),FAQ(3)
C
DATA OSIX/0.16666666666666667D0/
DATA OTWLV/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

```

```

Q4R 305
Q4R 310
Q4R 315
Q4R 320
Q4R 325
Q4R 330
Q4R 335
Q4R 340
Q4R 345
Q4R 350
Q4R 355
Q4R 360
Q4R 365
Q4R 370
Q4R 375
Q4R 380
Q4R 385
Q4R 390
Q4R 395
Q4R 400
Q4R 405
Q4R 410
Q4R 415
Q4R 420
Q4R 425
Q4R 430
Q4R 435
Q4R 440
Q4R 445
Q4R 450
Q4R 455
Q4R 460
Q4R 465
Q4R 470
Q4R 475
Q4R 480
Q4R 485
Q4R 490
Q3R 005
Q3R 010
Q3R 015
Q3R 020
Q3R 025
Q3R 030
Q3R 035
Q3R 040
Q3R 045
Q3R 050
Q3R 055
Q3R 060
Q3R 065
Q3R 070
Q3R 075
Q3R 080
Q3R 085
Q3R 090
Q3R 095
Q3R 100
Q3R 105
Q3R 110
Q3R 115
Q3R 120
Q3R 125
Q3R 130

```

C	AREA2=DABS (A (1)+A (2)+A (3))	Q3R	135
	AREA=AREA2/2.000	Q3R	140
C	DO 190 IQ=1,3	Q3R	145
	F1C(IQ)=FOR*HMEG(IQ)	Q3R	150
	F2C(IQ)=RQ(IQ)	Q3R	155
	F3C(IQ)=HMBQ(IQ)	Q3R	160
	F4C(IQ)=CSQ(IQ)	Q3R	165
	F5C(IQ)=HMBQ(IQ)*DHQ(IQ)	Q3R	170
	F6C(IQ)=FOR*RQ(IQ)+RROP*CSQ(IQ)	Q3R	175
C	F7C(IQ)=HOMHQ(IQ)	Q3R	180
	IF (HOMHQ(IQ).GT.0.0) F8Q(IQ)=COPQ	Q3R	185
	IF (HOMHQ(IQ).LE.0.0) F8Q(IQ)=RQ(IQ)	Q3R	190
	F9C(IQ)=HLMHQ(IQ)	Q3R	195
	IF (HLMHQ(IQ).GT.0.0) FAQ(IQ)=CLQP	Q3R	200
	IF (HLMHQ(IQ).LE.0.0) FAQ(IQ)=RQ(IQ)	Q3R	205
	190 CONTINUE	Q3R	210
C		Q3R	215
C	QRM=0.0	Q3R	220
	QDM=0.000	Q3R	225
	QDH=0.0	Q3R	230
	UPLKRN=0.0	Q3R	235
	DNLEKN=0.0	Q3R	240
	DO 290 IQ=1,3	Q3R	245
C		Q3R	250
	GO TO (230,250,270),IQ	Q3R	255
C		Q3R	260
230	I=1	Q3R	265
	J=2	Q3R	270
	K=3	Q3R	275
	GO TO 280	Q3R	280
C		Q3R	285
250	I=2	Q3R	290
	J=3	Q3R	295
	K=1	Q3R	300
	GO TO 280	Q3R	305
C		Q3R	310
270	I=3	Q3R	315
	J=1	Q3R	320
	K=2	Q3R	325
C		Q3R	330
280	QRM=QRM + AREA*(F1Q(I)*OSIX+OTNLVE*(F1Q(J)+F1Q(K)))*F2Q(I)	Q3R	335
	QDM=QDM + AREA*(F3Q(I)*OSIX+OTNLVE*(F3Q(J)+F3Q(K)))*F4Q(I)	Q3R	340
	QDH=QDH + AREA*(F5Q(I)*OSIX+OTNLVE*(F5Q(J)+F5Q(K)))*F6Q(I)	Q3R	345
	UPLKRN=UPLKRN - AREA*(F7Q(I)*OSIX+OTNLVE*(F7Q(J)+F7Q(K)))*F8Q(I)	Q3R	350
	DNLEKN=DNLEKN - AREA*(F9Q(I)*OSIX+OTNLVE*(F9Q(J)+F9Q(K)))*FAQ(I)	Q3R	355
C		Q3R	360
290	CONTINUE	Q3R	365
C		Q3R	370
	IF (SCSQP.GE.0.0) SOSM=-SOSQP*AREA	Q3R	375
	IF (SOSQP.LT.0.0) SOSM=-SOSQP*AREA*(RQ(1)+RQ(2)+RQ(3))/3.000	Q3R	380
C		Q3R	385
	RETURN	Q3R	390
	END	Q3R	395
		Q3R	400
		Q3R	405
		Q3R	410
		Q3R	415
	SUBROUTINE PRINTT(R,FX,FZ, PRATE,FLOW,TFLOW, TIME,DELT,KPR,KOUT,	PRIN	005
	> KCIAG,NAINP,NNP,IBAND, ITIM)	PRIN	010
C		PRIN	015
C		PRIN	020
C	FUNCTION OF SUBROUTINE--TO OUTPUT FLOWS, CONCENTRATIONS, MATERIAL	PRIN	025
C	FLUXES, WATER CONTENTS, DARCY VELOCITIES, PRESSURE HEADS, AND	PRIN	030
C	TOTAL HEADS AS SPECIFIED BY THE PARAMETER KPR.	PRIN	035

```

C
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      DIMENSION S(MAXNP),FX(MAXNP),FZ(MAXNP)
C      DIMENSION PRATE(14),FLOW(14),TFLOW(14)
C
C      IF (KPR.EQ.0) RETURN
C
C ***** PRINT FLOW SYSTEM PARAMETERS AND MASS BALANCE
C
C      KDIAG=KDIAG+1
C      PRINT 1000, KDIAG,TIME,DELT,ITIM
C      PRINT 1100, (PRATE(I),FLOW(I),TFLOW(I),I=1,14)
C      IF (KPR.EQ.1) RETURN
C
C ***** PRINT CONCENTRATION
C
C      KOUT=KOUT+1
C      KLINE=-1
C      DO 290 NI=1,NNP,8
C      NJHN=NI
C      NJHX=MIN0(NI+7,NNP)
C      KLINE=KLINE+1
C      IF (MOD(KLINE,50).EQ.0) PRINT 2000, KOUT,TIME,DELT,IBAND,ITIM
C      PRINT 2100, NI, (R(NJ),NJ=NJHN,NJHX)
C 290 CONTINUE
C      IF (KPR.EQ.2) RETURN
C
C ***** PRINT VERTICALLY INTEGRATED FLUX
C
C      KOUT=KOUT+1
C      KLINE=-1
C      DO 390 NI=1,NNP,4
C      NJHN=NI
C      NJHX=MIN0(NI+3,NNP)
C      KLINE=KLINE+1
C      IF (MOD(KLINE,50).EQ.0) PRINT 3000, KOUT,TIME,DELT,IBAND,ITIM
C      PRINT 3100, (NJ,FX(NJ),FZ(NJ),NJ=NJHN,NJHX)
C 390 CONTINUE
C
C      RETURN
C
C 1000 FORMAT(1X,52H*****
> 62H*****1*****
> 5H****//18H SYSTEM-FLOW TABLE,14,12H.. AT TIME =,1PD12.4,
> 9H ,(DELT =,1PD12.4,1H),6H ITIM=,110)
1100 FORMAT(//5X,13H TYPE OF FLOW,35X,9H RATE (M/T),6X,12H INC. FLOW (M),
> 7X,13HTOTAL FLOW (M)/5X,
1 4CH 1. DIRICHLET NODE FLOW . . . . . ,3(E12.4,8X)/5X,
2 4CH 2. OPEN BOUNDARY NODE FLOW IN . . . . ,3(E12.4,8X)/5X,
3 4CH 3. OPEN BOUNDARY NODE FLOW OUT . . . . ,3(E12.4,8X)/5X,
4 4CH 4. THROUGH UNSPECIFIED NODES . . . . ,3(E12.4,8X)/5X,
5 4CH 5. NET FLOW THROUGH ENTIRE BOUNDARY . ,3(E12.4,8X)/5X,
6 4CH 6. INCREASE IN MATERIAL (LIQUID) . . . ,3(E12.4,8X)/5X,
7 4CH 7. INCREASE IN MATERIAL (SOLID) . . . ,3(E12.4,8X)/5X,
8 4CH 8. RADIOACTIVE DECAY (LIQUID & SOLID) ,3(E12.4,8X)/5X,
9 4CH 9. LOSS TO COMP. OF SKELTON . . . . . ,3(E12.4,8X)/5X,
A 4CH 10. LOSS THROUGH DISSOLVED PHASE . . . ,3(E12.4,8X)/5X,
B 4CH 11. LOSS THROUGH ADSORBED PHASE . . . ,3(E12.4,8X)/5X,
C 4CH 12. LEAK THROUGH UPPER AQUIFERS . . . ,3(E12.4,8X)/5X,
D 4CH 13. LEAK THROUGH LOWER AQUIFERS . . . ,3(E12.4,8X)/5X,
E 4CH 14. RECHARGE (-), PUMPING (+) . . . . ,3(E12.4,8X)//5X
F 50H *** NOTE: (+) = OUT FROM, (-) INTO THE REGION . ,//)

```

```

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
PRIN 305
PRIN 310
PRIN 315
PRIN 320
PRIN 325
PRIN 330
PRIN 335
PRIN 340
PRIN 345
PRIN 350
PRIN 355

```

```

2000 FORMAT(13H OUTPUT TABLE, I4, 34H. CONCENTRATIONS (M/L**3) AT TIME =,
> 1PD12.4, 9H, (DELT =, 1PD12.4, 15H), (BAND WIDTH =, I4, 1H) // 5X,
> 12H * * ITIME =, I10 //
> 7H NODE I, 5X, 44H CONCENTRATION (M/L**3) AT NODES I, I+1, ..., I+7/)
2100 FORMAT(17, 8(1PD15.4))
3000 FORMAT(13H OUTPUT TABLE, I4, 46H. VERTICALLY INTEGRATED FLUX (M/L/T)
> AT TIME =, 1PD12.4, 9H, (DELT =, 1PD12.4, 15H), (BAND WIDTH =, I4, 1H)
> // 5X, 12H * * ITIME =, I10 // 1X,
> 4H NODE, 9X, 2HPX, 9X, 2HPZ, 6X, 4H NODE, 9X, 2HPX, 9X, 2HPZ, 6X, 4H NODE, 9X,
> 2HPX, 9X, 2HPZ, 6X, 4H NODE, 9X, 2HPX, 9X, 2HPZ /)
3100 FORMAT(4(I5, 2E11.3 5X))
C
      ZNL
                                PRIN 360
                                PRIN 365
                                PRIN 370
                                PRIN 375
                                PRIN 380
                                PRIN 385
                                PRIN 390
                                PRIN 395
                                PRIN 400
                                PRIN 405
                                PRIN 410
                                PRIN 415
                                PRIN 420

      SUBROUTINE STORE(X,Z,IE,R,FX,PZ,TITLE,NPROB,NNP,NEL,NTI,
> MAXNP,MAXEL,TIME)
                                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

C
C
C FUNCTION OF SUBROUTINE--TO STORE PERTINENT QUANTITIES ON AUXILIARY
C DEVICE FOR FUTURE USE, E.G. FOR PLOTTING. WHAT DEVICE IS TO BE
C USED MUST BE SPECIFIED BY APPROPRIATE JOB-CONTROL CARDS.
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION TITLE(9)
      DIMENSION X(MAXNP), Z(MAXNP), IE(MAXEL,5), R(MAXNP), FX(MAXNP),
> PZ(MAXNP)
                                STOR 135
                                STOR 140
                                STOR 145
                                STOR 150
                                STOR 155
                                STOR 160
                                STOR 165
                                STOR 170
                                STOR 175
                                STOR 180
                                STOR 185
                                STOR 190
                                STOR 195
                                STOR 200
                                STOR 205
                                STOR 210
                                STOR 215
                                STOR 220
                                STOR 225
                                STOR 230
                                STOR 235
                                STOR 240
                                STOR 245
                                STOR 250
                                STOR 255
                                STOR 260
                                STOR 265
                                STOR 270
                                STOR 275
                                STOR 280
                                STOR 285
                                STOR 290
                                STOR 295
                                STOR 300
                                STOR 305
                                STOR 310
                                STOR 315
                                STOR 320
                                STOR 325
                                STOR 330
                                STOR 335
                                STOR 340
                                STOR 345
                                STOR 350
                                STOR 355
                                STOR 360
                                STOR 365
                                STOR 370
                                STOR 375
                                STOR 380
                                STOR 385
                                STOR 390
                                STOR 395
                                STOR 400
                                STOR 405
                                STOR 410
                                STOR 415
                                STOR 420
                                STOR 425
                                STOR 430
                                STOR 435
                                STOR 440
                                STOR 445
                                STOR 450
                                STOR 455
                                STOR 460
                                STOR 465
                                STOR 470
                                STOR 475
                                STOR 480
                                STOR 485
                                STOR 490
                                STOR 495
                                STOR 500
                                STOR 505
                                STOR 510
                                STOR 515
                                STOR 520
                                STOR 525
                                STOR 530
                                STOR 535
                                STOR 540
                                STOR 545
                                STOR 550
                                STOR 555
                                STOR 560
                                STOR 565
                                STOR 570
                                STOR 575
                                STOR 580
                                STOR 585
                                STOR 590
                                STOR 595
                                STOR 600
                                STOR 605
                                STOR 610
                                STOR 615
                                STOR 620
                                STOR 625
                                STOR 630
                                STOR 635
                                STOR 640
                                STOR 645
                                STOR 650
                                STOR 655
                                STOR 660
                                STOR 665
                                STOR 670
                                STOR 675
                                STOR 680
                                STOR 685
                                STOR 690
                                STOR 695
                                STOR 700
                                STOR 705
                                STOR 710
                                STOR 715
                                STOR 720
                                STOR 725
                                STOR 730
                                STOR 735
                                STOR 740
                                STOR 745
                                STOR 750
                                STOR 755
                                STOR 760
                                STOR 765
                                STOR 770
                                STOR 775
                                STOR 780
                                STOR 785
                                STOR 790
                                STOR 795
                                STOR 800
                                STOR 805
                                STOR 810
                                STOR 815
                                STOR 820
                                STOR 825
                                STOR 830
                                STOR 835
                                STOR 840
                                STOR 845
                                STOR 850
                                STOR 855
                                STOR 860
                                STOR 865
                                STOR 870
                                STOR 875
                                STOR 880
                                STOR 885
                                STOR 890
                                STOR 895
                                STOR 900
                                STOR 905
                                STOR 910
                                STOR 915
                                STOR 920
                                STOR 925
                                STOR 930
                                STOR 935
                                STOR 940
                                STOR 945
                                STOR 950
                                STOR 955
                                STOR 960
                                STOR 965
                                STOR 970
                                STOR 975
                                STOR 980
                                STOR 985
                                STOR 990
                                STOR 995

      DATA NPPROB / -1 /
C
      IF (NPPROB.EQ.(-1)) REWIND 2
      IF (NPPROB.EQ.NPROB) GO TO 10
      WRITE(2) (TITLE(I), I=1,9), NPROB, NNP, NEL, NTI
      WRITE(2) (X(NP), NP=1, NNP), (Z(NP), NP=1, NNP), ((IE(M, IQ), M=1, NEL),
> IC=1, 5)
      NPERCB=NPROB
      10 WRITE(2) TIME, (R(NP), NP=1, NNP), (FX(NP), NP=1, NNP), (PZ(NP), NP=1, NNP)
      RETURN
      ENL

      SUBROUTINE READR(F, MAXMOD, NNP)
                                REDR 005
                                REDR 010
                                REDR 015
                                REDR 020
                                REDR 025
                                REDR 030
                                REDR 035
                                REDR 040
                                REDR 045
                                REDR 050
                                REDR 055
                                REDR 060
                                REDR 065
                                REDR 070
                                REDR 075
                                REDR 080
                                REDR 085
                                REDR 090
                                REDR 095
                                REDR 100
                                REDR 105
                                REDR 110
                                REDR 115

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION F(MAXMOD)
      NOIES=0
      150 READ 40, NI, NSEQ, NAD, FNI, PAD, PRD
      IF (NI.EQ.0) GO TO 170
      NJ=NI+NSEQ
      DO 160 M=NI, NJ
          NODES=NODES+1
          I=NI+(M-NI)*NAD
          IF (PRD.NE.0.0) GO TO 155
          F(I)=FNI+PAD*DFLOAT(M-NI)
          GO TO 160
      155 I1=I-NAD
          IF (M.EQ.NI) F(I)=FNI
          IF (M.EQ.NI) DINC=1.0D0
          IF (M.GT.NI) DINC=DINC*(1.0D0+PRD)
          IF (M.GT.NI) F(I)=F(I1)+PAD*DINC
      160 CONTINUE
          GO TO 150
      170 IF (NODES.EQ.NNP) GO TO 180
          PRINT 1100
          STOP

```

180 IF (NRP.LE.NAXNOD) GO TO 190	REDR 120
PRINT 1200	REDR 125
STCP	REDR 130
190 CONTINUE	REDR 135
C 40 ZOSMAT(3I5,5X,4D10.3)	REDR 140
1100 FOSMAT(1H1/1H,64H *** ERROR IN EXECUTING READR SINCE NODES .NE.	REDR 145
> NRP: STOP ***)	REDR 150
1200 FOSMAT(1H1/1H,46H *** NRP .GT. NAXNOD IN EXECUTING READR: STOP)	RLJR 155
C RETURN	REDR 160
END	REDR 165
	REDR 170
	REDR 175
SUBROUTINE READN(INDTYP,NXTYP,NTYPE)	REDN 005
IMPLICIT REAL*8(A-H,O-Z)	REDN 010
DIMENSION INDTYP(NXTYP)	REDN 015
C NTYPES=0	REDN 020
110 READ 30,N1,NSEQ,NAD,NITYP,NTYPAD	REDN 025
IF (N1.EQ.0) GO TO 130	REDN 030
NJ=N1+NSEQ	REDN 035
DO 120 N=N1,NJ	REDN 040
I=N1+(N-N1)*NAD	REDN 045
INDTYP(I)=NITYPE + NTYPAL*(N-N1)	REDN 050
NTYPES=NTYPES+1	REDN 055
120 CONTINUE	REDN 060
GO TO 110	REDN 065
130 IF (NTYPES.EQ.NTYPE) GO TO 140	REDN 070
PRINT 1100	REDN 075
STCP	REDN 080
140 IF (NTYPE.LE.NXTYP) GO TO 150	REDN 085
PRINT 1200	REDN 090
STCP	REDN 095
150 CONTINUE	REDN 100
C 30 FOSMAT(5I5)	REDN 105
1100 FOSMAT(1H1/1H,64H *** ERROR IN EXECUTING READN SINCE NTYPES .NE.	REDN 110
> NTYPE: STOP ***)	REDN 115
1200 FOSMAT(1H1/1H,46H *** NTYPE .GT. NXTYP IN EXECUTING READN: STOP)	REDN 120
C RETURN	REDN 125
END	REDN 130
	REDN 135
	REDN 140
	REDN 145
SUBROUTINE INTERP(FALL,TRF,RF,TIME,MXPR,MXDP,NPR,NDP)	INTE 005
IMPLICIT REAL*8(A-H,O-Z)	INTE 010
DIMENSION FALL(MXPE),TRF(MXPR,MXDP),RF(MXPR,MXDP)	INTE 015
C DO 160 I=1,NPR	INTE 020
DO 140 J=2,NDP	INTE 025
IF (TRF(I,J-1).LE.TIME .AND. TIME.LE.TRF(I,J)) GO TO 120	INTE 030
GO TO 140	INTE 035
120 FALL(I)=RF(I,J-1) + (TIME-TRF(I,J-1)) * (RF(I,J)-RF(I,J-1)) /	INTE 040
> (TRF(I,J)-TRF(I,J-1))	INTE 045
GO TO 160	INTE 050
140 CONTINUE	INTE 055
FALL(I)=0.0	INTE 060
160 CONTINUE	INTE 065
C RETURN	INTE 070
END	INTE 075
	INTE 080
	INTE 085

INTERNAL DISTRIBUTION

1. T. L. Ashwood	19. T. H. Row
2. S. I. Auerbach	20. E. D. Smith
3. W. J. Boegly, Jr.	21. B. P. Spalding
4. R. O. Chester	22. S. H. Stow
5. R. B. Clapp	23-32. L. E. Stratton
6. K. E. Cowser	33. D. M. Walls
7. N. H. Cutshall	34-58. G. T. Yeh
8. E. C. Davis	59. Central Research Library
9. C. S. Haase	60-84. ESD Library
10. S. G. Hildebrand	85-86. Laboratory Records Dept.
11-15. D. D. Huff	87. Laboratory Records, ORNL-RC
16. I. L. Larsen	88. ORNL Patent Section
17. R. S. Lowrie	89. ORNL Y-12 Technical Library
18. T. W. Oakes	

EXTERNAL DISTRIBUTION

90. M. U. Ahmad, Department of Geology, Ohio University, Athens, OH 45701
91. J. R. Albaugh, Program Manager, Solid Waste Processing and Disposal, Rockwell Hanford Operations, P.O. Box 800, Richland, WA 99352
92. E. L. Albenesius, Savannah River Laboratory, P.O. Box A, Aiken, SC 29801
93. M. Barainca, Program Manager, Low-Level Waste Management Program, U.S. Department of Energy, 550 Second Street, Idaho Falls, ID 83401
94. M. Bergeron, USGS, Water Resources Division, 521 West Seneca Street, Ithaca, NY 14850
95. Eshel Bresler, Soil Physicist, Agricultural Research Organization, The Volcanic Center, Institute of Soil and Water, P.O. Box 6, Bet Dagan, 50-250, Israel
96. T. W. Broyd, Atkins Research and Development, Woodcote Grove Ashley Rd., Epsom Surrey KT18 5BW England
97. W. Brutsaert, School of Civil Engineering, Cornell University, Ithaca, NY 14850
98. J. Thomas Callahan, Associate Director, Ecosystem Studies Program, Room 336, 1800 G Street, NW, National Science Foundation, Washington, DC 20550
99. S. T. Chau, Albert Environmental Protection Service, 14th Floor, Standard Life Center, 10405 Jasper Ave., Edmonton, Alberta, Canada
100. T. C. Chee, R&D and Byproducts Division, DP-123 (GTN), U.S. Department of Energy, Washington, DC 20545
101. S. W. Childs, Department of Soil Science, Oregon State University, Corvallis, OR, 97331
102. C. G. Clyde, Utah Water Resources Laboratory, Utah State University, Logan, UT 84322

103. Peter Colombo, Group Leader, Nuclear Waste Research, Brookhaven National Laboratory, Bldg. 701, Upton, NY 11973
104. E. F. Conti, Office of Nuclear Regulatory Research, Nuclear Regulatory Commission, MS-1130-SS, Washington, DC 20555
105. D. Cooper, School of Public Health, Harvard University, 665 Huntington Ave., Boston, 02115
106. D. G. de Marsily, Ecole Nationale, Supérieure des Mines, Centre d'Informatique Géologique, 35 Rue Saint-Honore, 77305 Fontainebleau, France
107. G. DePoorter, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545
108. A. K. Deb, Roy F. Weston, Inc., Weston Way, West Chester, PA 19380
109. John L. Deichman, Program Director, Waste Management Program Office, Rockwell Hanford Operations, P.O. Box 800, Richland, WA 99352
110. J. E. Dieckhoner, Acting Director, Operations and Traffic Division, DP-122 (GTN), U.S. Department of Energy, Washington, DC 20545
111. G. J. Foley, Office of Environmental Process and Effects Research, U.S. Environmental Protection Agency, 401 M Street, SW, RD-682, Washington, DC 20460
112. E. Fray, EG&G Idaho, P.O. Box 1625, Idaho Falls, ID 83401
113. R. A. Freeze, Dept. of Geological Sciences, University of British Columbia, Vancouver, B.C., Canada
114. W. J. Gburek, U.S. Department of Agriculture, Northwest Watershed Research Center, 110 Research Building A, University Park, PA 16802
115. Lynn W. Gelhar, Department of Geosciences, New Mexico Tech., Socorro, NM 87801
116. Carl Gertz, Chief, Radioactive Waste Technology Branch, Idaho Operations Office, U.S. Department of Energy, 550 Second Street, Idaho Falls, ID 83401
117. Robert W. Gillham, Department of Earth Sciences, University of Waterloo, Waterloo, Ontario, Canada 21361
118. Stephen Gorelick, USGS, Water Resources Division, MS-21, 345 Middlefield Road, Menlo Park, CA 94205
119. A. Berge Gureghian, Project Management Division, Performance Analysis Department, Battelle Columbus, 505 King Avenue, Columbus, OH 43201
120. David Hamilton, Water Management Division, P.O. Box 30028, Lansing, MI 48909
121. L. A. Hermann, Dept. of Geosciences, Louisiana Tech. University, Ruston, LA 71272
122. Beverly Herzog, Illinois State Geological Survey, Natural Resources Bldg., 615 E. Peabody Drive, Champaign, IL 61820
123. Richard Heystee, Ontario Hydro, 700 University Avenue, Toronto, Ontario M5G 1X6 Canada
124. J. A. Hoopes, 1261B Engineering Bldg., Department of Civil and Environmental Engineering, University of Wisconsin-Madison, Madison, WI 53706
125. J. Horyna, Department of Radiation Protection, Nuclear Research Institute, 250-68 Rez., Prague, Czechoslovakia
126. J. W. Huckabee, Project Manager, Environmental Assessment Department, Electric Power Research Institute, 3412 Hillview Avenue, P.O. Box 10412, Palo Alto, CA 94303
127. J. C. Hwang, Department of Civil Engineering, Drexel University, Philadelphia, PA 19104

128. E. A. Jennrich, Program Manager, Low-Level Waste Management Program, EG&G Idaho, Inc., P.O. Box 1625, Idaho Falls, ID 83415
129. J. J. Jicha, Director, R&D and Byproducts Division, DP-123 (GTN), U.S. Department of Energy, Washington, DC 20545
130. R. Johnson, Department of Environmental Sciences, Oregon Graduate Center, 19600 NW Walker Road, Beaverton, OR 97006
131. A. Jonasson, Department of Geology, Chalmers University of Technology, S-412, 96 Gothenburg, Sweden
132. E. A. Jordan, Low Level Waste Program Manager, Division of Storage and Treatment Projects, NE-25 (GTN), U.S. Department of Energy, Washington, DC 20545
133. George Y. Jordy, Director, Office of Program Analysis, Office of Energy Research, ER-30, G-226, U.S. Department of Energy, Washington, DC 20545
134. W. A. Jury, Soils Department, University of California, Riverside, CA 92501
135. D. Kent, Whitehurst Hall, Oklahoma University, Stillwater, OK 74078
136. J. Howard Kittel, Manager, Office of Waste Management Programs, Argonne National Laboratory, 9700 S. Cass Ave., Bldg. 205, Argonne, IL 60439
137. M. R. Kreiter, Waste Isolation, Pacific Northwest Laboratories, Richland, WA 99352
138. J. T. Kuo, Dept. of Civil Engineering, National Taiwan University, Taipei, Taiwan
139. Leonard Lane, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545
140. R. T. Lawson, Environmental Sciences Division, Australian Atomic Energy Commission, New Illanarra Rd., Lucas Heights, N.S.W., Australia
141. Denis R. LeBlanc, USGS, Water Resources Division, 150 Causeway Street, Suite 1001, Boston, MA 02114
142. D. B. Leclaire, Director, Office of Defense Waste and Byproducts Management, DP-12 (GTN), U.S. Department of Energy, Washington, DC 20545
143. Timothy C. Lederman, Radiological Sciences Institute, State of New York, Dept. of Health, The Governor Nelson A. Rockefeller Empire State Plaza, Albany, NY 12201
144. Joan Lesler, Environmental Review Division, Office of Testing and Evaluation, Office of Toxic Substances, Environmental Protection Agency, 401 M Street, SW, Washington, DC 20460
145. John J. Lentz, Applied Physics Laboratory, The Johns Hopkins University, Johns Hopkins Rd., Laurel, MD 20810
146. Y. C. Lu, Dept. of Civil Engineering, National Taiwan University, Taipei, Taiwan
147. A. Marples, Safety and Medical Services, British Nuclear Fuels Limited, Sellafield Seascale, Cumbria CA20 1PG, England
148. 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
149. Daene C. McKinney, Department of Environmental Resources Engineering, Hornbonds State University, Arcata, CA 95521
150. G. L. Meyer, Environmental Protection Agency, 401 M Street SW, MS-ANR459, Washington, DC 20460
151. Fred H. Molz, Director, School of Engineering and Engineering Experiment Station, Auburn University, Auburn, AL 36849

152. Harold A. Mooney, Department of Biological Sciences, Stanford University, Stanford, CA 94305
153. Jerry D. Nasson, NEBG Installation Representative, Mail Code 311, General Electric Company, 175 Curtner Ave., San Jose, CA 95125
154. Donald T. Oakley, Program Manager for Waste Management, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545
155. Edward O'Donnell, Division of Radiation Programs and Earth Sciences, U.S. Nuclear Regulatory Commission, MS 1130 SS, Washington, DC 20555
156. G. K. Oertel, Director, Defense Waste and By-Products Division, Department of Energy, DP-121, MS-B107, Headquarters, Washington, DC 20545
157. Peter M. Ornstein, MS-697, U.S. Nuclear Regulatory Commission, 7915 Eastern Ave., Silver Spring, MD 20910
158. C. A. Oster, Energy Systems Department, Pacific Northwest Laboratories, Richland, WA 99352
159. Frank Parker, Dept. of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN 37235
160. G. F. Pinder, Dept. of Civil Engineering, Princeton University, Princeton, NJ 08544
161. David Prudic, USGS, Water Resources Division, Room 229, Federal Building, Carson City, NV 89701
162. Ellen J. Quinn, Northeast Utilities, P.O. Box 270, Hartford, CT 06141
163. P.A.C. Raats, Institute for Soil Fertility, P.O. Box 30003, 9750 Ra Haren (Gr), Netherlands
164. Irwin Remson, Department of Applied Earth Sciences, Stanford University, Stanford, CA 94305
165. Jackson Robertson, USGS, 410 National Center, Reston, VA 22092
166. J. C. Rodgers, Group LS-6, Environmental Sciences Group, Los Alamos National Laboratory, Los Alamos, NM 87545
167. E. M. Romney, University of California, Los Angeles, 900 Veteran Avenue, Los Angeles, CA 90024
168. R. W. Root, Jr., Savannah River Laboratory, Aiken, SC 29801
169. E. R. Rothschild, CH₂M Hill, 310 W. Wisconsin Ave., Suite 700, P.O. Box 2090, Milwaukee, WI 53201
170. Jerry L. Schnoor, Division of Energy Engineering, The University of Iowa, Iowa City, IA 52242
171. Marios Sophocleous, Kansas Geological Survey, 1930 Avenue A, Campus West, The University of Kansas, Lawrence, KS 66044
172. Ricky H. Stacks, Department of Pollution Control & Ecology, State of Arkansas, 8001 National Drive, P.O. Box 9583, Little Rock, AR 72209
173. R. J. Starmer, HLW Technical Development Branch, Office of Nuclear Material Safety and Safeguards, Nuclear Regulatory Commission, Room 427-SS, Washington, DC 20555
174. J. G. Steger, Environmental Sciences Group, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545
175. Daniel B. Stephens, New Mexico Tech., Department of Geosciences, Socorro, NM 87801
176. R. J. Stern, Director, Office of Environmental Compliance, MS PE-25, FORRESTAL, U.S. Department of Energy, 1000 Independence Avenue, SW, Washington, DC 20585

177. J. A. Stone, Savannah River Laboratory, E. I. DuPont de Nemours and Company, Bldg. 773-A, Room E-112, Aiken, SC 29808
178. Steve Y. Tsai, Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439
179. M. van Genuchten, U.S. Salinity Laboratory, 4500 Gleadow Drive, Riverside, CA 92501
180. Jan Wagner, School of Chemical Engineering, Oklahoma State University, Stillwater, OK 74078
181. D. E. Wessol, Reactor Physics Branch, EG&G Idaho, Inc., P.O. Box 1625, Idaho Falls, ID 83415
182. J. B. Whitsett, Chief, Nuclear Fuel Cycle and Waste Management Branch, Department of Energy, Idaho Operations Office, 550 Second Street, Idaho Falls, ID 83401
183. D. Widmayer, U.S. Nuclear Regulatory Commission, 7915 Eastern Ave., Silver Spring, MD 20920
184. Raymond G. Wilhour, Chief, Air Pollution Effects Branch, Corvallis Environmental Research Laboratory, U.S. Environmental Protection Agency, 200 SW 35th Street, Corvallis, OR 97330
185. P. J. Wierenga, Department of Agronomy, New Mexico State University, P.O. Box 3Q, Las Cruces, NM 88003
186. Frank J. Wobber, Ecological Research Division, Office of Health and Environmental Research, Office of Energy Research, MS-E201, Department of Energy, Washington, DC 20545
187. Gordon M. Wolman, The Johns Hopkins University, Department of Geography and Environmental Engineering, Baltimore, MD 21218
188. Steve Yabusaki, Geophysical Fluid Dynamics Section, Pacific Northwest Laboratories, Richland, WA 99352
189. Feng-quing Zhuang, Institute of Mechanics, Chinese Academy of Sciences, Beijing, Peoples' Republic of China
190. Office of Assistant Manager for Energy Research and Development, Oak Ridge Operations, P.O. Box E, U.S. Department of Energy, Oak Ridge, TN 37831
- 191-217. Technical Information Center, Oak Ridge, TN 37831