ornl

OAK
RIDGE
NATIONAL
LABORATORY

UNION
CARBIDE

ORNL-5601

FEMWASTE: A Finite-Element Model of WASTE Transport Through Saturated- Unsaturated Porous Media

G. T. Yeh
D. S. Ward

ENVIRONMENTAL SCIENCES DIVISION
Publication No. 1462

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A08 Microfiche A01

ORNL-5601
Distribution Category UC-70

Contract No. W-7405-eng-26

FEMWASTE: A FINITE-ELEMENT MODEL OF WASTE TRANSPORT THROUGH SATURATED-UNSATURATED POROUS MEDIA

G. T. Yeh and D. S. Ward*

ENVIRONMENTAL SCIENCES DIVISION
Publication No. 1462

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

*Present Address: INTERA Environmental Consultants, Inc., 11511 Katy Freeway, Suite 630, Houston, TX 77079

Date Published: April 1981

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

ACKNOWLEDGEMENT

The financial support for this study came from the Radionuclide Movement from Burial Grounds and Development of Technology for Land Burial programs of the Department of Energy.

Duguid and Reeves' original work provided the basis and stimulation for this study. Private discussions with T. Tamura on the fundamental properties of soil absorption led to the reformulation of distribution coefficient and retardation factor.

ABSTRACT

YEH, G. T., and D. S. WARD. 1981. FEMWASTE: A finite-element model of waste transport through saturated-unsaturated porous media. ORNL-5601. Oak Ridge National Laboratory, Oak Ridge, Tennessee. 148 pp.

A two-dimensional transient model for the transport of dissolved constituents through porous media originally developed at Oak Ridge National Laboratory (ORNL) has been expanded and modified. Transport mechanisms include: convection, hydrodynamic dispersion, chemical sorption, and first-order decay. Implementation of quadrilateral iso-parametric finite elements, bilinear spatial interpolation, asymmetric weighting functions, several time-marching techniques, and Gaussian elimination are employed in the numerical formulation. A comparative example is included to demonstrate the difference between the new and original models. Results from 12 alternative numerical schemes of the new model are compared. The waste transport model is compatible with the water flow model developed at ORNL for predicting convective Darcy velocities in porous media which may be partially saturated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure		Page

I. INTRODUCTION

The transport of a dissolved constituent by ground water through sorbing porous media is controlled by carrier fluid advection, hydrodynamic dispersion, molecular diffusion, chemical reactions, and sorption on the media and decay. Mathematical equations to describe these phenomena have been formulated for various physical and chemical properties by many investigators. Reeves and Duguid (1975) and Duguid and Reeves (1976) have provided full background information concerning transport equation development and Galerkin finite-element techniques. Emphasis here will be placed upon modification and expansion to their original work.

Motivation for this work was three-fold. First, we believed that modification to the original material transport code (Duguid and Reeves 1976) was necessary for the definition of solute sorption processes by incorporating a moisture-dependent retardation factor. Second, the computation of waste flux requires the application of the finite-element method in such a way that mass balance over the whole region can be preserved. Changes in the way properties for a given element are computed were needed to achieve this objective. Third, in attempts to increase model applicability and improve overall accuracy, several numerical solution schemes were incorporated. Employing asymmetric weighting functions in the spatial discretization (Huyakorn and Nilkuku 1979, Heinrich et al. 1977), the new program overcomes the problem of numerical oscillation associated with advectively dominated flow. This is an essential aspect in the application of the model to field situations in which the scale length of dispersivities is small relative to the spatial discretization (Ergatoudis et al. 1968, Finlayson 1972, Zienkiewicz 1977).

II. MATHEMATICAL STATEMENTS

Except for modifications and expansions (Sections 2, 4, 6, 11, 12, 13, and part of Section 5), the original work (Duguid and Reeves 1976) is followed very closely in the following statements of the problem. However, in the derivation of finite-element approximation, matrix component representation is used rather than the matrix in its entirety. This component representation is believed to be more readily understood.

1. Governing Equations and Initial and Boundary Conditions

The governing equations to describe the distribution of a pollutant constituent in a two-dimensional subsurface porous system is obtained from the law of mass balance. This can be written in the form (Duguid and Reeves 1976):

$$L(c) = \frac{\partial}{\partial t}(\theta c + \rho s) + (\theta c + \rho s)\alpha'\frac{\partial h}{\partial t} + \left(\frac{\partial V_x c}{\partial x} + \frac{\partial V_z c}{\partial z}\right)$$
$$- \left[\frac{\partial}{\partial x}\left(\theta D_{xx}\frac{\partial c}{\partial x} + \theta D_{xz}\frac{\partial c}{\partial z}\right) + \frac{\partial}{\partial z}\left(\theta D_{zx}\frac{\partial c}{\partial x} + \theta D_{zz}\frac{\partial c}{\partial z}\right)\right] \quad (1)$$
$$+ \lambda(\theta c + \rho s) - M \quad ,$$

where θ is the moisture content; c is the concentration of dissolved constituent in the water; ρ is the bulk density of the solid; s is the concentration of the constitutent that is adsorbed on the solid; α' is the modified coefficient of compressibility of the medium; h is pressure head of the water; D_{xx}, D_{xz}, D_{zx}, and D_{zz} are the dispersion coefficient tensor components; V_x and V_z are the Darcian velocity components in the x- and z-directions, respectively; λ is the decay constant; M is the artificial source; x and z are the horizontal and vertical coordinates, respectively; t is the time; and L is an operator.

Equation (1) expresses the mass balance in an initially small bulk volume. The first term represents the rate of change of total mass (including dissolved and adsorbed) in the element volume. The second term is the mass change due to the change of the bulk volume under pressure. The third and fourth terms represent the mass fluxes out and into the volume by advection and dispersion, respectively. The fifth term is the mass change due to decay while the last term is the artificial input or withdrawal. Variables, θ, h, V_x, and V_z, in Eq. (1) can be obtained from the hydrodynamics of subsurface flow system (Reeves and Duguid 1975, Yeh and Ward 1980). The dispersion coefficient tensor may be related to flow field and media properties as (Bear 1972):

$$\theta D_{xx} = a_T V + (a_L - a_T) V_x^2 / V + D_m \tau , \tag{2a}$$

$$\theta D_{xz} = \theta D_{zx} = (a_L - a_T) V_x V_z / V \quad , \tag{2b}$$

and

$$\theta D_{zz} = a_T V + (a_L - a_T) V_z^2 / V + D_m \tau , \tag{2c}$$

where $V = \sqrt{V_x^2 + V_z^2}$; a_T and a_L are the transverse and longitudinal dispersivities, respectively; τ is the tortuosity; and D_m is the molecular diffusion coefficient. The decay constant, λ, is a property of the constituent and ρ and α' are the properties of the porous media under consideration. The independent variables include: x, z, and t. Thus, there are two dependent unknowns, c and s, in

Eq. (1) to be determined. An additional equation is required to completely define the system. It is assumed that the adsorption of the constituent by the solid is to occur at a rapid rate (i.e., a fast exchange reaction) such that the dissolved material is in equilibrium with the material adsorbed by the solid. This is expressed by the linear equation:

$$s = K_d c \quad , \tag{3}$$

where K_d is the distribution coefficient. Substituting Eq. (3) into Eq. (1), one obtains:

$$L(c) = \theta R_d \frac{\partial c}{\partial t} + \left(\frac{\partial V_x c}{\partial x} + \frac{\partial V_z c}{\partial z}\right) - \left[\frac{\partial}{\partial x}\left(\theta D_{xx} \frac{\partial c}{\partial x} + \theta D_{xz} \frac{\partial c}{\partial z}\right)\right.$$
$$\left. + \frac{\partial}{\partial z}\left(\theta D_{zx} \frac{\partial c}{\partial x} + \theta D_{zz} \frac{\partial c}{\partial z}\right)\right] + \left(\frac{\partial \theta}{\partial t} + \alpha' \theta R_d \frac{\partial h}{\partial t} + \lambda \theta R_d\right) c - M = 0 \ , \tag{4}$$

where

$$R_d = 1 + \frac{\rho K_d}{\theta} \tag{5}$$

is the retardation factor, which is a measure of the delay of the breakthrough of the dissolved constituent.

The initial condition of Eq. (4) is assumed to be known as:

$$c = c_0(x,z) \qquad \text{at } t = 0 \text{ and } (x,z) \text{ in } R \ , \tag{6}$$

where c_0 is a given function of spatial coordinates, x and z; R is a region bounded by the curve, B(x,z) = 0, as shown in Fig. 1. This c_0 may also be obtained by simulating the steady version of Eq. (4) with

ORNL-DWG 78-20110R

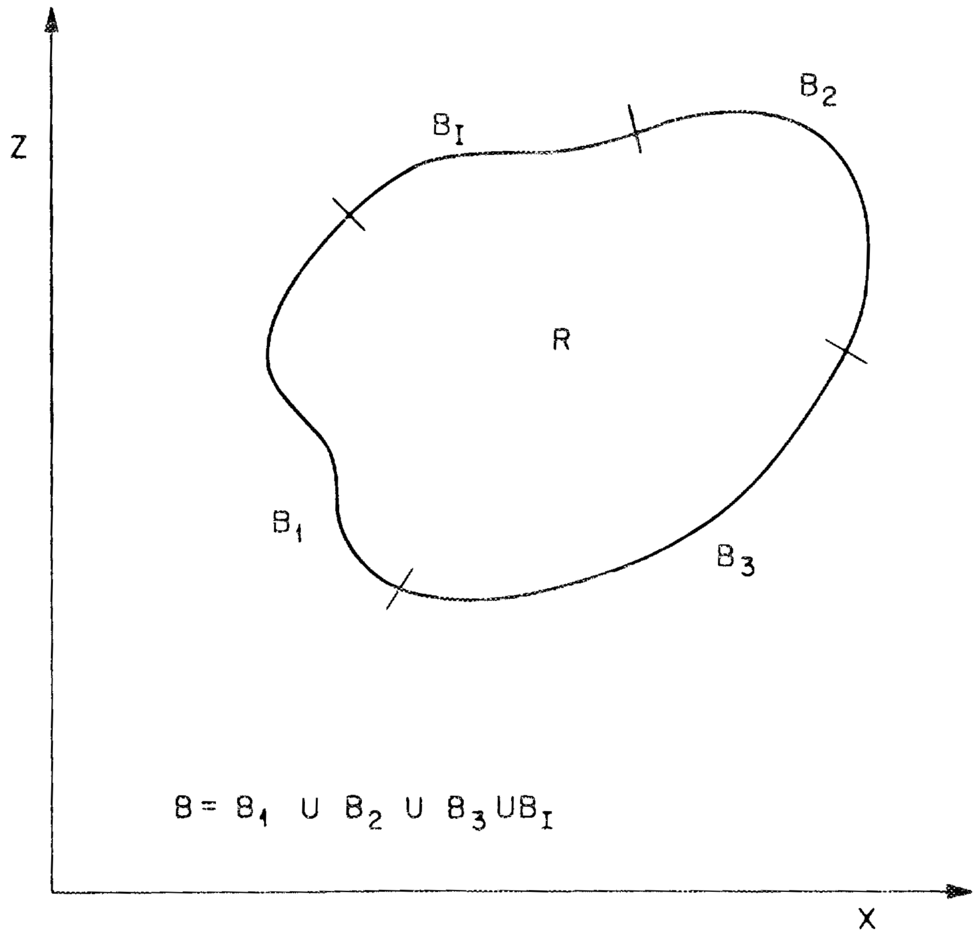

Fig. 1. Spatial boundaries of flow region, R.

steady boundary conditions and ground-water flow field. Three types of boundary conditions may be specified depending on the physical constraint. The first one is the Dirichlet boundaries on which the concentration is prescribed:

$$c = c_1(x,z,t) \quad \text{on } B_1 \quad , \tag{7}$$

where B_1 is a portion of B, and c_1 is a given function of time and (x,z) on B_1. The second one is the Neumann boundaries on which the normal gradient of the concentration is prescribed:

$$- (\theta D_{xx} \frac{\partial c}{\partial x} + \theta D_{xz} \frac{\partial c}{\partial z} - V_x c)n_x - (\theta D_{zx} \frac{\partial c}{\partial x} + \theta D_{zz} \frac{\partial c}{\partial z} - V_z c)n_z = q_2(x,z,t) + (V_x n_x + V_z n_z)c \quad \text{on } B_2 \quad , \tag{8}$$

where n_x and n_z are the directional cosines of the outward unit vector normal to the B_2 portion of the curve B; $q_2(x,z,t)$ is the given function of time, t, and (x,z) on B_2. The right-hand side of Eq. (8) is zero when it is applied to impervious boundaries on which both $q_2(x,z,t)$ and the normal velocity, $(V_x n_x + V_z n_z)$, are equal to zero. If it is applied to the flow-through boundaries with outflows from the region, it becomes a concentration-dependent boundary condition and $q_2(x,z,t)$ is normally set to zero. When it is applied to the flow-through boundaries with inflows into the region, it degenerates into a third type (Cauchy) boundary condition:

$$- (D_{xx} \theta \frac{\partial c}{\partial x} + D_{xz} \theta \frac{\partial c}{\partial z} - V_x c)n_x - (D_{zx} \theta \frac{\partial c}{\partial x} + D_{zz} \theta \frac{\partial c}{\partial z} - V_z c)n_z = q_3(x,z,t) \quad \text{on } B_3 \quad , \tag{9}$$

where q_3 is a given function of time and points (x,z) on B_3 portion of B. The boundaries, B_1, B_2, B_3, and the imp vious boundary B_I consititute the whole boundary B(x,z) = 0 as shown in Fig. 1.

2. Distribution Coefficient and Retardation Factor

The transport mechanisms included in this report are chemical sorption, first-order decay, advection (convection), and hydrodynamic dispersion. The mitigation of the dissolved waste in the water is accomplished through chemical sorption and the first-order decay. The advection is only to move the waste to where it should be. Hydrodynamic dispersion spreads the waste over a wide region with respect to its mean position of advection. These two last mechanisms do not contribute anything directly toward reducing the total amount of waste dissolved in the water but they do affect the concentration distribution. Thus, proper formulation of the chemical sorption by soil matrix is vital to the study of the mitigation and transport of dissolved waste. As stated in the last section, this process is assumed to be characterized by the distribution coefficient, K_d, and retardation factor, R_d. Hence, the problem is to appropriately formulate K_d and R_d as a function of flow dynamics. Duguid and Reeves assumed that K_d is directly proportional to the moisture content, θ(Reeves and Duguid 1975). This assumption yields unrealistic transport in the unsaturated zone after extended simulation periods. In the absence of competing ions, K_d is a function of the soil and the constituent. The absence of water does not necessarily imply the distribution coefficient is small. The dependence of K_d on the moisture content is very complicated but also not unique and varies with the type of soils and constituent involved (Kokotov and Popova 1962, and Prokhorov and Chaai 1963). Several investigators have simply assumed that K_d is independent of moisture content (Van Genuchten et al. 1977, Van Genuchten and Pinder 1978). A moisture-independent K_d yields a more realistic concentration distribution in the unsaturated zone for soils we have studied. Therefore, this assumption is adopted here. Figure 2 shows the differences in formulating K_d and R_d between the original model (Duguid and Reeves 1976) and the present work.

Perhaps it is worthwhile to deviate here to discuss the effect of the moisture content on the waste transport. When the moisture content is extremely unsaturated, it may be imagined that not all the soil grains are effectively participating the absorption of a constituent. Recall in the derivation of Eq. (1), ρs represents the total amount of constituent absorbed in the soil matrix per unit bulk volume. Thus for extreme unsaturation, ρs should be replaced by $\rho_e s$ where ρ_e is defined as the "effective" or "absorbable" bulk density and may depend on the moisture content. Physically, ρ_e represents the density of the effective soil matrix that participates in absorption. This definition of the effective density is very similar to the definition of effective or drainable porosity used in the ground-water movement study. The effective porosity physically represents that portion of the pore space usable in transporting water flow (Eagleson 1970). From

ORNL-DWG 79-13049R

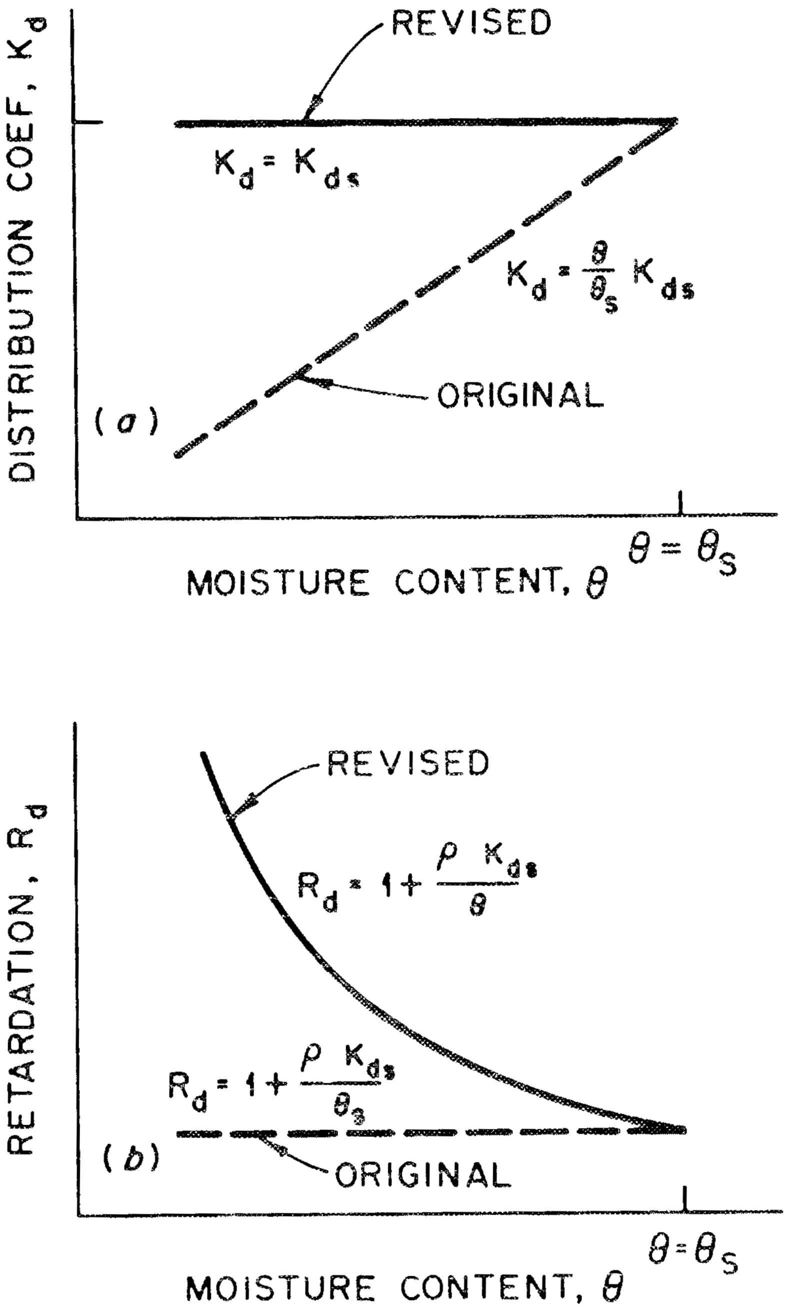

Fig. 2. Comparison of (a) the distribution coefficient, K_d, and (b) retardation factor, R_d, as formulated in the original model (-----) and the present model(_____).

this discussion, it is obvious that one should use a moisture-dependent ρ_e in Eq. (1) to replace ρ under general conditions:

$$\rho = \rho_e \tag{10}$$

However, it is noted that in Eq. (1), the bulk density in all terms comes together with the absorbed concentration, s. Thus, to assume a moisture-dependent distribution coefficient is mathematically indistinguishable from assuming a moisture-dependent effective bulk density, although Physical implications are different. A moisture dependent effective bulk density is much easier to conceptualize. In fact, one would imagine that ρ_e/ρ as function of θ would be equal to 1 long before the soil is completely saturated as shown in Fig. 3. There will exist a critical moisture content, θ_c, below which not all the soil grains are effective in absorbing. This critical θ_c warrants further research. This report will deal with the case of $\theta > \theta_c$ only, since this is equivalent to the assumption that K_d is independent of moisture content and yields plausible concentration distribution in the unsaturated zone for our case study.

3. Finite-Element Approximations

Equations (4) through (10) will be integrated in the spatial dimensions by the weighted-residual method in conjunction with finite elements. Because the formulation and use of the finite-element weighted-residual method has been well addressed (Huebner 1975), the theoretical basis of the method will not be repeated here. Only the numerical procedures are summarized in the following. The region of

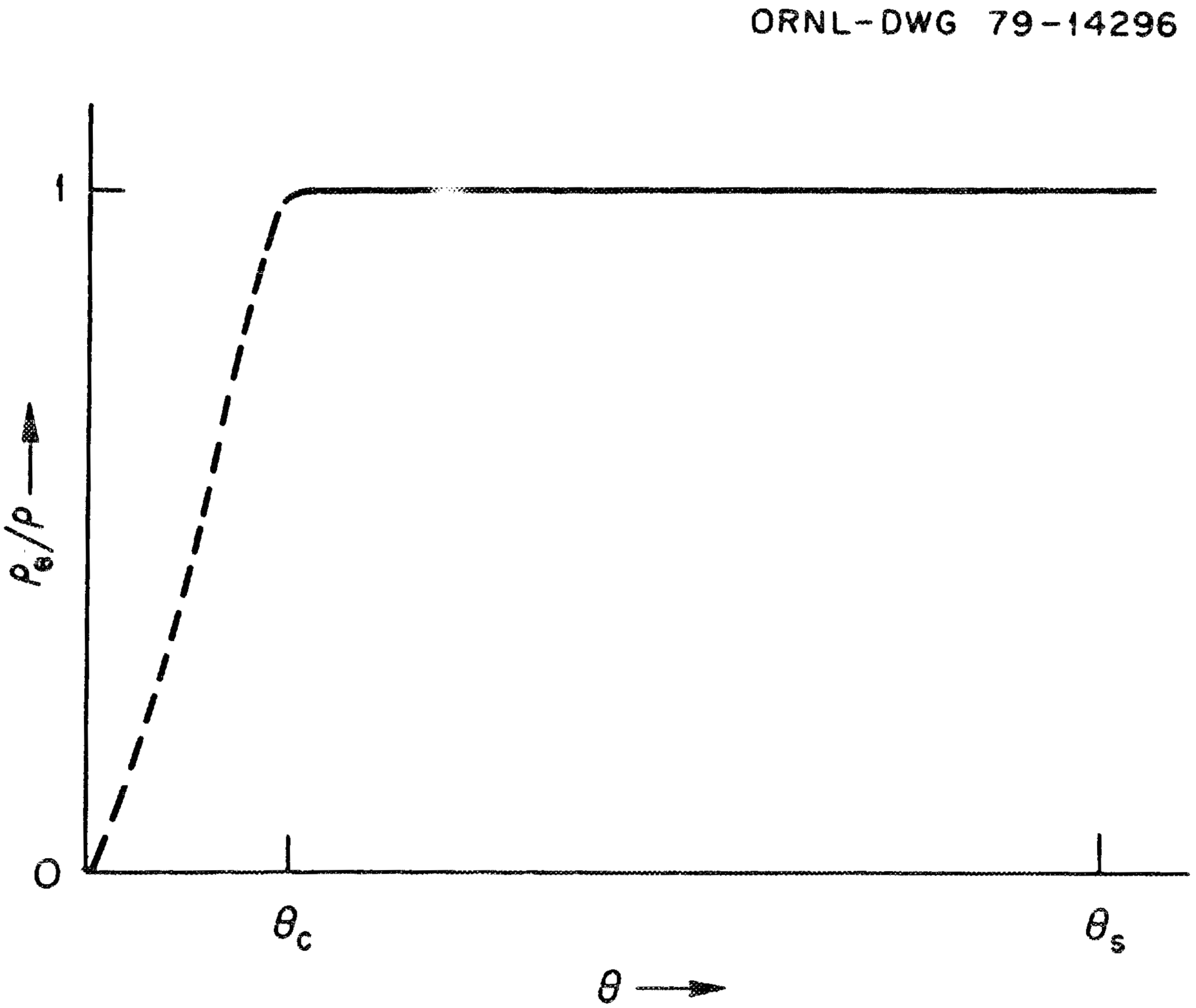

Fig. 3. Sketch of possible dependence of effictive bulk density on the moisture content.

interest is subdivided into an assemblage of smaller sub-domains called elements. Following the procedure of finite-element weighted-residual method, and employing quadrilateral bilinear elements for spatial discretization, approximate formulation of the concentration distribution, c, will be obtained. Thus, let the variable, c, be approximated in an element, e, by:

$$c \cong \hat{c} = \sum_{j=1}^{4} c_j(t)\, N_j \quad , \tag{11}$$

where N_j and c_j are the base functions (interpolating functions) of element, e, and the magnitude of c, respectively, at nodal point, j. Upon substituting Eq. (11) into Eq. (5) and applying the weighted-residual equation,

$$\int_{R_e} W_i \cdot L(c)\, dR = 0, \; i = 1,2,3,4 \quad , \tag{12}$$

one obtains the following element matrix equation for element e:

$$\left[M_{ij}\right]\left\{\dot{c}_j\right\} + \left[S_{ij}\right]\left\{c_j\right\} + \left\{D_i\right\} + \left\{Q_i\right\} = 0 \quad , \tag{13}$$

where W_i is the weighting function of nodal point i; dR is an differential area within element e; R_e is the region of element e; and

$$\dot{c}_j = \frac{dc_j}{dt} \tag{14}$$

The element mass matrix, $\left[M_{ij}\right]$, is given by the following equation:

$$M_{ij} = \int_{R_e} \theta W_i R_d N_j \, dR \tag{15}$$

The element stiff matrix, $[S_{ij}]$, is defined by

$$S_{ij} = \int_{R_e} \left\{ \frac{\partial W_i}{\partial x}\left(\theta D_{xx} \frac{\partial N_j}{\partial x} + \theta D_{xz} \frac{\partial N_j}{\partial z}\right) + \frac{\partial W_i}{\partial z}\left(\theta D_{zx} \frac{\partial N_j}{\partial x} + \theta D_{zz} \frac{\partial N_j}{\partial z}\right) \right.$$

$$\left. - \left(\frac{\partial W_i}{\partial x} V_x N_j + \frac{\partial W_i}{\partial z} V_z N_j\right) + W_i\left(\frac{\partial \theta}{\partial t} + \alpha' \theta R_d \frac{\partial h}{\partial t} + \lambda \theta R_d\right) N_j \right\} dR \quad . \qquad (16)$$

The element column matrices, $\{D_i\}$ and $\{Q_i\}$, are given by the following equations,

$$D_i = \int_{R_e} -W_i M \, dR \qquad (17)$$

and

$$Q_i = \int_{B_e} W_i \left\{ \left(-\theta D_{xx} \frac{\partial c}{\partial x} - \theta D_{xz} \frac{\partial c}{\partial z} + V_x c\right) n_x + \right.$$

$$\left. \left(-\theta D_{zx} \frac{\partial c}{\partial x} - \theta D_{zz} \frac{\partial c}{\partial z} + V_z c\right) n_z \right\} dB \quad , \qquad (18)$$

in which B_e is the boundary of the element e, and n_x and n_z are directional cosines. Equation (18) applies to only those elements having one or more sides on either B_2 or B_3. For all interior elements, Q_i's are set equal to 0 because they would cancel each other when the global matrix is obtained by assembling the element matrix. The incorporation of Q_i will be addressed later.

4. Mass Lumping Option

Referring to the element mass matrix, $[M_{ij}]$, one may note that this is a unit matrix if the finite-difference formulation is adopted in the spatial discretization. Hence by proper scaling, the 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 (Gureghian et al., submitted). Under such circumstances, it is preferred to the consistent mass matrix (mass matrix without lumping). Therefore, an option is provided in this report for the lumping of mass matrix, $\left[M_{ij}\right]$. More explicitly, $\left[M_{ij}\right]$ will be lumped according to

$$M_{ii} = \sum_{j=1}^{4} \int_{R_e} \theta W_i \, R_d N_j \, dR \tag{19}$$

and

$$M_{ij} = 0 \text{ if } i \neq j \;, \tag{20}$$

where R_e is the region of element e, and no summation is taken over i in Eq. (19).

5. Time-Marching Methods

Two most important advantages in finite-element approximation over the finite-difference approximation are the intrinsic abilities to handle the complex boundaries and the normal derivatives therein. In the time dimension, none of these advantages is evident. Thus, finite-difference methods are normally used in approximating the time derivative and in marching the solution. Two time-marching methods were adopted in the material transport model (Duguid and Reeves 1976). The first one is the central or Crank-Nicolson formulation:

$$\left[M_{ij}\right]\left(\left\{c_j\right\}_{t+\Delta t} - \left\{c_j\right\}_t\right)/\Delta t + \frac{1}{2} \bullet \left[S_{ij}\right]\left(\left\{c_j\right\}_{t+\Delta t} + \left\{c_j\right\}_t\right) + \left\{D_i\right\} + \left\{Q_i\right\} = 0 \quad , \tag{21}$$

where $\left[M_{ij}\right]$, $\left[S_{ij}\right]$, $\left\{D_i\right\}$, and $\left\{Q_i\right\}$ are evaluated at time, $t + \Delta t/2$. The second method is the backward-difference formulation,

$$\left[M_{ij}\right]\left(\left\{c_j\right\}_{t+\Delta t} - \left\{c_j\right\}_t\right)/\Delta t + \left[S_{ij}\right]\left\{c_j\right\}_{t+\Delta t} + \left\{D_i\right\} + \left\{Q_i\right\} = 0 \quad , \tag{22}$$

where $\left[M_{ij}\right]$, $\left[S_{ij}\right]$, $\left\{D_i\right\}$, and $\left\{Q_i\right\}$ are evaluated at time, $t + \Delta t$. A third optional time marching is also provided in this report. In this method, the values of unknowns are assumed to vary linearly with time in the time interval, Δt. It is thus called the mid-difference method and the recurrence formulae are:

$$\left(2\left[M_{ij}\right]/\Delta t + \left[S_{ij}\right]\right)\left\{c_j\right\}_{t+\Delta t/2} - \frac{2}{\Delta t} \bullet \left[M_{ij}\right]\left\{c_j\right\}_t + \left\{D_i\right\} + \left\{Q_i\right\} = 0 \tag{23}$$

and

$$\left\{c_j\right\}_{t+\Delta t} = 2\left\{c_j\right\}_{t+\Delta t/2} - \left\{c_j\right\}_t \quad , \tag{23a}$$

where $\left[M_{ij}\right]$, $\left[S_{ij}\right]$, $\left\{D_i\right\}$, and $\left\{Q_i\right\}$ are all evaluated at time, $t + \Delta t/2$. This option has been shown superior to the central or backward-difference formulation, if the mass matrix is not lumped (Gureghian et al. 1979). The element matrix equations, Eq. (21), (22), and (23), may be written as:

$$\left[C_{ij}\right]\left\{c_j\right\} = \left\{R_i\right\} - \left\{Q_i\right\} \quad , \tag{24}$$

where $\left[C_{ij}\right]$ is the element coefficient matrix, $\left\{c_j\right\}$ is the unknown vector to be found, and $\left\{R_i\right\}$ is the element load vector. Take, for example, Eq. (21), $\left[C_{ij}\right]$, $\left\{c_j\right\}$, and $\left\{R_j\right\}$ represent the following:

$$\left[C_{ij}\right] = \left[M_{ij}\right]/\Delta t + \frac{1}{2} \bullet \left[S_{ij}\right] . \tag{21a}$$

$$\left\{c_j\right\} = \left\{c_j\right\}_{t+\Delta t} , \tag{21b}$$

and

$$\left\{R_j\right\} = \left[M_{ij}\right]\left\{c_j\right\}_t/\Delta t + \frac{1}{2} \bullet \left[S_{ij}\right]\left\{c_j\right\}_t - \left\{D_i\right\} , \tag{21c}$$

respectively.

6. Base and Upstream Weighting Functions

For a quadrilateral element with four corner nodes, a bilinear polynomial base function for the i-th node may be written in terms of local normalized coordinates as:

$$N_i = \frac{1}{4} \bullet (1 + \xi_i\xi) \bullet (1 + \eta_i\eta) \quad i=1,2,3,4 , \tag{25}$$

where ξ_i and η_i are the local coordinates of the corner nodes, which are numbered 1 to 4 progressing around the element in a counterclockwise direction as shown in Fig. 4.

When the advection terms in Eq. (5) are as important as the dispersion terms, it is sometimes advantageous to use a weighting function different from the base function. For example, upstream weighting functions have been shown to be adequate (Huyakorn and Pinder 1977). In this report two options are provided: one is the Galerkin

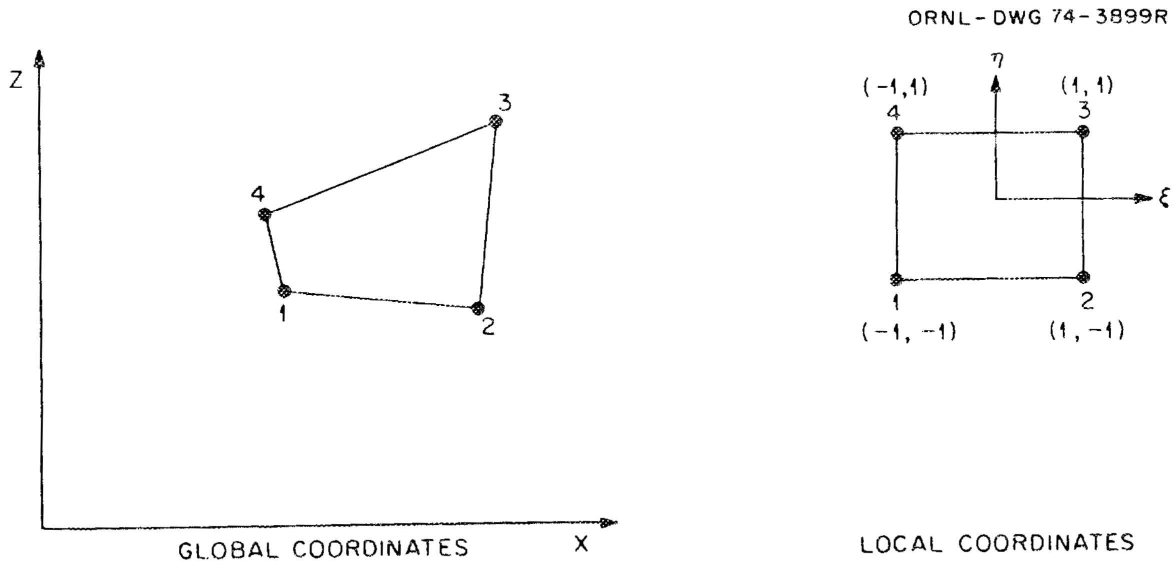

Fig. 4. A typical finite-element in global and local coordinates.

finite-element option and the other is the upwind weighting finite-element option. The function, W_i, for the former will be set equal to the base function, N_i, i.e.,

$$W_i = N_i, \quad i = 1,2,3,4 \ . \tag{26}$$

The weighting function for the later case has been derived elsewhere (Huyakorn and Pinder 1977, Huyakorn and Nilkuha 1974) and will be used here:

$$\begin{aligned}
W_1 &= \frac{1}{16}\left[(1+\eta)(3\beta_2 \eta - 3\beta_2 - 2) + 4\right]\left[(1+\xi)(3\alpha_1\xi - 3\alpha_1 - 2) + 4\right] , \\
W_2 &= \frac{1}{16}\left[(1+\eta)(3\beta_1 \eta - 3\beta_1 - 2) + 4\right]\left[(1+\xi)(-3\alpha_1\xi + 3\alpha_1 + 2)\right] , \\
W_3 &= \frac{1}{16}\left[(1+\eta)(-3\beta_1 \eta + 3\beta_1 + 2)\right]\left[(1+\xi)(-3\alpha_2\xi + 3\alpha_2 + 2)\right] , \\
W_4 &= \frac{1}{16}\left[(1+\eta)(-3\beta_2 \eta + 3\beta_2 + 2)\right]\left[(1+\xi)(3\alpha_2\xi - 3\alpha_2 - 2) + 4\right] ,
\end{aligned} \tag{27}$$

where α_1, α_2, β_1, and β_2 are the weighting factors assigned to the sides 12, 43, 23, and 14, respectively, of the element, e, as shown in Fig. 4. The exact solution to the original differential equation can be obtained at the nodal points if (Christie et al. 1976)

$$\alpha = \coth\left[\frac{uL}{2D}\right] - \frac{2D}{uL} \ , \tag{28}$$

where α may denote α_1, α_2, β_1, or β_2; and u, L, and D are the corresponding velocity component, length, and dispersion component, respectively, along the corresponding side (Fig. 5). Figure 6 shows the optimal values of α as a function of the Peclet number, uL/D (Kantorovich and Krylov 1964). It should be noted that α in Eq. (28)

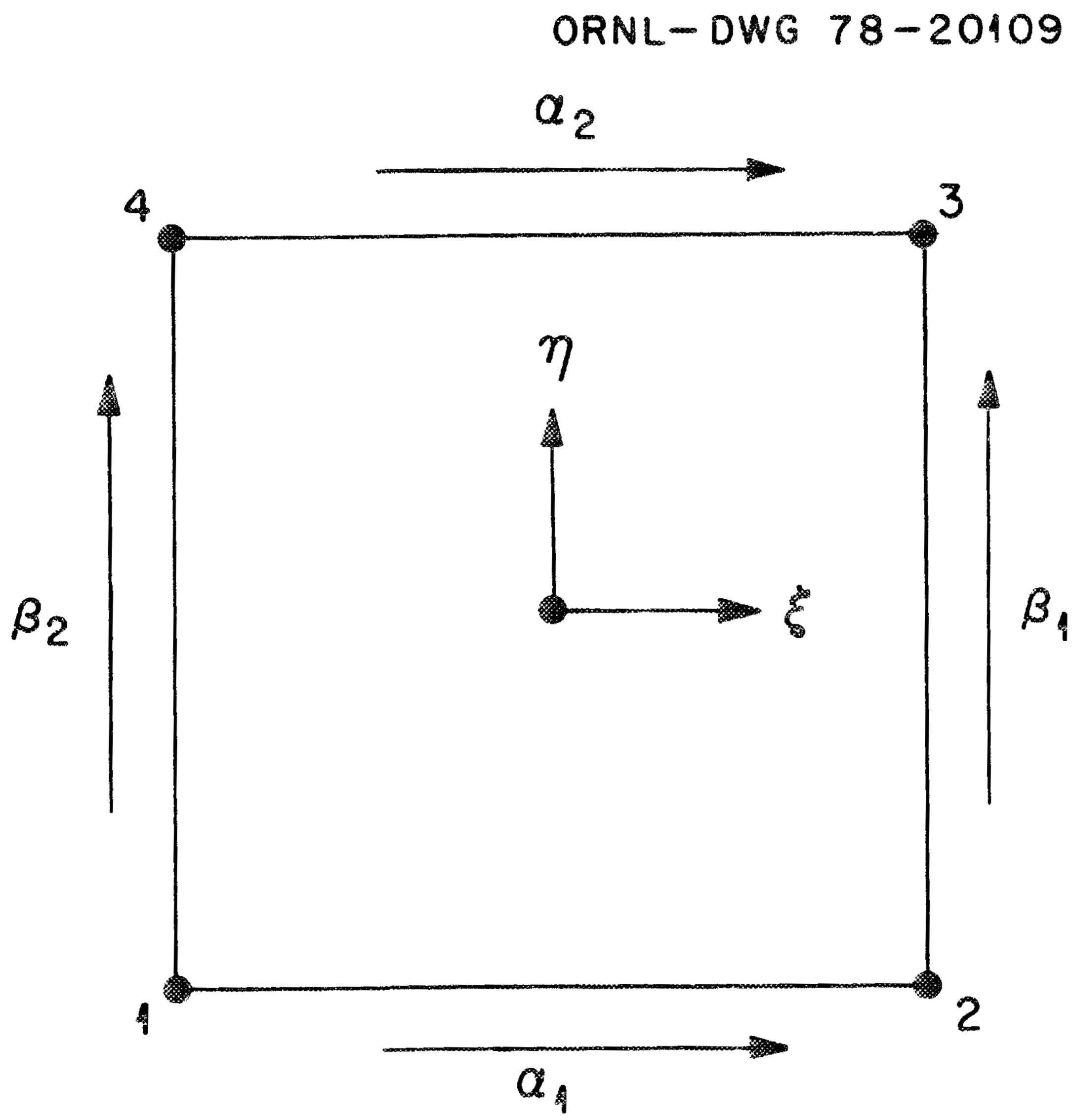

Fig. 5. Location and direction of upwind weighting functions along the edge of a transformed element.

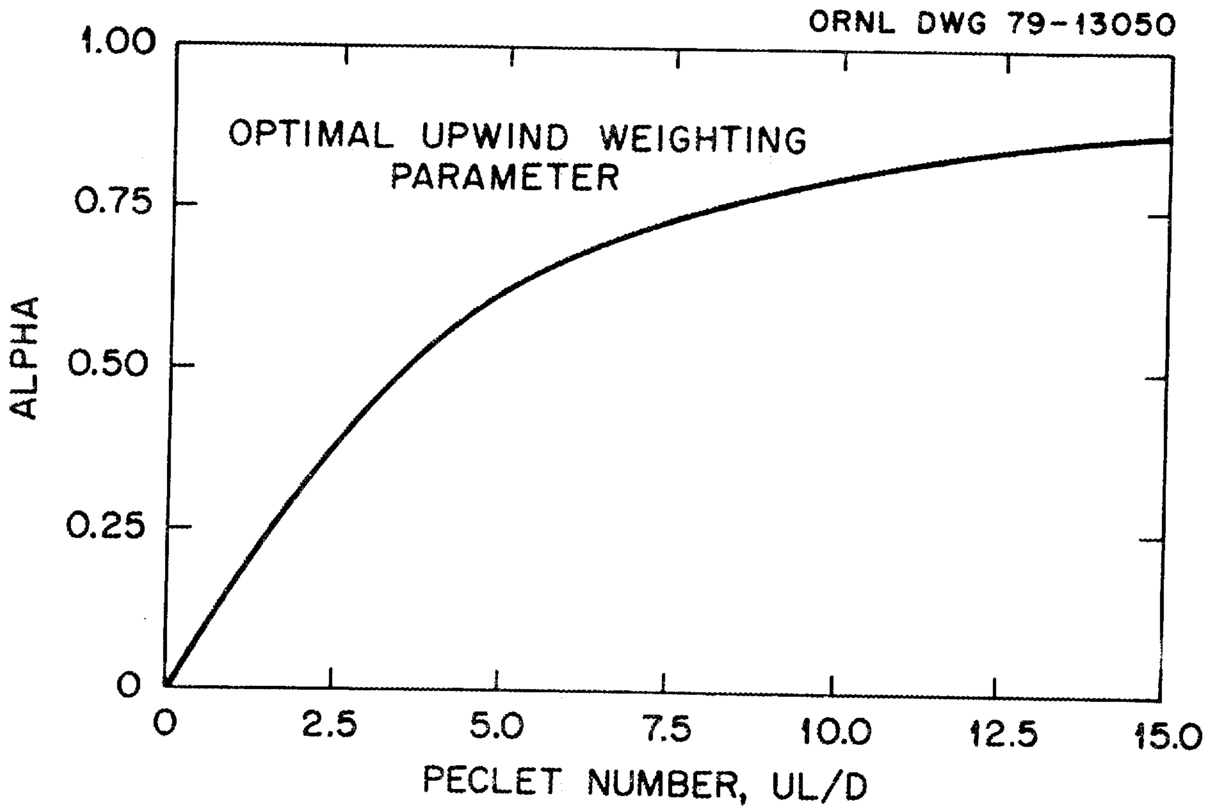

Fig. 6. Optimal values of the upwind weighting parameter for different values of Peclet number.

was derived for steady transport. For the transient transport, dimensional analysis would yield the dependence of α on $u\Delta t/L$ as well as uL/D (Zienkiewicz and Parekh 1970). The relationship between α and two dimensionless parameters, $u\Delta T/L$ and uL/D, has to be a further topic of research.

7. Numerical Integration

In the local coordinate system the element is square regardless of the shape of the quadrilateral in the global coordinates. The global coordinates at any point within the element, e, are given in terms of local coordinates by the relationship:

$$x = \sum_{j=1}^{4} x_j N_j$$

and (29)

$$z = \sum_{j=1}^{4} z_j N_j \quad ,$$

where x_j and z_j are the global coordinates of the nodes, and N_j, which depends on the local coordinates ξ and η is the shape function. The shape function, N_j, of the coordinate transformation is taken the same as the base functions; hence, this element formulation is termed isoparametric. The Jacobian for the transformation from global to the local coordinates is expressed as:

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{bmatrix} \qquad (30)$$

Substitution of Eq. (29) into the determinant of this expression yields:

$$J = \text{Det}\left[J\right] = (x_j \frac{\partial N_j}{\partial \xi}) \bullet (z_k \frac{\partial N_k}{\partial \eta}) - (z_j \frac{\partial N_j}{\partial \xi}) \bullet (x_k \frac{\partial N_k}{\partial \eta}) \quad . \tag{31}$$

The integrals of Eq. (15), (16), and (17) over the area of the e-th finite element may now be written in local coordinates using the determinant of the Jacobian to transform the elemental area:

$$M_{ij} = \int_{-1}^{1} \int_{-1}^{1} W_i R_d \theta N_j \; J_d \xi d\eta \tag{32}$$

$$S_{ij} = \int_{-1}^{1} \int_{-1}^{1} \left\{ \frac{\partial W_i}{\partial x}(\theta D_{xx} \frac{\partial N_j}{\partial x} + \theta D_{xz} \frac{\partial N_j}{\partial z}) + \frac{\partial W_i}{\partial z} (\theta D_{zx} \frac{\partial N_j}{\partial x} + \theta D_{zz} \frac{\partial N_j}{\partial z}) - (\frac{\partial W_i}{\partial x} \bullet V_x + \frac{\partial W_i}{\partial z} \bullet V_z)N_j + W_i(\frac{\partial \theta}{\partial t} + \alpha' \theta R_d \frac{\partial h}{\partial t} + \lambda \theta R_d)N_j \right\} Jd\xi d\eta \tag{33}$$

and

$$D_i = \int_{-1}^{1} \int_{-1}^{1} - W_i M \; Jd\xi d\eta \quad . \tag{34}$$

Integration of these equations is easily performed using 2 x 2 Gaussian integration. A linear algebraic equation, Eq. (24), results since $\left\{c_j\right\}$ is a function of time only and the matrices, $\left[M_{ij}\right]$ and $\left[S_{ij}\right]$, and the vector $\left\{D_i\right\}$ are evaluated from the output of ground-water hydrodynamics.

In order to evaluate $\left[S_{ij}\right]$, expressions for the spatial derivative of the interpolation function and weighting function are necessary. The chain rule

$$\begin{Bmatrix} \dfrac{\partial}{\partial \xi} \\ \\ \dfrac{\partial}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} J \end{bmatrix} \begin{Bmatrix} \dfrac{\partial}{\partial x} \\ \\ \dfrac{\partial}{\partial z} \end{Bmatrix} , \tag{35}$$

may be inverted to yield

$$\begin{Bmatrix} \dfrac{\partial}{\partial x} \\ \\ \dfrac{\partial}{\partial z} \end{Bmatrix} = \frac{1}{J} \cdot \begin{bmatrix} \dfrac{\partial z}{\partial \eta} & -\dfrac{\partial z}{\partial \xi} \\ \\ -\dfrac{\partial x}{\partial \eta} & \dfrac{\partial x}{\partial \xi} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial}{\partial \xi} \\ \\ \dfrac{\partial}{\partial \eta} \end{Bmatrix} , \tag{36}$$

using the definition of $\begin{bmatrix} J \end{bmatrix}$ in Eq. (30).

When the top row of Eq. (36) is applied to the base function, N_i, the following is obtained:

$$\frac{\partial N_i}{\partial x} = \frac{1}{J} \left[(z_j \frac{\partial N_j}{\partial \eta}) \cdot \frac{\partial N_i}{\partial \xi} - (z_j \frac{N_j}{\partial \xi}) \cdot \frac{N_i}{\partial \eta} \right] . \tag{37}$$

Similarly,

$$\frac{\partial W_i}{\partial x} = \frac{1}{J} \left[(z_j \frac{\partial N_j}{\partial \eta}) \cdot \frac{\partial W_i}{\partial \xi} - (z_j \frac{\partial N_j}{\partial \xi}) \cdot \frac{\partial W_i}{\partial \eta} \right] . \tag{38}$$

Finally, applying the bottom row of Eq. (36) to N_i and W_i, one obtains the following expressions:

$$\frac{\partial N_i}{\partial z} = \frac{1}{J} \left[(x_j \frac{\partial N_j}{\partial \eta}) \cdot \frac{\partial N_i}{\partial \xi} - (x_j \frac{\partial N_j}{\partial \xi}) \cdot \frac{\partial N_i}{\partial \eta} \right] \tag{39}$$

and

$$\frac{\partial W_i}{\partial z} = \frac{1}{J}\left[(x_j \frac{\partial N_j}{\partial \eta}) \cdot \frac{\partial W_i}{\partial \xi 25} - (x_j \frac{\partial N_j}{\partial \xi}) \cdot \frac{\partial W_i}{\partial \eta} \right] . \qquad (40)$$

Equations (37) through (40) are in a form suitable for numerical integration. The derivatives of W_i and N_i with respect to ξ and η can be obtained by the partial differentiation of Eqs. (25) and (27), respectively.

8. Assembly of Element Matrix

Equation (24) is evaluated for each element, and the direct stiff method is adopted to assemble them to form a system of algebraic equations as:

$$\left[T_{ij}\right]\left\{c_j\right\} = \left\{X_i\right\} - \left\{B_i\right\} = \left\{Y_i\right\} , \qquad (4[illegible])$$

where $\left[T_{ij}\right]$ is the global coefficient matrix and $\left\{Y_i\right\}$ is the global load vector. The detailed discussion of the assembly of the element matrix into a global matrix has been presented (Desai and Abel 1972, Duguid and Reeves 1976).

9. Application of Boundary Conditions

Surfaces on which third-type boundary conditions, Eq. (9), are imposed, of course, yield concentration-independent entries in the element column matrix $\left\{Q_i\right\}$. These entries are evaluated by the direct application of substituting Eq. (9) into Eq. (17) to yield element normal fluxes. This is then followed by assembling overall boundary elements having one or more sides on the boundaries B_3 of B to yield a global column matrix $\{B_i\}$. The results are then subtracted from the $\{X_i\}$ to form $\{Y_i\}$.

Surfaces on which the Neumann boundary conditions are imposed are either the flow-through boundaries with outflows from the region or the impervious boundaries. For the latter case, the entries $\{B_i\}$ are zero and no boundary conditions have to be applied. For the former case, the corresponding entries in $\{B_i\}$ are then linear functions of the unknown concentration. Such terms must therefore be incorporated into the $[T_{ij}]$ matrix in Eq. (41). Substituting Eq. (8) with $q_2 = 0$ into Eq. (17), one obtains:

$$\{Q_i\} = [E_{ij}]\{c_j\} \quad i = 1,2 \quad \text{and} \quad j = 1,2 \quad , \tag{42}$$

where

$$E_{ij} = \int_{B_e} W_i \cdot (V_x n_x + V_z n_z) \cdot N_j \, dB \quad . \tag{43}$$

For the backward-difference and mid-difference time-marching algorithm, $[E_{ij}]$ after assembly over all boundary elements of B_2 is added to the matrix $[T_{ij}]$. For the central-difference time-marching algorithm, one-half of $[E_{ij}]$ after assembly over boundary elements is added to the matrix $[T_{ij}]$ while the other half is multiplied by $\{c_j\}$ of the previous time and then substracted from $\{X_i\}$.

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

10. Solution of the Assembled Equations

The Gaussian elimination algorithm is used to solve the matrix equation, Eq. (41).

11. Mass Balance Computation

The mass balance over the whole region of interest is obtained by integrating Eq. (1):

$$\int_R \left[\frac{\partial \theta c}{\partial t} + \rho_b K_d \frac{\partial c}{\partial t} + \lambda (\theta + \rho_b K_d) c + \alpha' (\theta + \rho_b K_d) \frac{\partial h}{\partial t} c \right] dR = \int_B F_n \, dB \quad , \tag{44}$$

where F_n is the total flux (advective and dispersive fluxes) through the global boundary, $B(x,z) = 0$. In fact, F_n denotes:

$$F_n = \left[\left(\theta D_{xx} \frac{\partial c}{\partial x} + \theta D_{xz} \frac{\partial c}{\partial z} - V_x c \right) n_x + \left(\theta D_{zx} \frac{\partial c}{\partial x} + \theta D_{zz} \frac{\partial c}{\partial z} - V_z c \right) n_z \right] \quad . \tag{45}$$

Having obtained the concentration field, c, one could integrate the right- and left-hand sides of Eq. (44) independently. If the solution for c is free of error, one would expect the equality of two integrals. In the present report, the integral of the right-hand side is broken into several components:

$$F_D = \int_{B_1} F_n dB \quad , \tag{46}$$

$$F_N = \int_{B_3} F_n dB \quad , \tag{47}$$

$$F_O = \int_{B_2} F_n dB \quad , \text{ and} \qquad (48)$$

$$F_I = \int_{B_I} F_n dB \quad , \qquad (49)$$

where F_D, F_N, F_O, and F_I represent the flux through the prescribed Dirichlet boundary, B_1, the Neumann-degenerated Cauchy boundary with prescribed total flux, B_3, the Neumann boundary with flow going out, B_2, and the impervious boundary, B_I, respectively. On the other hand, the left-hand side of Eq. (44) is divided into four terms,

$$F_W = \int_R \frac{\partial \theta c}{\partial t} dR \quad , \qquad (50)$$

$$F_A = \int_R \rho K_d \frac{\partial c}{\partial t} dR \quad , \qquad (51)$$

$$F_L = \int_R \lambda(\theta + \rho K_d) c \, dR \quad , \text{ and} \qquad (52)$$

$$F_S = \int_R \alpha' (\theta + \rho K_d) \frac{\partial h}{\partial t} c \, dR \quad , \qquad (53)$$

where F_W, F_A, F_L, and F_S denote the rate change in the region due to the dissolved concentration in the water, the absorbed concentration by the solid, the decay, and the concentration change by the compression of the soil matrix, respectively.

For an exact solution, the net influx across the whole boundary, $B(x,z) = 0$, defined by

$$F_{net} = (F_D + F_N + F_O + F_I) \quad , \qquad (54)$$

should be equal to the total volumetric increase, F_V, defined by

$$F_V = F_W + F_A + F_L + F_S \quad . \tag{55}$$

In addition, F_I should theoretically be equal to zero. However, in any practical simulation, F_{net} will not be equal to F_V, and F_I will be non-zero. Nevertheless, the mass balance computation should provide the means to check the numerical scheme and the consistency in the computer code.

12. Computation of Total Waste Flux

The waste flux at any point in the region is due to two mechanisms, advection and dispersion. After the concentration field is obtained and knowing the Darcy's velocity field, the flux components in the x- and z-direction, respectively, F_x and F_z may be obtained by

$$F_x = -\left(\theta D_{xx} \frac{\partial c}{\partial x} + \theta D_{xz} \frac{\partial c}{\partial z}\right) + V_x c \qquad (56$$

and

$$F_z = -\left(\theta D_{zx} \frac{\partial c}{\partial x} + \theta D_{zz} \frac{\partial c}{\partial z}\right) + V_x c \quad . \tag{57}$$

Duguid and Reeves (1976) numerically evaluate Eqs. (56) and (57) by

$$F_x = -\left\{\left(\theta D_{xx} \frac{\partial N_i}{\partial x} + \theta D_{xz} \frac{\partial N_i}{\partial z}\right) - V_x N_i\right\} c_i \tag{58}$$

and

$$F_z = -\left\{\left(\theta D_{zx} \frac{\partial N_i}{\partial x} + \theta D_{zz} \frac{\partial N_i}{\partial z}\right) - V_z N_i\right\} c_i \quad . \tag{59}$$

This formulation would result in the discontinuity of waste flux at nodal points and element boundaries. In the new model, it is proposed that an alternative approach be made by applying the finite-element technique to Eqs. (56) and (57). Thus, one obtains for an element e:

$$\left[S'_{ij}\right]\left\{F_{xj}\right\} = \left\{R_{xi}\right\} \quad (60)$$

and

$$\left[S'_{ij}\right]\left\{F_{zj}\right\} = \left\{R_{zi}\right\} \quad , \quad (61)$$

where

$$S'_{ij} = \int_{R_e} W_i N_j \, dR \quad , \quad (62)$$

$$R_{xi} = - \int_{R_e} W_i \left\{ \theta D_{xx} \frac{\partial N_j}{\partial x} + \theta D_{xz} \frac{\partial N_j}{\partial z} - V_x N_j \right\} c_j \, dR \quad , \text{ and} \quad (63)$$

$$R_{zi} = - \int_{R_e} W_i \left\{ \theta D_{zx} \frac{\partial N_j}{\partial x} + \theta D_{zz} \frac{\partial N_j}{\partial z} - V_z N_j \right\} c_j \, dR \quad . \quad (64)$$

In Eqs. (60) and (61), F_{xj} and F_{zj} are the values of F_x and F_z, respectively, at the nodal point j. This approach yields the continuous flux field ,

$$F_x = \sum_{i=1}^{4} F_{xi} N_i \quad (65)$$

and

$$F_z = \sum_{i=1}^{4} F_{zi} N_i \quad , \quad (66)$$

for all elements in the region of interest. It is an approach consistent with the finite-element spirit.

13. Alternative Numerical Schemes

Depending on the type of weighting functions (Galerkin or upstream weighting), the methods of time marching (Crank-Nicolson, or backward-difference, or mid-difference), and the treatment of mass matrix (lumping or no-lumping), there are 12 optional finite-element numerical schemes in the present work as shown in Table 1. Schemes 1 and 2 were reported by Duguid and Reeves (1976). Schemes 3 through 12 are the expanded ones investigated by several authors.

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

III. COMPUTER PROGRAM MODIFICATION AND EXPANSION

The overall program organization is shown in Figs. 7 and 8. Except for the names of subroutines, the original computer code (Duguid and Reeves 1976) has been almost completely overhauled. The overhaul is necessary to accomplish: (1) the theoretical modification in formulating moisture-dependent retardation factor, R_d, (2) the application of finite-element method to compute the dispersive part of the total material flux (advective and dispersive), (3) the provision of 12 alternative numerical schemes, (4) the reduction of storage by compressing all arrays containing boundary variables, and (5) the adoption of variable arrays in all subroutines.

A short main program is written to dimensionalize and initialize all arrays and to specify the maximum dimension in any of the arrays. The program is then passed to subroutine GM2DXZ, which was the main program in the computer code developed earlier (Duguid and Reeves 1976).

Considerable additions and changes have been made to the subroutine DATAIN. The purpose is to compress the arrays that specify Dirichlet boundary, surface source term (or Neumann boundary), and the mixed boundary conditions. DATAIN is no longer to call subroutine SURF to determine the information of boundary elements and nodes. Instead, having been determined in the water flow code (Yeh and Ward 1980), they are read in via disk unit 1.

Two new subroutines, AFABTA and SHAPE, are made. AFABTA is to computes the weighting parameters, α_1, α_2, β_1 and β_2. SHAPE evaluates the base and weighting functions at the Gaussian integration point.

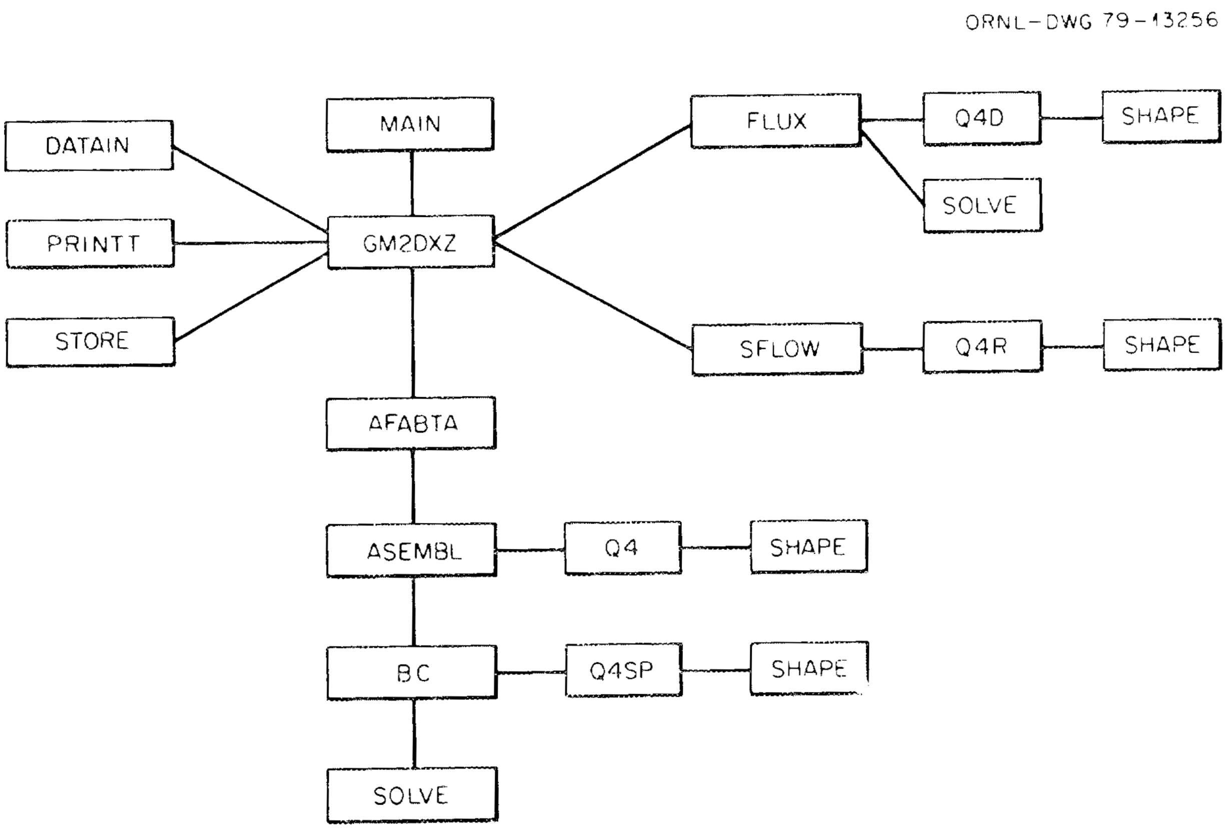

Fig. 7. Subroutine chart of waste transport program.

ORNL-DWG 75-5574R2

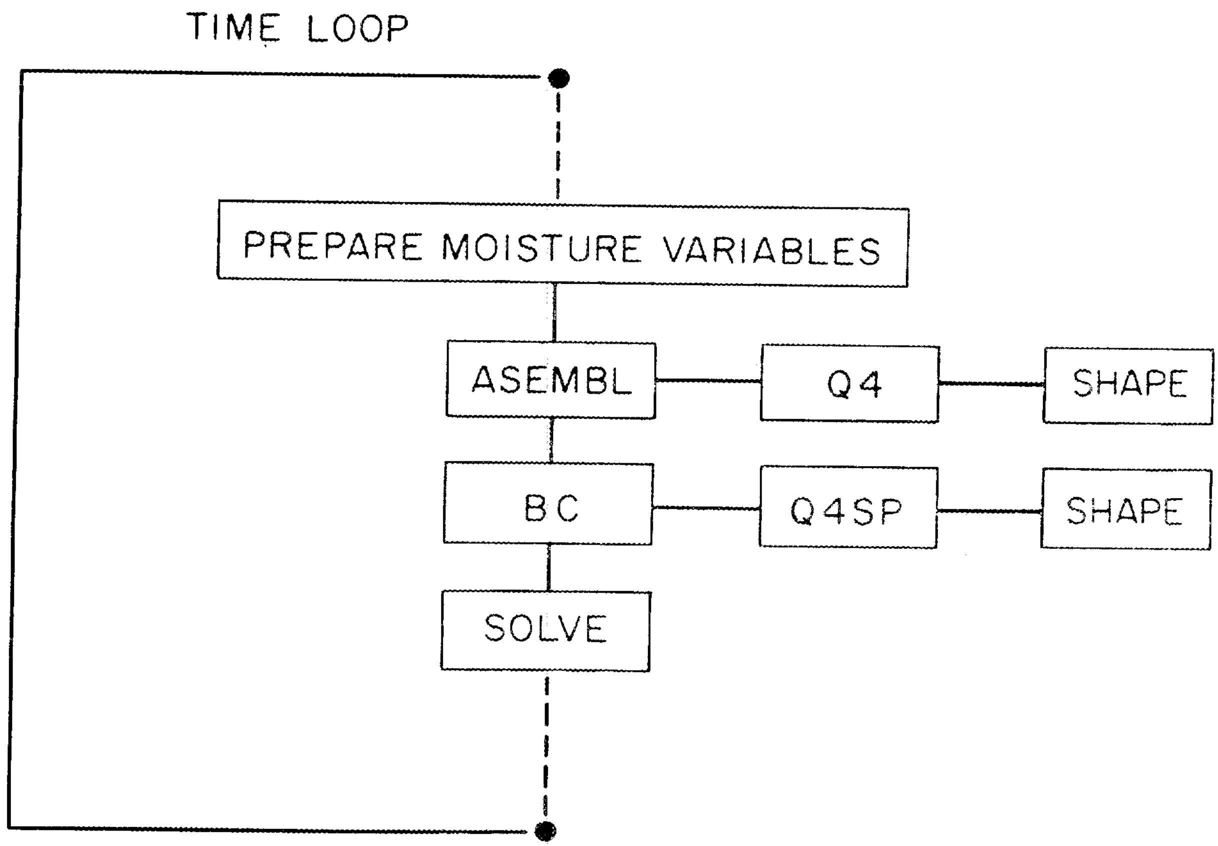

Fig. 8. Subroutine chart for the time-iteration loop.

The subroutines FLUX and Q4D have been rewritten. FLUX in the new model is used to sum over the element matrix $\left[S'_{ij}\right]$. Subroutine FLUX subsequently calls the subroutine SOLVE to yield the solution for F_{xj} and F_{zj} by the Gaussian elimination method. Q4D is called upon by FLUX to evaluate the element matrix $\left[S'_{ij}\right]$ and the load vectors $\left[R_{xi}\right]$ and $\left[R_{zi}\right]$. The total flux field is returned to calling subroutine GM2DXZ through the arguments of FLUX. This flux is then passed to the subroutine SFLOW to compute various surface fluxes via various types of bou.daries. Thus SFLOW no longer has to call Q4S, which is eliminated in the present model.

The theoretical modification of the model is performed in the subroutines, ASEMBL ana Q4. Associated with the theoretical revision, the materials dissolved in the water and adsorbed by the soil matrix have to be reevaluated. This is accomplished through subroutines, SFLOW and Q4R.

On the alternative numerical schemes, the lumping option is determined in the subroutine, Q4. The upstream weighting option is provided in subroutine SHAPE. To evaluate the base and weighting functions, AFABTA must be called at the beginning of each time step. Finally, the time-marching alternatives are accomplished in the subroutines ASEMBL and GM2DXZ.

Subroutines PRINTT and STORE are modified for better display of printout and selectively storing the output on logical unit 2. The standard subroutine, SOLVE, remains intact.

IV. RESULTS

The sample problem of transport from a seepage pond reported by Duguid and Reeves (1976) is used to compare the simulation by the original computer codes (Reeves and Duguid 1975, Duguid and Reeves 1976) with that by the new waste transport code coupled with the revised water-flow code (Yeh and Ward 1980). The seepage pond is assumed to be situated near a stream as shown in Fig. 9. The system is composed of a highly permeable sand with soil properties shown in Fig. 10. A flux of 4.0×10^{-4} cm^3 cm^{-2} sec^{-1}, directed vertically downward from bottom of the pond, provides the only driving force for moving the contaminant toward the stream.

Prior to the simulation of waste transport, hydrodynamic variables are required and computed by flow models to yield pressure distribution. Flow patterns are then derived either by simply taking the derivatives of potential (Reeves and Duguid 1975) or by applying the finite-element technique to Darcy's law (Yeh and Ward 1980). These flow variables are Darcy's velocity, pressure head, total head, and moisture content, which provide the essential input to the waste transport, and are shown in Figs. 11 and 12. Figure 11 displays the results by the original water movement model (Reeves and Duguid 1975) while Fig. 12 describes the results using the revised water flow model (Yeh and Ward 1980). It is interesting to note that flow field is not unique as simulated by the original model (Fig. 11). This deficiency is eliminated by the revised model (Fig. 12)(Yeh and Ward 1980).

ORNL-DWG 79-9954R

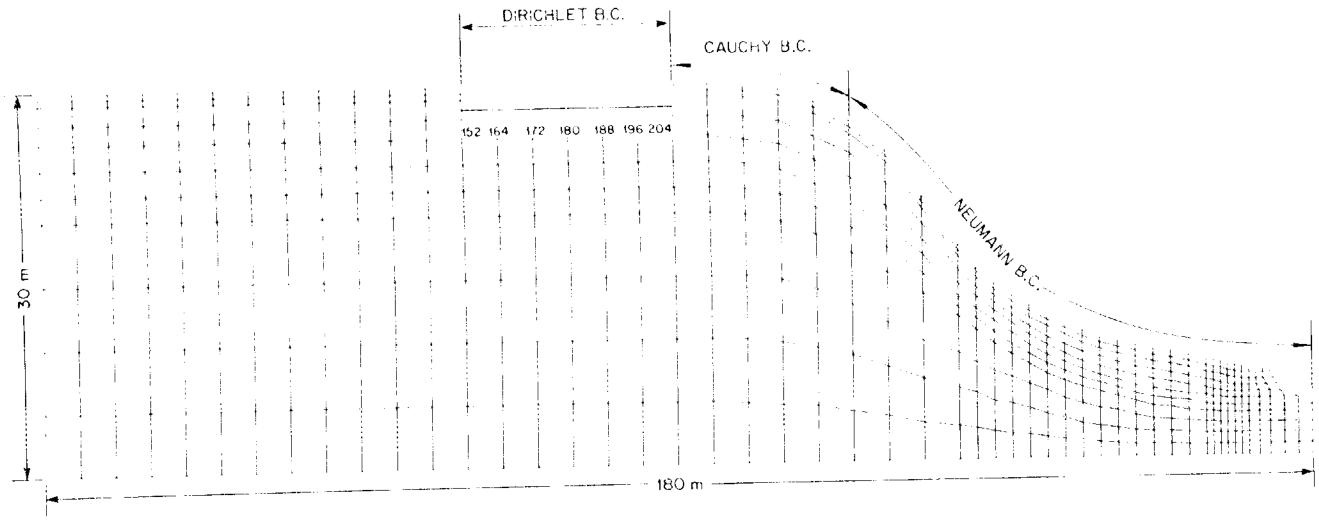

Fig. 9. Finite-element discretization of the seepage pond problem.

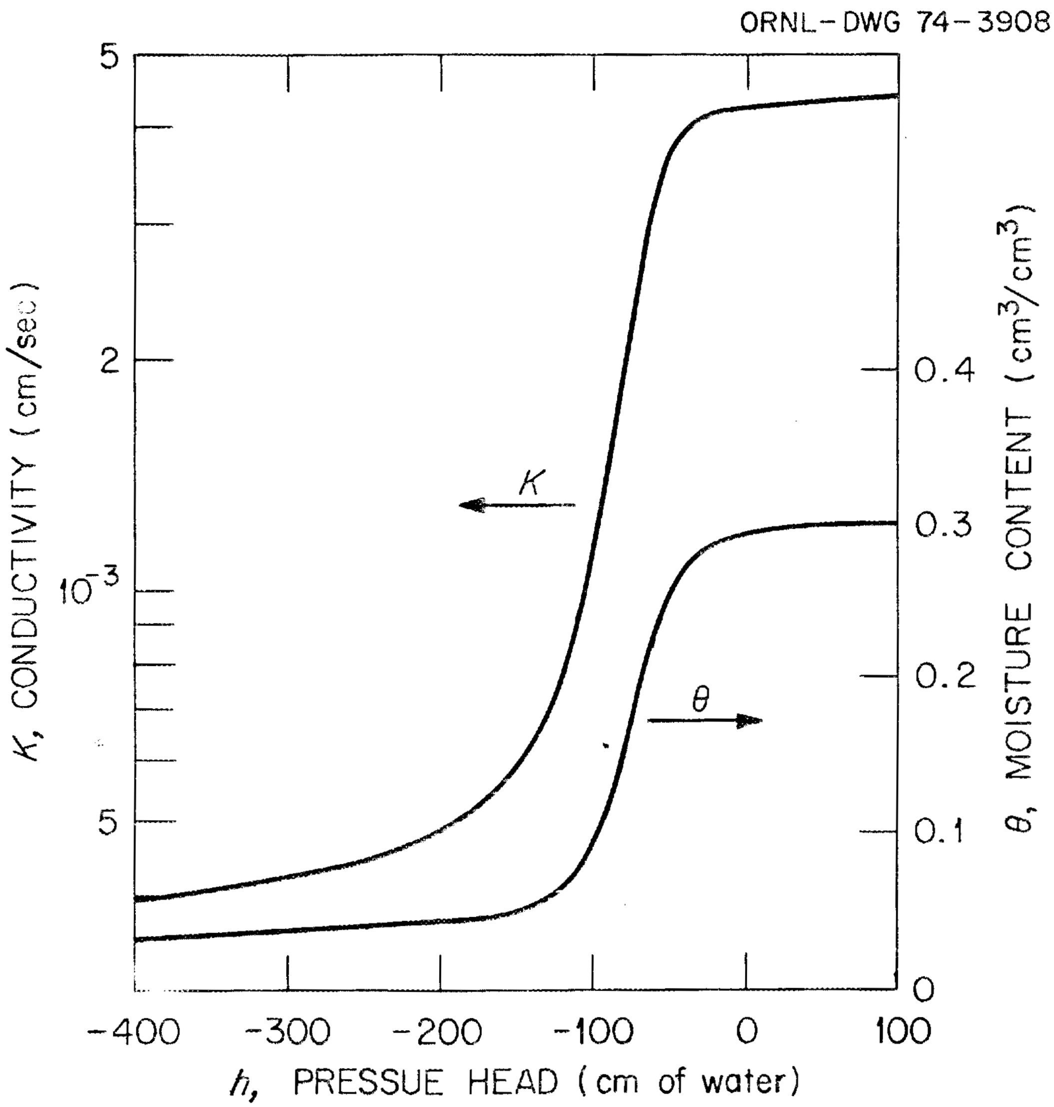

Fig. 10. Hydraulic conductivity and soil-moisture characteristics for a hypothetical sandy soil.

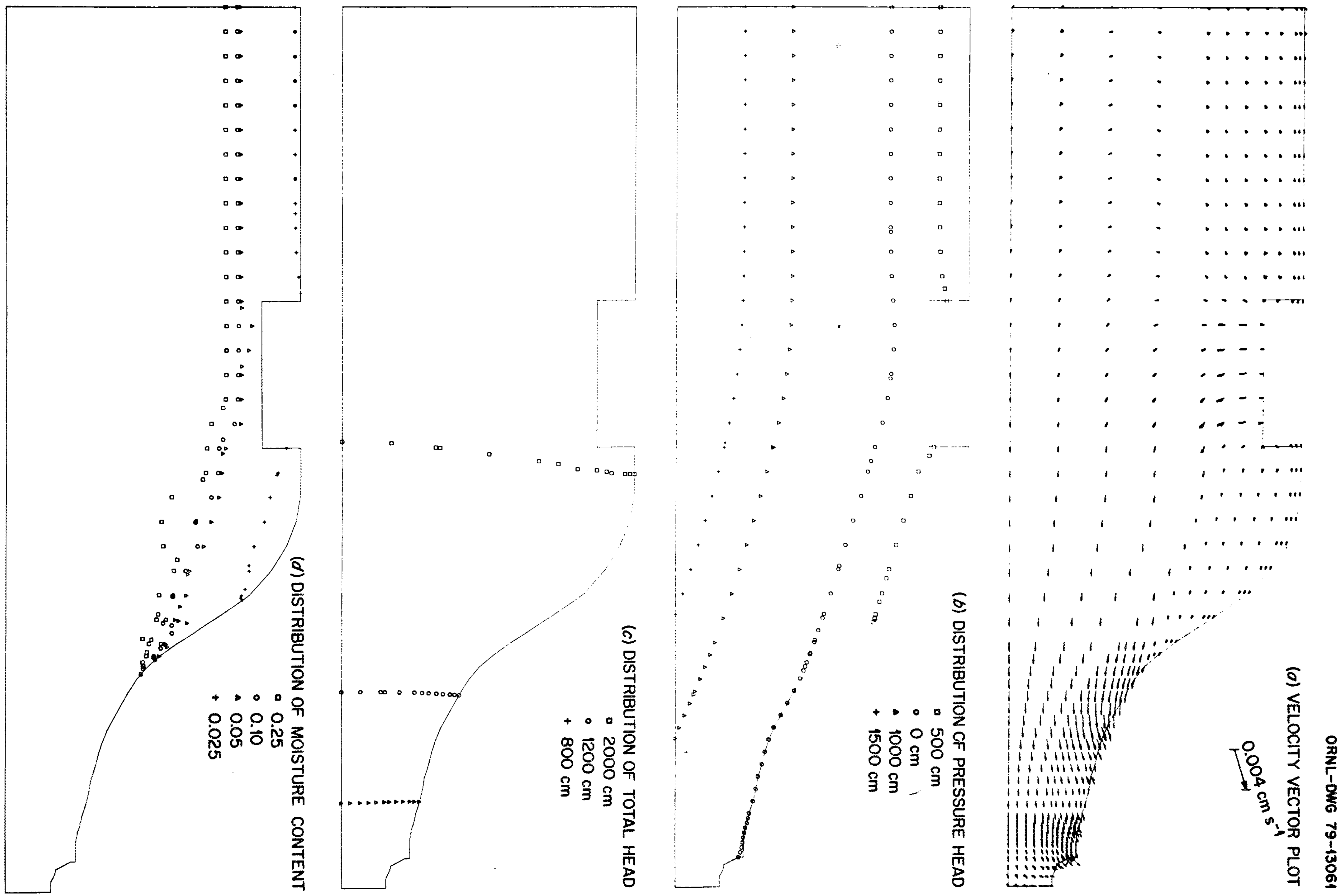

Fig. 11. Flow variables simulated by the Reeves and Duguid model: (a) velocity vector, (b) pressure head, (c) total head, (d) moisture content.

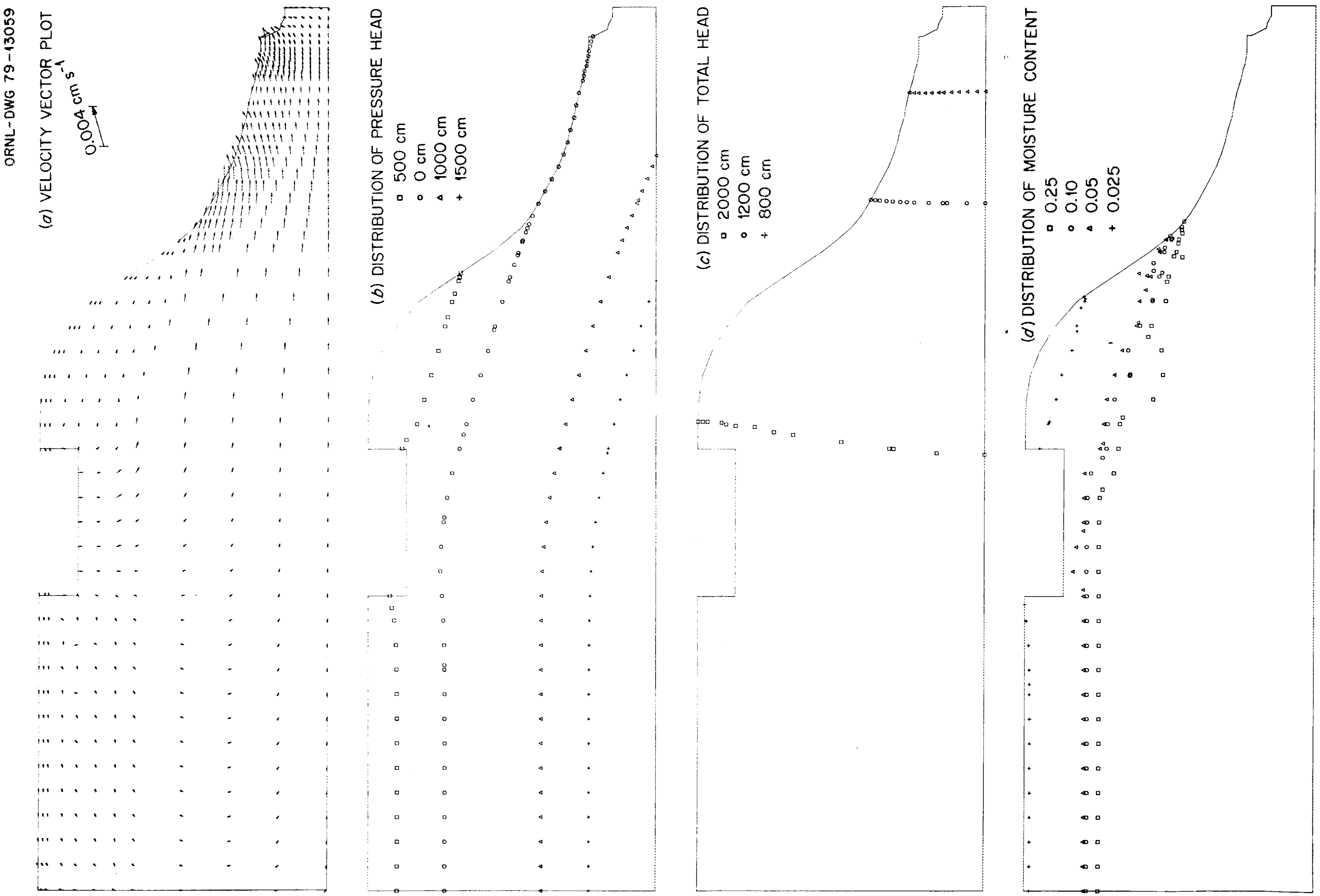

Fig. 12. Flow variables simulated by the Yeh and Ward model: (a) velocity vector, (b) pressure head, (c) total head, (d) moisture content.

Figures 13a and 13b show the contaminant concentration distributions, which were obtained by applying the original and the new models, respectively. The distribution coefficient, $K_d = 100\ cm^3/g$ was used. Longitudinal and transverse dispersivities were taken from Pinder's (1973) study of chromium transport in Long Island, New York: $a_L = 21.3$ m and $a_T = 4.27$ m. It is seen from Fig. 13a that the waste travels as fast in the unsaturated zone as in the saturated zone. This overprediction of concentration distribution from the original model in the unsaturated zone resulted from the unrealistic formulation of a moisture-dependent distribution coefficient, K_d, and a noncontinuous flow field. Figure 13b indicates that a more plausible result was obtained by using the assumption included in present model.

As an academic interest, Figs. 14a and 14b depict two other concentration distributions. Figure 14a was obtained by the old material transport model using the flow field obtained by the revised water flow program. Figure 14b was obtained by the new waste transport model using the flow field obtained by the old water movement program. Comparing Fig. 14a with Fig. 13a or Fig. 14b with Fig. 13b, it is seen that the noncontinous velocity field contributes partially to the overprediction of concentration in the unsaturated zone, but the assumption concerning the distribution coefficient is the most important.

Table 2 shows the comparison of mass balance as predicted by the original and new models, respectively, at the end of the 19.34-year simulation time. The total net mass through all boundaries is only

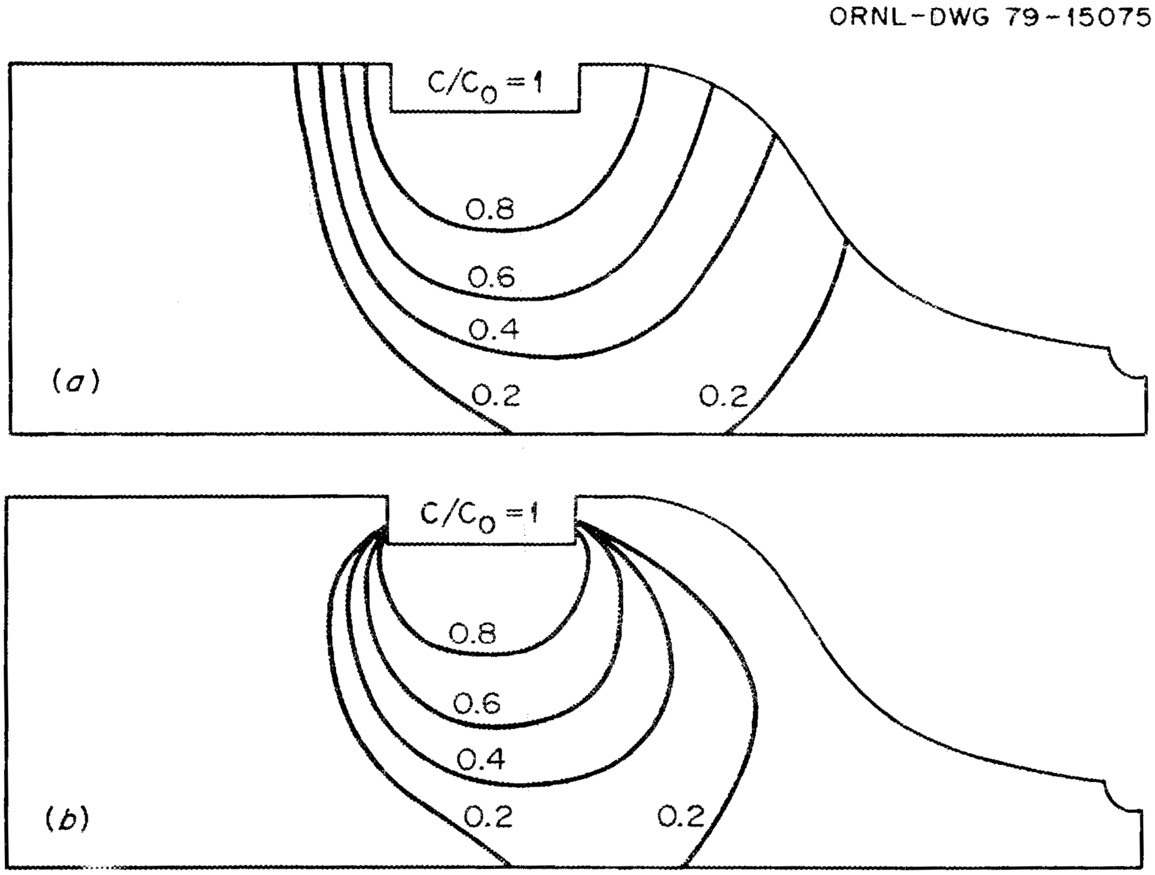

Fig. 13. Normalized concentration distribution for a dissolved constituent with a distribution coefficient of 100 ml/g computed with (a) original models (b) new models.

ORNL-DWG 79-15076

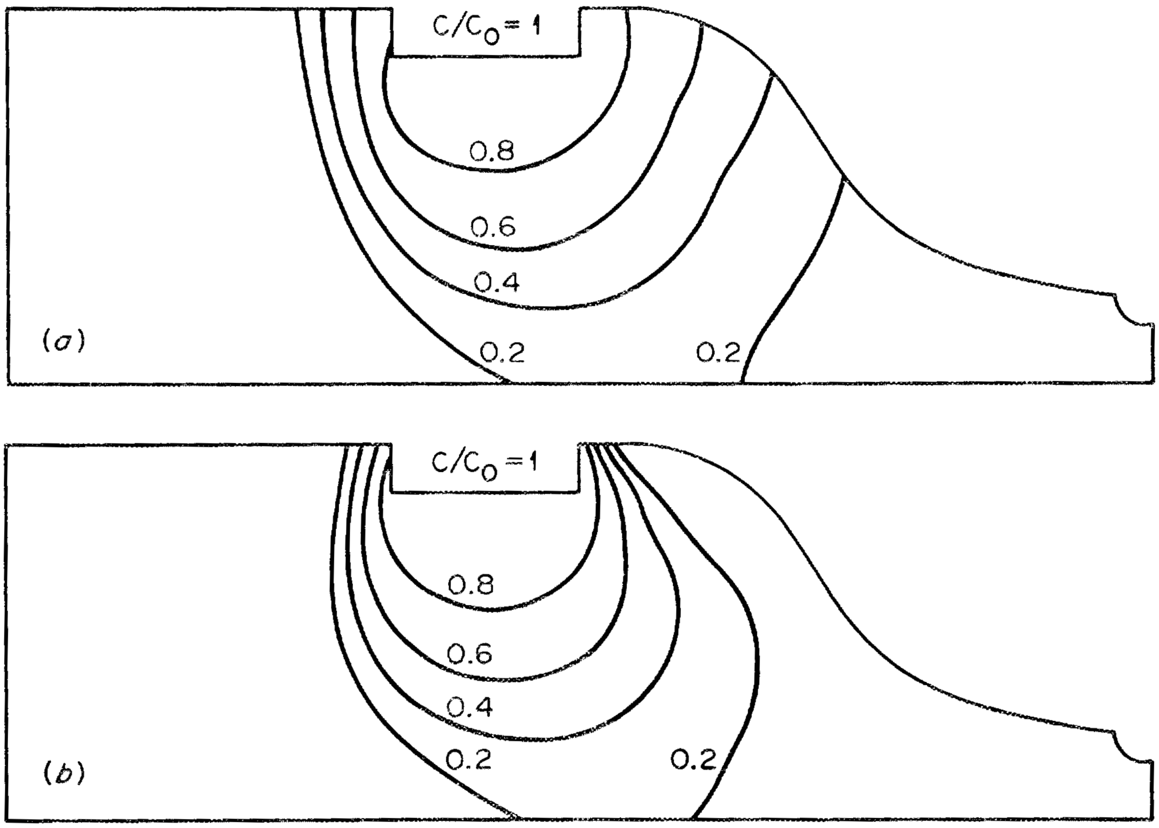

Fig. 14. Concentration distribution for a dissolved constituent with a distribution coefficient of 100 ml/g computed with (a) original material transport model with flow variables from revised water flow model, (b) new waste transport model with flow variables from original water movement model.

Table 2. Comparison of percentage of mass-balance error as simulated by alternative numerical schemes of the original and new models, respectively

Model	Schemes 1	2	3	4	5	6	7	8	9	10	11	12
Old model	17.66	17.02	N/A[a]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
New model	-1.08	-2.16	-0.81	4.06	2.44	-1.[illegible]5	1.30	2.81	-1.08	-0.83	2.44	4.06

[a]Not applicable

about 82.34% of the mass accumulated in the media by numerical scheme 1 of the original model. In other words, 17.66% of the mass has not been accounted for (i.e., has been lost through the boundaries.) On the other hand, numerical scheme 1 of the new model shows a 1.08% mass gain (- 1.08% of mass loss). For numerical scheme 2, the original and new models render a 17.02% of mass loss and 2.16% of mass gain, respectively. Regardless of the mass gain or mass loss, the error in mass balance by the new model is much smaller than that by the old model. Figures 15 and 16 illustrate the comparison of mass balance rate history as simulated by the original and new (present) models with numerical schemes 1 and 2, respectively. A positive value means mass loss while a negative value indicate mass gain. They show that the mass-balance-rate error by the original model is large and, after initial decrease, increasing with time. On the other hand, the mass-balance-rate error as simulated by the new model is relatively small and decreasing with time.

Table 2 also shows the percentage of mass-balance error by all alternative numerical schemes. Because an analytical solution for the concentration distribution for this type of problem is not available, the comparison of concentration distribution as simulated by alternative schemes with that by analytical solution is not possible. Therefore, the percentage of mass-balance error is used as a criterion to ascertain the superiority of one method over the other. From this point of view, it is seen that numerical schemes 3 and 10 give the best results for the example problem. At any rate, all 12 schemes of the new model are considered acceptable.

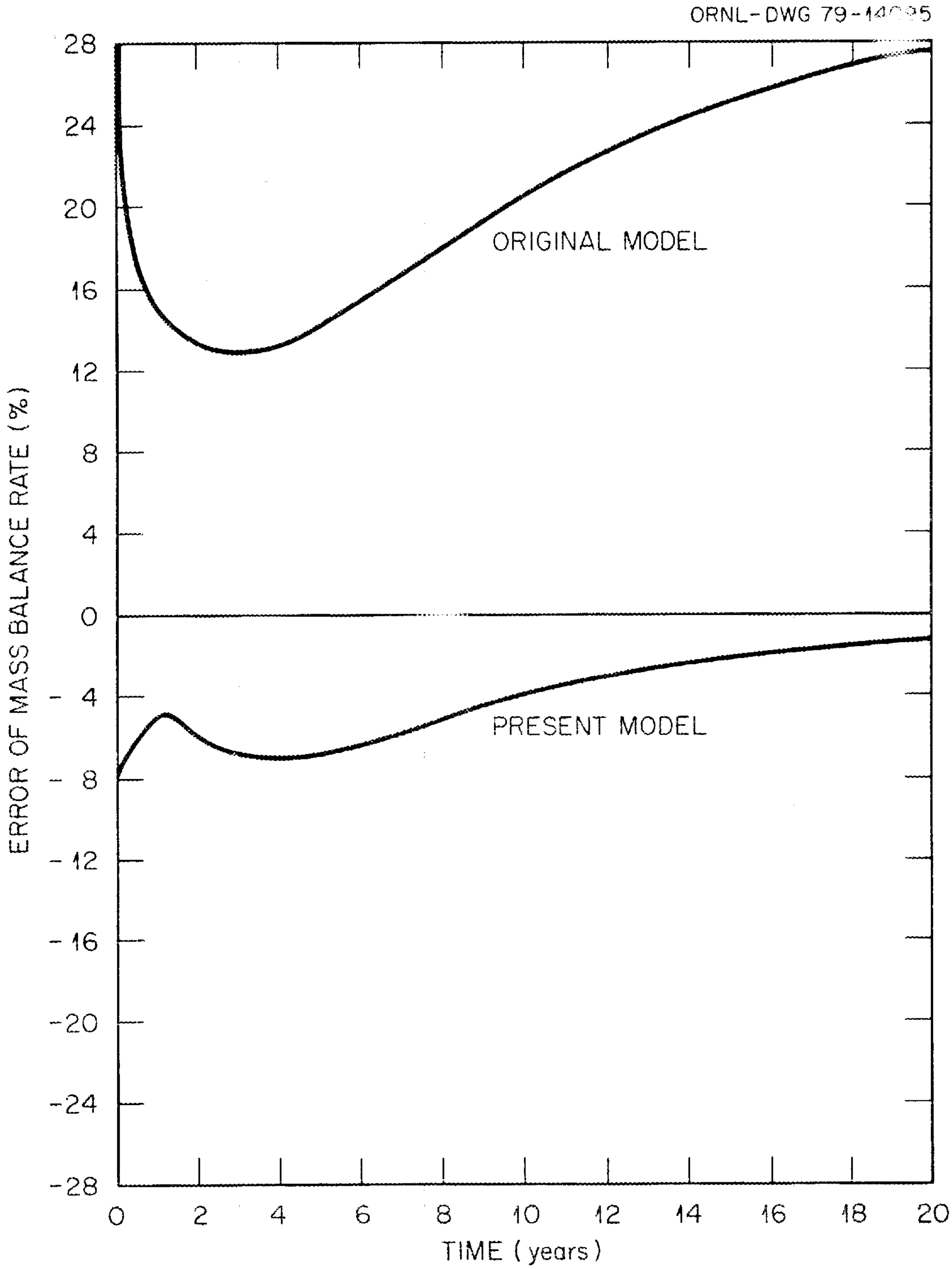

Fig. 15. Comparison of mass-balance-rate error history as simulated by numerical scheme 1 of the original and new models, respectively.

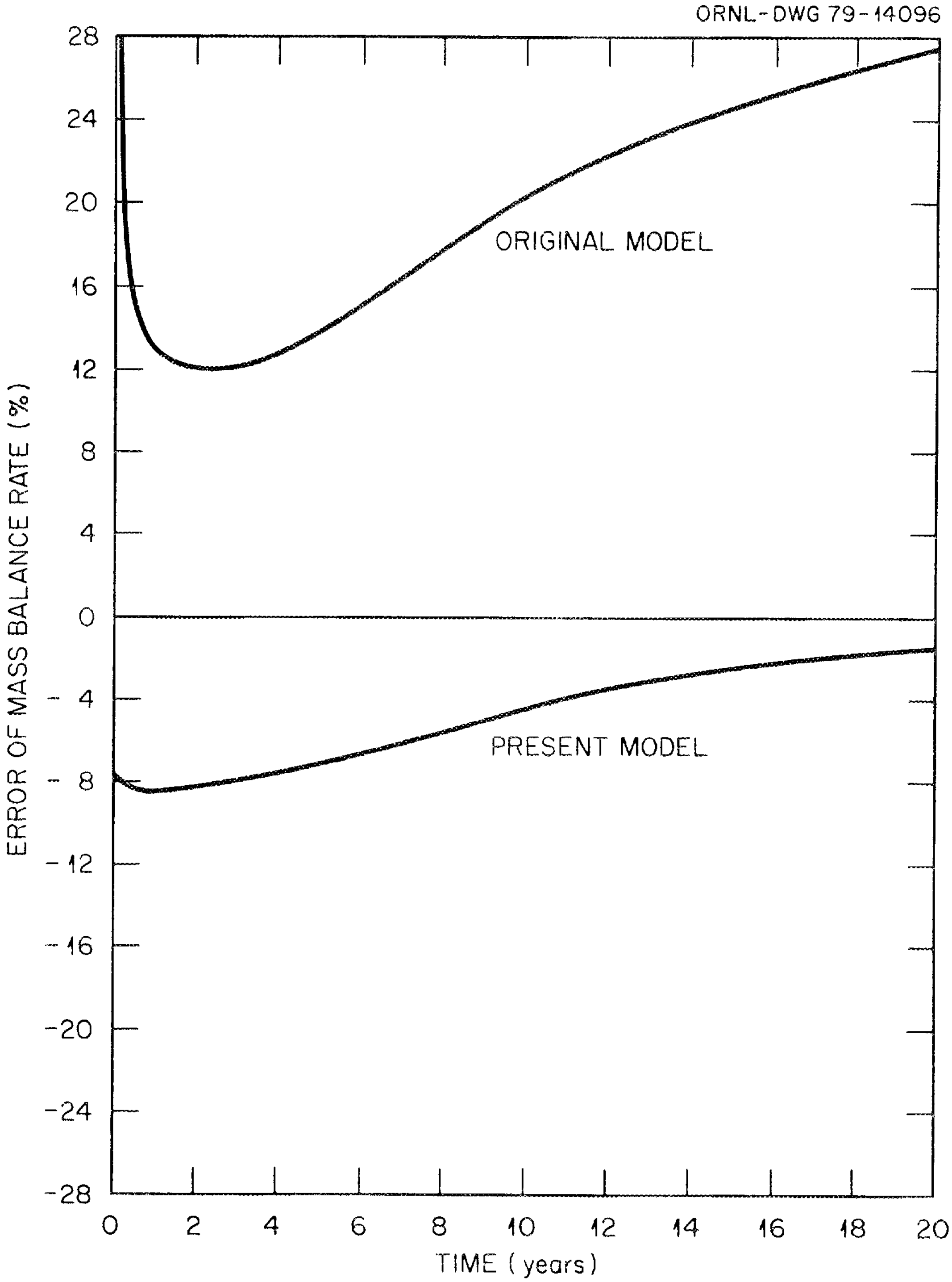

Fig. 16. Comparison of mass-balance-rate error history as simulated by numerical scheme 1 of the original and new models, respectively.

Figs. (17a) through (17d) show the plots of the concentration distribution as simulated by numerical scheme 1 at time equal to 1.01, 7.67, 14.3, and 19.3 years, respectively. They demonstrate how the pollutant is spreading away from the source with time. Figs. 18 through 28 show the concentration distribution by all other alternative numerical schemes. They show that the variation is no more than 10% among them at long simulation times . We consider those results are comparable.

To recapitulate, the modified model has accomplished the following: (1) overprediction of movement rate of contaminant in the unsaturated zone is eliminated, (2) the error of mass balance is significantly reduced, (3) a continuous waste flux field, including the nodal points and element boundaries, is produced, and (4) 10 additional alternative numerical schemes, all of which are operational and render comparable results, are provided.

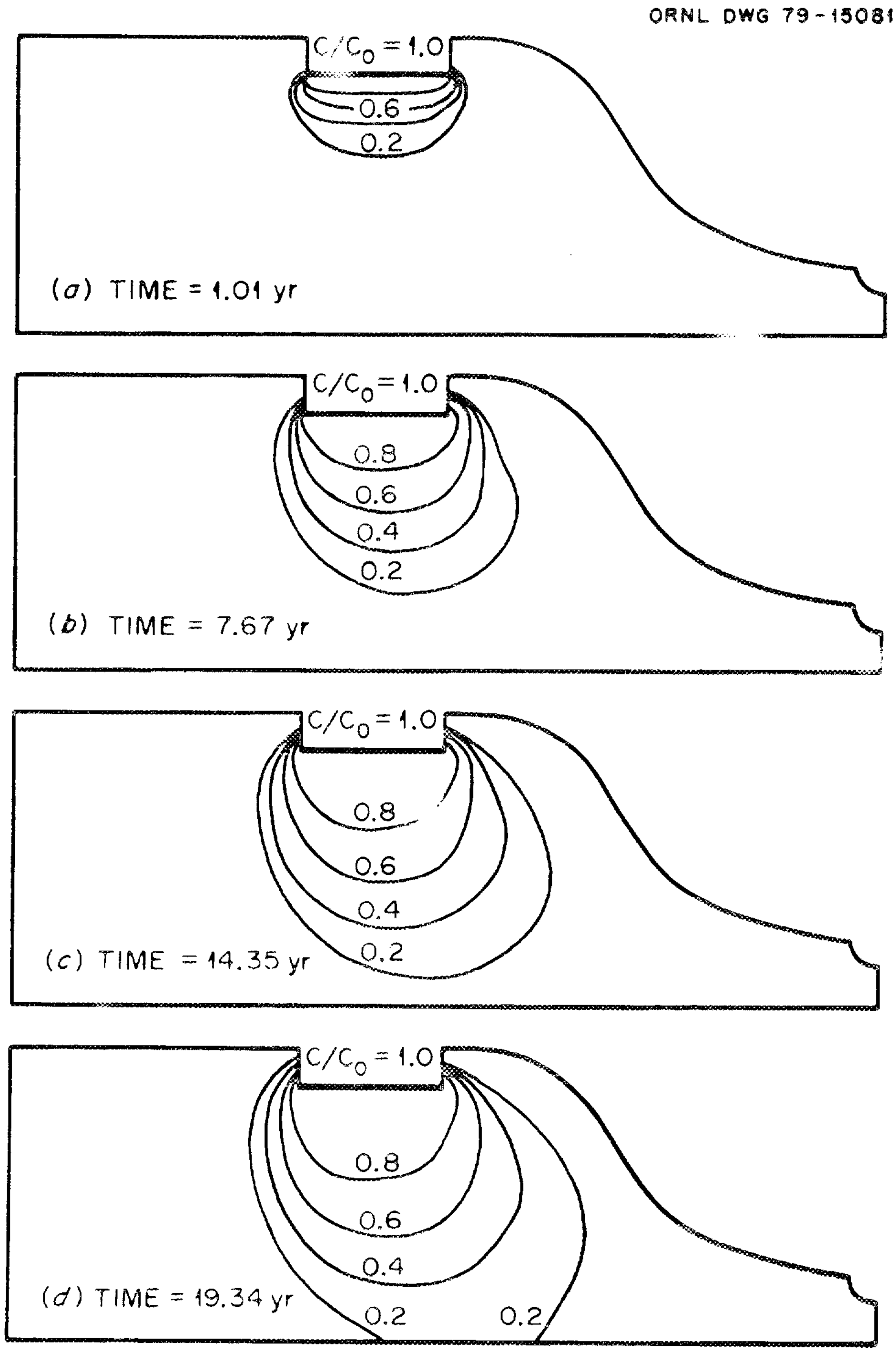

Fig. 17. Concentration distribution as simulated by numerical scheme 1 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

ORNL DWG 79-15077

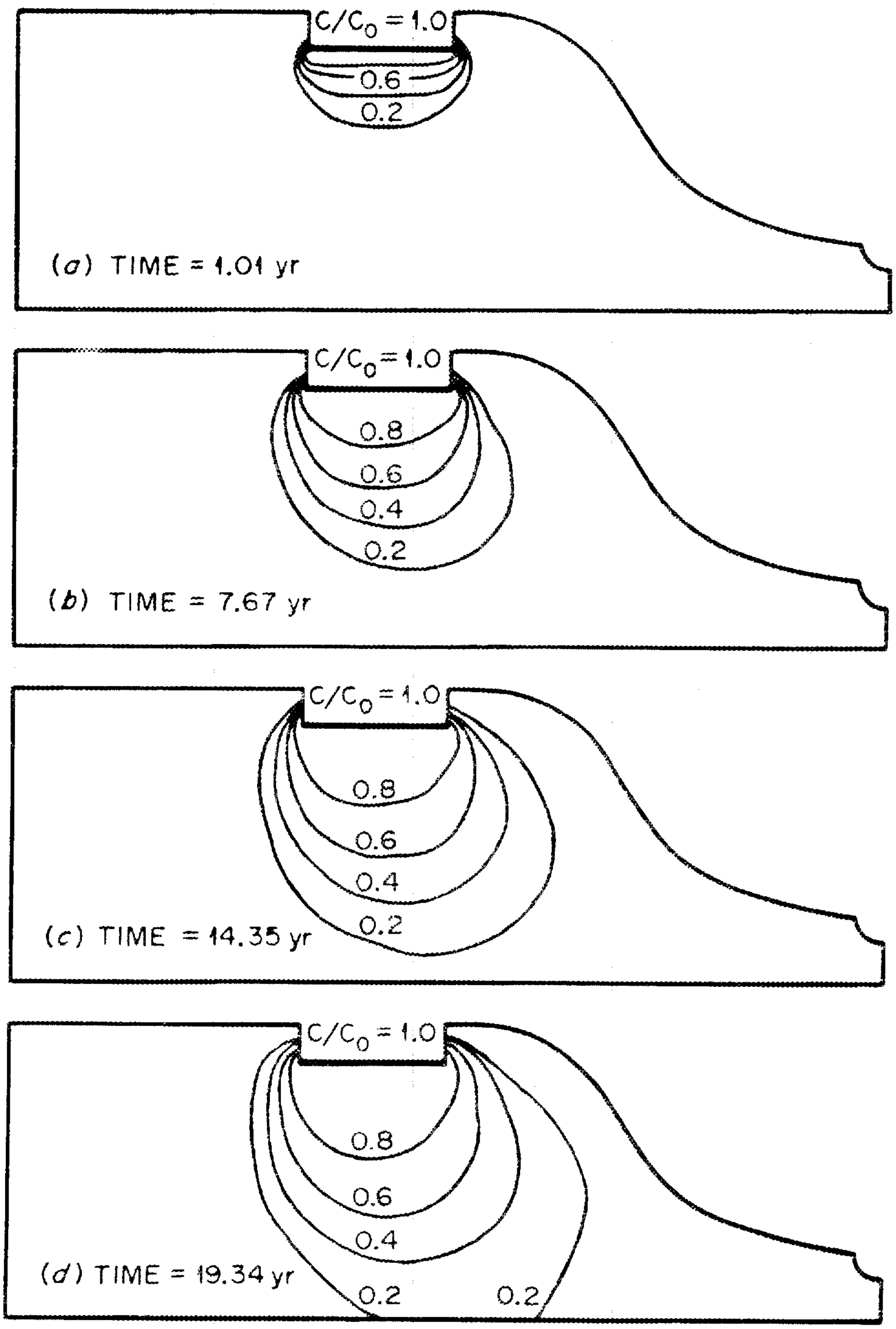

Fig. 18. Concentration distribution as simulated by numerical scheme 2 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

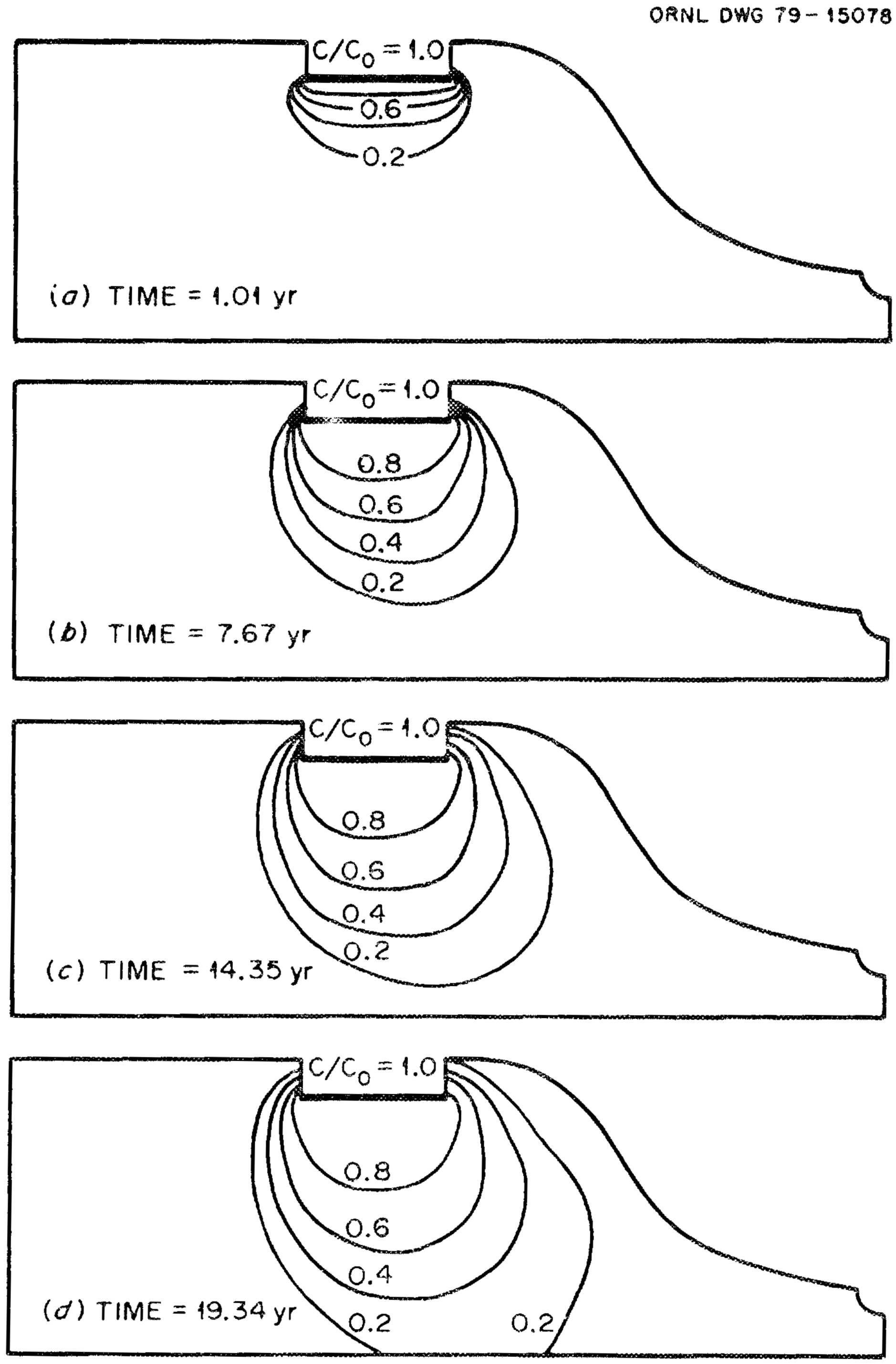

Fig. 19. Concentration distribution as simulated by numerical scheme 3 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

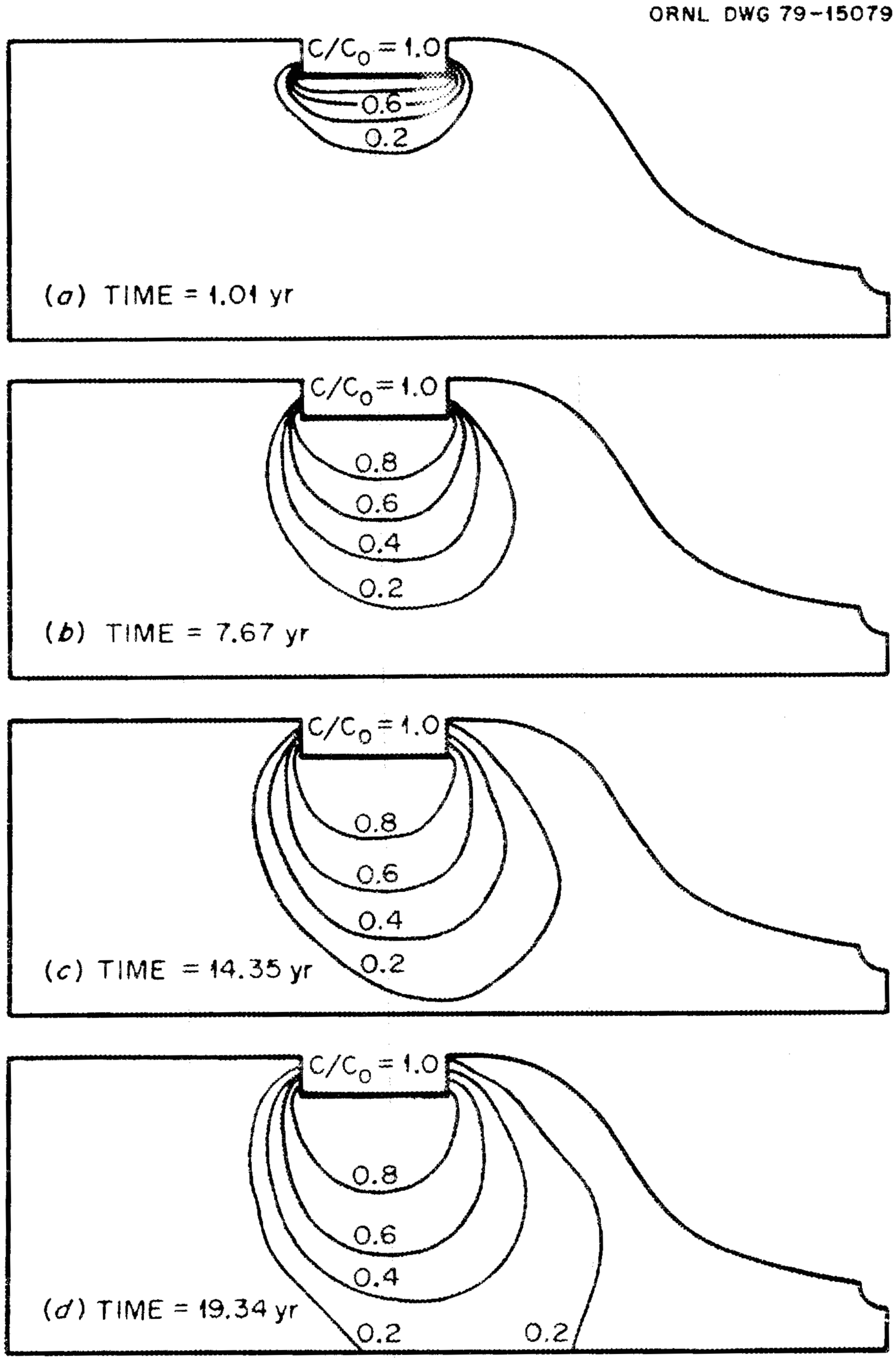

Fig. 20. Concentration distribution as simulated by numerical scheme 4 of the new model at time equal to (a) 1.01, (fb) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

ORNL DWG 79-15080

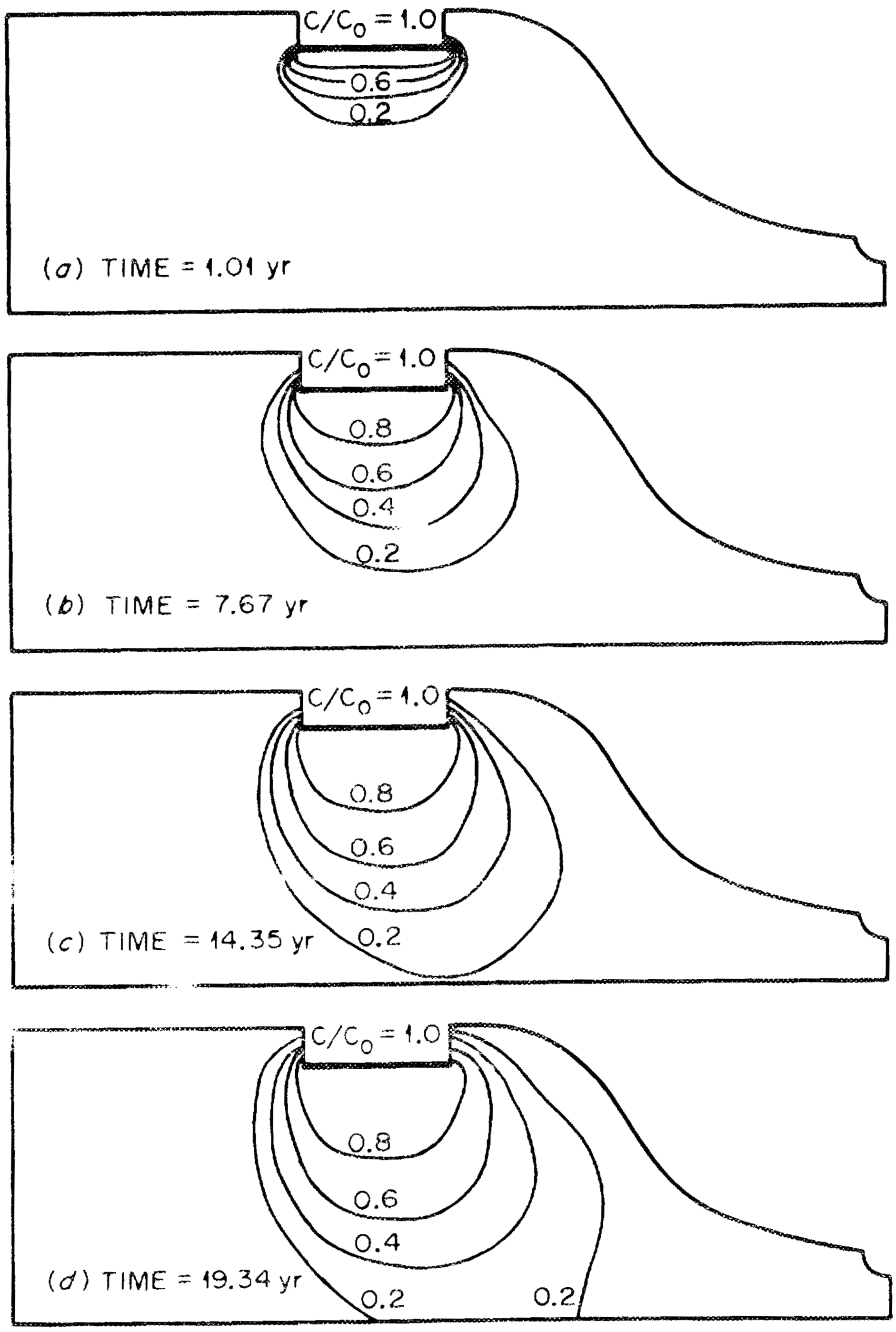

Fig. 21. Concentration distribution as simulated by numerical scheme 5 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

ORNL DWG 79-15082

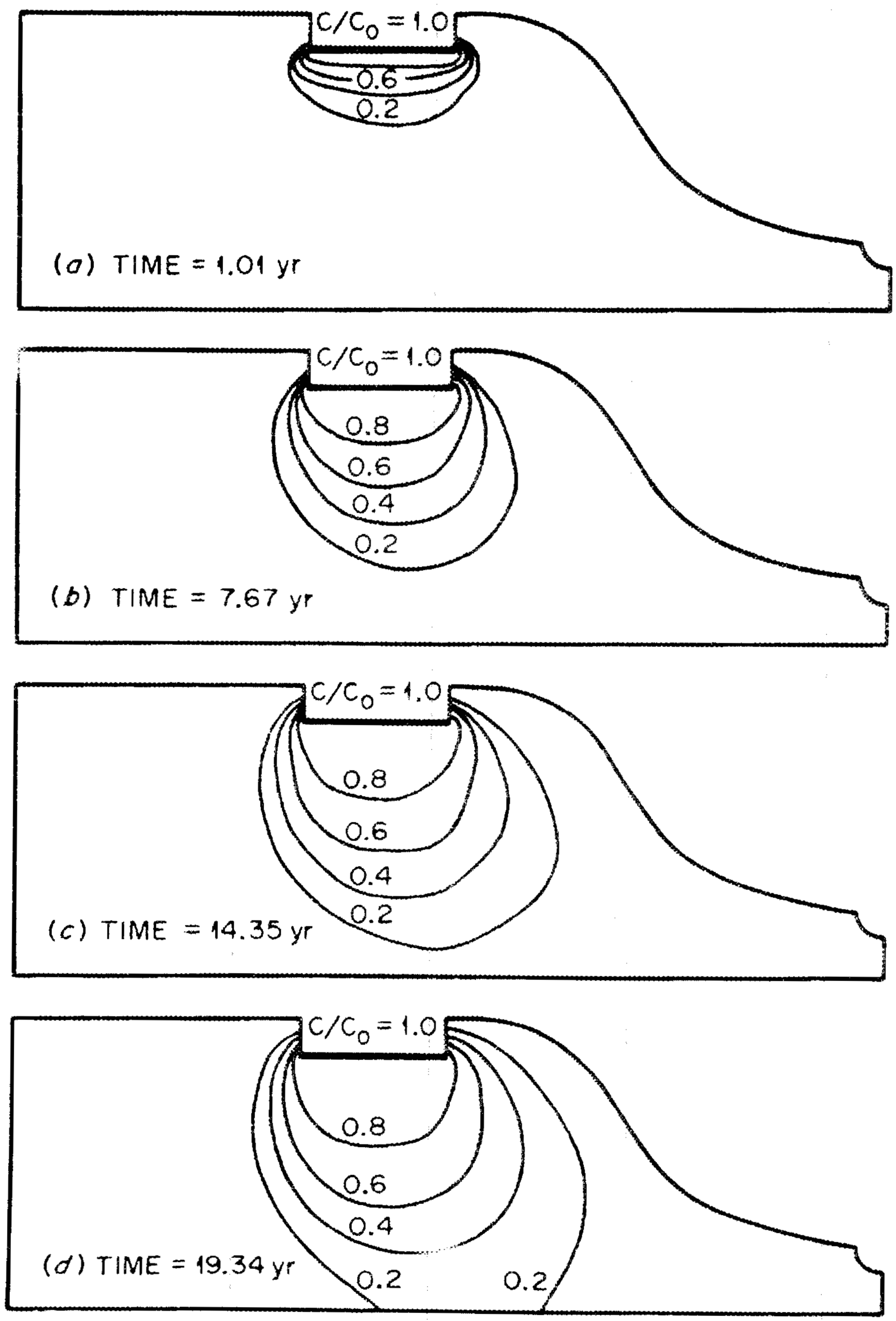

Fig. 22. Concentration distribution as simulated by numerical scheme 6 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

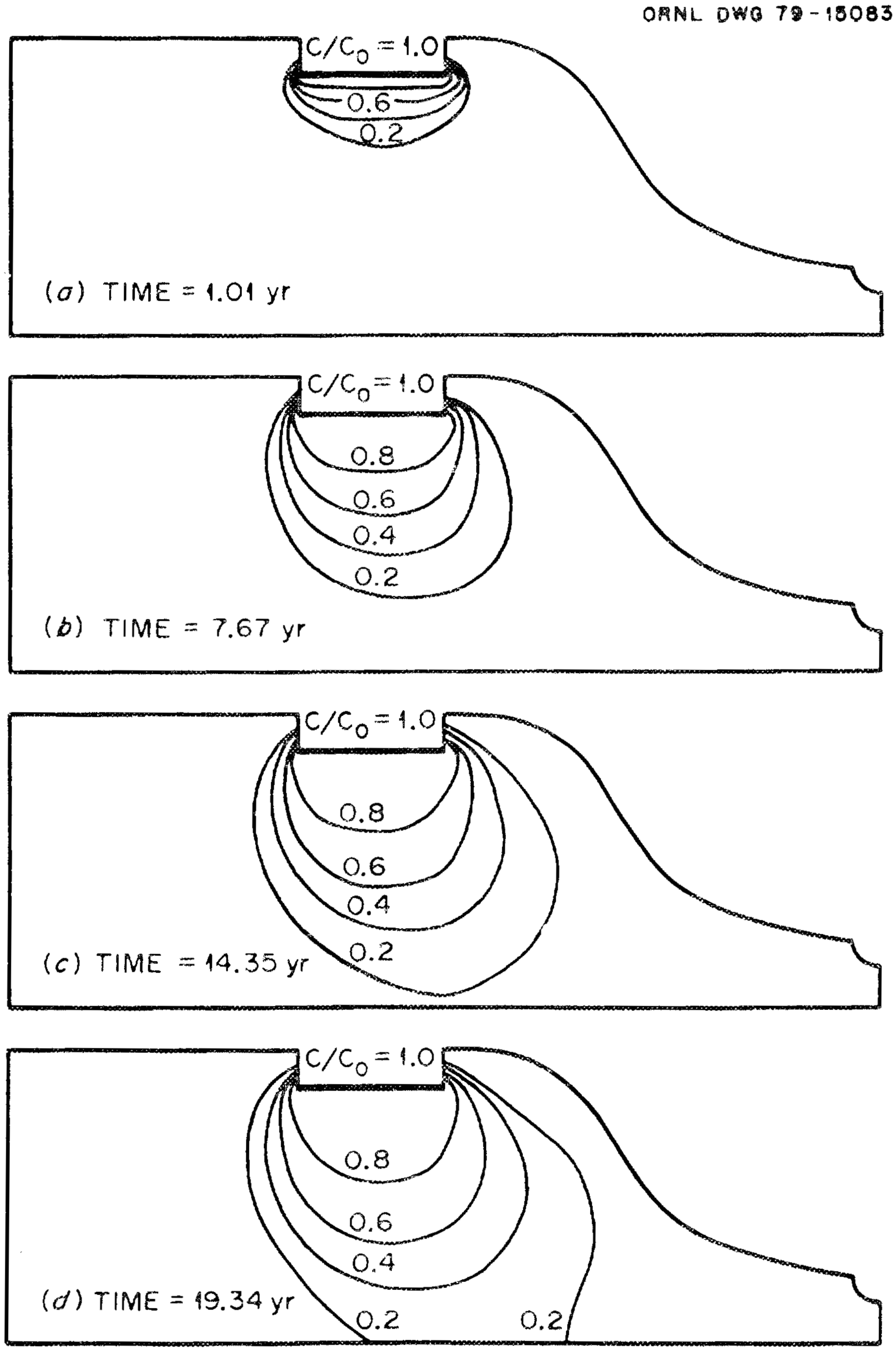

Fig. 23. Concentration distribution as simulated by numerical scheme 7 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

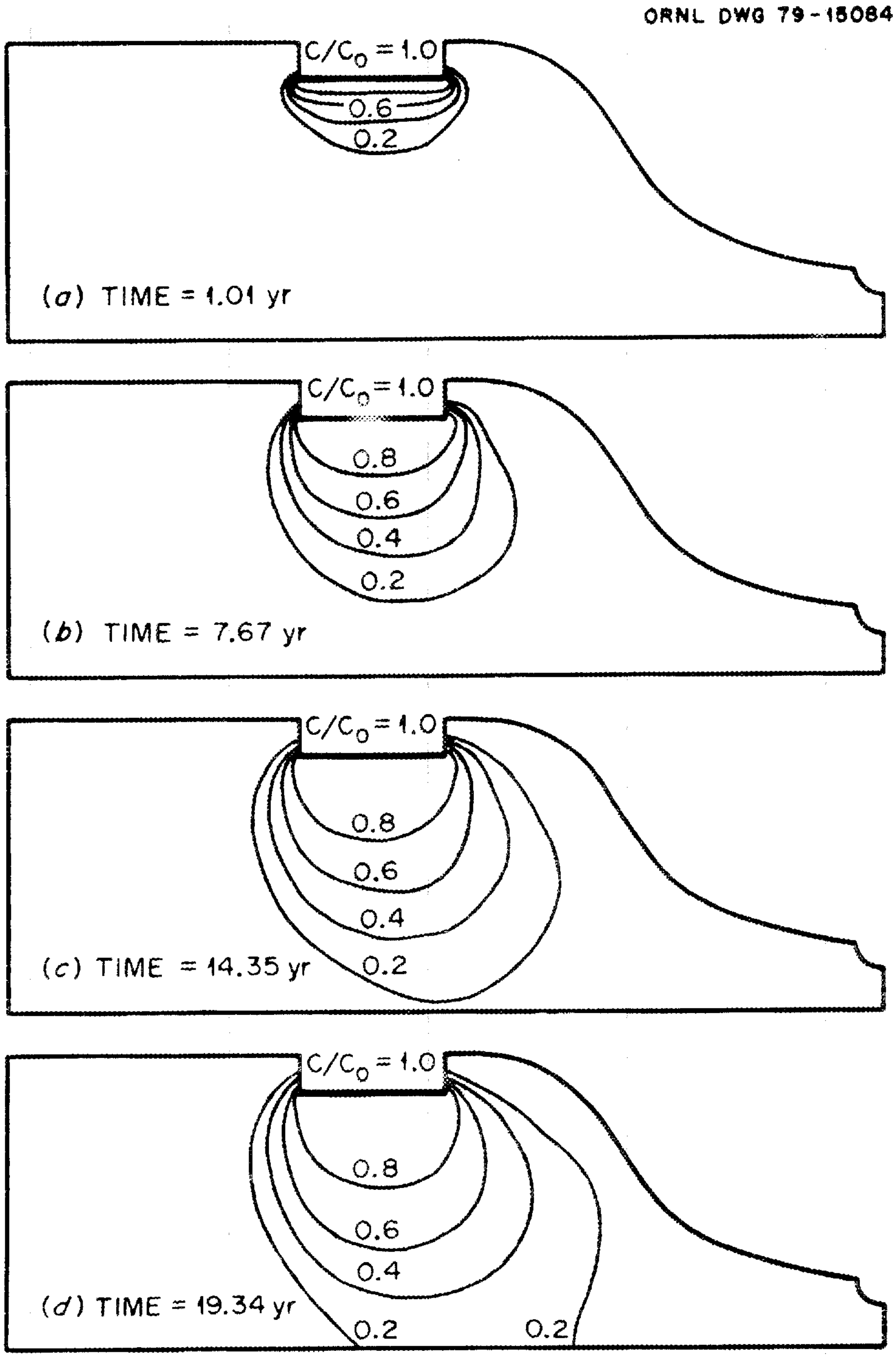

Fig. 24. Concentration distribution as simulated by numerical scheme 8 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

ORNL DWG 79-15085

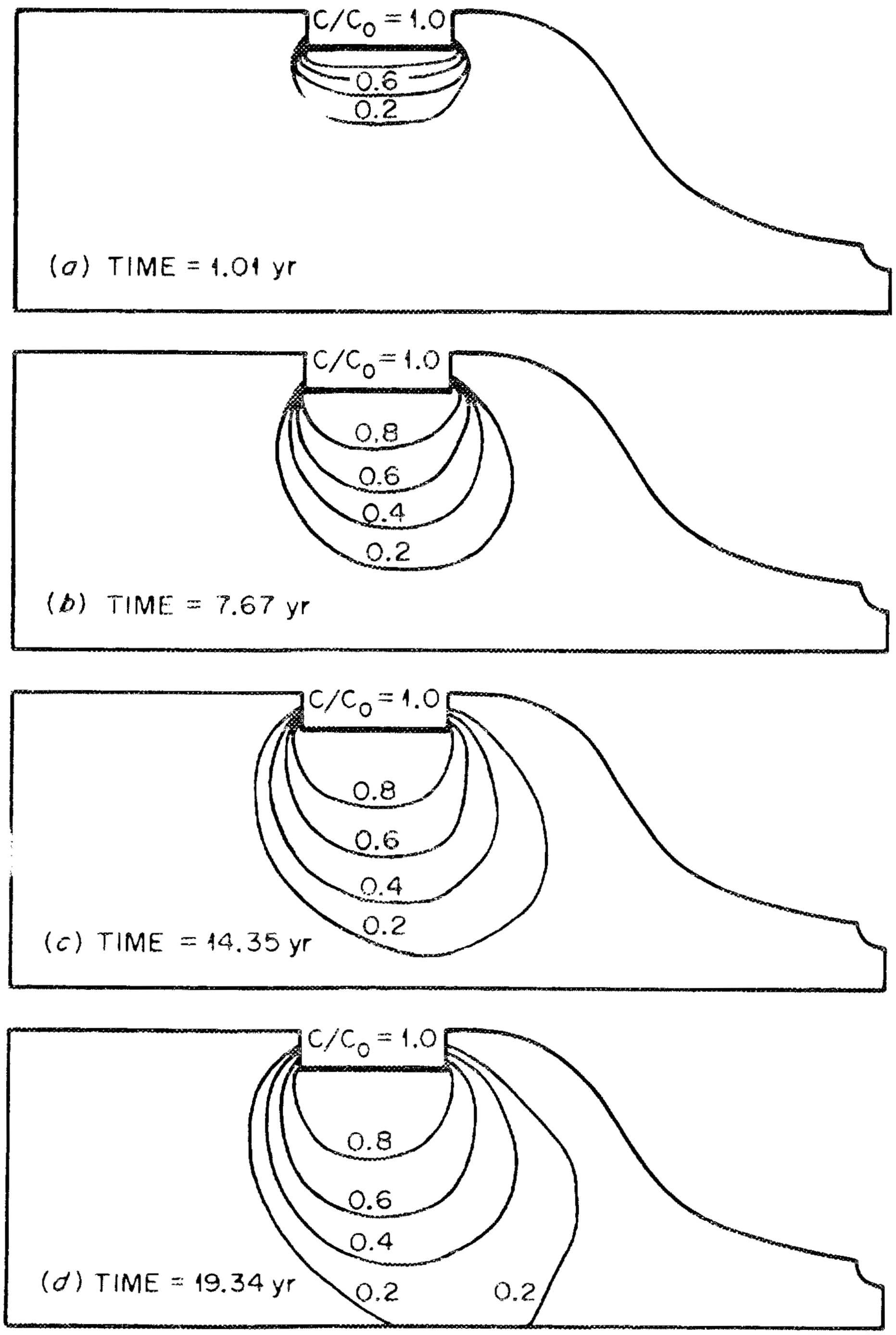

Fig. 25. Concentration distribution as simulated by numerical scheme 9 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

ORNL DWG 79-15086

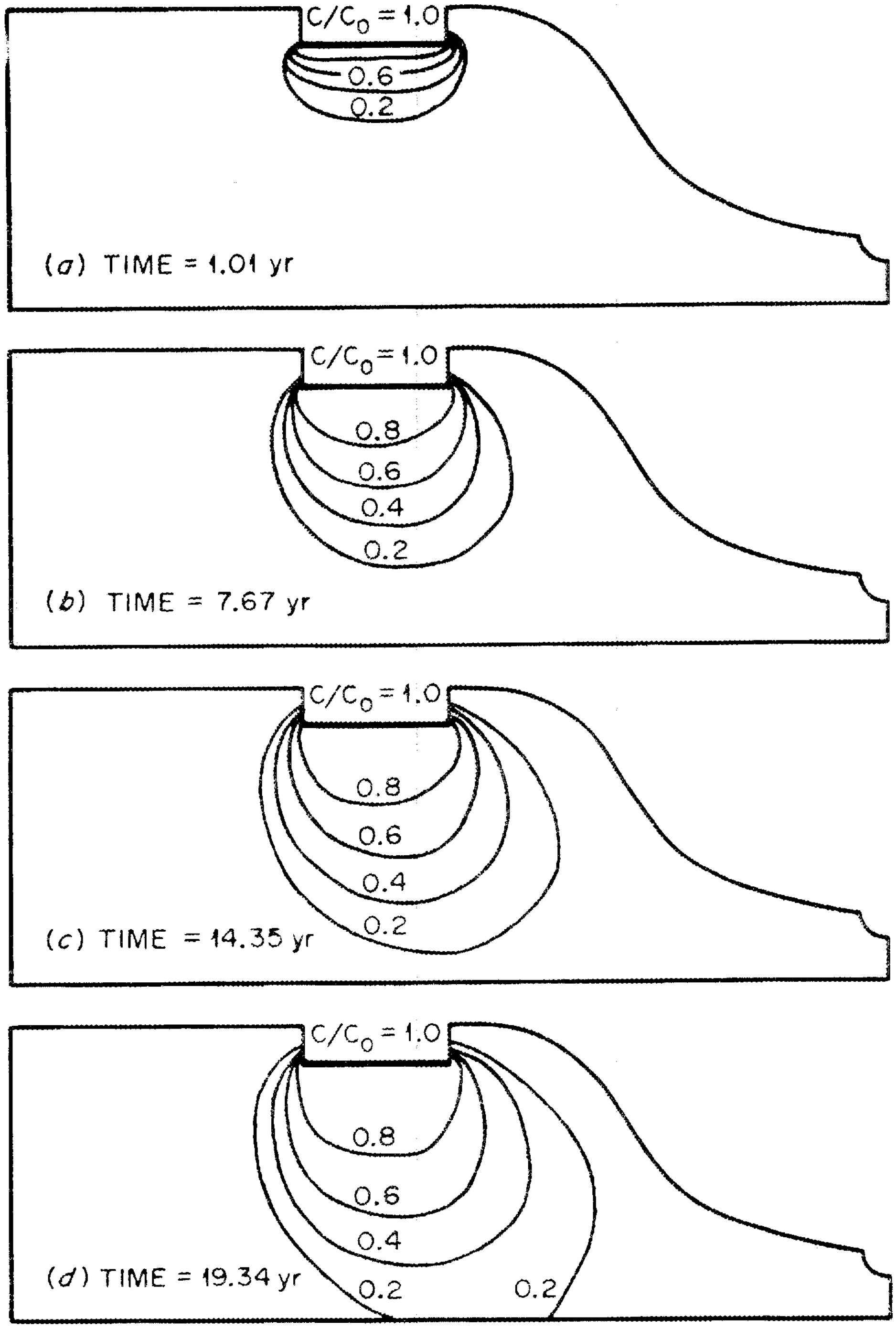

Fig. 26 Concentration distribution as simulated by numerical scheme 10 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

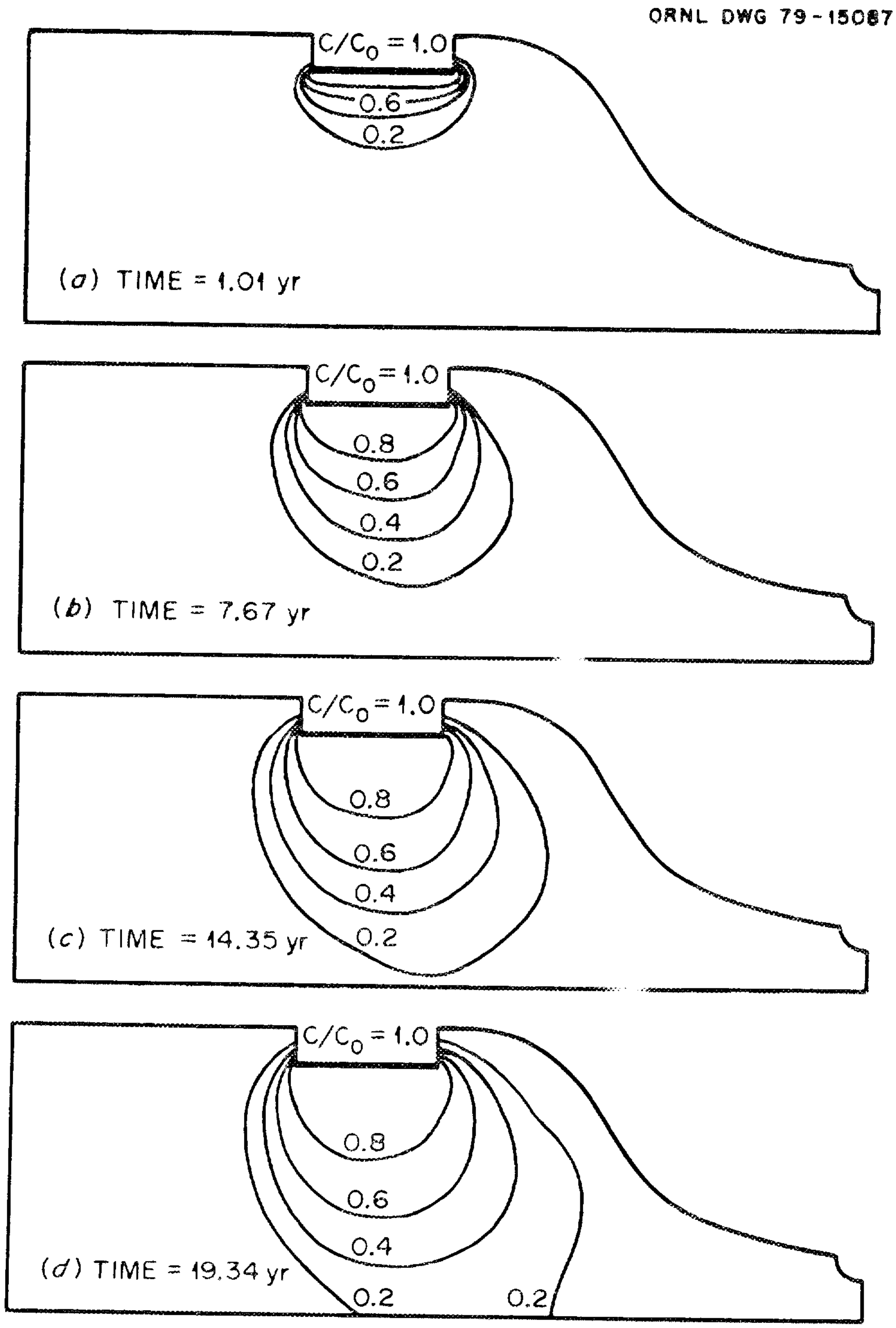

Fig. 27. Concentration distribution as simulated by numerical scheme 11 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

ORNL DWG 79-15088

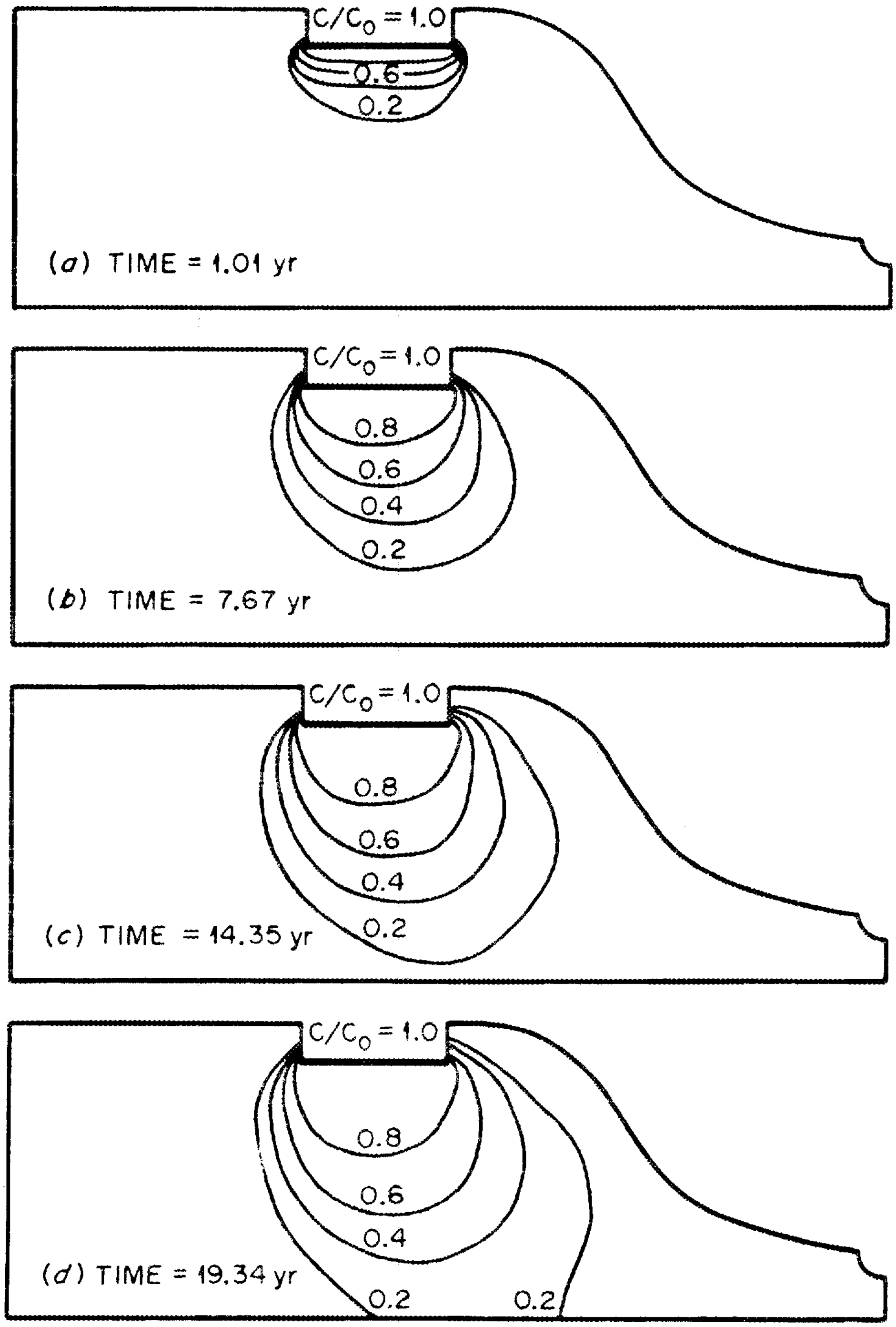

Fig. 28. Concentration distribution as simulated by numerical scheme 12 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively.

V. NOTATIONS

a_L	Longitudinal dispersivity (L)
a_T	Transverse dispersivity (L)
B_1	Domain boundary with Dirichlet conditions
B_2	Domain boundary with Neumann conditions
B_3	Domain boundary with Cauchy conditions
c	Concentration of the dissolved constituent (M/L^3)
$\dot{c}$	Time derivative of the concentration (M/L^3T)
D_m	Molecular diffusion coefficient (L^2/T)
$\{D_i\}$	Element load vector from source term
$[E_{ij}]$	Element matrix resulting Cauchy-type boundary conditions
h	Fluid pressure head (L)
J	Determinant of $[J]$
$[J]$	Jacobian matrix
K_ds	Saturated distribution coefficient
K_d	Unsaturated distribution coefficient = K_{ds}
L	An operator
M	Artificial source or sink
$[M_{ij}]$	Element mass matrix
N_i	Base functions
$\{Q_i\}$	An element column vector defined by Eq. 18
R_d	Retardation factor
R_e	Region of element e
$\{R_i\}$	An element column vector defined by Eq. 21c
$\{S_{ij}\}$	Element stiffness matrix

s	Solid-phase concentration of adsorbed constituent (M/M)
t	Time (T)
$\left[T_{ij}\right]$	Global coefficient matrix
V	Volume of the system (L^3)
V_x, V_z	Darcy velocity components in x and z directions (L/T)
V	Magnitude of Darcy velocity (L/T)
W_i	Weighting function
x	Lateral coordinate (L)
$\{x\}$	Global coordinates of the nodes of the r-th element
$\{Y\}$	Assembled load vector containing boundary fluxes and time-integration components
z	Vertical coordinate (L)
$\{z\}$	Global coordinates of the nodes of the r-th element
α_i	Upwind weighting parameter, i = 1 or 2
α'	Modified coefficient of compressibility of the medium
β_i	Upwind weighting parameter, i = 1 or 2
Δt	Time increment (T)
θ	Moisture content (L^3/L^3)
λ	Radioactive decay constant (1/T)
η	Local coordinate
ρ	Bulk density of the medium (M/L^3)
$\mathbf{T}$	Tortuosity
ξ	Local coordinate

VI. REFERENCES

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

Christie, L., D. F. Griffiths, A. R. Mitchell, and O. C. Zienkiewicz. 1976. Finite element methods for second order differential equations with significant first derivatives. Int. J. Num. Meth. Engrg. 10:1389-1396.

Clough, R. W. 1971. Analysis of structural vibrations and dynamic response. IN R. H. Gallagher, Y. Yamada, and J. T. Oden (eds.), Recent Advances in Matrix Methods of Structural Analysis and Design. University of Alabama Press, Huntsville, Alabama.

Desai, C., and J. Abel. 1972. Introduction to the Finite Element Method: A Numerical Method for Engineering Analysis. Van Nostrand Reinhold, New York.

Duguid, J. O., and M. Reeves. 1976. Material transport in porous media: A finite-element Galerkin model. ORNL-4928. Oak Ridge National Laboratory, Oak Ridge, Tennessee. 198 pp.

Eagleson, P. S. 1970. Dynamic Hydrology. McGraw-Hill Book Company, New York. 462 pp.

Ergatoudis, J., B. M. Irons, and O. C. Zienkiewicz. 1968. Curved isoparametric, quadrilateral elements for finite-element analysis. Int. J. Solids Struct. 4:31-42.

Finlayson, B. A. 1972. Method of Weighted Residuals and Variational Principles with Application in Fluid Mechanics, Heat and Mass Transfer. Academic Press, New York. 412 pp.

Gureghian, A. B., D. S. Ward, and R. W. Cleary. A finite-element model for the migration of leachate from a sanitary landfill in Long Island, New York. Water Resour. Bull. (submitted).

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

Heinrich, J. C., P. S. Huyakorn, O. C. Zienkiewicz, and A. R. Mitchell. 1977. An upwind finite element scheme for two-dimensional convective transport equations. Int. J. Num. Meth. Engrg. 11:131-143.

Huyakorn, P. S., and G. F. Pinder. 1977. A pressure enthalpy finite element model for simulating hydrothermal reservoirs. Paper presented at the Second International Symposium on Computer Methods for Partial Differential Equations. Lehigh University, Bethlehem, Pennsylvania, June 22-24, 1977.

Huyakorn, P. S., and K. Nilkuha. 1979. Solution of transient transport equation using an upstream finite element scheme. Appl. Math. Model. Vol. 3. IPC Business Press, February 1979.

Kantorovich, L. V., and V. I. Krylov. 1964. Approximate method of higher analysis. Noordhoff, Netherlands. 681 pp.

Kokotov, Y. A., and R. F. Popova. 1962. Sorption of long-lived fission products by soils and argillaceous minerals. Radiokhimiya 4(3):328-336.

Pinder, G. 1973. A Galerkin-finite element simulation of groundwater contamination on Long Island, New York. Water Resour. Res. 9:1657-1670.

Prokhorov, V. M., and T. Y. Chaai. 1963. Diffusion of cesium-137 into the soil. Radiokhimiga 5(5):639-642.

Reeves, M., and J. O. Duguid. 1975. Water movement through saturated-unsaturated porous media: A finite-element Galerkin model. ORNL-4927. Oak Ridge National Laboratory, Oak Ridge, Tennessee. 232 pp.

Van Genuchten, M. Th., G. F. Pinder, and W. F. Sankin. 1977. Modeling of leachate and soil interactions in an aquifer. pp. 95-103. IN S. K. Banerji (ed.), Leachate in Landfills. EPA-600/9-77-026. Office of Research and Development, U. S. Environmental Protection Agency, Cincinnati, Ohio.

Van Genuchten, M. Th., and G. F. Pinder. 1978. Mass trasnport in saturated-unsaturated media. II. Two-dimensional solutions. Research Report 178-WR-12. Water Resources Program, Department of Civil Engineering, Princeton University, Princeton, New Jersey.

Wang, J. D., and J. J. Connor. 1975. Mathematical modeling of near coastal circulation. Report No. MITSG 75-13. Massachussetts Instutute of Technology, Cambridge, Massachussetts.

Yeh, G. T., and D. S. Ward. 1980. FEMWATER: A Finite-Element Model of WATER Flow through saturated-unsaturated porous media. ORNL-5567. Oak Ridge National Laboratory, Oak Ridge, Tennessee.

Zienkiewicz, O. C. 1977. The Finite Element Method in Engineering Science. McGraw-Hill, New York. 521 pp.

Zienkiewicz, O. C., and C. J. Parekh. 1970. Transient field problems, two-dimensional and three-dimensional analysis by isoparametric finite elements. Int. J. Num. Meth. Engrg. 2:61-71.

APPENDIX A

DATA INPUT GUIDE

APPENDIX A
DATA INPUT GUIDE

Data Set 1 - Problem Identification. FORMAT(I5,9A8)

One card is required per job.

Card	Variable	Column	Format	Definition
1	NPROB	1-5	I5	Problem number
	TITLE	6-77	9A8	Array for the title of the problem

Data Set 2 - Integer Control Parameters. FORMAT(16I5)

One card is needed per job

Card	Variable	Column	Format	Definition
1	NNP	1-5	I5	Number of nodal points
	NEL	6-10	I5	Number of elements
	NMAT	11-15	I5	Number of materials
	NCM	16-20	5	Number of elements with corrected material properties
	NTI	21-25	I5	Number of time increments
	NBC	26-30	I5	Number of Dirichlet nodes
	NST	31-35	I5	Number of surface term element-sides with Cauchy (mixed) boundary condition
	NRSEL	36-40	I5	Number of rainfall-seepage element-sides. Either Neumann or Cauchy boundary conditions can be applied depending on if the flow is out from or into the region.
	KVI	41-45	I5	Flow field input control, 1 = steady-state flow, 2 = transient-state flow field
	KSTR	46-50	I5	Auxilary storage control; 0 = no storage, 1 = stored on unit 2
	KSS	51-55	I5	Steady state control; 0 = steady state solution, 1 = transient state solution
	NMPPM	56-60	I5	Number of material properties per material

Data Set 2 (continued)

Card	Variable	Column	Format	Definition
	IWET	61-65	I5	Weighting function control; 0 = Galerkin, 1 = upstream weighting
	ILUMP	66-70	I5	Matrix lumping control, 0 = no lumping, 1 = lumping
	IMID	71-75	I5	Mid-difference contr[illegible] 0 = not mid-differenc[illegible], 1 = yes

Data Set 3 - Basic Real Parameters. FORMAT(8F10.0)

One card per problem is required.

Card	Variable	Column	Format	Definition
1	DELT	1-10	F10.0	Initial time step size, (T)
	CHNG	11-20	F10.0	Multiplies for increasing time step size
	DELMAX	21-30	F10.0	Maximum time step size allowed, (T)
	TMAX	1-40	F10.0	Value of maximum simulation time, (T)
	W	41-50	F10.0	Time derivative weighting; 0.5 = for Crank-Nicolson, 1.0 = for backward

Data Set 4 - Printer Output Control. FORMAT(80I1)

The number of cards depends on the number of time increments, NTI. Total number of cards = (NTI + 1)/80 + 1

Card	Variable	Column	Format	Definition
1	KPRO	1	I1	Printer control for steady-state and initial conditions; 0 = print nothing, 1 = print flow rate, 2 = also concentration, 3 = also material flux,
	KPR(1)	2	I1	
	KPR(2)	3	I1	Similar to KPRO but as a function of time index ITM
	.	.	.	
	.	.	.	
	.	.	.	
	KPR(ITM)		I1	

Data Set 4A - Auxiliary storage control. FORMAT(80I1)

Total number of cards required is same as Data Set 4

Card	Variable	Column	Format	Definition
1	KDSKO	1	I1	Auxiliary storage control for steady-state and initial conditions, 0 = no auxiliary storage, 1 = storage on Unit 1
	KDSK(1)	2	I2	
	.	.	.	
	.	.	.	Similar to KDSKO as a function of time index ITM
	.	.	.	
	KDSK(NTI)			

Data Set 5 - Material Properties. FORMAT(8F10.0)

A total of NMAT groups of cards are required, one group for each material. The number of cards in each group depends on NMPPM

Card	Variable	Column	Format	Definition
Group J	PROP(J,1)	1-10	F10.0	Distribution coefficient for material J, (L^3/M)
	PROP(J,2)	11-20	F10.0	Bulk density for material J, (M/L^3)
	PROP(J,3)	21-30	F10.0	Longitudinal dispersivity for material J, (L)
	PROP(J,4)	31-40	F10.0	Transverse dispersivity for material J, (L)
	PROP(J,5)	41-50	F10.0	Decay constant in material J, (T^{-1})
	PROP(J,6)	51-60	F10.0	Porosity for material J
	PROP(J,7)	61-70	F10.0	Modified coefficient of compressibility of media J, (L^{-1})
	.			.
	.			.
	PROP (J,NMPPM)			.

Data Set 6 - Geometrical Data

This data set is read in via Logical Unit 1. It includes the nodal-point and element definitions and boundary-side information. No card is required. One should refer to the sequence number DATA 335 to DATA 360 of the subroutine DATAIN in APPENDIX C

```
REWIND 1
READ(1) (TITLEM(I),I=1,9),NPROBM,NNP,NEL,NBN,NBEL,NTIM
READ(1) (X(NP),NP=1,NNP),(Z(NP),NP=1,NNP),((IE(M,IQ),M=1,NEL),
IQ=  1,4),(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),
(NPB(NP),NP=1,   NBN)
```

Data Set 7 - Material Correction. FORMAT(16I5)

This data set is required only if NCM 0. Normally, one card is required per material change. However, in those cases where number of the affected elements ranged from a lower limit of MI to an upper limit of MK with an increment of MINC, automatic correction may be used. Fields MK and MINC are left blank if the automatic generation facility is not used.

Card	Variable	Column	Format	Definition
I	MI	1-5	I5	Material correction element number
	MTYP	6-10	I5	Type of material correction element MI
	MK	11-15	I5	Upper limit of automatic correction
	MINC	16-20	I5	Element Increment of automatic correction (MK = 0, MINC = 0 for no automatically generated correction)

Data Set 8 - Initial Conditions. FORMAT(I5,5X, F10.0)

In the most general case there is one card per node, i.e., a total of NNP cards are needed. Frequently, however, groups of neighboring nodal points NJ have identical values R(NJ). If a gap is recognized in the input sequence of nodal numbers, the initial concentration are assumed to be identical to the concentration at the lower bound of the gap.

Card	Variable	Column	Format	Definition
NJ	NJ	1-5	I5	Node number
	RP(NJ)	11-20	F10.0	Initial concentration for Node NJ

Data Set 9 - Dirichlet Boundary Conditions. FORMAT(2I5,F10.0)

This data set is required only if NBC > 0. If automatic generation is not used (NPINC = 0), NBC cards are required. If NPINC > 0, automatic generation proceeds in the same manner as Data Set 8

Card	Variable	Column	Format	Definition
NPN	NPN	1-5	I5	Dirichlet Node number
	NPINC	6-10	I5	Automatic generation increment to NPN
	BB	11-20	F10.0	Dirichlet concentration at node NPN

Data Set 10 - Cauchy Boundary Condition. FORMAT(3I5,5X.2F10.0)

This data set is required only if NST > 0. If KINC = 0, automatic generation will not be performed and NST cards are required. Each NST card represents an element-boundary side. If KINC > 0, then automatic generation will be made.

Card	Variable	Column	Format	Definition
MPP	NI	1-5	I5	Cauchy flux node number at one end of an element-boundary side
	NJ	6-10	I5	Cauchy flux node number at other end of an element-boundary side
	KINC	11-15	I5	Automatic generation increment for NI and NJ
	EI	21-30	F10.0	Dot product of flux at NI with outwardly directed unit vector normal to element-boundary side (NI, NJ)
	EJ	31-40	F10.0	Similar to EI

Data Set 11 - Neumann Boundary Condition. FORMAT(16I5)

This data set is required only if NRSEL > 0. The data set actually contains the rainfall-seepage element-boundary sides. If the flow is directed out from the region, the side is a Neumann boundary with concentration dependent condition and is applied as such If the flow is directed into the region, the side is a Cauchy boundary with total flux equal to zero and thus the application of boundary condition is not necessary. This is why this data set is termed Neumann boundary condition data. Typically, one card is required for each element-boundary side. However, if KINC > 0, automatic generation may be made.

Card	Variable	Column	Format	Definition
MPP	NRSE(MP)	1-5	I5	Element number of the element-boundary side MP
	IS(MP,1)	6-10	I5	First node number of the element-boundary side, MP
	IS(MP,2)	11-15	I5	Second node number of the element-boundary side MP
	KINC	16-20	I5	Automatic generation increment for NRSE and IS

Data Set 12 - Initial Flow Variable.

This data set is read in via Logical Unit 1. The variables include the time, TIMEM, pressure head, H, moisture-content, TH, and Darcy's velocity components, VX and VZ. One is referred to the sequence number GM2D 390 and GM2D 395 in the subroutine GM2DXZ of APPENDIX C.

```
READ(1) TIMEM, (H(NP),NP=1,NNP),(HT(NP),NP=1,NNP),((TH(M,IQ),
M=1, NEL),IQ=1,4),(VX(NP),NP=1,NNP),(VZ(NP),NP=1,NNP)
```

Data Set 13 - Transient Flow Variables

This data set is required only if KVI = 2. It is read via Logical Unit 1. The variables included in this data is the same as Data Set 12. It is also read in the same manner. One is referred to the sequence number GM2D 660 and GM2D 665 in the subroutine GM2DXZ of the APPENDIX C.

APPENDIX B

SAMPLE INPUT OF THE SEEPAGE POND PROBLEM

APPENDIX B

SAMPLE INPUT OF THE SEEPAGE POND PROBLEM

```
C
C
C ------- DATA SET 1: PROBLEM IDENTIFICATION
C
 1273  WASTE TRANSPORT FROM SEEPAGE POND INTO THE AQUIFER                  CARD 001
C
C
C ------- DATA SET 2: INTEGER CONTROL PARAMETERS
C
  595  528    1    0  250    7    0   38    1    1    1    9    0    0    0 CARD 002
C
C
C ------- DATA SET 3: BASIC REAL NUMBERS
C
300.       .3        5256000.  630720000. 0.5                              CARD 003
C
C
C ------- DATA SET 4: PRINTER OUTPUT CONTROL
C
511                  1                   1         2         1             CARD 004
1                    2                   1                   1         2   CARD 005
1                    1                    2                  1             CARD 006
1         2                                                                CARD 007
C
C
C ------- DATA SET 4A: AUXILIARY STORAGE OUTPUT CONTROL
C
1                    1                   1                   1             CARD 008
1                    1                   1                   1         1   CARD 009
                                                                           CARD 010
          1                                                                CARD 011
C
C
C ------- DATA SET 5: MATERIAL PROPERTIES
C
100.      1.75      2130.     427.      0.         .3       0.        0.   CARD 012
0.                                                                         CARD 013
C
C
C ------- DATA SET 6: GEOMETRICAL DATA IS INPUT VIA LOGICAL UNIT 1
C
C
C
C ------- DATA SET 7: MATERIAL CORRECTION IS NOT REQUIRED SINCE NCM=0
C
C
C
C ------- DATA SET 8: INITIAL CONDITIONS
C
    1     0.                                                               CARD 014
```

APPENDIX B (continued)

```
  595     0.                                                        CARD 015
C
C
C ------- DATA SET 9: DIRICHLET BOUNDARY CONDITIONS
C
  152     0 1.                                                      CARD 016
  164     0 1.                                                      CARD 017
  204     9 1.                                                      CARD 018
C
C
C ------- DATA SET 10: CAUCHY BOUNDARY CONDITIONS ARE NOT NEEDED
C ------- SINCE NST=0
C
C
C
C ------- DATA SET 11: NEUMANN BOUNDARY CONDITIONS
C
  185  208  220                                                     CARD 019
  515  568  580    1                                                CARD 020
  515  579  580    0                                                CARD 021
  514  578  579    0                                                CARD 022
  513  577  578    0                                                CARD 023
  512  576  577    0                                                CARD 024
  522  576  588    0                                                CARD 025
  522  587  588    0                                                CARD 026
  528  587  595    0                                                CARD 027
C
C
C ------- DATA SET 12: INITIAL FLOW VARIABLES ARE READ IN VIA LOGICAL UNIT 1
C
C
C
C ------- DATA SET 13: TRANSIENT FLOW VARIABLES ARE READ IN VIA LOGICAL UNIT 1
C
C
C
C ------- FINALLY A BLANK CARD TO END THE JOB
                                                                    CARD 028

IHC002I STOP       0
```

APPENDIX C

LISTING OF FORTRAN IV SOURCE PROGRAM

APPENDIX C
LISTING OF FORTRAN IV SOURCE PROGRAM

```
C
C
C    THIS COMPUTER CODE IS CONTAINED IN THE FOLLOWING REPORT:
C
C    YEH, G. T. AND D. S. WARD, 1979.  "FEMWASTE: A FINITE-ELEMENT
C     MODEL OF WASTE TRANSPORT THROUGH SATURATED-UNSATURATED POROUS MEDIA"
C    ORNL-5601, OAK RIDGE NATIONAL LABORATORY, OAK RIDGE, TN 37830
C
C    A SLIGHTLY UPDATED VERSION IS CONTAINED IN "FECWASTE: USER'S MANUAL
C    OF A FINITE ELEMENT CODE FOR SIMULATING WASTE TRANSPORT THROUGH
C    SATURATED-UNSATURATED POROUS MEDIA, ORNL/TM ----"
C
C    FOR ANY QUESTION, PLEASE CONTACT DR. G. T. YEH AT (615) 574-7285
C
C
C    ADDITIONAL REFERENCES IS:
C
C    DUGUID, J. AND M. REEVES, 1976. "MATERIAL TRANSPORT THROUGH
C    SATURATED-UNSATURATED POROUS MEDIA: A GALERKIN FINITE ELEMENT MODEL",
C    ORNL 4928, OAK RIDGE NATIONAL LABORATORY, OAK RIDGE, TENESSEE 37830
C
C
C    MAIN PROGRAM
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 PMAT
C
      DIMENSION X(595),Z(595),I[illegible]28,5)
      DIMENSION C(595,27),R(595),RP(595),WETAB(528,4)
      DIMENSION FX(595),FZ(595)
C
      DIMENSION VX(595),VXP(595),VZ(595),VZP(595),H(595),HP(595),
     > DH(595),HT(595),TH(528,4),THP(528,4),DTH(528,4)
C
      DIMENSION DLB(199),DCOSXB(199),DCOSZB(199),BFLX(200),BFLXP(200),
     > NBE(199),ISB(199,4),NPB(200)
C
      DIMENSION DL(99),DCOSX(99),DCOSZ(99),NRSE(99),IS(99,4),NPRS(100)
      DIMENSION NPN(30),BB(30),NPST(40),DP(40)
      DIMENSION PMAT(3,9),PROP(3,9)
      DIMENSION KPR(500),KDSK(500)
      DIMENSION FRATE(10),FLOW(10),TFLOW(10)
C
      COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
      COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTH,[illegible]T
      COMMON /CONTRL/ NTI,NSTR,KSTR,KPRO,KDSKO,KSS,KVI
      COMMON /PARM/ DELT,CHNG,DELMAX,TMAX
      COMMON /MTL/ NMAT,NMPPM,NCM
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
```

```
      DATA MAXNP,MAXEL,MAXBW,MAXNM /595,528,27,500/
      DATA MAXMAT,MXMPPM /3,9/
      DATA MAXBEL,MAXBNP /199,200/
      DATA MXRSEL,MXRSNP /99,100/
      DATA MXSTNP,MAXBCN /0,30/
C
      DATA PMAT/4H    ,4   KD,4H    ,4H    ,4HRHOB,4H    ,4H    ,
     > 4H  AL,4H    ,4H    ,4H  A[illegible]4H    ,4H   L,4HAMBD,4HA   ,4H    ,
     > 4HPOR ,4H    ,4H    ,4HALP ,4H    ,4H    ,4H  AM,4H    ,4H    ,
     > 4H TAU,4H    /
C
C
C    ----- INITIATE ARRAY  FOR NODAL POINTS
C
      DO 100 NP=1,MAXNP
         X(NP)=0.0
         Z(NP)=0.0
         R(NP)=0.0
         RP(NP)=0.0
         VX(NP)=0.0
         VXP(NP)=0.0
         VZ(NP)=0.0
         VZP(NP)=0.0
         H(NP)=0.0
         HP(NP)=0.0
         DH(NP)=0.0
         HT(NP)=0.0
         FX(NP)=0.0
         FZ(NP)=0.0
         DO 100 IB=1,MAXBW
  100       C(NP,IB)=0.0
C
C ------- INITIATE ARRAYS FOR ELEMENTS
C
      DO 150 MP=1,MAXEL
C
         DO 120 IQ=1,5
  120       IE(MP,IQ)=0
C
         DO 140 IQ=1,4
            TH(MP,IQ)=0.0
            THP(MP,IQ)=0.0
            WETAB(MP,IQ)=0.0
  140       DTH(MP,IQ)=0.0
C
  150    CONTINUE
C
C
C ------- INITIATE ARRAYS FOR BOUNDARY ELEMENTS
C
```

```
      DO 200 MP=1,MAXBEL
         DLB(MP)=0.0
         DCOSXB(MP)=0.0
         DCOSZB(MP)=0.0
         NBE(MP)=0
         DO 200 IQ=1,4
            ISB(MP,IQ)=0
  200    CONTINUE
C
C
C ------- INITIATE ARRAYS FOR BOUNDARY NODAL POINTS
C
      DO 250 NP=1,MAXBNP
         BFLX(NP)=0.0
         BFLXP(NP)=0.0
  250    NPB(NP)=0
C
C
C ------- INITIATE ARRAYS FOR INFLUX-OUTFLUX BOUNDARY ELEMENTS
C
      DO 300 MP=1,MXRSEL
         DL(MP)=0.0
         DCOSX(MP)=0.0
         DCOSZ(MP)=0.0
         NRSE(MP)=0
         DO 300 IQ=1,4
            IS(MP,IQ)=0
  300    CONTINUE
C
C
C ------- INITIATE ARRAYS FOR INFLUX-OUTFLUX BOUNDARY NODAL POINTS
C
      DO 350 NP=1,MXRSNP
  350    NPRS(NP)=0
C
C
C ------- INITIATE ARRAYS FOR SURFACCE TERM POINT FLUX
C
      DO 500 NP=1,MXSTNP
         NPST(NP)=0
  500    DP(NP)=0.0
C
C
C ------- INITIATE ARRAYS FOR DIRICHLENT BOUNDARY CONDITIONS
C
      DO 510 NP=1,MAXBCN
         BB(NP)=0.0
  510    NPN(NP)=0
C
C
```

APPENDIX C (continued)

```
C ------- INITIATE ARRAYS FOR MATERIAL PROPERTIES
C
      DO 650 I=1,MAXMAT
         DO 610 J=1,MXMPPM
  610       PROP(I,J)=0.0
  650    CONTINUE
C
C
C ------- INITIATE ARRAYS FOR FLOW THROUGH VARIOUS TYPES OF BOUNDARIES
C
      DO 700 I=1,10
         FRATE(I)=0.0
         FLOW(I)=0.0
  700    TFLOW(I)=0.0
C
C
C ------- INITIATE ARRAYS FOR PRINT AND WRITE ON DISK CONTROL
C
      DO 800 NP=1,MAXNTI
         KPR(NP)=0
  800    KDSK(NP)=0
C
C
C ------- PASS THE PROGRAM TO GM2DXZ
C
      CALL GM2DXZ(X,Z,IE,WETAB, C,R,RP,VX,VXP,VZ,VZP,H,HP,DH,HT,TH,THP,
     > DTH,FX,FZ, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,BFLX,BFLXP, DL,DCOSX,
     > DCOSZ,NRSE,IS,NPRS,NPN,BB,NPST,DP,KPR,KDSK,FRATE,FLOW,TFLOW,
     > PMAT,PROP,MAXNP,MAXEL,MAXBW,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,
     > MAXBCN,MAXNTI,MAXMAT,MXMPPM)
C
      STOP
      END
```

```
      SUBROUTINE GM2DXZ(X,Z,IE,WETAB, C,R,RP,VX,VXP,VZ,VZP,H,HP,DH,HT,
     > TH,THP,DTH,FX,FZ, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,BFLX,BFLXP, DL,
     > DCOSX,DCOSZ,NRSE,IS,NPRS,NPN,BB,NPST,DP,KPR,KDSK,FRATE,FLOW,TFLOW
     > ,PMAT,PROP,MAXNP,MAXEL,MAXBW,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,
     > MAXBCN,MAXNTI,MAXMAT,MXMPPM)
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 PMAT
C
      DIMENSION TITLE(9)
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)
      DIMENSION C(MAXNP,MAXBW),R(MAXNP),RP(MAXNP),WETAB(MAXEL,4)
      DIMENSION FX(MAXNP),FZ(MAXNP)
C
      DIMENSION VX(MAXNP),VXP(MAXNP),VZ(MAXNP),VZP(MAXNP),H(MAXNP),
     > HP(MAXNP),DH(MAXNP),HT(MAXNP),TH(MAXEL,4),THP(MAXEL,4),
     > DTH(MAXEL,4)
C
      DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),BFLX(MAXBNP),
     > BFLXP(MAXBNP),NBE(MAXBEL),ISB(MAXBEL,4),NPB(MAXBNP)
C
      DIMENSION DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),NRSE(MXRSEL),
     > IS(MXRSEL,4),NPRS(MXRSNP)
C
      DIMENSION NPN(MAXBCN),BB(MAXBCN),NPST(MXSTNP),DP(MXSTNP)
C
      DIMENSION PMAT(MAXMAT,MXMPPM),PROP(MAXMAT,MXMPPM)
      DIMENSION KPR(MAXNTI),KDSK(MAXNTI)
C
      DIMENSION FRATE(10),FLOW(10),TFLOW(10)
C
      COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
      COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN,NST
      COMMON /CONTRL/ NTI,NSTR,KSTR,KPRO,KDSKO,KSS,KVI
      COMMON /PARM/ DELT,CHNG,DELMAX,TMAX
      COMMON /MTL/ NMAT,NMPPM,NCM
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
C
C     INPUT PROBLEM IDENTIFICATION AND DESCRIPTION
C
   10 READ 10000,NPROB,(TITLE(I),I=1,9)
      IF (NPROB.LE.0) GO TO 250
      PRINT 10100,NPROB,(TITLE(I),I=1,9)
C
C     READ AND PRINT INPUT DATA
C
      KOUT=0
      KDIG=0
C
```

```
      CALL DATAIN(X,Z,IE,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,DL,DCOSX,DCOSZ,
     > NRSE,IS,NPRS,NPN,BB,NPST,DP,RP,PMAT,PROP,KPR,KDSK,MAXNP,MAXEL,
     > MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,MAXNTI,
     > MAXDIF,W,ISTOP)
C
      IF (ISTOP.GT.0) GO TO 240
C
C  COMPUTE BAND-WIDTH VARIABLES
C
      IHALFB=MAXDIF
      IBAND=2*IHALFB+1
      IHBP=IHALFB+1
      IF (IBAND.GT.MAXBW) GO TO 230
C
C  PREPARE INITIAL VARIABLES
C
      TIME=0.
C
C  READ INITIAL VELOCITIES, PRESSURES, AND WATER CONTENTS, IF NECESSARY
C
      DO 20 M=1,NEL
         MTYP=IE(M,5)
         POR=PROP(MTYP,6)
         DO 20 IQ=1,4
            TH(M,IQ)=POR
   20       CONTINUE
C
   60 READ(1) TIMEM,(H(NP),NP=1,NNP),(HT(NP),NP=1,NNP),((TH(M,IQ), M=1,
     > NEL),IQ=1,4),(VX(NP),NP=1,NNP),(VZ(NP),NP=1,NNP)
C
      DO 95 NP=1,NNP
         VXP(NP)=VX(NP)
   95    VZP(NP)=VZ(NP)
C
      CALL AFABTA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,
     > MAXMAT,MXMPPM, NEL, W)
C
      PRINT 10600
      KLINE=0
      DO 100 MP=1,NEL,2
         NJMN=MP
         NJMX=MIN0(MP+1,NEL)
         PRINT 10700, (NJ,(WETAB(NJ,IQ),IQ=1,4),NJ=NJMN,NJMX)
         KLINE=KLINE+1
         IF(MOD(KLINE,50).EQ.0) PRINT 10600
  100    CONTINUE
C
      CALL FLUX(X,Z,IE,WETAB,C,FX,FZ,RP,VX,VZ,PROP,MAXNP,MAXEL,MAXBW,
     > MAXMAT,MXMPPM)
C
```

```
      CALL SFLOW(X,Z,IE,WETAB,RP,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,NBE,
     > ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,MAXEL,
     > MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,DELT,DH)
C
C
C  PRINT INITIAL VARIABLES
C
      CALL PRINTT(RP,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPRO,KOUT,KDIG,
     > MAXNP,NNP,IBAND)
C
      IF(KSTR.EQ.1 .AND. KDSKO.EQ.1) CALL STORE(X,Z,IE,RP,FX,FZ,TITLE,
     > NPROB,NNP,NEL,NTI,MAXNP,MAXEL,TIME)
C
C  PERFORM TRANSIENT-STATE CALCULATION
C
      TIME=DELT
      W1=W
      W2=1.-W
C
C  READ TIME-DEPENDENT VELOCITIES, AS REQUIRED
C
      DO 220 ITM=1,NTI
         DO 110 NP=1,NNP
            VXP(NP)=VX(NP)
            VZP(NP)=VZ(NP)
  110       HP(NP)=H(NP)
         DO 130 M=1,NEL
            DO 130 IQ=1,4
  130          THP(M,IQ)=TH(M,IQ)
         IF(KVI.NE.2) GO TO 170
C
         READ(1) TIMEM,(H(NP),NP=1,NNP),(HT(NP),NP=1,NNP),((TH(M,IQ), M=
     >     1,NEL),IQ=1,4),(VX(NP),NP=1,NNP),(VZ(NP),NP=1,NNP)
C
C  ASSEMBLE COEFFICIENT MATRICES A, B, AND C, AND CONSTRUCT
C  LOAD VECTOR R
C
  170    DO 180 NP=1,NNP
  180       DH(NP)=(H(NP)-HP(NP))/DELT
         DO 200 M=1,NEL
            DO 200 IQ=1,4
  200          DTH(M,IQ)=(TH(M,IQ)-THP(M,IQ))/DELT
C
         IF(KVI.NE.2) GO TO 201
         CALL AFABTA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,
     >               MAXMAT,MXMPPM, NEL, W)
C
  201    CALL ASEMBL(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,TH,THP,DH,DTH,
     >      PROP,W,MAXNP,MAXEL,MAXBW,MAXMAT,MXMPPM)
C
```

```
C     APPLY BOUNDARY CONDITIONS
C
            CALL BC(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,NPN,BB,NPST,DP,
     >       DL,DCOSX,DCOSZ,NRSE,IS, W, MAXNP,MAXEL,MAXBW,MAXBCN,MXSTNP,
     >       MXRSEL)
C
C     TRIANGULARIZE C MATRIX
C
            CALL SOLVE(1,C,R,NNP,IHALFB,MAXNP,MAXBW)
C
C     BACK SUBSTITUTE
C
            CALL SOLVE(2,C,R,NNP,IHALFB,MAXNP,MAXBW)
C
            IF(IMID.EQ.0) GO TO 208
            DO 205 NP=1,NNP
  205          R(NP)=2.0D0*R(NP)-RP(NP)
            DO 206 NPP=1,NBC
               NI=NPN(NPP)
  206          R(NI)=BB(NPP)
  208       CONTINUE
C
C     CALCULATE MATERIAL FLUX FX(NP) AND FZ(NP)
C
            CALL FLUX(X,Z,IE,WETAB, C,FX,FZ, R,VX,VZ, PROP, MAXNP,MAXEL,
     >                MAXBW,MAXMAT,MXMPPM)
C
C
C     DETERMINE BOUNDARY FLOWS
C
            CALL SFLOW(X,Z,IE,WETAB,R,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,NBE,
     >       ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,MAXEL,
     >       MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,DELT,
     >       DH)
C
C     PRINT VARIABLES AT CURRENT TIME STEP
C
            CALL PRINTT(R,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPR(ITM),KOUT,
     >       KDIG,MAXNP,NNP,IBAND)
C
         IF(KSTR.EQ.1 .AND. KDSK(ITM).EQ.1) CALL STORE(X,Z,IE,R,FX,FZ,
     1  TITLE,NPROB,NNP,NEL,NTI,MAXNP,MAXEL,TIME)
C
C     PREPARE FOR NEXT TIME STEP
C
            IF (KSS.EQ.0) GO TO 10
            IF (TIME.GT.TMAX) GO TO 10
            DELT=DELT*(1.+CHNG)
            DELT=DMIN1(DELT,DELMAX)
            TIME=TIME+DELT
```

APPENDIX C (continued)

```
      DO 210 NP=1,NNP
  210     RP(NP)=R(NP)
  220   CONTINUE
      GO TO 10
  230 PRINT 10400, IBAND, MAXBW
  240 PRINT 10500, ISTOP
  250 RETURN
C
10000 FORMAT(I5,9A8)
10100 FORMAT(/8H1PROBLEM,I5,3H.. ,9A8/)
10200 FORMAT(8F10.0)
10300 FORMAT(////35H1ERROR IN VELOCITY INPUT AT ELEMENT, I5///)
10400 FORMAT(////12H BANDWIDTH =,I4,25H EXCEEDS MAX. ALLOWABLE =,I4///)
10500 FORMAT(1H0,4X,'ISTOP = ',I5)
10600 FORMAT(1H1,'TABLE OF WEIGHTING FACTORS OF EVERY ELEMENTS'//)
10700 FORMAT(1H ,I5,4E12.4,5X,I5,4E12.4)
      END
```

```
      SUBROUTINE DATAIN(X,Z,IE,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,DL,DCOSX,
     > DCOSZ,NRSE,IS,NPRS,NPN,BB,NPST,DP,RP,PMAT,PROP,KPR,KDSK,MAXNP,
     > MAXEL,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,
     > MAXNTI,MAXDIF,W,ISTOP)
C
C
C
C    FUNCTION OF SUBROUTINE--TO READ, PRINT, AND CHECK VARIABLES
C    PERTAINING TO SIMULATION TIME, GEOMETRY OF THE SYSTEM, BOUNDARY-
C    INITIAL CONDITIONS, AND PROPERTIES OF BOTH THE MATERIAL BEING
C    TRANSPORTED AND THE POROUS MEDIA.
C
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 PMAT
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)
C
      DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),NBE(MAXBEL),
     > ISB(MAXBEL,4),NPB(MAXBNP)
C
      DIMENSION DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),NRSE(MXRSEL),
     > IS(MXRSEL,4),NPRS(MXRSNP)
C
      DIMENSION NPN(MAXBCN),BB(MAXBCN),DP(MXSTNP),NPST(MXSTNP)
      DIMENSION RP(MAXNP),PMAT(MAXMAT,MXMPPM),PROP(MAXMAT,MXMPPM)
      DIMENSION KPR(MAXNTI),KDSK(MAXNTI)
      DIMENSION TITLEM(9)
C
      COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
      COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN,NST
      COMMON /CONTRL/ NTI,NSTR,KSTR,KPRO,KDSKO,KSS,KVI
      COMMON /PARM/ DELT,CHNG,DELMAX,TMAX
      COMMON /MTL/ NMAT,NMPPM,NCM
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
      ISTOP=0
      READ 10900,NNP,NEL,NMAT,NCM,NTI,NBC,NST,NRSEL,KVI,KSTR,KSS,NMPPM,
     1 IWET,ILUMP,IMID
      READ 11000, DELT,CHNG,DELMAX,TMAX,W
      READ 11100,KPRO,(KPR(ITM),ITM=1,NTI)
      READ 11100, KDSKO,(KDSK(ITM),ITM=1,NTI)
      IF(TMAX.LE.0.0) TMAX=1.0E50
C
      PRINT 10000,NNP,NEL,NMAT,NCM,NTI,NBC,NST,NRSEL,KVI,KSTR,KSS,DELT,
     > CHNG,DELMAX,TMAX,W
      PRINT 10001,IWET,ILUMP,IMID
      PRINT 10100
      PRINT 11200,KPRO,(KPR(ITM),ITM=1,NTI)
```

```
      PRINT 10100
      PRINT 11200, KDSK0,(KDSK(ITM),ITM=1,NTI)
C
C
C  READ AND PRINT MATERIAL PROPERTIES
C
  100 IF (NMPPM.LE.0) GO TO 120
      PRINT 10200,((PMAT(I,J),I=1,3),J=1,NMPPM)
      DO 110 I=1,NMAT
         READ 11000,(PROP(I,J),J=1,NMPPM)
  110    PRINT 11300,I,(PROP(I,J),J=1,NMPPM)
  120 CONTINUE
C
C  READ NODAL-POINT AND ELEMENT DATA FROM AUXILIARY STORAGE AND PRINT,
C  IF NECESSARY
C
      REWIND 1
      READ(1) (TITLEM(I),I=1,9),NPROBM,NNP,NEL,NBN,NBEL,NTIM
      READ(1) (X(NP),NP=1,NNP),(Z(NP),NP=1,NNP),((IE(M,IQ),M=1,NEL), IQ=
     > 1,4),(DLB(M),M=1,NBEL),(DCOSXB(M),M=1,NBEL),(DCOSZE(M),M=1,NBEL),
     > (NBE(M),M=1,NBEL),((ISB(M,IQ),M=1,NBEL),IQ=1,4),(NPB(NP),NP=1,
     > NBN)
C
      PRINT 10300
      KLINE=-1
      DO 130 NP=1,NNP
         KLINE=KLINE+1
         IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) PRINT 10300
  130    PRINT 11500,NP,X(NP),Z(NP)
      PRINT 10400
      KLINE=-1
      MAXDIF=0
      DO 150 M=1,NEL
         IE(M,5)=1
         MND=0
         DO 140 IQ=1,3
            IQ1 = IQ + 1
            DO 140 JQ=IQ1,4
               ND = IABS(IE(M,IQ)-IE(M,JQ))
               MND = MAX0(ND,MND)
  140          MAXDIF = MAX0(ND,MAXDIF)
         KLINE=KLINE+1
         IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) PRINT 10400
         PRINT 11600, M,(IE(M,I),I=1,5),MND
  150    CONTINUE
C
C
C  MODIFY MATERIAL TYPES FOR SELECTED ELEMENTS, IF NECESSARY
C
      IF (NCM.LE.0) GO TO 370
```

```
      PRINT 10500
      L=0
  340 READ 10900, MI,MTYP,MK,MINC
      IE(M,5) = MTYP
      PRINT 11700, MI,IE(MI,5)
      L = L + 1
      IF (MK.LE.MI) GO TO 360
      IF (MINC.LE.0) MINC = 1
      MI = MI + MINC
      DO 350 MJ=MI,MK,MINC
         IE(MJ,5) = MTYP
         PRINT 11700, MJ,IE(MJ,5)
  350    L = L + 1
  360 IF (L.LT.NCM) GO TO 340
C
C  CHECK MATERIAL TYPES FOR EACH ELEMENT
C
  370 DO 380 M=1,NEL
         MTYP=IE(M,5)
         IF (MTYP.GT.C.AND.MTYP.LE.NMAT) GO TO 380
         PRINT 14100,M
         ISTOP=ISTOP+1
  380    CONTINUE
C
C  READ INITIAL CONDITIONS
C
      IF (ISTOP.EQ.0) GO TO 390
      PRINT 13600,ISTOP
  390 NI=0
      NJ=0
  400 IF (NJ.EC.NNP) GO TO 440
      READ 11800,NJ,RP(NJ)
  410 NI=NI+1
      IF (NI.GT.1) GO TO 420
      IF (NJ.EQ.1) GO TO 420
      PRINT 13500,NJ
      ISTOP=ISTOP+1
      GO TO 820
  420 IF (NJ.EQ.NI) GO TO 400
      IF (NJ.GT.NI) GO TO 430
      PRINT 13500,NJ
      ISTOP=ISTOP+1
      GO TO 820
  430 RP(NI)=RP(NI-1)
      GO TO 410
C
  440 CONTINUE
      DO 450 NP=1,NSTN
  450    DP(NP)=0.
      IF (NBC.EQ.0) GO TO 550
```

APPENDIX C (continued)

```
C
C  READ CONSTANT-CONCENTRATION DIRICHLET CONDITIONS BB(NPP) TO BE
C  APPLIED AT NODES NPN(NPP)
C
      NPP=0
  460 IF (NPP.EQ.NBC) GO TO 520
      IF (NPP.LT.NBC) GO TO 470
      PRINT 12500,NBC
      ISTOP=ISTOP+1
      GO TO 520
  470 READ 11900,NI,NPINC,BBI
      IF (NPINC.GT.0) GO TO 490
  480 NPP=NPP+1
      NPN(NPP)=NI
      BB(NPP)=BBI
      GO TO 460
  490 IF (NPP.GT.0) GO TO 500
      ISTOP=ISTOP+1
      PRINT 13900
  500 NJ=NPN(NPP)+NPINC
      BBJ=BB(NPP)
      NK=NI-1
      DO 510 NP=NJ,NK,NPINC
         NPP=NPP+1
         NPN(NPP)=NP
  510    BB(NPP)=BBJ
      GO TO 480
  520 PRINT 10600
      DO 530 NPP=1,NBC
  530    PRINT 12000,NPN(NPP),BB(NPP)
C
C  APPLY DIRICHLET BOUNDARY SPECIFICATIONS TO THE INITIAL CONDITIONS
C
      DO 540 NPP=1,NBC
         NP=NPN(NPP)
  540    RP(NP)=BB(NPP)
  550 IF (NST.LE.0) GO TO 650
C
C  READ SURFACE-TERM FLUXES EI AND EJ TO BE APPLIED AT BOUNDARY
C  NODES NI AND NJ, RESPECTIVELY
C
      NPP=0
      MP=0
      PRINT 10700
  560 IF (MP.EQ.NST) GO TO 610
      READ 12100,NI,NJ,KINC,EI,EJ
      IF (KINC.GT.0) GO TO 580
  570 MP=MP+1
      DX=X(NI)-X(NJ)
      DZ=Z(NI)-Z(NJ)
```

```
      EL=DSQRT(DX*DX+DZ*DZ)
      IF(MP.GT.1) GO TO 571
      NPP=NPP+1
      NPST(NPP)=NI
      NII=NPP
      NPP=NPP+1
      NPST(NPP)=NJ
      NJJ=NPP
      GO TO 578
  571 DO 572 I=1,NPP
         IJ=NPST(I)
         IF(IJ.EQ.NI) GO TO 573
  572    CONTINUE
      NPP=NPP+1
      NPST(NPP)=NI
      NII=NPP
      GO TO 574
  573 NII=I
  574 DO 575 J=1,NPP
         IJ=NPST(J)
         IF(IJ.EQ.NJ) GO TO 576
  575    CONTINUE
      NPP=NPP+1
      NPST(NPP)=NJ
      NJJ=NPP
      GO TO 578
  576 NJJ=J
  578 DP(NII)=DP(NII)+EI*EL/3.0+EJ*EL/6.0
      DP(NJJ)=DP(NJJ)+EI*EL/6.0+EJ*EL/3.0
      EK=EJ
      PRINT 12200,NI,NJ,EI,EJ
      GO TO 560
  580 IF (MP.GT.0) GO TO 590
      ISTOP=ISTOP+1
      PRINT 14000
  590 NPINC=IABS(NJ-NI)
      NPMIN=MAX0(NPST(NPP),NPST(NPP-1))
      NPMAX=MIN0(NI,NJ)-1
      DO 600 NK=NPMIN,NPMAX,NPINC
         NL=NK+NPINC
         PRINT 12200,NK,NL,EK,EK
         MP=MP+1
         DX=X(NK)-X(NL)
         DZ=Z(NK)-Z(NL)
         EL=DSQRT(DX*DX+DZ*DZ)
         IF(MP.GT.1) GO TO 591
         NPP=NPP+1
         NPST(NPP)=NK
         NKK=NPP
         NPP=NPP+1
```

```
      NPST(NPP)=NL
      NLL=NPP
      GO TO 598
591   DO 592 K=1,NPP
         KL=NPST(K)
         IF(KL.EQ.NK) GO TO 593
592      CONTINUE
      NPP=NPP+1
      NPST(NPP)=NK
      NKK=NPP
      GO TO 594
593   NKK=K
594   DO 595 L=1,NPP
         KL=NPST(L)
         IF(KL.EQ.NL) GO TO 596
595      CONTINUE
      NPP=NPP+1
      NPST(NPP)=NL
      NLL=NPP
      GO TO 598
596   NLL=L
598   DP(NKK)=DP(NKK)+EK*EL/2.0
      DP(NLL)=DP(NLL)+EK*EL/2.0
600   CONTINUE
610 NSTN=NPP
C
C  READ NUMBERS OF ELEMENTS AND SIDES TO WHICH OPEN BOUNDARU CONDITIONS
C  ARE TO BE APPLIED
C
650 IF(NRSEL .LE.0) GO TO 820
    NPP=0
    MPI=0
660 IF(MPI.EQ.NRSEL) GO TO 710
    READ 10900,MI,IS1,IS2,KINC
    IF (KINC.GT.0) GO TO 680
670 MPI=MPI+1
    NRSE(MPI)=MI
    IS(MPI,1)=IS1
    IS(MPI,2)=IS2
    IF(MPI.GT.1) GO TO 671
    NPP=NPP+1
    NPRS(NPP)=IS1
    NPP=NPP+1
    NPRS(NPP)=IS2
    GO TO 678
671 DO 672 I=1,NPP
      IJ=NPRS(I)
      IF(IJ.EQ.IS1) GO TO 673
672   CONTINUE
    NPP=NPP+1
```

APPENDIX C (continued)

```
      NPRS(NPP)=IS1
      GO TO 674
  673 CONTINUE
  674 DO 675 J=1,NPP
         IJ=NPRS(J)
         IF(IJ.EQ.NJ) GO TO 676
  675    CONTINUE
      NPP=NPP+1
      NPRS(NPP)=IS2
      GO TO 678
  676 CONTINUE
  678 CONTINUE
      GO TO 660
  680 IF (MPI.GT.0) GO TO 690
      ISTOP=ISTOP+1
      PRINT 13700
  690 NPINC=IS(MPI,2)-IS(MPI,1)
      MINC=IABS(NPINC)-1
      MJ=NRSE(MPI)+MINC
      MK=MI-1
      DO 700 M=MJ,MK,MINC
         MPJ=MPI
         MPI=MPI+1
         NRSE(MPI)=M
         IS(MPI,1)=IS(MPJ,2)
         NK=IS(MPI,1)
         IS(MPI,2)=IS(MPI,1)+NPINC
         NL=IS(MPI,2)
         IF(MPI.GT.1) GO TO 691
         NPP=NPP+1
         NPRS(NPP)=NK
         NPP=NPP+1
         NPRS(NPP)=NL
         GO TO 698
  691    DO 692 K=1,NPP
            KL=NPRS(K)
            IF(KL.EQ.NK) GO TO 693
  692       CONTINUE
         NPP=NPP+1
         NPRS(NPP)=NK
         GO TO 694
  693    CONTINUE
  694    DO 695 L=1,NPP
            KL=NPRS(L)
            IF(KL.EQ.NL) GO TO 696
  695       CONTINUE
         NPP=NPP+1
         NPRS(NPP)=NL
      GO TO 698
  696    CONTINUE
```

```
  698 CONTINUE
  700     CONTINUE
      GO TO 670
  710 NRSN=NPP
      PRINT 10800
      DO 720 MP=1,NRSEL
         M=NRSE(MP)
  720    PRINT 11600,M,IS(MP,1),IS(MP,2)
C
C  DETERMINE DIRECTION COSINES DCOSX(MP) AND DCOSZ(MP) FOR THE
C  OPEN BOUNDARY SIDES
C
      DO 810 MPI=1,NRSEL
         MI=NRSE(MPI)
         DO 800 MPJ=1,NBEL
            MJ=NBE(MPJ)
            IF (MJ.NE.MI) GO TO 800
            IF (ISB(MPJ,1).EQ.IS(MPI,1).AND.ISB(MPJ,2).EQ.IS(MPI,2)) GO
     >       TO 780
            IF (ISB(MPJ,1).EQ.IS(MPI,2).AND.ISB(MPJ,2).EQ.IS(MPI,1)) GO
     >       TO 780
            GO TO 800
  780       DO 790 J=1,4
  790          IS(MPI,J)=ISB(MPJ,J)
            DL(MPI)=DLB(MPJ)
            DCOSX(MPI)=DCOSXB(MPJ)
            DCOSZ(MPI)=DCOSZB(MPJ)
            GO TO 810
  800       CONTINUE
         ISTOP=ISTOP+1
         PRINT 13800,MI
  810 CONTINUE
  820 IF(ISTOP.EQ.0) GO TO 830
      PRINT 13600, ISTOP
  830 RETURN
C
10000 FORMAT(35H0INPUT TABLE 1.. BASIC PARAMETERS    // 5X,
     > 40H NUMBER OF NODAL POINTS. . . . . . . . . ..I5/ 5X,
     > 40H NUMBER OF ELEMENTS. . . . . . . . . . . ..I5/ 5X,
     > 40H NUMBER OF DIFFERENT MATERIALS . . . . . ..I5/ 5X,
     > 40H NUMBER OF CORRECTION MATERIALS. . . . . ..I5/ 5X,
     > 40H NUMBER OF TIME INCREMENTS . . . . . . . ..I5/ 5X,
     > 40H NUMBER OF BOUNDARY CONDITIONS . . . . . ..I5/ 5X,
     > 40H NUMBER OF SURFACE TERMS . . . . . . . . ..I5/ 5X,
     > 40H NUMBER OF SEEPAGE SURFACE TERMS . . . . ..I5/ 5X,
     > 40H VELOCITY INPUT CONTROL. . . . . . . . . ..I5/ 5X,
     > 40H AUXILIARY STORAGE CONTROL . . . . . . . ..I5/ 5X,
     > 40H STEADY-STATE CONTROL. . . . . . . . . . ..I5/ 5X,
     > 40H TIME INCREMENT. . . . . . . . . . . . . ..F10.6/ 5X,
     > 40H MULTIPLIER FOR INCREASING DELT. . . . . ..F10.6/ 5X,
```

```
     > 40H MAXIMUM VALUE  F DELT . . . . . . . . . . ..D10.4/ 5X,
     > 40H MAXIMUM VALUE  F TIME . . . . . . . . . . ..D10.4/ 5X,
     > 40H TIME-INTEGRATI N PARAMETER. . . . . . . ..F10.6)
10001 FORMAT(1H ,5X,
     > 40H UPSTREAM WEIGH ING INDICATOR, IWET. . . ..I5/ 5X,
     > 40H LUMPING INDICA OR, ILUMP. . . . . . . . ..I5/ 5X,
     > 40H TIME-DIFFERENC  INDICATOR, IMID . . . ..I5)
10100 FORMAT(//6X,14HOUTP T CONTROL)
10200 FORMAT(36H1INPUT TA LE 2.. MATERIAL PROPERTIES// 9H MAT. NO., 9(
     > 3A4))
10300 FORMAT(34H1INPUT TA LE 3.. NODAL POINT DATA // 7X,4HNODE, 8X,1HX,
     > 14X,1HZ)
10400 FORMAT(34H1INPUT TA LE 4.. ELEMENT DATA     // 11X,
     > 31HGLOBAL INDICES OF ELEMENT NODES/7X,7HELEMENT, 3X,1H1,7X,1H2,
     > 7X,1H3,7X,1H4,6X,8HMATERIAL,6X, 10HNODE DIFF. )
10500 FORMAT(//52H CORRECTIONS TO MATERIAL TYPES FOR SELECTED ELEMENTS/)
10600 FORMAT(44H1INPUT TABLE 5.. BOUNDARY CONDITIONS OF FORM, 5H R=BB//
     > 6H  NODE,7X,2HBB/)
10700 FORMAT(32H1INPUT TABLE 6.. SURFACE TERMS , 20H E=EI AT NODE NI, E=
     >  13HEJ AT NODE NJ//3X,2HNI,3X,2HNJ,6X,2HEI,13X,2HEJ/)
10800 FORMAT(44H1INPUT TABLE 7.. SEEPAGE-SURFACE INFORMATION//5X
     > 14HELEMENT NODE 1,2X,6HNODE 2)
10900 FORMAT(16I5)
11000 FORMAT(8F10.0)
11100 FORMAT(80I1)
11200 FORMAT(10X,101I1)
11300 FORMAT(I8,9D12.4)
11400 FORMAT(I5,2F10.3)
11500 FORMAT(I10,2D15.4)
11600 FORMAT(I10,4I8,I10,I15)
11700 FORMAT(I10,32X,I10)
11800 FORMAT(I5,5X,F10.0)
11900 FORMAT(2I5,F10.0)
12000 FORMAT(I5,D15.4)
12100 FORMAT(3I5,5X,2F10.0)
12200 FORMAT(2I5,2(1PD15.4))
129000 FORMAT(////36H CHECK BOUNDARY CONDITIONS, MAXIMUM=,I5///)
13300 FORMAT(////30H ERROR IN NODAL-POINT CARD NO.,I5///)
13400 FORMAT(////26H ERROR IN ELEMENT CARD NO.,I5///)
13500 FORMAT(////36H ERROR IN INITIAL-CONDITION CARD NO.,I5///)
13600 FORMAT(////45H ASSEMBLY AND SOLUTION WILL NOT BE PERFORMED.,I5,
     > 19H  FATAL CARD ERRORS///)
13700 FORMAT(////38H ERROR IN FIRST TRANSIENT-SURFACE CARD///)
13800 FORMAT(////44H ERROR IN TRANSIENT-SURFACE CARD FOR ELEMENT,I5///)
13900 FORMAT(////49H ERROR IN FIRST R=BB TYPE BOUNDARY-CONDITION CARD //
     >  /)
14000 FORMAT(////33H ERROR IN FIRST SURFACE-TERM CARD///)
14100 FORMAT(////40H ERROR IN MATERIAL TYPE CODE FOR ELEMENT ,I5///)
      END
```

```
      SUBROUTINE AFABTA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,
     > MAXMAT,MXMPPM, NEL, W)
C
C  FUNCTION OF THE SUBROUTINE-TO CALCULATE THE WEIGHTING FACTORS
C  ON FOUR SIDES OF EACH OF THE ELEMENTS
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
      DIMENSION VX(MAXNP),VXP(MAXNP),VZ(MAXNP),VZP(MAXNP)
      DIMENSION PROP(MAXMAT,MXMPPM)
      DIMENSION APHA(4),BETA(4),MDIM(4)
C
      W1=W
      W2=1.0-W
      IF(KSS.EQ.0) GO TO 100
      W1=1.0
      W2=0.0
  100 CONTINUE
      DO 500 M=1,NEL
      MTYP=IE(M,5)
      AL=PROP(MTYP,3)
      AT=PROP(MTYP,4)
      AM=PROP(MTYP,8)
      TAU=PROP(MTYP,9)
      DD=TAU*AM
C
      M1=IE(M,1)
      M2=IE(M,2)
      M3=IE(M,3)
      M4=IE(M,4)
C
      A=Z(M2)-Z(M1)
      B=X(M2)-X(M1)
      BETA(1)=DATAN2(A,B)
      A=Z(M3)-Z(M2)
      B=X(M3)-X(M2)
      BETA(2)=DATAN2(A,B)
      A=Z(M3)-Z(M4)
      B=X(M3)-X(M4)
      BETA(3)=DATAN2(A,B)
      A=Z(M4)-Z(M1)
      B=X(M4)-X(M1)
      BETA(4)=DATAN2(A,B)
      MDIM(1)=M1
      MDIM(2)=M2
      MDIM(3)=M3
      MDIM(4)=M4
C
      DO 400 J=1,4
```

```
      JQ=J                                                              AFAB 255
      JQP=JQ+1                                                          AFAB 260
      IF(JQ.EQ.4) JQP=1                                                 AFAB 265
      M1=MDIM(J)                                                        AFAB 270
      IF(J-3) 200,200,210                                               AFAB 275
  200 M2=MDIM(J+1)                                                      AFAB 280
      GO TO 300                                                         AFAB 285
  210 M2=MDIM(1)                                                        AFAB 290
  300 CONTINUE                                                          AFAB 295
      ALENG=DSQRT((X(M2)-X(M1))**2+(Z(M2)-Z(M1))**2)                    AFAB 300
      CBETA=DCOS(BETA(J))                                               AFAB 305
      SBETA=DSIN(BETA(J))                                               AFAB 310
      VXX=0.5*((VX(M1)+VX(M2))*W1+(VXP(M1)+VXP(M2))*W2)                 AFAB 315
      VZZ=0.5*((VZ(M1)+VZ(M2))*W1+(VZP(M1)+VZP(M2))*W2)                 AFAB 320
      VAL=VXX*CBETA+VZZ*SBETA                                           AFAB 325
      VV=DSQRT(VXX*VXX+VZZ*VZZ)                                         AFAB 330
      VVI=1.0/VV                                                        AFAB 335
      DLL=(AL*VXX*VXX + AT*VZZ*VZZ)*VVI + DD                            AFAB 340
      DLT=(AL-AT)*VXX*VZZ*VVI                                           AFAB 345
      DTT=(AL*VZZ*VZZ + AT*VXX*VXX)*VVI + DD                            AFAB 350
      DAL=DABS(DLL*CBETA*CBETA+2.0D0*CBETA*SBETA*DLT+DTT*SBETA*SBETA)   AFAB 355
      VEL=VAL*ALENG                                                     AFAB 360
      DISP=2.0*DAL                                                      AFAB 365
      APHA(J)=1.0/DTANH(VEL/DISP)- DISP/VEL                             AFAB 370
      IF(APHA(J).LT.0.0) APHA(J)=0.0                                    AFAB 375
  400 CONTINUE                                                          AFAB 380
      WETAB(M,1)=APHA(1)                                                AFAB 385
      WETAB(M,2)=APHA(2)                                                AFAB 390
      WETAB(M,3)=APHA(3)                                                AFAB 395
      WETAB(M,4)=APHA(4)                                                AFAB 400
  500 CONTINUE                                                          AFAB 405
      RETURN                                                            AFAB 410
      END                                                               AFAB 415
```

```
      SUBROUTINE FLUX(X,Z,IE,WETAB, C,FX,FZ, R,VX,VZ,PROP,
     > MAXNP,MAXEL,MAXBW,MAXMAT,MXMPPM)
C
C FUNCTION OF SUBROUTINE TO COMPUTE DARCY MATERIAL FLUX FX AND FZ,
C RESPECTIVELY
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
      DIMENSION C(MAXNP,MAXBW),FX(MAXNP),FZ(MAXNP)
      DIMENSION R(MAXNP),VX(MAXNP),VZ(MAXNP),PROP(MAXMAT,MXMPPM)
      DIMENSION QB(4,4),RQ(4),XQ(4),ZQ(4),CRQ(4),VXQ(4),VZQ(4)
C
      COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
C INITIALIZE THE FLUX FX(NP) AND FZ(NP)
C
      DO 100 NP=1,NNP
        FX(NP)=0.0
100     FZ(NP)=0.0
C
C COMPUTE THE FLUX COMPONENTS BY APPLYING THE FINITE ELEMENT METHOD
C TO THE DISPERSION TERMS
C
      IHALFB=(IBAND-1)/2
      IHBP=IHALFB+1
      DO 300 IXZ=1,2
        DO 110 NP=1,NNP
          DO 110 IB=1,IBAND
110         C(NP,IB)=0.0
        DO 180 M=1,NEL
          APHA1=WETAB(M,1)
          APHA2=WETAB(M,3)
          BETA1=WETAB(M,2)
          BETA2=WETAB(M,4)
          MTYP=IE(M,5)
          AL=PROP(MTYP,3)
          AT=PROP(MTYP,4)
          AM=PROP(MTYP,8)
          TAU=PROP(MTYP,9)
          DD=AM*TAU
          DO 120 IQ=1,4
            NP=IE(M,IQ)
            XQ(IQ)=X(NP)
            ZQ(IQ)=Z(NP)
            CRQ(IQ)=R(IQ)
            VXQ(IQ)=VX(NP)
            VZQ(IQ)=VZ(NP)
120       CONTINUE
```

APPENDIX C (continued)

```
C
C     COMPUTE THE ELEMENT MATRIX QB(IQ,JQ) AND QQ(IQ)
C
               CALL Q4D(QB,RQ,XQ,ZQ,VXQ,VZQ,CRQ,DD,AL,AT,SNFE,CSFE,IXZ)
C
C
C     ASSEMBLE QB(IQ,JQ) INTO THE GLOBAL MATRIX C(NP,IB) AND
C     FORM THE LOAD VECTOR FX(NP) OR FZ(NP)
C
               DO 140 IQ=1,4
                  NI=IE(M,IQ)
                  DO  130 JQ=1,4
                     NJ=IE(M,JQ)
                     IB=NJ-NI+IHBP
                     C(NI,IB)=C(NI,IB)+QB(IQ,JQ)
  130                CONTINUE
C
                  IF(IXZ.EQ.2) GO TO 135
                  FX(NI)=FX(NI)+RQ(IQ)
                  GO TO 140
  135             FZ(NI)=FZ(NI)+RQ(IQ)
  140             CONTINUE
  180          CONTINUE
C
C     SOLVE THE MATRIX EQUATION CX=B
C
            IF(IXZ.EQ.2) GO TO 200
            CALL SOLVE(1,C,FX,NNP,IHALFB,MAXNP,MAXBW)
            CALL SOLVE(2,C,FX,NNP,IHALFB,MAXNP,MAXBW)
            GO TO 300
  200       CALL SOLVE(1,C,FZ,NNP,IHALFB,MAXNP,MAXBW)
            CALL SOLVE(2,C,FZ,NNP,IHALFB,MAXNP,MAXBW)
  300       CONTINUE
         RETURN
         END
```

```
      SUBROUTINE Q4D(QB,RQ,XQ,ZQ,VXQ,VZQ,CRQ,DD,AL,AT,SNFE,CSFE,IND)
C
C  FUNCTION OF SUBROUTINE-TO EVALUATE THE MATRIX QUADRATURE OVER THE
C  AREA OF ONE ELEMENT.  THESE INTEGRALS ARISE THROUGH THE
C  APPLICATION OF THE GALERKIN INTEGRATION SCHEME
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N(4)
      DIMENSION QB(4,4),RQ(4),XQ(4),ZQ(4),CRQ(4),VXQ(4),VZQ(4)
      DIMENSION S(4),T(4),W(4),DNX(4),DNZ(4),DWX(4),DWZ(4)
      DIMENSION PJAB(2,2),DNSS(4),DNTT(4),DWSS(4),DWTT(4)
C
      DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-
     > 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /
C
C  INITIALIZE MATRICES QB(IQ,JQ) AND QQ(IQ)
C
      DO 100 IQ=1,4
         RQ(IQ)=0.0
         DO 100 JQ=1,4
  100       QB(IQ,JQ)=0.0
C
C  SUMMATION OF THE INTEGRAND OVER THE GAUSSIAN POINTS
C
      DO 400 KG=1,4
C
C  DETERMINE LOCAL COORDINATE (SS,TT) OF
C  GAUSS-INTEGRATION POINT KG
C
         SS=P*S(KG)
         TT=P*T(KG)
C
C  CALCULATE THE VALUES OF BASIS FUNCTIONS, N(IQ), AND WEIGHTING
C  W(IQ) AND THEIR DERIVATIVES WITH RESPECTIVE TO X AND Z, RESPECTIVELY
C  AT THE GAUSSIAN POINT KG
C
         CALL SHAPE(N,W,DNSS,DNTT,DWSS,DWTT,SS,TT)
         DO 110 I=1,2
            DO 110 J=1,2
  110          PJAB(I,J)=0.0
         DO 120 I=1,4
            PJAB(1,1)=PJAB(1,1)+ZQ(I)*DNTT(I)
            PJAB(1,2)=PJAB(1,2)-ZQ(I)*DNSS(I)
            PJAB(2,1)=PJAB(2,1)-XQ(I)*DNTT(I)
  120       PJAB(2,2)=PJAB(2,2)+XQ(I)*DNSS(I)
         DJAC=PJAB(2,2)*PJAB(1,1)-PJAB(1,2)*PJAB(2,1)
         DJACI=1.0/DJAC
         DO 130 I=1,2
            DO 130 J=1,2
  130          PJAB(I,J)=PJAB(I,J)*DJACI
```

```
          DO 140 I=1,4
             DNX(I)=DNSS(I)*PJAB(1,1)+DNTT(I)*PJAB(1,2)
             DNZ(I)=DNSS(I)*PJAB(2,1)+DNTT(I)*PJAB(2,2)
             DWX(I)=DWSS(I)*PJAB(1,1)+DWTT(I)*PJAB(1,2)
  140        DWZ(I)=DWSS(I)*PJAB(2,1)+DWTT(I)*PJAB(2,2)
C
C  INTERPOLATE WITH THE BASIS FUNCTIONS N(IQ) TO OBTAIN VALUES OF
C  VELOCITY COMPONENTS VXK AND VZK AT GAUSSIAN POINT
C
          VXK=0.0
          VZK=0.0
          DO 150 IQ=1,4
             VXK=VXK+VXQ(IQ)*N(IQ)
             VZK=VZK+VZQ(IQ)*N(IQ)
  150     CONTINUE
C
          VK=DSQRT(VXK*VXK+VZK*VZK)
          VKI=1.0/VK
          AKXK=(AL*VXK*VXK+AT*VZK*VZK)*VKI + DD
          AKZK=(AL*VZK*VZK+AT*VXK*VXK)*VKI + DD
          AKXZK=(AL-AT)*VXK*VZK*VKI
C
C  ACCUMULATE THE SUMS TO OBTAIN THE MATRIX INTEGRALS QB(IQ,JQ)
C  AND QQ(IQ)
C
          DO 300 IQ=1,4
             DO 300 JQ=1,4
                QB(IQ,JQ)=QB(IQ,JQ)+W(IQ)*N(JQ)*DJAC
                IF(IND.EQ.2) GO TO 200
                RQ(IQ)=RQ(IQ)-W(IQ)*CRQ(JQ)*(AKXK*DNX(JQ)+AKXZK*DNZ(JQ))*
     >           DJAC + W(IQ)*CRQ(JQ)*VXK*N(JQ)*DJAC
                GO TO 300
  200           RQ(IQ)=RQ(IQ)-W(IQ)*CRQ(JQ)*(AKXZK*DNX(JQ)+AKZK*DNZ(JQ))*
     >           DJAC + W(IQ)*CRQ(JQ)*VZK*N(JQ)*DJAC
C
  300           CONTINUE
  400     CONTINUE
       RETURN
       END
```

APPENDIX C (continued)

```
      SUBROUTINE ASEMBL(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,TH,THP,DH,DTH,
     > PROP,W,MAXNP,MAXEL,MAXBW,MAXMAT,MXMPPM)
C
C
C  FUNCTION OF SUBROUTINE--TO ASSEMBLE THE GLOBAL COEFFICIENT MATRIX
C  C(NP,IB) AND LOAD VECTOR R(NP) FROM THE ELEMENT MATRICES QA(IQ,JQ)
C  AND QB(IQ,JQ).
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KD,LAMBDA
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
      DIMENSION C(MAXNP,MAXBW),R(MAXNP),RP(MAXNP),VX(MAXNP),VXP(MAXNP),
     > VZ(MAXNP),VZP(MAXNP),TH(MAXEL,4),THP(MAXEL,4),DH(MAXNP),
     > DTH(MAXEL,4),PROP(MAXMAT,MXMPPM)
C
      DIMENSION QA(4,4),QB(4,4),VXQ(4),VZQ(4),XQ(4),ZQ(4),DHQ(4),
     > DTHQ(4),THQ(4)
C
      COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
      COMMON /CONTRL/ NTI,NSTR,KSTR,KPRO,KDSKO,KSS,KVI
      COMMON /PARM/ DELT,CHNG,DELMAX,TMAX
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
C
      IHALFB=(IBAND-1)/2
      IHBP=IHALFB+1
C
      DELTI=1./DELT
      W1=W
      W2=1.-W
      IF (KSS.NE.0) GO TO 10
      DELTI=0.
      W1=1.
      W2=0.
C
C   INITIALIZE MATRICES C(NP,IB) AND R(NP)
C
   10 DO 20 NP=1,NNP
         R(NP)=0.
         DO 20 IB=1,IBAND
   20       C(NP,IB)=0.0
C
C   COMPUTE MATRICES QA(IQ,JQ) AND QB(IQ,JQ) FOR EACH ELEMENT M
C
      DO 50 M=1,NEL
         APHA1=WETAB(M,1)
         APHA2=WETAB(M,3)
         BETA1=WETAB(M,2)
```

```
      BETA2=WETAB(M,4)
      MTYP=IE(M,5)
      KD=PROP(MTYP,1)
      RHOB=PROP(MTYP,2)
      AL=PROP(MTYP,3)
      AT=PROP(MTYP,4)
      LAMBDA=PROP(MTYP,5)
      POR=PROP(MTYP,6)
      ALP=PROP(MTYP,7)
      AM=PROP(MTYP,8)
      TAU=PROP(MTYP,9)
      DO 30 IQ=1,4
         NP=IE(M,IQ)
         XQ(IQ)=X(NP)
         ZQ(IQ)=Z(NP)
         DHQ(IQ)=DH(NP)
         THQ(IQ)=TH(M,IQ)*W1+THP(M,IQ)*W2
         DTHQ(IQ)=DTH(M,IQ)
         VXQ(IQ)=VX(NP)*W1+VXP(NP)*W2
         VZQ(IQ)=VZ(NP)*W1+VZP(NP)*W2
   30 CONTINUE
C
      CALL Q4(QA,QB,XQ,ZQ,VXQ,VZQ,THQ,     DTHQ,DHQ,KD,RHOB,AL,AT,
     >  LAMBDA,POR,ALP,AM,TAU)
C
C   ASSEMBLE QA(IQ,JQ) AND QB(IQ,JQ) INTO THE GLOBAL MATRIX
C   C(NP,IB) = W1*B + A/DELT AND FORM THE LOAD VECTOR
C   R(NP) = (A/DELT - W2*B)*RP.  MATRIX C IS ASYMMETRIC DUE TO
C   THE ADVECTION TERM.
C
      IF(IMID.EQ.1) GO TO 41
      IF(IMID.EQ.1) GO TO 41
      DO 40 IQ=1,4
         NI=IE(M,IQ)
         DO 40 JQ=1,4
            NJ=IE(M,JQ)
            QA(IQ,JQ)=QA(IQ,JQ)*DELTI
            R(NI)=R(NI)+(QA(IQ,JQ)-W2*QB(IQ,JQ))*RP(NJ)
            IB=NJ-NI+IHBP
            C(NI,IB)=C(NI,IB)+QA(IQ,JQ)+W1*QB(IQ,JQ)
   40       CONTINUE
      GO TO 50
C
   41 DO 43 IQ=1,4
         NI=IE(M,IQ)
         DO 43 JQ=1,4
            NJ=IE(M,JQ)
            QA(IQ,JQ)=2.0D0*QA(IQ,JQ)*DELTI
            R(NI)=R(NI) + QA(IQ,JQ)*RP(NJ)
            IB=NJ-NI+IHBP
```

APPENDIX C (continued)

```
            C(NI,IB)=C(NI,IB) + QA(IQ,JQ) + QB(IQ,JQ)
43          CONTINUE
50       CONTINUE
      RETURN
      END
```

```
      SUBROUTINE Q4(QA,QB,XQ,ZQ,VXQ,VZQ,THQ,     DTHQ,DHQ,KD,RHOB,AL,AT,
     >    LAMBDA,POR,ALP,AM,TAU)
C
C
C  FUNCTION OF SUBROUTINE--TO EVALUATE THE MATRIX QUADRATURES QA(IQ,JQ)
C  AND QB(IQ,JQ) OVER THE AREA OF ONE ELEMENT.
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N(4),NN,KD,LAMBDA
C
      DIMENSION W(4),DNX(4),DNZ(4),DWX(4),DWZ(4)
      DIMENSION PJAB(2,2),DNSS(4),DNTT(4),DWSS(4),DWTT(4)
      DIMENSION QA(4,4),QB(4,4),VXQ(4),VZQ(4),XQ(4),ZQ(4),DHQ(4),
     > DTHQ(4),THQ(4)
      DIMENSION S(4),T(4)
C
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
      DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-
     > 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /
C
      DD=AM*TAU
C
C  INITIALIZE MATRICES QA(IQ,JQ) AND QB(IQ,JQ)
C
      DO 10 IQ=1,4
         DO 10 JQ=1,4
            QA(IQ,JQ)=0.0
10          QB(IQ,JQ)=0.0
C
      AREA=0.0D0
C
      DO 40 KG=1,4
C
C  DETERMINE THE LOCAL COORDINATES (SS,TT) AND EVALUATE THE JACOBIAN AT
C  EACH GAUSS-INTEGRATION POINT KG
C
         SS = P*S(KG)
         TT = P*T(KG)
C
C  CALCULATE VALUES OF BASIS AND WEIGHTING FUNCTIONS N(IQ) AND W(IQ)
C  AND THEIR DERIVATIVES DWX DNX AND DWZ DNZ WITH RESPECT TO X AND Z,
C  RESPECTIVELY, AT THE GAUSS POINT KG
C
         CALL SHAPE(N,W,DNSS,DNTT,DWSS,DWTT,SS,TT)
C
C
C  COMPUTE THE INVERSE TRANSFORMATION MATRIX I.E. THE INVERSE JACOBIAN:
C  PJAB(1,1)=(DZ/DT)/DJAC, PJAB(1,2)=-(DZ/DS)/DJAC,
```

```
C   PJAB(2,1)=-(DX/DT)/DJAC, PJAB(2,2)=(DX/DS)/DJAC, WHERE
C   DJAC IS THE DETERMINANT OF THE JACOBIAN
C
          DO 11 I=1,2
             DO 11 J=1,2
                PJAB(I,J)=0.0
   11           CONTINUE
C
          DO 16 I=1,4
             PJAB(1,1)=PJAB(1,1) + ZQ(I)*DNTT(I)
             PJAB(1,2)=PJAB(1,2) - ZQ(I)*DNSS(I)
             PJAB(2,1)=PJAB(2,1) - XQ(I)*DNTT(I)
             PJAB(2,2)=PJAB(2,2) + XQ(I)*DNSS(I)
   16        CONTINUE
C
C   COMPUTE THE DETERMINANT OF THE JACOBIAN
C
          DJAC=PJAB(2,2)*PJAB(1,1) - PJAB(1,2)*PJAB(2,1)
          DJACI=1.0D0/DJAC
          AREA=AREA + DJAC
C
          DO 17 I=1,2
             DO 17 J=1,2
   17           PJAB(I,J)=PJAB(I,J)*DJACI
C
          DO 18 I=1,4
             DNX(I)=PJAB(1,1)*DNSS(I) + PJAB(1,2)*DNTT(I)
             DNZ(I)=PJAB(2,1)*DNSS(I) + PJAB(2,2)*DNTT(I)
             DWX(I)=PJAB(1,1)*DWSS(I) + PJAB(1,2)*DWTT(I)
   18        DWZ(I)=PJAB(2,1)*DWSS(I) + PJAB(2,2)*DWTT(I)
C
C   INTERPOLATE WITH THE BASIS-INTERPOLATION FUNCTIONS N(IQ) TO OBTAIN
C   THE ADVECTIVE VELOCITY AT EACH GAUSS INTEGRATION POINT
C
          DHK=0.
          THK=0.
          DTHK=0.
          VXK=0.
          VZK=0.
          DO 20 IQ=1,4
             DHK=DHK+N(IQ)*DHQ(IQ)
             THK=THK+N(IQ)*THQ(IQ)
             DTHK=DTHK+N(IQ)*DTHQ(IQ)
             VXK=VXK+N(IQ)*VXQ(IQ)
   20        VZK=VZK+N(IQ)*VZQ(IQ)
C
          VK=DSQRT(VXK*VXK+VZK*VZK)
          VKI=1./VK
          A=DJAC*(THK+RHOB*KD)
          DXX=DJAC*((AL*VXK*VXK+AT*VZK*VZK)*VKI+DD)
```

```
      DZZ=DJAC*((AL*VZK*VZK+AT*VXK*VXK)*VKI+DD)
      DXZ=DJAC*(AL-AT)*VXK*VZK*VKI
      C=DJAC*(DTHK+ALP*(THK+RHOB*KD)*DHK+LAMBDA*(THK+RHOB*KD))
      VXK=VXK*DJAC
      VZK=VZK*DJAC
C
C     ACCUMULATE THE SUMS TO EVALUATE THE MATRIX INTEGRALS QA(IQ,JQ)
C     AND QB(IQ,JQ)
C
C
      DO 30 IQ=1,4
         DO 30 JQ=1,4
            WN=W(IQ)*N(JQ)
            DWXDNX=DWX(IQ)*DNX(JQ)
            DWZDNZ=DWZ(IQ)*DNZ(JQ)
            DWXN=DWX(IQ)*N(JQ)
            DWZN=DWZ(IQ)*N(JQ)
            DWZDNX=DWZ(IQ)*DNX(JQ)
            DWXDNZ=DWX(IQ)*DNZ(JQ)
            QA(IQ,JQ)=QA(IQ,JQ)+A*WN
   30       QB(IQ,JQ)=QB(IQ,JQ)+DWXDNX*DXX+DXZ*(DWXDNZ+DWZDNX)+DZZ*
     >                DWZDNZ + C*WN - (VXK*DWXN+VZK*DWZN)
   40    CONTINUE
C
      IF(ILUMP.NE.0) GO TO 50
      RETURN
C
   50 CONTINUE
      DO 52 I=1,4
         SUM=0.0
         DO 52 J=1,4
            SUM=SUM+QA(I,J)
   51       QA(I,J)=0.0
         QA(I,I)=SUM
   52    CONTINUE
      RETURN
      END
```

APPENDIX C (continued)

```
      SUBROUTINE SHAPE(N,W,DNSS,DNTT,DWSS,DWTT,SS,TT)
C
C  FUNCTION OF THE SUBROUTINE-TO COMPUTE THE VALUES OF BASIS FUNCTIONS
C  AND WEIGHTING FUNCTIONS IN LOCAL COORDINATES
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N(4)
C
      DIMENSION DNSS(4),DNTT(4),DWSS(4),DWTT(4),W(4)
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
      SM=1.0-SS
      SP=1.0+SS
      TM=1.0-TT
      TP=1.0+TT
      N(1)=0.25*SM*TM
      N(2)=0.25*SP*TM
      N(3)=0.25*SP*TP
      N(4)=0.25*SM*TP
      DNSS(1)=-0.25*TM
      DNSS(2)=0.25*TM
      DNSS(3)=0.25*TP
      DNSS(4)=-0.25*TP
      DNTT(1)=-0.25*SM
      DNTT(2)=-0.25*SP
      DNTT(3)=0.25*SP
      DNTT(4)=0.25*SM
C
      IF(IWET.EQ.0) GO TO 100
C
      W(1)=0.0625*(TP*(3.0*BETA2*(-TM)-2.0)+4.0)*(SP*(3.0*APHA1*(-SM)-
     1     2.0)+4.0)
      W(2)=0.0625*(TP*(3.0*BETA1*(-TM)-2.)+4.0)*(SP*(3.0*APHA1*SM+2.0))
      W(3)=0.0625*(TP*(3.0*BETA1*TM+2.0))*(SP*(3.0*APHA2*SM+2.0))
      W(4)=0.0625*(TP*(3.0*BETA2*TM+2.0))*(SP*(3.0*APHA2*(-SM)-2.0)+4.0)
      DWSS(1)=0.125*(TP*(3.0*BETA2*(-TM)-2.0)+4.0)*(3.0*APHA1*SS-1.0)
      DWSS(2)=-0.125*(TP*(3.0*BETA1*(-TM)-2.0)+4.0)*(3.0*APHA1*SS-1.0)
      DWSS(3)=-0.125*(TP*(3.0*BETA1*TM +2.0))*(3.0*APHA2*SS-1.0)
      DWSS(4)=0.125*(TP*(3.0*BETA2*TM+2.0))*(3.0*APHA2*SS-1.0)
      DWTT(1)=0.125*(SP*(3.0*APHA1*(-SM)-2.0)+4.0)*(3.0*BETA2*TT-1.0)
      DWTT(2)=0.125*(SP*(3.0*APHA1*SM+2.0))*(3.0*BETA1*TT-1.0)
      DWTT(3)=-0.125*(SP*(3.0*APHA2*SM+2.0))*(3.0*BETA1*TT-1.0)
      DWTT(4)=-0.125*(SP*(3.0*APHA2*(-SM)-2.0)+4.0)*(3.0*BETA2*TT-1.0)
      RETURN
C
  100 DO 200 I=1,4
      W(I)=N(I)
      DWSS(I)=DNSS(I)
  200 DWTT(I)=DNTT(I)
      RETURN
      END
```

```
      SUBROUTINE BC(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,NPN,BB,NPST,DP,
     > DL,DCOSX,DCOSZ,NRSE,IS, W, MAXNP,MAXEL,MAXBW,MAXBCN,MXSTNP,
     > MXRSEL)
C
C
C  FUNCTION OF SUBROUTINE--TO APPLY CONSTANT-CONCENTRATION DIRICHLET
C  CONDITIONS AND BOTH CONSTANT AND TRANSIENT NEUMANN SURFACE
C  BOUNDARY CONDITIONS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KD
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
      DIMENSION C(MAXNP,MAXBW),R(MAXNP),RP(MAXNP),VX(MAXNP),VXP(MAXNP),
     > VZ(MAXNP),VZP(MAXNP), NPN(MAXBCN),BB(MAXBCN),NPST(MXSTNP),
     > DP(MXSTNP), DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),NRSE(MXRSEL),
     > IS(MXRSEL,4)
C
      DIMENSION KQ(2),VXQ(4),VZQ(4),XQ(4),ZQ(4),DFLXQ(2,2)
C
      COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
      COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN,NST
      COMMON /CONTRL/ NTI,NSTR,KSTR,KPRO,KDSKO,KSS,KVI
      COMMON /PARM/ DELT,CHNG,DELMAX,TMAX
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
      IHALFB=(IBAND-1)/2
      IHBP=IHALFB+1
C
C  APPLY CONSTANT-CONCENTRATION DIRICHLET BOUNDARY CONDITIONS
C
      IF (NBC.EQ.0) GO TO 100
      DO 90 NPP=1,NBC
C
C  MODIFY ROW NPN(NPP) OF MATRIX C(NP,IB)
C
         NI=NPN(NPP)
         DO 10 IB=1,IBAND
  10        C(NI,IB)=0.0
         C(NI,IHBP)=1.0
         R(NI)=BB(NPP)
C
C  MODIFY LOAD VECTOR R(NP) FOR NON-ZERO BB(NPP)
C
         IF (BB(NPP).EQ.0.0) GO TO 50
         DO 20 IB=1,IHALFB
            NJ=NI-IB
            IF (NJ.LT.1) GO TO 30
            JB=IHBP+IB
  20        R(NJ)=R(NJ)-BB(NPP)*C(NJ,JB)
```

```
   30     DO 40 IB=1,IHALFB
             NJ=NI+IB
             IF (NJ.GT.NNP) GO TO 50
             JB=IHBP-IB
   40        R(NJ)=R(NJ)-BB(NPP)*C(NJ,JB)
C
C   ZERO COLUMN NPN(NPP) OF MATRIX C(NP,IB)
C
   50     DO 60 IB=1,IHALFB
             NJ=NI-IB
             IF (NJ.LT.1) GO TO 70
             JB=IHBP+IB
   60        C(NJ,JB)=0.0
   70     DO 80 IB=1,IHALFB
             NJ=NI+IB
             IF (NJ.GT.NNP) GO TO 90
             JB=IHBP-IB
   80        C(NJ,JB)=0.0
   90     CONTINUE
C
  100 IF(NRSEL.LE.0) GO TO 140
C
C   ENTER SEEPAGE TERMS IN MATRIX C(NP,IB)
C
      W1=W
      W2=1.-W
      IF (KSS.NE.0) GO TO 110
      W1=1.
      W2=0.
  110 DO 130 MP=1,NRSEL
          M=NRSE(MP)
          APHA1=WETAB(M,1)
          APHA2=WETAB(M,3)
          BETA1=WETAB(M,2)
          BETA2=WETAB(M,4)
          NI=IS(MP,1)
          NJ=IS(MP,2)
          KQ(1)=IS(MP,3)
          KQ(2)=IS(MP,4)
          DO 120 IQ=1,4
              NP=IE(M,IQ)
              XQ(IQ)=X(NP)
              ZQ(IQ)=Z(NP)
              VXQ(IQ)=VX(NP)*W1+VXP(NP)*W2
              VZQ(IQ)=VZ(NP)*W1+VZP(NP)*W2
  120         CONTINUE
C
          CALL Q4SP(DFLXQ,KQ,DL(MP),DCOSX(MP),DCOSZ(MP),XQ,ZQ,VXQ, VZQ)
C
          IB=IHBP
```

```
      JB=IHBP+(NJ-NI)
C
      IF(IMID.NE.0) GO TO 121
C
      C(NI,IB)=C(NI,IB)+W1*DFLXQ(1,1)
      C(NI,JB)=C(NI,JB)+W1*DFLXQ(1,2)
      R(NI)=R(NI)-W2*(DFLXQ(1,1)*RP(NI)+DFLXQ(1,2)*RP(NJ))
      GO TO 123
  121 C(NI,IB)=C(NI,IB) + DFLXQ(1,1)
      C(NI,JB)=C(NI,JB) + DFLXQ(1,2)
C
  123 JB=IHBP
      IB=IHBP+(NI-NJ)
      IF(IMID.NE.0) GO TO 125
C
      C(NJ,JB)=C(NJ,JB)+W1*DFLXQ(2,2)
      C(NJ,IB)=C(NJ,IB)+W1*DFLXQ(2,1)
      R(NJ)=R(NJ)-W2*(DFLXQ(2,1)*RP(NI)+DFLXQ(2,2)*RP(NJ))
      GO TO 130
  125 C(NJ,JB)=C(NJ,JB) + DFLXQ(2,2)
      C(NJ,IB)=C(NJ,IB) + DFLXQ(2,1)
  130 CONTINUE
C
  140 IF (NST.LE.0) GO TO 160
C
C MODIFY LOAD VECTOR FOR SURFACE TERMS OF THE FORM DR/DN=C
C
      DO 150 NPP=1,NSTN
      NP=NPST(NPP)
  150 R(NP)=R(NP)+DP(NPP)
C
  160 RETURN
      END
```

```
      SUBROUTINE Q4SP(DFLXQ,KQ,DL,DCOSXQ,DCOSZQ,XQ,ZQ,VXQ,VZQ)
C
C
C     FUNCTION OF SUBROUTINE--TO EVALUATE THE SEEPAGE-FLUX INTEGRALS
C     ALONG THE BOUNDARY LINE EXTENDING FROM NODE LQ TO NODE MQ.
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N(4)
C
      DIMENSION W(4),DNX(4),DNZ(4),DWX(4),DWZ(4)
      DIMENSION PJAB(2,2),DNSS(4),DNTT(4),DWSS(4),DWTT(4)
      DIMENSION S(4,4),T(4,4),DFLXQ(2,2),KQ(2),SSA(2),TTA(2),
     > XQ(4),ZQ(4),AKXQ(4),AKZQ(4),VXQ(4),VZQ(4)
C
      DATA S/0.D0,-.57735D0,0.D0,-1.D0,.57735D0,0.D0,1.D0,0.D0, 0.D0,
     > 1.D0,0.D0,-.57735D0,-1.D0,0.D0,.57735D0,0.D0/, T/0.D0,-1.D0,0.D0,
     >  -.57735D0,-1.D0,0.D0,-.57735D0,0.D0,0.D0, .57735D0,0.D0,1.D0,
     > .57735D0,0.D0,1.D0,0.D0/
C
C     INITIALIZE NODAL COMPONENTS OF LINE INTEGRAL
C
      DO 10 JQ=1,2
         DO 10 IQ=1,2
10          DFLXQ(IQ,JQ)=0.
C
C     DETERMINE LOCAL COORDINATES OF GAUSS-INTEGRATION POINTS KG
C
      LQ=KQ(1)
      MQ=KQ(2)
      SSA(1)=S(LQ,MQ)
      TTA(1)=T(LQ,MQ)
      SSA(2)=S(MQ,LQ)
      TTA(2)=T(MQ,LQ)
      DO 40 KG=1,2
         SS = SSA(KG)
         TT = TTA(KG)
C
C     CALCULATE VALUES OF BASIS AND WEIGHTING FUNCTIONS N(IQ) AND W(IQ)
C     AND THEIR DERIVATIVES DWX DNX AND DWZ DNZ WITH RESPECT TO X AND Z
C     RESPECTIVELY, AT THE GAUSS POINT KG
C
         CALL SHAPE(N,W,DNSS,DNTT,DWSS,DWTT,SS,TT)
C
C
C     COMPUTE THE INVERSE TRANSFORMATION MATRIX I.E. THE INVERSE JACOBIAN:
C     PJAB(1,1)=(DZ/DT)/DJAC, PJAB(1,2)=-(DZ/DS)/DJAC,
C     PJAB(2,1)=-(DX/DT)/DJAC, PJAB(2,2)=(DX/DS)/DJAC, WHERE
C     DJAC IS THE DETERMINANT OF THE JACOBIAN
C
```

```
      DO 11 I=1,2
         DO 11 J=1,2
            PJAB(I,J)=0.0
 11         CONTINUE
C
      DO 16 I=1,4
         PJAB(1,1)=PJAB(1,1) + ZQ(I)*DNTT(I)
         PJAB(1,2)=PJAB(1,2) - ZQ(I)*DNSS(I)
         PJAB(2,1)=PJAB(2,1) - XQ(I)*DNTT(I)
         PJAB(2,2)=PJAB(2,2) + XQ(I)*DNSS(I)
 16      CONTINUE
C
C
C     COMPUTE THE DETERMINANT OF THE JACOBIAN
C
      DJAC=PJAB(2,2)*PJAB(1,1) - PJAB(1,2)*PJAB(2,1)
      DJACI=1.0D0/DJAC
C
      DO 17 I=1,2
         DO 17 J=1,2
 17         PJAB(I,J)=PJAB(I,J)*DJACI
C
      DO 18 I=1,4
         DNX(I)=PJAB(1,1)*DNSS(I) + PJAB(1,2)*DNTT(I)
         DNZ(I)=PJAB(2,1)*DNSS(I) + PJAB(2,2)*DNTT(I)
         DWX(I)=PJAB(1,1)*DWSS(I) + PJAB(1,2)*DWTT(I)
 18      DWZ(I)=PJAB(2,1)*DWSS(I) + PJAB(2,2)*DWTT(I)
C
C
C     INTERPOLATE WITH FUNCTIONS N(IQ), DNX(IQ), DNZ(IQ) TO OBTAIN
C     VALUES OF DARCY VELOCITIES VXQP AND VZQP
C
      VXK=0.
      VZK=0.
      DO 20 IQ=1,4
         VXK=VXK+VXQ(IQ)*N(IQ)
 20      VZK=VZK+VZQ(IQ)*N(IQ)
C
C     EVALUATE THE NORMAL DARCY VELOCITY AT THE GAUSS POINT AND ACCUMULATE
C     THE INTEGRAL SUMS
C
      VNK=VXK*DCOSXQ+VZK*DCOSZQ
      IF(VNK.LE.0.0) GO TO 40
      DO 30 JQ=1,2
         MQ=KQ(JQ)
         DO 30 IQ=1,2
            LQ=KQ(IQ)
 30         DFLXQ(IQ,JQ)=DFLXQ(IQ,JQ)+W(LQ)*VNK*N(MQ)
 40   CONTINUE
   DO 50 JQ=1,2
```

```
      DO 50 IQ=1,2
50       DFLXQ(IQ,JQ)=.5*DL*DFLXQ(IQ,JQ)
   RETURN
   END
```

```
      SUBROUTINE SOLVE(KKK,C,R,NNP,IHALFB,MAXNP,MAXBW)
C
C
C     FUNCTION OF SUBROUTINE--TO SOLVE THE MATRIX EQUATION CX = R,
C     RETURNING THE SOLUTICN X IN R.  IT IS ASSUMED THAT THE ARRAY C(NP,IB)
C     CONTAINS THE FULL BAND OF AN ASYMMETRIC MATRIX.
C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION C(MAXNP,MAXBW),R(MAXNP)
C
      IHBP=IHALFB+1
C
C     IF KKK = 1, THEN TRIANGULARIZE THE BAND MATRIX C(NP,IB), BUT
C     IF KKK = 2, THEN SIMPLY SOLVE WITH THE RIGHT-HAND SIDE R(NP)
C
      IF (KKK.EQ.2) GO TO 50
C
C     TRIANGULARIZE MATRIX C(NP,IB)
C
      NU=NNP-IHALFB
      DO 20 NI=1,NU
          PIVOTI=1./C(NI,IHBP)
          NJ=NI+1
          IB=IHBP
          NK=NI+IHALFB
          DO 10 NL=NJ,NK
              IB=IB-1
              A=-C(NL,IB)*PIVOTI
              C(NL,IB)=A
              JB=IB+1
              KB=IB+IHALFB
              LB=IHBP-IB
              DO 10 MB=JB,KB
                  NB=LB+MB
   10             C(NL,MB)=C(NL,MB)+A*C(NI,NB)
   20     CONTINUE
      NR=NU+1
      NU=NNP-1
      NK=NNP
      DO 40 NI=NR,NU
          PIVOTI=1./C(NI,IHBP)
          NJ=NI+1
          IB=IHBP
          DO 30 NL=NJ,NK
              IB=IB-1
              A=-C(NL,IB)*PIVOTI
              C(NL,IB)=A
              JB=IB+1
```

APPENDIX C (continued)

```
              KB=IB+IHALFB
              LB=IHBP-IB
              DO 30 MB=JB,KB
                 NB=LB+MB
   30            C(NL,MB)=C(NL,MB)+A*C(NI,NB)
   40       CONTINUE
         RETURN
C
C     MODIFY LOAD VECTOR R(NP)
C
   50    NU=NNP+1
         IBAND=2*IHALFB+1
         DO 70 NI=2,IHBP
            IB=IHBP-NI+1
            NJ=1
            SUM=0.0
            DO 60 JB=IB,IHALFB
               SUM=SUM+C(NI,JB)*R(NJ)
   60          NJ=NJ+1
   70       R(NI)=R(NI)+SUM
         IB=1
         NL=IHBP+1
         DO 90 NI=NL,NNP
            NJ=NI-IHBP+1
            SUM=0.0
            DO 80 JB=IB,IHALFB
               SUM=SUM+C(NI,JB)*R(NJ)
   80          NJ=NJ+1
   90       R(NI)=R(NI)+SUM
C
C     BACK SOLVE
C
         R(NNP)=R(NNP)/C(NNP,IHBP)
         DO 110 IB=2,IHBP
            NI=NU-IB
            NJ=NI
            MB=IHALFB+IB
            SUM=0.0
            DO 100 JB=NL,MB
               NJ=NJ+1
  100          SUM=SUM+C(NI,JB)*R(NJ)
  110       R(NI)=(R(NI)-SUM)/C(NI,IHBP)
         MB=IBAND
         DO 130 IB=NL,NNP
            NI=NU-IB
            NJ=NI
            SUM=0.0
            DO 120 JB=NL,MB
               NJ=NJ+1
  120          SUM=SUM+C(NI,JB)*R(NJ)
```

```
130    R(NI)=(R(NI)-SUM)/C(NI,IHBP)                                    SOLV 505
      RETURN                                                           SOLV 510
      END                                                              SOLV 515
```

```
      SUBROUTINE SFLOW(X,Z,IE,WETAB,R,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,
     > NBE,ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,
     > MAXEL,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,
     > DELT,DH)
C
C   FUNCTION OF SUBROUTINE--TO COMPUTE BOUNDARY FLUXES, FLOW RATES,
C   INCREMENTAL FLOWS OCCURRING DURING TIME DELT, TOTAL FLOWS SINCE
C   TIME ZERO, AND THE CHANGE IN MOISTURE CONTENT FOR THE ENTIRE
C   SYSTEM DURING TIME DELT.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KD,LAMBDA
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
      DIMENSION R(MAXNP),FX(MAXNP),FZ(MAXNP),DH(MAXNP),TH(MAXEL,4)
      DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),BFLX(MAXBNP),
     > BFLXP(MAXBNP),NBE(MAXBEL),ISB(MAXBEL,4),NPB(MAXBNP)
      DIMENSION NPRS(MXRSNP),NPST(MXSTNP),NPN(MAXBCN)
      DIMENSION PROP(MAXMAT,MXMPPM)
      DIMENSION FRATE(10),FLOW(10),TFLOW(10)
C
      DIMENSION XQ(4),ZQ(4),RQ(4),DHQ(4),THRQ(4),THQ(4)
C
      COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
      COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN,NST
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
      DATA QR,QD,QL,QDH /0.0D0,0.0D0,0.0D0,0.0D0/
C
C
C   CALCULATE NODAL FLOW RATES
C
      DO 10 NP=1,NBN
         BFLXP(NP)=BFLX(NP)
   10    BFLX(NP)=0.
C
      DO 30 MP=1,NBEL
         M=NBE(MP)
         NI=ISB(MP,1)
         NJ=ISB(MP,2)
         DO 20 I=1,NBN
            IJ=NPB(I)
            IF(IJ.NE.NI) GO TO 20
            NII=I
            GO TO 22
   20       CONTINUE
   22    DO 25 J=1,NBN
            IJ=NPB(J)
            IF(IJ.NE.NJ) GO TO 25
            NJJ=J
```

```
            GO TO 27
25          CONTINUE
27       CONTINUE
         FNI=(FX(NI)*DCOSXB(MP)+FZ(NI)*DCOSZB(MP))*DLB(MP)
         FNJ=(FX(NJ)*DCOSXB(MP)+FZ(NJ)*DCOSZB(MP))*DLB(MP)
         BFLX(NII)=BFLX(NII)+FNI/3.0+FNJ/6.0
         BFLX(NJJ)=BFLX(NJJ)+FNJ/3.0+FNI/6.0
30       CONTINUE
C
C
C     DETERMINE FLOWS AND FLOW RATES THRU THE VARIOUS
C     TYPES OF BOUNDARY NODES, STARTING WITH THE
C     NET FLOWS THROUGH ALL BOUNDARY NODES
C
      S=0.
      SP=0.
      DO 40 NP=1,NBN
         S=S+BFLX(NP)
40       SP=SP+BFLXP(NP)
      FRATE(5)=S
      FLOW(5)=.5*(S+SP)*DELT
C
C     CONSTANT DIRICHLET BOUNDARY NODES
C
      FRATE(1)=0.
      FLOW(1)=0.
      IF (NBC.LE.0) GO TO 60
      S=0.
      SP=0.
      DO 50 NPP=1,NBC
         NP=NPN(NPP)
         DO 45 I=1,NBN
            IJ=NPB(I)
            IF(IJ.NE.NP) GO TO 45
            NII=I
            GO TO 46
45          CONTINUE
46       CONTINUE
         S=S+BFLX(NII)
50       SP=SP+BFLXP(NII)
      FRATE(1)=S
      FLOW(1)=.5*(S+SP)*DELT
C
C     CONSTANT NEUMANN BOUNDARY NODES
C
60    FRATE(2)=0.
      FLOW(2)=0.
      IF (NST.LE.0) GO TO 80
      S=0.
      SP=0.
```

```
      DO 70 NPP=1,NSTN
         NP=NPST(NPP)
         DO 65 I=1,NBN
            IJ=NPB(I)
            IF(IJ.NE.NP) GO TO 65
            NII=I
            GO TO 66
   65       CONTINUE
   66    CONTINUE
         S=S+BFLX(NII)
   70    SP=SP+BFLXP(NII)
      FRATE(2)=S
      FLOW(2)=.5*(S+SP)*DELT
C
C  TRANSIENT SEEPAGE BOUNDARY NODES
C
   80 FRATE(3)=0.
      FLOW(3)=0.
      IF(NRSEL.LE.0) GO TO 100
      S=0.
      SP=0.
      DO 90 NPP=1,NRSN
         NP=NPRS(NPP)
         DO 85 I=1,NBN
            IJ=NPB(I)
            IF(IJ.NE.NP) GO TO 85
            NII=I
            GO TO 86
   85       CONTINUE
   86    CONTINUE
         S=S+BFLX(NII)
   90    SP=SP+BFLXP(NII)
      FRATE(3)=S
      FLOW(3)=.5*(S+SP)*DELT
C
C  NUMERICAL FLOW THROUGH UNSPECIFIED BOUNDARY NODES
C
  100 S=0.
      SP=0.
      DO 110 I=1,3
         S=S+FRATE(I)
  110    SP=SP+FLOW(I)
      FRATE(4)=FRATE(5)-S
      FLOW(4)=FLOW(5)-SP
C
C  CALCULATE THE INCREASES IN THE INTEGRATED MATERIAL CONTENTS FOR THE
C  FLUID AND THE SOLID PHASES AND DETERMINE LOSSES DUE TO RADIOACTIVE
C  DECAY
C
      QRP=QR
```

```
      QDP=QD
      QLP=QL
      QDHP=QDH
      QR=0.
      QD=0.
      QL=0.
      QDH=0.0
      DO 130 M=1,NEL
         APHA1=WETAB(M,1)
         APHA2=WETAB(M,3)
         BETA1=WETAB(M,2)
         BETA2=WETAB(M,4)
         MTYP=IE(M,5)
         KD=PROP(MTYP,1)
         RHOB=PROP(MTYP,2)
         LAMBDA=PROP(MTYP,5)
         POR=PROP(MTYP,6)
         ALP=PROP(MTYP,7)
         DO 120 IQ=1,4
            NP=IE(M,IQ)
            XQ(IQ)=X(NP)
            ZQ(IQ)=Z(NP)
            DHQ(4)=DH(NP)
            THQ(IQ)=TH(M,IQ)
            RQ(IQ)=R(NP)
  120       THRQ(IQ)=THQ(IQ)*RQ(IQ)
C
C
         CALL Q4R(QRM,QDM,QDHM,XQ,ZQ,RQ,DHQ,THRQ,THQ,RHOB,KD)
         QR=QR+QRM
         QD=QD+QDM*RHOB*KD
         QLM=QRM+QDM
         QL=QL+LAMBDA*QLM
         QDH=QDH+ALP*QDHM
  130    CONTINUE
      FLOW(6)=QR-QRP
      FRATE(6)=FLOW(6)/DELT
      FLOW(7)=QD-QDP
      FRATE(7)=FLOW(7)/DELT
      FRATE(8)=.5*(QL+QLP)
      FLOW(8)=DELT*FRATE(8)
      FRATE(9)=QDH
      FLOW(9)=DELT*FRATE(9)
      DO 140 I=1,9
  140    TFLOW(I)=TFLOW(I)+FLOW(I)
      RETURN
      END
```

APPENDIX C (continued)

```
      SUBROUTINE Q4R(QRM,QDM,QDHM,XQ,ZQ,RQ,DHQ,THRQ,THQ,RHOB,KD)
C
C
C     FUNCTION OF SUBROUTINE--TO EVALUATE THE CONCENTRATION INTEGRAL
C     OVER THE AREA OF ONE ELEMENT.
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N(4),KD
C
      DIMENSION XQ(4),ZQ(4),RQ(4),DHQ(4),THRQ(4),THQ(4)
      DIMENSION S(4),T(4)
      DIMENSION W(4)
      DIMENSION PJAB(2,2),DNSS(4),DNTT(4),DWSS(4),DWTT(4)
C
      DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-
     > 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /
C
C     EVALUATE QUANTITIES FOR USE IN THE JACOBIAN DJAC, BELOW, NECESSARY
C     FOR TRANSFORMATION FROM GLOBAL TO LOCAL COORDINATES
C
      QRM=0.
      QDM=0.0
      QDHM=0.0
      DO 20 KG=1,4
C
C     DETERMINE LOCAL COORDINATES (SS,TT) OF GAUSS-INTEGRATION POINT KG
C
         SS = P*S(KG)
         TT = P*T(KG)
C
C     EVALUATE THE JACOBIAN DJAC
C
         CALL SHAPE(N,W,DNSS,DNTT,DWSS,DWTT,SS,TT)
C
         DO 11 I=1,2
            DO 11 J=1,2
               PJAB(I,J)=0.0
 11            CONTINUE
         DO 16 I=1,4
            PJAB(1,1)=PJAB(1,1) + ZQ(I)*DNTT(I)
            PJAB(1,2)=PJAB(1,2) - ZQ(I)*DNSS(I)
            PJAB(2,1)=PJAB(2,1) - XQ(I)*DNTT(I)
            PJAB(2,2)=PJAB(2,2) + XQ(I)*DNSS(I)
 16         CONTINUE
         DJAC=PJAB(2,2)*PJAB(1,1) - PJAB(1,2)*PJAB(2,1)
C
C     INTERPOLATE TO OBTAIN THE CONCENTRATION RQP AT THE GAUSS POINT KG
C
         RQP=0.
```

```
      DHQP=0.0
      THRQP=0.0
      THQP=0.0
      DO 10 IQ=1,4
         RQP=RQP+RQ(IQ)*N(IQ)
         DHQP=DHQP+DHQ(IQ)*N(IQ)
         THQP=THQP+THQ(IQ)*N(IQ)
10       THRQP=THRQP+THRQ(IQ)*N(IQ)
C
C  ACCUMULATE THE SUM TO EVALUATE THE INTEGRAL QRM
C
      QRM=QRM+THRQP*DJAC
      QDM=QDM+RQP*DJAC
      QDHM=QDHM+(THQP+RHOB*KD)*DHQP*RQP*DJAC
20    CONTINUE
    RETURN
    END
```

```
      SUBROUTINE PRINTT(R,FX,FZ, FRATE,FLOW,TFLOW, TIME,DELT,KPR,KOUT,
     > KDIG,MAXNP,NNP,IBAND)
C
C
C   FUNCTION OF SUBROUTINE--TO OUTPUT FLOWS, CONCENTRATIONS, MATERIAL
C   FLUXES, WATER CONTENTS, DARCY VELOCITIES, PRESSURE HEADS, AND
C   TOTAL HEADS AS SPECIFIED BY THE PARAMETER KPR.
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION R(MAXNP),FX(MAXNP),FZ(MAXNP)
      DIMENSION FRATE(10),FLOW(10),TFLOW(10)
      IMPLICIT REAL*8 (A-H,O-Z)
      IF (KPR.EQ.0) RETURN
      IF (KOUT.EQ.0) GO TO 10
C
C   PRINT DIAGNOSTIC FLOW INFORMATION
C
      KDIG=KDIG+1
      PRINT 10600,KDIG,TIME,DELT
      PRINT 10500,(FRATE(I),FLOW(I),TFLOW(I),I=1,9)
   10 IF (KPR.EQ.1) RETURN
C
C   PRINT CONCENTRATIONS
C
      KOUT=KOUT+1
      PRINT 10000,KOUT,TIME,DELT,IBAND
      KLINE=-1
      DO 20 NI=1,NNP,8
         NJMN=NI
         NJMX=MIN0(NI+7,NNP)
         KLINE=KLINE+1
         IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) PRINT 10000,KOUT,
     >     TIME,DELT,IBAND
   20    PRINT 10100,NI,(R(NJ),NJ=NJMN,NJMX)
      IF (KPR.EQ.2) RETURN
C
C   PRINT MATERIAL FLUX
C
      KOUT=KOUT+1
      PRINT 10200,KOUT,TIME,DELT,IBAND
      KLINE=-1
      DO 30 NI=1,NNP,4
         NJMN=NI
         NJMX=MIN0(NI+3,NNP)
         KLINE=KLINE+1
         IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) PRINT 10200,KOUT,
     >     TIME,DELT,IBAND
   30    PRINT 10300,(NJ,FX(NJ),FZ(NJ),NJ=NJMN,NJMX)
      RETURN
```

```
C
10000 FORMAT(13H1OUTPUT TABLE,I4,27H.. CONCENTRATIONS AT TIME =,
     > 1PD12.4,9H ,(DELT =,1PD12.4,15H),(BAND WIDTH =,I4,1H)//
     > 7H NODE I,5X,36HCONCENTRATION AT NODES I,I+1,....,I+7/)
10100 FORMAT(I7,8(1PD15.4))
10200 FORMAT(13H1OUTPUT TABLE,I4,26H.. MATERIAL FLUX AT TIME =, 1PD12.4,
     >  9H ,(DELT =,1PD12.4,15H),(BAND WIDTH =,I4,1H)//1X,
     > 4HNODE,9X,2HFX,9X,2HFZ,6X,4HNODE,9X,2HFX,9X,2HFZ,6X,4HNODE,9X,
     > 2HFX,9X,2HFZ,6X,4HNODE,9X,2HFX,9X,2HFZ/)
10300 FORMAT(4(I5,2E11.3,5X))
10500 FORMAT(//5X,13H TYPE OF FLOW,35X,4HRATE,8X,9HINC. FLOW,7X,
     > 10HTOTAL FLOW/5X,40H CONSTANT-CONCENTRATION NODE FLOW. . . .,3(
     > E12.4,5X)/5X,40H CONSTANT-FLUX-NODE FLOW . . . . . . . . .,3(E12.4,
     >  5X)/5X,40H SEEPAGE FLUX-NODE FLOW. . . . . . . . . .,3(E12.4,5X)/
     > 5X,40H NUMERICAL LOSSES. . . . . . . . . . . . .,3(E12.4,5X)/5X,
     > 40H NET FLOW. . . . . . . . . . . . . . . . . .,3(E12.4,5X)/5X,
     > 40H INCREASE IN MATERIAL CONTENT (LIQUID) .,3(E12.4,5X)/5X,
     > 40H INCREASE IN MATERIAL CONTENT (SOLID). .,3(E12.4,5X)/5X,
     > 40H RADIOACTIVE LOSSES (LIQUID AND SOLID) .,3(E12.4,5X)/5X,
     > 40H INCREASE DUE TO COMP. OF SKELTON ..... ,3(E12.4,5X))
10600 FORMAT(1H1,52H****************************************************
     > 62H**************************************************************
     > 5H*****///18H SYSTEM-FLOW TABLE,I4,12H.. AT TIME =,1PD12.4,
     > 9H ,(DELT =,1PD12.4,1H))
      END
```

APPENDIX C (continued)

```
      SUBROUTINE STORE(X,Z,IE,R,FX,FZ,TITLE,NPROB,NNP,NEL,NTI,
     > MAXNP,MAXEL,TIME)
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),
     > FZ(MAXNP)
C
      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),
     > IQ=1,4)
      NPPROB=NPROB
   10 WRITE(2) TIME,(R(NP),NP=1,NNP),(FX(NP),NP=1,NNP),(FZ(NP),NP=1,NNP)
      RETURN
      END
```

ORNL-5601
Distribution Category UC-70

INTERNAL DISTRIBUTION

1-2. S. I. Auerbach
3. R. E. Blanco
4. J. H. Coobs
5. R. B. Craig
6. N. H. Cutshall
7. R. B. Fitts
8. D. D. Huff
9. E. M. King
10. M. E. LaVerne
11. C. A. Little
12. A. L. Lotts
13-14. R. S. Lowrie
15. R. J. Luxmoore
16. M. S. Moran
17. K. J. Notz
18 T. W. Oakes
19. H. A. Pfuderer
20. R. A. Robinson
21. T. H. Row
22. B. P. Spalding
23. S. Stow
24. L. E. Stratton
25. T. Tamura
26. D. B. Trauger
27. E. K. Triegel
28. J. E. Vath
29. N. D. Vaughan
30-34. G. T. Yeh
35-36. Central Research Library
37. Engineering Physics Information Center (EPIC)
38-48. ESD Library
49-50. Laboratory Records Dept.
51 Laboratory Records, ORNL-RC
52. ORNL Y-12 Technical Library
53. ORNL Patent Office

EXTERNAL DISTRIBUTION

54. J. Apel, Pacific Marine Environmental Research Laboratory, NOAA, 3711-15th Ave. NE, Seattle, WA 98105
55. R. G. Baca, Energy System Group, Rockwell Hanford Operation, Richland, WA 99352
56. R. W. Brocksen, Electric Power Research Institute, 3412 Hillview Ave., P.O. Box 10412, Palo Alto, CA 94303
57. W. Brutsaert, School of Civil Engineering, Cornell University, Ithaca, NY 14850
58. S. L. Chiu, Department of Civil Engineering, University of Pittsburg, Pittsburg, PA 15219
59. Ven Te Chow, Department of Civil Engineering, University of Illinois, Champaign/Urbana, IL 61801
60. David S. L. Chu, Taiwan Power Company, Taipei, Taiwan
61. J. J. Connor, Department of Civil Engineering, MIT, Cambridge, MA 02139
62. William J. Coppoc, Texaco, Inc., P.O. Box 509, Beacon, NY 12508
63. G. H. Daly, Acting Director, Technology Division, Office of Waste Operations and Technology, Department of Energy, MS B-107, Washington, DC 20545
64. J. L. Dieckhoner, Director, Operations Division, Office of Waste Operations and Technology, Department of Energy, Washington, DC 20545
65. J. O. Duguid, Battelle Columbus Laboratories, 505 King Avenue, Columbus, OH 43201
66. A. L. Edwards, Lawrence Livermore Laboratory, University of California Livermore, Livermore, CA 94550
67. E. Fray, EG&G Idaho, P.O. Box 1625, Idaho Falls, ID 83401

68. J. F. Geiger, Atlantic Richfield Hanford Company, P.O. Box 250, Richland, WA 99352
69. Donald R. F. Harleman, Ralph M. Parsons Laboratory, Department of Civil Engineering, MIT, Cambridge, MA 02139
70. W. A. Jack, Senior Vice President, The Ralph M. Parsons Company 100 West Walnut Street, Pasadena, CA 91124
71 D. Jacobs, Evaluation Research Corporation, Oak Ridge, TN 37830
72. Jack Kennedy, Institute of Hydraulics, Iowa State University, Iowa City, IA 52240
73-77. M. L. Krupinski, Program Manager, National Low Level Waste Program, EG&G Idaho, P.O. Box 1625, Idaho Falls, ID 83401
78. D. Jacobs, Evaluation Research Corporation, Oak Ridge, TN 37830
79. Elizabeth Jordan, Operations Division, Office of Waste Operations and Technology, Department of Energy, MS B-107, Washington, DC 20545
80-85. D. E. Large, Manager, ORO Low-Level Radioactive Waste Management Programs, Department of Energy, Oak Ridge Operations, P.O. Box E, Oak Ridge, TN 37830
86. L. Law, Energy System Group, Rockwell Hanford Operation, Richland, WA 99352
87. Joan Lefler, Environmental Review Division, Office of Testing and Evaluation, Office of Toxic Substances, Environmental Protection Agency, 401 M Street, SW, Washington, DC 20460
88. J. A. Lenhard, Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations, P.O. Box E, Oak Ridge, TN 37830
89. J. W. Lentsch, Portland Electric Company, 121 S. W. Salmon St., Portland, OR 97201
90. G. B. Levin, EG&G Idaho, P.O. Box 1625, Idaho Falls, ID 83401
91. Robert A. Lewis, EV-34, GTN, Department of Energy, Washington, DC 20545
92. J. A. Liggett, School of Civil Engineering, Cornell University, Ithaca, NY 14850
93. Ray K. Linsley, Department of Civil Engineering, Stanford University, Stanford, CA 94305
94. Y. C. Lu, Department of Civil Engineering, National Taiwan University, Taipei, Taiwan
95. A. D. Martin, Las Arenas (Bilbao), Avda. Del Trinnfo, 56, SENER, Tecnica Industrial Y Naval S.A., Spain
96. Helen McCammon, Director, Division of Ecological Research, Office of Health and Environmental Research, Department of Energy, Washington, DC 20545
97. J. D. McKinney, EG&G Idaho, Inc., P.O. Box 1265, Idaho Falls, ID 83401
98. R. W. Nelson, Battelle Northwest, Richland, WA 99352
99. R. Nelson, Program Manager, Radioactive Waste Programs Branch, Nuclear Fuel Cycle Division, Idaho Operations Office, Department of Energy, 550 Second Street, Idaho Falls, ID 83401

100. G. Oertel, Director, Office of Waste Operations and Technology, Department of Energy, MS B-107, Washington, DC 20545
101. William S. Osburn, Jr., Division of Ecological Research, Office of Health and Environmental Research, Department of Energy, Washington, DC 20545
102. S. S. Papadoperlos, U. S. Geological Survey, Reston, VA 22092
103. A. Perge, Office of the Deputy Assistant Secretary for Nuclear Waste Management, Department of Energy, Washington, DC 20545
104. G. Pinder, Dept. of Civil Engineering, Princeton University, Princeton, NJ 08540
105. A. E. Potter, Johnson Space Center, NASA, Houston, TX 77058
106. James J. Reisa, Office of Toxic Substances, Division of Environmental Review, 401 M Street, SW, Environmental Protection Agency, Washington, DC 20460
107. Paul G. Risser, Department of Botany and Microbiology, University of Oklahoma, Norman, OK 73019
108. J. Rubin, U. S. Geological Survey, Menlo Park, CA 94306
109. H. S. Shen, Department of Civil Engineering, Colorado University, Fort Collins, CO 80521
110. George R. Shepherd, Department of Energy, Rm 4G-052, MS 4G-085, Forestal Building, Washington, DC 20545
111. J. G. Steger, Health Research Division, University of California, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87545
112. Victor L. Streeter, Department of Civil Engineering, University of Michigan, Ann Arbor, MI 48106
113. R. C. Swartz, U.S. Environmental Protection Agency at the Corvallis Environmenmtal Research Laboratory, 200 SW 35th St., Corvallis, OR 97330
114. W. Viessman, Jr., Department of Civil Engineering, University of Nebraska, Lincoln, NE 68501

115-119. David S. Ward, INTERA Environmental Consultants, Inc., 11511 Katy Freeway, Suite 630, Houston, TX 77079

120. Richard H. Waring, Department of Forest Science, Oregon State University, Corvallis, OR 97331
121. Robert L. Watters, Office of Health and Environmental Research, Department of Energy, Washington, DC 20545
122. J. B. Whitsett, Chief, Radioactive Waste Programs Branch, Nuclear Fuel Cycle Division, Idaho Operations Office, Department of Energy, 550 Second Street, Idaho Falls, ID 83401
123. P. A. Witherspoon, Lawrence Berkeley Laboratory, University of California, Berkeley, CA 94720
124. Robert W. Wood, Office of Health and Environmental Research, Department of Energy, Washington, DC 20545
125. O. C. Zienkiewicz, Department of Civil Engineering, University of Wales, Swansea, England

126-486. Given distribution as shown in TIC-4500 under UC-70, Nuclear Waste Management

☆ U.S. GOVERNMENT PRINTING OFFICE: 1981-740-062/3