365
4/30/81

U.S.
ORNL

OAK
RIDGE
NATIONAL
LABORATORY

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

Dr. 26-29

2

R-39170

MASTER

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

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

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

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

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 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

	<u>Page</u>
ABSTRACT	v
LIST OF TABLES	ix
LIST OF FIGURES	xi
I. INTRODUCTION	1
II. MATHEMATICAL STATEMENTS	3
1. Governing Equations and Initial and Boundary Conditions	3
2. Distribution Coefficient and Retardation Factor . .	8
3. Finite-Element Approximations.	11
4. Mass Lumping Option.	14
5. Time-Marching Methods	15
6. Base and Upstream Weighting Functions	17
7. Numerical Integration	22
8. Assembly of Element Matrix	25
9. Application of Boundary Conditions	25
10. Solution of Assembled Equations	27
11. Mass Balance Computation	27
12. Computation of Total Waste Flux	29
13. Alternative Numerical Schemes	30
III. COMPUTER PROGRAM MODIFICATION AND EXPANSION	33
IV. RESULTS.	37
V. NOTATIONS	62
VI. REFERENCES	64
APPENDIX A. DATA INPUT GUIDE.	69
APPENDIX B. SAMPLE INPUT OF THE SEEPAGE POND PROBLEM	81
APPENDIX C. LISTING OF FORTRAN IV SOURCE PROGRAM	85

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Optional finite element schemes	32
2	Comparison of percentage of mass-balance error as simulated by alternative numerical schemes of the original and new models, respectively	45

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1	Spatial boundaries of flow region, R	6
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 (——)	10
3	Sketch of possible dependence of effective bulk density on the moisture content	12
4	A typical finite-element in global and local coordinates	18
5	Location and direction of upwind weighting functions along the edge of a transformed element	20
6	Optimal values of the upwind weighting parameter for different values of Peclet number	20
7	Subroutine chart of waste transport program.	34
8	Subroutine chart for the time-iteration loop	35
9	Finite-element discretization of the seepage pond problem	38
10	Hydraulic conductivity and soil-moisture characteristics for a hypothetical sandy soil	39
11	Flow variables simulated by the Reeves and Duguid model: (a) velocity vector, (b) pressure head, (c) total head, (d) moisture content	40
12	Flow variables simulated by the Yeh and Ward model: (a) velocity vector, (b) pressure head, (c) total head, (d) moisture content	41
13	Normalized concentration distribution for a dissolved constituent with a distribution coefficient of 100 ml/g computed with (a) original models (b) new models	43

<u>Figure</u>		<u>Page</u>
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	44
15	Comparison of mass-balance-rate error history as simulated by numerical scheme 1 of the original and new models, respectively	47
16	Comparison of mass-balance-rate error history as simulated by numerical scheme 2 of the original and new models, respectively	48
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	50
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	51
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	52
20	Concentration distribution as simulated by numerical scheme 4 of the new model at time equal to (a) 1.01, (b) 7.67, (c) 14.35, and (d) 19.34 years, respectively	53
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	54
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	55
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	56

<u>Figure</u>		<u>Page</u>
24	Concentration distribution as simulated by numerical scheme 8 of the new model at time equal to (a) 1.01, (b) 7.57, (c) 14.35, and (d) 19.34 years, respectively	57
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	58
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	59
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	60
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	61

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 Nijkamp 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):

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

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 constituent 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):

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

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

and

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

where $V = \sqrt{V_x^2 + V_z^2}$; a_T and a_L are the transverse and longitudinal dispersivities, respectively; T 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:

$$s = K_d c, \quad (3)$$

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

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

where

$$R_d = 1 + \frac{\rho K_d}{\theta} \quad (5)$$

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:

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

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:

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

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:

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

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:

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

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 impervious boundary B_1 constitute 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 Chaii 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), ρ_s represents the total amount of constituent absorbed in the soil matrix per unit bulk volume. Thus for extreme unsaturation, ρ_s 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:

$$\rho = \rho_e \quad (10)$$

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

ORNL-DWG 79-14296

Fig. 3. Sketch of possible dependence of effective 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:

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

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,

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

one obtains the following element matrix equation for element e :

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

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

$$\dot{c}_j = \frac{dc_j}{dt} \quad (14)$$

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

$$M_{ij} = \int_{R_e} \theta W_i R_d N_j dR \quad (15)$$

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

$$S_{ij} = \int_{R_e} \left\{ \frac{\partial W_i}{\partial x} \left(\theta D_{xx} \frac{\partial N_j}{\partial x} + \theta D_{xz} \frac{\partial N_j}{\partial z} \right) + \frac{\partial W_i}{\partial z} \left(\theta D_{zx} \frac{\partial N_j}{\partial x} + \theta D_{zz} \frac{\partial N_j}{\partial z} \right) \right. \\ \left. - \left(\frac{\partial W_i}{\partial x} V_x N_j + \frac{\partial W_i}{\partial z} V_z N_j \right) + W_i \left(\frac{\partial \theta}{\partial t} + \alpha' \theta R_d \frac{\partial h}{\partial t} + \lambda \theta R_d \right) N_j \right\} dR \quad (16)$$

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

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

and

$$Q_i = \int_{B_e} W_i \left\{ (-\theta D_{xx} \frac{\partial c}{\partial x} - \theta D_{xz} \frac{\partial c}{\partial z} + V_x c) n_x + \right. \\ \left. (-\theta D_{zx} \frac{\partial c}{\partial x} - \theta D_{zz} \frac{\partial c}{\partial z} + V_z c) n_z \right\} dB \quad (18)$$

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, $[M_{ij}]$. More explicitly, $[M_{ij}]$ will be lumped according to

$$M_{ij} = \sum_{j=1}^4 \int_{R_e} \theta W_i R_d N_j dR \quad (19)$$

and

$$M_{ij} = 0 \text{ if } i \neq j, \quad (20)$$

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:

$$\begin{aligned} [M_{ij}] (\{c_j\}_{t+\Delta t} - \{c_j\}_t) / \Delta t + \frac{1}{2} \cdot [S_{ij}] (\{c_j\}_{t+\Delta t} + \{c_j\}_t) \\ + \{D_i\} + \{Q_i\} = 0 \end{aligned} \quad (21)$$

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

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

where $[M_{ij}]$, $[S_{ij}]$, $\{D_i\}$, and $\{Q_i\}$ 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:

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

and

$$\{c_j\}_{t+\Delta t} = 2 \{c_j\}_{t+\Delta t/2} - \{c_j\}_t \quad (23a)$$

where $[M_{ij}]$, $[S_{ij}]$, $\{D_i\}$, and $\{Q_i\}$ 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:

$$[C_{ij}] \{c_j\} = \{R_i\} - \{Q_i\} \quad (24)$$

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

$$[C_{ij}] = [M_{ij}]/\Delta t + \frac{1}{2} \cdot [S_{ij}] \quad (21a)$$

$$\{c_j\} = \{c_j\}_{t+\Delta t} \quad (21b)$$

and

$$\{R_j\} = [M_{ij}]\{c_j\}_t/\Delta t + \frac{1}{2} \cdot [S_{ij}]\{c_j\}_t - \{D_i\} \quad (21c)$$

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:

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

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.,

$$W_i = N_i, \quad i = 1, 2, 3, 4. \quad (26)$$

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

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

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)

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

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)

ORNL-DWG 78-20109

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 $\omega\Delta t/L$ as well as uL/D (Zienkiewicz and Parekh 1970). The relationship between α and two dimensionless parameters, $\omega\Delta t/L$ 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)

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

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:

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

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

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

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:

$$M_{ij} = \int_{-1}^1 \int_{-1}^1 W_i R_d \theta N_j J d\xi d\eta \quad (32)$$

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

and

$$D_i = \int_{-1}^1 \int_{-1}^1 -W_i M J d\xi d\eta \quad (34)$$

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

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

$$\begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{pmatrix} = [J] \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial z} \end{pmatrix}, \quad (35)$$

may be inverted to yield

$$\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial z} \end{pmatrix} = \frac{1}{J} \cdot \begin{bmatrix} \frac{\partial z}{\partial \eta} & -\frac{\partial z}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{pmatrix}, \quad (36)$$

using the definition of $[J]$ in Eq. (30).

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

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

Similarly,

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

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

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

and

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

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:

$$[T_{ij}] \{C_j\} = \{X_i\} - \{B_i\} = \{Y_i\} , \quad (41)$$

where $[T_{ij}]$ is the global coefficient matrix and $\{Y_i\}$ 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 $\{Q_i\}$. 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 over "

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:

$$\{Q_i\} = [E_{ij}] \{C_j\} \quad i = 1, 2 \quad \text{and} \quad j = 1, 2, \quad (42)$$

where

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

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 subtracted 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):

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

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

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

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:

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

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

$$F_0 = \int_{B_2} F_n dB \quad , \text{ and} \quad (48)$$

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

where F_D , F_N , F_0 , 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,

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

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

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

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

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

$$F_{\text{net}} = (F_D + F_N + F_0 + F_I) \quad , \quad (54)$$

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

$$F_v = F_w + F_A + F_L + F_S \quad (55)$$

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

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

and

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

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

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

and

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

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

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

where

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

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

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

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

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

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 compute the weighting parameters, α_1 , α_2 , β_1 and β_2 . SHAPE evaluates the base and weighting functions at the Gaussian integration point.

ORNL-DWG 79-13256

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 $[S'_{ij}]$. 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 $[S'_{ij}]$ and the load vectors $[R_{xi}]$ and $[R_{zi}]$. 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 boundaries. 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 and 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 $4.0 \times 10^{-4} \text{ cm}^3 \text{ cm}^{-2} \text{ sec}^{-1}$, 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 head, (c) total,

40

(a) VELOCITY VECTOR PLOT

0.004 cm s⁻¹

(b) DISTRIBUTION OF PRESSURE HEAD

- 500 cm
- 0 cm
- ▲ 1000 cm
- + 1500 cm

(c) DISTRIBUTION OF TOTAL HEAD

- 2000 cm
- 1200 cm
- + 800 cm

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

ORNL-5601

(a) VELOCITY VECTOR PLOT

0.004 cm s⁻¹

(b) DISTRIBUTION OF PRESSURE HEAD

- 500 cm
- 0 cm
- △ 1000 cm
- + 1500 cm

(c) DISTRIBUTION OF TOTAL HEAD

- 2000 cm
- 1200 cm
- + 800 cm

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

■ 2000 cm
 ○ 1200 cm
 + 800 cm

(d) DISTRIBUTION OF MOISTURE CONTENT

□ 0.25
 ○ 0.10
 △ 0.05
 + 0.025

velocity vector, (b) pressure head,

Figures 13a and 13b show the contaminant concentration distributions, which were obtained by applying the original and the new models, respectively. The distribution coefficient, $K_d = 100 \text{ cm}^3/\text{g}$ was used. Longitudinal and transverse dispersivities were taken from Pinder's (1973) study of chromium transport in Long Island, New York: $\alpha_L = 21.3 \text{ m}$ and $\alpha_T = 4.27 \text{ m}$. 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 noncontinuous 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

ORNL-DWG 79-15075

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	-1.85	1.30	2.81	-1.08	-0.83	2.44	4.06

^aNot 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 indicates 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.

ORNL DWG 79-12081

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.

ORNL DWG 79-15079

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

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.

ORNL DWG 79-18084

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.

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 ³ T)
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)
$[T_{ij}]$	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)
τ	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 (clerkin 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. Radiokhimika 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 transport 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. Massachusetts Institute of Technology, Cambridge, Massachusetts.
- Yeh, G. T., and D. S. Ward. 1980. FEMWATER: A Finite-Element Model of WATER Flow through saturated-unsaturated porous media. ORNL-5567. Oak Ridge National Laboratory, Oak Ridge, Tennessee.
- 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

68

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(1615)

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	I5	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	Auxiliary 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 control; 0 = not mid-difference, 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	KDSK0	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 KDSKC 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 normal-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

80

APPENDIX B SAMPLE INPUT OF THE SEEPAGE POND PROBLEM.

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

APPENDIX B (continued)

C	595	0.		CARD 015	
C					
C	----- DATA SET 9: DIRICHLET BOUNDARY CONDITIONS				
C	152	0	1.	CARD 016	
	164	0	1.	CARD 017	
	204	5	1.	CARD 018	
C					
C	----- DATA SET 10: CAUCHY BOUNDARY CONDITIONS ARE NOT NEEDED				
C	----- SINCE NST=0				
C					
C	----- DATA SET 11: NEUMANN BOUNDARY CONDITIONS				
C	135	209	220	0	CARD 019
	515	569	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	597	588	0	CARD 026
	528	587	595	0	CARD 027
C					
C	----- DATA SET 12: INITIAL FLOW VARIABLES ARE READ IN VIA LOGICAL UNIT 1				
C					
C	----- DATA SET 13: TRANSIENT FLOW VARIABLES ARE READ IN VIA LOGICAL UNIT 1				
C					
C	----- FINALLY A BLANK CARD TO END THE JOB				
				CARD 028	
	IMC0021 STOP 0				

ORNL-5601

APPENDIX C
LISTING OF FORTRAN IV SOURCE PROGRAM

84

APPENDIX C LISTING OF FORTRAN IV SOURCE PROGRAM

THIS COMPUTER CODE IS CONTAINED IN THE FOLLOWING REPORT:

YEH, G. T. AND D. S. WARD, 1979. "FEMWASTE: A FINITE-ELEMENT
MODEL OF WASTE TRANSPORT THROUGH SATURATED-UNSATURATED POROUS MEDIA"
ORNL-5601, OAK RIDGE NATIONAL LABORATORY, OAK RIDGE, TN 37830

A SLIGHTLY UPDATED VERSION IS CONTAINED IN "FECWASTE: USER'S MANUAL
OF A FINITE ELEMENT CODE FOR SIMULATING WASTE TRANSPORT THROUGH
SATURATED-UNSATURATED POROUS MEDIA, ORNL/TM ----"

FOR ANY QUESTION, PLEASE CONTACT DR. G. T. YEH AT (615) 574-7285

ADDITIONAL REFERENCES IS:

DUGUID, J. AND M. REEVES, 1976. "MATERIAL TRANSPORT THROUGH
SATURATED-UNSATURATED POROUS MEDIA: A GALERKIN FINITE ELEMENT MODEL",
ORNL 4928, OAK RIDGE NATIONAL LABORATORY, OAK RIDGE, TENNESSEE 37830

MAIN PROGRAM

IMPLICIT REAL*8(A-H,O-Z)
REAL*4 PMAT

DIMENSION X(595),Z(595),IE(528,5)
DIMENSION C(595,27),R(595),RP(595),WETAB(528,4)
DIMENSION FX(595),FZ(595)

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)

DIMENSION DLB(199),DCOSXB(199),DCOSZB(199),BFLX(200),BFLXP(200),
> NDB(199),ISB(199,4),NP3(200)

DIMENSION DL(99),DCOSX(99),DCOSZ(99),NRSE(99),IS(99,4),NPRS(100)
DIMENSION NPN(30),BB(30),NPST(40),OP(40)
DIMENSION PMAT(3,9),PRCP(3,9)
DIMENSION KPR(500),KDSK(500)
DIMENSION FRATE(10),FLOW(10),TFLOW(10)

COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN,NST
COMMON /CONTRL/ NTI,NSTR,KSTR,KPRO,KOSK0,KSS,KVI
COMMON /PARM/ DELT,CHNG,DELMAX,TMAX
COMMON /MTL/ NMAT,NMPPN,NCH
COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID

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

ORNL-5601

85

APPENDIX C (continued)

```

      DO 200 NP=1,MAXBEL
        DLB(NP)=0.0
        DCOSXT(NP)=0.0
        DCOSZB(NP)=0.0
        NBE(NP)=0
        DO 200 IQ=1,4
          ISB(NP,IQ)=0
200    CONTINUE
C
C
C----- INITIATE ARRAYS FOR BOUNDARY NODAL POINTS
      DO 250 NP=1,MAXBNP
        BFLX(NP)=0.0
        BFLXP(NP)=0.0
        NPB(NP)=0
250    CONTINUE
C
C
C----- INITIATE ARRAYS FOR INFLUX-OUTFLUX BOUNDARY ELEMENTS
      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
      DO 350 NP=1,MXRSNP
        NPRS(NP)=0
350    CONTINUE
C
C
C----- INITIATE ARRAYS FOR SURFACE TERM POINT FLUX
      DO 500 NP=1,MXSTNP
        NPST(NP)=0
        DP(NP)=0.0
500    CONTINUE
C
C
C----- INITIATE ARRAYS FOR DIRICHLENT BOUNDARY CONDITIONS
      DO 510 NP=1,MAXBCN
        BB(NP)=0.0
        NBN(NP)=0
510    CONTINUE
C

```

```

MAIN 505
MAIN 510
MAIN 515
MAIN 520
MAIN 525
MAIN 530
MAIN 535
MAIN 540
MAIN 545
MAIN 550
MAIN 555
MAIN 560
MAIN 565
MAIN 570
MAIN 575
MAIN 580
MAIN 585
MAIN 590
MAIN 595
MAIN 600
MAIN 605
MAIN 610
MAIN 615
MAIN 620
MAIN 625
MAIN 630
MAIN 635
MAIN 640
MAIN 645
MAIN 650
MAIN 655
MAIN 660
MAIN 665
MAIN 670
MAIN 675
MAIN 680
MAIN 685
MAIN 690
MAIN 695
MAIN 700
MAIN 705
MAIN 710
MAIN 715
MAIN 720
MAIN 725
MAIN 730
MAIN 735
MAIN 740
MAIN 745
MAIN 750

```

APPENDIX C (continued)

```

C ----- INITIATE ARRAYS FOR MATERIAL PROPERTIES
C
      DO 650 I=1,MAXNAT
      DO 610 J=1,MXMPPM
610        PROP(I,J)=0.0
650      CONTINUE

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 ----- INITIATE ARRAYS FOR PRINT AND WRITE ON DISK CONTROL
C
      DO 800 NP=1,MAXNTI
      KPR(NP)=0
800      KDSK(NP)=0

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,PRCP,MAXNP,MAXEL,MAXBW,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,
> MAXBCN,MAXNTI,MAXNAT,MXMPPM)

C
      STOP
      END

```

```

MAIN 755
MAIN 760
MAIN 765
MAIN 770
MAIN 775
MAIN 780
MAIN 785
MAIN 790
MAIN 795
MAIN 800
MAIN 805
MAIN 810
MAIN 815
MAIN 820
MAIN 825
MAIN 830
MAIN 835
MAIN 840
MAIN 845
MAIN 850
MAIN 855
MAIN 860
MAIN 865
MAIN 870
MAIN 875
MAIN 880
MAIN 885
MAIN 890
MAIN 895
MAIN 900
MAIN 905
MAIN 910
MAIN 915

```

APPENDIX C (continued)

	SUBROUTINE GM2DXZ(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,H,HP,DH,HT,	GM2D 005
	> TH,THP,DTH,FX,FZ,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,BFLX,BFLXP,DL,	GM2D 010
	> DCOSX,DCOSZ,NRSE,IS,NPRS,NPN,BB,NPST,DP,KPR,KOSK,FRATE,FLOW,TFLOW,	GM2D 015
	> PMAT,PROP,MAXNP,MAXTL,MAXBW,MAXBNP,MAXBEL,MXRSP,MXRSEL,MXSTNP,	GM2D 020
	> MAXBCN,MAXNTI,MAXMAT,MXMPM)	GM2D 025
C	IMPLICIT REAL*8(A-H,O-Z)	GM2D 030
	REAL*4 PMAT	GM2D 035
C	DIMENSION TITLE(9)	GM2D 040
	DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)	GM2D 045
	DIMENSION C(MAXNP,MAXBW),R(MAXNP),RP(MAXNP),WETAB(MAXEL,4)	GM2D 050
	DIMENSION FX(MAXNP),FZ(MAXNP)	GM2D 055
C	DIMENSION VX(MAXNP),VXP(MAXNP),VZ(MAXNP),VZP(MAXNP),H(MAXNP),	GM2D 060
	> HP(MAXNP),DH(MAXNP),HT(MAXNP),TH(MAXEL,4),THP(MAXEL,4),	GM2D 065
	> DTH(MAXEL,4)	GM2D 070
C	DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),BFLX(MAXBNP),	GM2D 075
	> BFLXP(MAXBNP),NBE(MAXBEL),ISB(MAXBEL,4),NPB(MAXBNP)	GM2D 080
C	DIMENSION DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),NRSE(MXRSEL),	GM2D 085
	> IS(MXRSEL,4),NPRS(MXRSP)	GM2D 090
C	DIMENSION NPN(MAXBCN),BB(MAXBCN),NPST(MXSTNP),DP(MXSTNP)	GM2D 095
C	DIMENSION PMAT(MAXMAT,MXMPM),PROP(MAXMAT,MXMPM)	GM2D 100
	DIMENSION KPR(MAXNTI),KOSK(MAXNTI)	GM2D 105
C	DIMENSION FRATE(10),FLOW(10),TFLOW(10)	GM2D 110
C	COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IEAND	GM2D 115
	COMMON /BNDY/ NBEL,NBN,NRSEL,NRSP,NBC,NSTN,NST	GM2D 120
	COMMON /CONTRL/ NTI,NSTR,KSTR,KPRO,KOSKO,KSS,KVI	GM2D 125
	COMMON /PARM/ DELT,CHNG,DELMAX,TMAX	GM2D 130
	COMMON /NTL/ NMAT,NMPPM,NCM	GM2D 135
	COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUM,IMID	GM2D 140
C	INPUT PROBLEM IDENTIFICATION AND DESCRIPTION	GM2D 145
C	10 READ 10000,NPROB,(TITLE(I),I=1,9)	GM2D 150
	IF (NPROB.LE.0) GO TO 250	GM2D 155
	PRINT 10100,NPROB,(TITLE(I),I=1,9)	GM2D 160
C	READ AND PRINT INPUT DATA	GM2D 165
C	KOUT=0	GM2D 170
	KDIG=0	GM2D 175
C		GM2D 180
		GM2D 185
		GM2D 190
		GM2D 195
		GM2D 200
		GM2D 205
		GM2D 210
		GM2D 215
		GM2D 220
		GM2D 225
		GM2D 230
		GM2D 235
		GM2D 240
		GM2D 245
		GM2D 250

APPENDIX C (continued)

```

C      CALL DATAIN(X,Z,IE,DLB,DCOSXB,DCOSZE,NBE,ISB,NPB,DL,DCOSX,DCOSZ,
C      > NRSE,IS,NPRS,NPN,99,NPST,OP,RP,PMAT,PROP,KPR,KOSK,MAXNP,MAXEL,
C      > MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,MAXTIT,
C      > MAXDIF,W,ISTOP)
C      IF (ISTOP.GT.0) GO TO 240
C      COMPUTE BAND-WIDTH VARIABLES
C      IHALFB=MAXDIF
C      IBAND=2*IHALFB+1
C      IGBP=IHALFB+1
C      IF (IBAND.GT.MAXBW) GO TO 230
C      PREPARE INITIAL VARIABLES
C      TIME=0.
C      READ INITIAL VELOCITIES, PRESSURES, AND WATER CONTENTS, IF NECESSARY
C      DO 20 M=1,NEL
C      MTYP=IE(M,3)
C      POR=PROP(MTYP,6)
C      DO 20 IQ=1,4
C      TH(M,IQ)=POR
C      20 CONTINUE
C      60 READ(1) TIMEN,(H(NP),NP=1,NNP),(HT(NP),NP=1,NNP),(TH(M,IQ), M=1,
C      > NEL),IQ=1,4),(VX(NP),NP=1,NNP),(VZ(NP),NP=1,NNP)
C      DO 95 NP=1,NNP
C      VXP(NP)=VX(NP)
C      95 VZP(NP)=VZ(NP)
C      CALL AFABTA(X,Z,IE,WETAB,VX,VXP,VZ,VZP,PROP,MAXNP,MAXEL,
C      > MAXMAT,MXMPPM,NEL,W)
C      PRINT 10600
C      KLINE=0
C      DO 100 MP=1,NEL,2
C      NJMN=MP
C      NJMX=MINO(MP+1,NEL)
C      PRINT 10700,(NJ,(WETAB(NJ,IQ),IQ=1,4),NJ=NJMN,NJMX)
C      KLINE=KLINE+1
C      IF(MOD(KLINE,50).EQ.0) PRINT 10600
C      100 CONTINUE
C      CALL FLUX(X,Z,IE,WETAB,C,F,X,FZ,RP,VX,VZ,PROP,MAXNP,MAXEL,MAXBW,
C      > MAXMAT,MXMPPM)

```

```

GM2D 255
GM2D 260
GM2D 265
GM2D 270
GM2D 275
GM2D 280
GM2D 285
GM2D 290
GM2D 295
GM2D 300
GM2D 305
GM2D 310
GM2D 315
GM2D 320
GM2D 325
GM2D 330
GM2D 335
GM2D 340
GM2D 345
GM2D 350
GM2D 355
GM2D 360
GM2D 365
GM2D 370
GM2D 375
GM2D 380
GM2D 385
GM2D 390
GM2D 395
GM2D 400
GM2D 405
GM2D 410
GM2D 415
GM2D 420
GM2D 425
GM2D 430
GM2D 435
GM2D 440
GM2D 445
GM2D 450
GM2D 455
GM2D 460
GM2D 465
GM2D 470
GM2D 475
GM2D 480
GM2D 485
GM2D 490
GM2D 495
GM2D 500

```

ORNL-5601

APPENDIX C (continued)

```

      CALL SFLOW(X,Z,IE,WETAB,RP,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,NBE,
> ISB,NPB,BFLX,BFLXP,PPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,MAXEL,
> MAXBNP,MAXBEL,MXRSN,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,DELT,DH)
C
C
C PRINT INITIAL VARIABLES
      CALL PRINTT(RP,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPRO,KOUT,KDIG,
> MAXNP,NNP,IBAND)
C
      IF(KSTR.EQ.1 .AND. KOSKO.EQ.1) CALL STORE(X,Z,IE,RP,FX,FZ,TITLE,
> NPROB,NNP,NEL,NTI,MAXNP,MAXEL,TIME)
C
C PERFORM TRANSIENT-STATE CALCULATION
      TIME=DELT
      W1=W
      W2=1.-W
C
C READ TIME-DEPENDENT VELOCITIES, AS REQUIRED
      DO 220 ITM=1,NTI
      DO 110 NP=1,NNP
        VXP(NP)=VX(NP)
        VZP(NP)=VZ(NP)
        HP(NP)=H(NP)
      110 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) TIMEN,(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
      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 AFASTA(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

```

```

GM2D 505
GM2D 510
GM2D 515
GM2D 520
GM2D 525
GM2D 530
GM2D 535
GM2D 540
GM2D 545
GM2D 550
GM2D 555
GM2D 560
GM2D 565
GM2D 570
GM2D 575
GM2D 580
GM2D 585
GM2D 590
GM2D 595
GM2D 600
GM2D 605
GM2D 610
GM2D 615
GM2D 620
GM2D 625
GM2D 630
GM2D 635
GM2D 640
GM2D 645
GM2D 650
GM2D 655
GM2D 660
GM2D 665
GM2D 670
GM2D 675
GM2D 680
GM2D 685
GM2D 690
GM2D 695
GM2D 700
GM2D 705
GM2D 710
GM2D 715
GM2D 720
GM2D 725
GM2D 730
GM2D 735
GM2D 740
GM2D 745
GM2D 750

```

APPENDIX C (continued)

```

C   APPLY BOUNDARY CONDITIONS
C
C       CALL BC(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,NPN,BB,NPST,DP,
C       >       DL,DCOSX,DCOSZ,NRSE,IS,W,MAXNP,MAXEL,MAXBW,MAXBCN,MXSTNP,
C       >       MXRSEL)
C
C   TRIANGULARIZE C 'ATR'
C
C       CALL SOLVE(1,C,R,NNP,IHALFB,MAXNP,MAXBW)
C
C   BACK SUBSTITUTE
C
C       CALL SOLVE(2,C,R,NNP,IHALFB,MAXNP,MAXBW)
C
C       IF (IMID.EQ.0) GO TO 208
C       DO 205 NP=1,NNP
C       R(NP)=2.000*R(NP)-RP(NP)
C       DO 206 NPP=1,NBC
C       NI=NON(NPP)
C       R(NI)=BB(NPP)
C       206
C       208 CONTINUE
C
C   CALCULATE MATERIAL FLUX FX(NP) AND FZ(NP)
C
C       CALL FLUX(X,Z,IE,WETAB,C,FX,FZ,R,VX,VZ,PROP,MAXNP,MAXEL,
C       >       MAXBW,MAXMPT,MX:PPH)
C
C   DETERMINE BOUNDARY FLOWS
C
C       CALL SFLOW(X,Z,IE,WETAB,K,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,NBE,
C       >       ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,MAXEL,
C       >       MAXBNP,MAXBEL,MXRSPN,MXRSEL,MXSTNP,MAXBCN,MAXVAT,MXNPPH,DELT,
C       >       DH)
C
C   PRINT VARIABLES AT CURRENT TIME STEP
C
C       CALL PRINTT(R,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPR(ITN),KOUT,
C       >       KDIG,MAXNP,NNP,IBAND)
C
C       IF (KSTR.EQ.1 .AND. KDSK(ITN).EQ.1) CALL STORE(X,Z,IE,R,FX,FZ,
C       1 TITLE,NPROB,NNP,NEL,NTI,MAXNP,MAXEL,TIME)
C
C   PREPARE FOR NEXT TIME STEP
C
C       IF (KSS.EQ.0) GO TO 10
C       IF (TIME.GT.TMAX) GO TO 10
C       DELT=DELT*(1.+CHNG)
C       DELT=OMINI(DELT,DELMAX)
C       TIME=TIME+DELT

```

GM2D 755
 GM2D 760
 GM2D 765
 GM2D 770
 GM2D 775
 GM2D 780
 GM2D 785
 GM2D 790
 GM2D 795
 GM2D 800
 GM2D 805
 GM2D 810
 GM2D 815
 GM2D 820
 GM2D 825
 GM2D 830
 GM2D 835
 GM2D 840
 GM2D 845
 GM2D 850
 GM2D 855
 GM2D 860
 GM2D 865
 GM2D 870
 GM2D 875
 GM2D 880
 GM2D 885
 GM2D 890
 GM2D 895
 GM2D 900
 GM2D 905
 GM2D 910
 GM2D 915
 GM2D 920
 GM2D 925
 GM2D 930
 GM2D 935
 GM2D 940
 GM2D 945
 GM2D 950
 GM2D 955
 GM2D 960
 GM2D 965
 GM2D 970
 GM2D 975
 GM2D 980
 GM2D 985
 GM2D 990
 GM2D 995
 GM2D 1000

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(15,9A8)
10100 FORMAT(/8H1PROBLEM,15,3H..,9A8/)
10200 FORMAT(2F10.0)
10300 FORMAT(////35H1ERROR IN VELOCITY INPUT AT ELEMENT, 15///)
10400 FORMAT(////12H BANDWIDTH =,14.25H EXCEEDS MAX. ALLOWABLE =,14///)
10500 FORMAT(1H0.4X,'ISTOP' = ',15)
10600 FORMAT(1H1,'TABLE OF WEIGHTING FACTORS OF EVERY ELEMENTS'///)
10700 FORMAT(1H,15.4E12.4,5X,15.4E12.4)
      END

```

```

GM2D1005
GM2D1010
GM2D1015
GM2D1020
GM2D1025
GM2D1030
GM2D1035
GM2D1040
GM2D1045
GM2D1050
GM2D1055
GM2D1060
GM2D1065
GM2D1070
GM2D1075
GM2D1080
GM2D1085

```

APPENDIX C (continued)

```

SUBROUTINE DATAINI,Y,Z,IE,DLB,DCOSXB,DCOSZB,NBE,ISB,NPE,DL,DCOSX,
> OCOZ, NRSE, IS, NPRS, NPN, BB, NPST, OP, RP, PMAT, PROP, KPR, KOSK, MAXNP,
> MAXEL, MAXBNP, MAXBEL, MXRSNP, MXRSEL, MXSTNP, MAXBCN, MAXVAT, MXMPPM,
> MAXNTI, MAXOIF, W, ISTOP)

FUNCTION OF SUBROUTINE--TO READ, PRINT, AND CHECK VARIABLES
PERTAINING TO SIMULATION TIME, GEOMETRY OF THE SYSTEM, BOUNDARY-
INITIAL CONDITIONS, AND PROPERTIES OF BOTH THE MATERIAL BEING
TRANSPORTED AND THE POROUS MEDIA.

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 PMAT

      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)
      DIMENSION OLB(MAXBEL),DCOSXB(MAXBEL),OCOSZB(MAXBEL),NBE(MAXBEL),
> ISB(MAXBEL,4),NPB(MAXBNP)
      DIMENSION OL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),NRSE(MXRSEL),
> IS(MXRSEL,4),NPRS(MXRSNP)

      DIMENSION NPN(MAXBCN),BB(MAXBCN),OP(MXSTNP),NPST(MXSTNP)
      DIMENSION RP(MAXNP),PMAT(MAXMAT,MXMPPM),PROP(MAXMAT,MXMPPM)
      DIMENSION KPR(MAXNTI),KOSK(MAXNTI)
      DIMENSION TITLE(9)

      COMMON /GEOM/ SNFE,CSFE,NPN,NEL,IBAND
      COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN,NST
      COMMON /CONTR/ NTI,NSTR,KSTR,KPRO,KOSKO,KSS,KVI
      COMMON /PARM/ DELT,CHNG,DELMAX,TMAX
      COMMON /MTL/ NMAT,MXMDPM,NCM
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, I WET,ILUMP,IMID

      ISTOP=0
      READ 10900,NPN,NEL,NMAT,NCM,NTI,NBC,NST,NRSEL,KVI,KSTR,KSS,MXMDPM,
      1 I WET,ILUMP,IMID
      READ 11000,DELT,CHNG,DELMAX,TMAX,W
      READ 11100,KPRO,(KPR(I TM),I TM=1,NTI)
      READ 11100,KOSKO,(KOSK(I TM),I TM=1,NTI)
      IF(TMAX*LE.0.0) TMAX=1.0E50

      PRINT 10000,NPN,NEL,NMAT,NCM,NTI,NBC,NST,NRSEL,KVI,KSTR,KSS,DELT,
> CHNG,DELMAX,TMAX,W
      PRINT 10001,I WET,ILUMP,IMID
      PRINT 10100
      PRINT 11200,KPRO,(KPR(I TM),I TM=1,NTI)

```

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

APPENDIX C (continued)

```

      PRINT 10100
      PRINT 11200, KDSKO, (KOSK(ITM), ITM=1, NTI)

C
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
      LEWIND 1
      READ(1) (TITLE(I), I=1,9), NPROB, 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)

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 I=1,3
              IQ1 = IQ + 1
              DO 140 JQ=IQ1,4
                  ND = IABS(IE(M,IQ)-IE(M,JQ))
                  MND = MAX0(ND, MND)
              140 KMAXDIF = 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=1,3), MND
      150 CONTINUE

C
C
C
C  MODIFY MATERIAL TYPES FOR SELECTED ELEMENTS, IF NECESSARY
      IF (NCM.LE.0) GO TO 370

```

```

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

```

APPENDIX C (continued)

```

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     UP(NP)=0.
    IF (NBC.EQ.0) GO TO 550

```

```

DATA 505
DATA 510
DATA 515
DATA 520
DATA 525
DATA 530
DATA 535
DATA 540
DATA 545
DATA 550
DATA 555
DATA 560
DATA 565
DATA 570
DATA 575
DATA 580
DATA 585
DATA 590
DATA 595
DATA 600
DATA 605
DATA 610
DATA 615
DATA 620
DATA 625
DATA 630
DATA 635
DATA 640
DATA 645
DATA 650
DATA 655
DATA 660
DATA 665
DATA 670
DATA 675
DATA 680
DATA 685
DATA 690
DATA 695
DATA 700
DATA 705
DATA 710
DATA 715
DATA 720
DATA 725
DATA 730
DATA 735
DATA 740
DATA 745
DATA 750

```

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
      .STOP=ISTOP+1
      GO TO 520
470 READ 11900,NI,NPINC,BB1
      IF (NPINC.GT.0) GO TO 490
480 NPP=NPP+1
      NPN(NPP)=NI
      BB(NPP)=BB1
      GO TO 460
490 IF (NPP.GT.0) GO TO 500
      ISTOP=ISTOP+1
      PRINT 13900
500 NJ=NPN(NPP)+NPINC
      BB1=BB(NPP)
      NK=NI-1
      DO 510 NP=NJ,NK,NPINC
          NPP=NPP+1
          NPN(NPP)=NP
          BB(NPP)=BB1
510 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)
          RP(NP)=BB(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
      NP=0
      PRINT 10700
560 IF (NP.EQ.NST) GO TO 610
      READ 12100,NI,NJ,KINC,EI,EJ
      IF (KINC.GT.0) GO TO 580
570 NP=NP+1
      DX=X(NI)-X(NJ)
      DZ=Z(NI)-Z(NJ)

```

```

DATA 755
DATA 760
DATA 765
DATA 770
DATA 775
DATA 780
DATA 785
DATA 790
DATA 795
DATA 800
DATA 805
DATA 810
DATA 815
DATA 820
DATA 825
DATA 830
DATA 835
DATA 840
DATA 845
DATA 850
DATA 855
DATA 860
DATA 865
DATA 870
DATA 875
DATA 880
DATA 885
DATA 890
DATA 895
DATA 900
DATA 905
DATA 910
DATA 915
DATA 920
DATA 925
DATA 930
DATA 935
DATA 940
DATA 945
DATA 950
DATA 955
DATA 960
DATA 965
DATA 970
DATA 975
DATA 980
DATA 985
DATA 990
DATA 995
DATA 1000

```

APPENDIX C (continued)

```

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=IAES(NJ-NI)
    NPMIN=MAXO(NPST(NPP),NPST(NPP-1))
    NPMAX=MINO(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

```

```

DATA1005
DATA1010
DATA1015
DATA1020
DATA1025
DATA1030
DATA1035
DATA1040
DATA1045
DATA1050
DATA1055
DATA1060
DATA1065
DATA1070
DATA1075
DATA1080
DATA1085
DATA1090
DATA1095
DATA1100
DATA1105
DATA1110
DATA1115
DATA1120
DATA1125
DATA1130
DATA1135
DATA1140
DATA1145
DATA1150
DATA1155
DATA1160
DATA1165
DATA1170
DATA1175
DATA1180
DATA1185
DATA1190
DATA1195
DATA1200
DATA1205
DATA1210
DATA1215
DATA1220
DATA1225
DATA1230
DATA1235
DATA1240
DATA1245
DATA1250

```

APPENDIX C (continued)

```

      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      READ NUMBERS OF ELEMENTS AND SIDES TO WHICH OPEN BOUNDARY 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,M1,IS1,IS2,KINC
      IF (KINC.GT.0) GO TO 680
670  MPI=MPI+1
      NRSE(MPI)=M1
      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

```

DATA1258
 DATA1260
 DATA1265
 DATA1270
 DATA1275
 DATA1280
 DATA1285
 DATA1290
 DATA1295
 DATA1300
 DATA1305
 DATA1310
 DATA1315
 DATA1320
 DATA1325
 DATA1330
 DATA1335
 DATA1340
 DATA1345
 DATA1350
 DATA1355
 DATA1360
 DATA1365
 DATA1370
 DATA1375
 DATA1380
 DATA1385
 DATA1390
 DATA1395
 DATA1400
 DATA1405
 DATA1410
 DATA1415
 DATA1420
 DATA1425
 DATA1430
 DATA1435
 DATA1440
 DATA1445
 DATA1450
 DATA1455
 DATA1460
 DATA1465
 DATA1470
 DATA1475
 DATA1480
 DATA1485
 DATA1490
 DATA1495
 DATA1500

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)=IC2
    GO TO 678
676 CONTINUE
678 CONTINUE
    GO TO 680
680 IF (MPI.GT.0) GO TO 690
    ISTOP=ISTOP+1
    PRINT 13700
690 NPINC=IS(MPI.2)-IS(MPI.1)
    MINC=ABS(NPINC)-1
    NJ=NRSE(MPI)+MINC
    M=1
    DO 691 N=NJ,NK,MINC
        NPJ=MPI
        MPI=MPI+1
        NRSE(MPI)=M
        IS(MPI.1)=IS(NPJ.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

```

```

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

```


APPENDIX C (continued)

```

698 CONTINUE
700 CONTINUE
GO TO 670
710 NRSN=NPP
PRINT 10800
DO 720 MP=1,NRSEL
N=NRSE(MP)
720 PRINT 11600,M,I (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,NBFL
MJ=NRSE(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
DO 790 J=1,4
IS(MPI,J)=ISB(MPJ,J)
780 DL(MPI)=DLB(MPJ)
790 DCOSX(MPI)=DCOSXB(MPJ)
DCOSZ(MPI)=DCOSZB(MPJ)
GO TO 810
800 CONTINUE
ISTOP=ISTOP+1
PRINT 13800,M,I
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. . . . . // 15/ 5X,
> 40H NUMBER OF ELEMENTS. . . . . // 15/ 5X,
> 40H NUMBER OF DIFFERENT MATERIALS. . . . . // 15/ 5X,
> 40H NUMBER OF CORRECTION MATERIALS. . . . . // 15/ 5X,
> 40H NUMBER OF TIME INCREMENTS. . . . . // 15/ 5X,
> 40H NUMBER OF BOUNDARY CONDITIONS. . . . . // 15/ 5X,
> 40H NUMBER OF SURFACE TERMS. . . . . // 15/ 5X,
> 40H NUMBER OF SEEPAGE SURFACE TERMS. . . . . // 15/ 5X,
> 40H VELOCITY INPUT CONTROL. . . . . // 15/ 5X,
> 40H AUXILIARY STORAGE CONTROL. . . . . // 15/ 5X,
> 40H STEADY-STATE CONTROL. . . . . // 15/ 5X,
> 40H TIME INCREMENT. . . . . // F10.6/ 5X,
> 40H MULTIPLIER FOR INCREASING DELT. . . . . // F10.6/ 5X,

```

```

DATA1755
DATA1760
DATA1765
DATA1770
DATA1775
DATA1780
DATA1785
DATA1790
DATA1795
DATA1800
DATA1805
DATA1810
DATA1815
DATA1820
DATA1825
DATA1830
DATA1835
DATA1840
DATA1845
DATA1850
DATA1855
DATA1860
DATA1865
DATA1870
DATA1875
DATA1880
DATA1885
DATA1890
DATA1895
DATA1900
DATA1905
DATA1910
DATA1915
DATA1920
DATA1925
DATA1930
DATA1935
DATA1940
DATA1945
DATA1950
DATA1955
DATA1960
DATA1965
DATA1970
DATA1975
DATA1980
DATA1985
DATA1990
DATA1995
DATA2000

```

APPENDIX C (continued)

```

> 40H MAXIMUM VALUE OF DELT . . . . .D10.4/ 5X.
> 40H MAXIMUM VALUE OF TIME . . . . .D10.4/ 5X.
> 40H TIME-INTEGRATION PARAMETER. . . . .F10.6)
10001 FORMAT(1H,5X,
> 40H UPSTREAM WEIR FLOW INDICATOR, IWET. . . . .IS/ 5X.
> 40H LUMPING INDICATOR, ILUMP. . . . .IS/ 5X.
> 40H TIME-DIFFERENCE INDICATOR, IMID . . . . .IS)
10100 FORMAT(//6X,14HOUTLET CONTROL)
10200 FORMAT(36H1INPUT FILE 2.. MATERIAL PROPERTIES// 9H MAT. NO., 9(
> 3A4))
10300 FORMAT(34H1INPUT FILE 3.. NODAL POINT DATA // 7X,4HNODE, 8X,1HX,
> 14X,1H2)
10400 FORMAT(34H1INPUT FILE 4.. ELEMENT DATA // 11X,
> 31HGLOBAL INDICES OF ELEMENT NODES/7X,7HELEMENT, 3X,1H1,7X,1H2,
> 7X,1H3,7X,1H4,6X,1H5MATERIAL,6X,10HNODE DIFF. )
10500 FORMAT(//52H CORRELATIONS TO MATERIAL TYPES FOR SELECTED ELEMENTS/)
10600 FORMAT(44H1INPUT FILE 5.. BOUNDARY CONDITIONS OF FORM, 5H R=BB//
> 6H NODE,7X,2HBB)
10700 FORMAT(32H1INPUT FILE 6.. SURFACE TERMS , 20H E=E1 AT NODE N1, E=
> 13HEJ AT NODE N2/3X,2HNI,3X,2HNJ,6X,2HEI,13X,2HEJ/)
10800 FORMAT(44H1INPUT FILE 7.. SEEPAGE-SURFACE INFORMATION//5X
> 14HELEMENT NODE 1,2X,6HNODE 2)
10900 FORMAT(16I5)
11000 FORMAT(8F10.0)
11100 FORMAT(80I1)
11200 FORMAT(10X,10I11)
11300 FORMAT(18,9D12.4)
11400 FORMAT(12,2F10.3)
11500 FORMAT(110,2D15.4)
11600 FORMAT(110,4I8,110,115)
11700 FORMAT(110,32X,110)
11800 FORMAT(15,5X,F10.0)
11900 FORMAT(215,F10.6)
12000 FORMAT(12,D15.4)
12100 FORMAT(315,5X,2F10.0)
12200 FORMAT(215,2(1PD15.4))
129000 FORMAT(////36H CHECK BOUNDARY CONDITIONS, MAXIMUM=.15///)
13300 FORMAT(////30H ERROR IN NODAL-POINT CARD NO.,15///)
13400 FORMAT(////26H ERROR IN ELEMENT CARD NO.,15///)
13500 FORMAT(////36H ERROR IN INITIAL-CONDITION CARD NO.,15///)
13600 FORMAT(////45H ASSEMBLY AND SOLUTION WILL NOT BE PERFORMED.,15.
> 19H FATAL CARD ERRORS///)
13700 FORMAT(////36H ERROR IN FIRST TRANSIENT-SURFACE CARD///)
13800 FORMAT(////44H ERROR IN TRANSIENT-SURFACE CARD FOR ELEMENT,15///)
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 ,15///)
END

```

```

DATA2005
DATA2010
DATA2015
DATA2020
DATA2025
DATA2030
DATA2035
DATA2040
DATA2045
DATA2050
DATA2055
DATA2060
DATA2065
DATA2070
DATA2075
DATA2080
DATA2085
DATA2090
DATA2095
DATA2100
DATA2105
DATA2110
DATA2115
DATA2120
DATA2125
DATA2130
DATA2135
DATA2140
DATA2145
DATA2150
DATA2155
DATA2160
DATA2165
DATA2170
DATA2175
DATA2180
DATA2185
DATA2190
DATA2195
DATA2200
DATA2205
DATA2210
DATA2215
DATA2220
DATA2225
DATA2230
DATA2235
DATA2240
DATA2245

```

APPENDIX C (continued)

```

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

```

```

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

```

APPENDIX C (continued)

```

JQ=J
JQP=JQ+1
IF(JQ.EQ.4) JQP=1
N1=NDIN(J)
IF(J-3) 300,200,210
200 N2=NDIN(J+1)
GO TO 300
210 N2=NDIN(1)
300 CONTINUE
ALENG=DSQRT((X(N2)-X(N1))**2+(Z(N2)-Z(N1))**2)
CBETA=DCOS(BETA(J))
SBETA=DSIN(BETA(J))
VXX=0.5*((VX(N1)+VX(N2))*W1+(VXP(N1)+VXP(N2))*W2)
VZZ=0.5*((VZ(N1)+VZ(N2))*W1+(VZP(N1)+VZP(N2))*W2)
VAL=VXX*CBETA+VZZ*SBETA
VV=DSQRT(VXX*VXX+VZZ*VZZ)
VVI=1.0/VV
OLL=(AL*VXX*VXX + AT*VZZ*VZZ)*VVI + DD
DLT=(AL-AT)*VXX*VZZ*VVI
OTT=(AL*VZZ*VZZ + AT*VXX*VXX)*VVI + DD
DAL=DABS(OLL*CBETA*CBETA+2.0D0*CBETA*SBETA*DLT+OTT*SBETA*SBETA)
VEL=VAL*ALENG
DISP=2.0*DAL
APHA(J)=1.0/DTANH(VEL/DISP)- DISP/VEL
IF(APHA(J).LT.0.0) APHA(J)=0.0
400 CONTINUE
WETAB(N,1)=APHA(1)
WETAB(N,2)=APHA(2)
WETAB(N,3)=APHA(3)
WETAB(N,4)=APHA(4)
500 CONTINUE
RETURN
END

```

```

AFAB 255
AFAB 260
AFAB 265
AFAB 270
AFAB 275
AFAB 280
AFAB 285
AFAB 290
AFAB 295
AFAB 300
AFAB 305
AFAB 310
AFAB 315
AFAB 320
AFAB 325
AFAB 330
AFAB 335
AFAB 340
AFAB 345
AFAB 350
AFAB 355
AFAB 360
AFAB 365
AFAB 370
AFAB 375
AFAB 380
AFAB 385
AFAB 390
AFAB 395
AFAB 400
AFAB 405
AFAB 410
AFAB 415

```

APPENDIX C (continued)

```

SUBROUTINE FLUX(X,Z,IE,WETAB,C,FX,FZ,R,VX,VZ,PROP,
> MAXNP,MAXEL,MAXBW,MAXMAT,MAXMPPM)
C
C FUNCTION OF SUBROUTINE TO COMPUTE DARCY MATERIAL FLUX FX AND FZ,
C RESPECTIVELY
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
C DIMENSION C(MAXNP,MAXBW),FX(MAXNP),FZ(MAXNP)
C DIMENSION R(MAXNP),VX(MAXNP),VZ(MAXNP),PROP(MAXMAT,MAXMPPM)
C DIMENSION QB(4,4),RQ(4),XQ(4),ZQ(4),CRQ(4),VXQ(4),VZQ(4)
C
C COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
C COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
C INITIALIZE THE FLUX FX(NP) AND FZ(NP)
C
C DO 100 NP=1,NNP
C   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
C IHALFB=(IBAND-1)/2
C IHNP=IHALFB+1
C DO 300 IXZ=1,2
C   DO 110 NP=1,NNP
C     DO 110 IB=1,IBAND
110       C(NP,IB)=0.0
C     DO 180 M=1,NEL
C       APHA1=WETAB(M,1)
C       APHA2=WETAB(M,3)
C       BETA1=WETAB(M,2)
C       BETA2=WETAB(M,4)
C       MTYP=IE(M,5)
C       AL=PROP(MTYP,3)
C       AT=PROP(MTYP,4)
C       AM=PROP(MTYP,6)
C       TAU=PROP(MTYP,9)
C       DD=AM*TAU
C       DO 120 IO=1,4
C         NP=IE(M,IO)
C         XQ(IO)=X(NP)
C         ZQ(IO)=Z(NP)
C         CRQ(IO)=R(IO)
C         VXQ(IO)=VX(NP)
C         VZQ(IO)=VZ(NP)
120       CONTINUE

```

```

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

```

APPENDIX C (continued)

```

C      COMPUTE THE ELEMENT MATRIX QB(IQ,JQ) AND QG(IQ)
C      CALL Q4D(QB,RQ,XQ,ZQ,VXQ,VZQ,CRQ,DD,AL,AT,SNFE,CSFE,IXZ)
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+IMBP
          C(NI,IB)=C(NI,IB)+QB(IQ,JQ)
130        CONTINUE
      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,1HALFB,MAXNP,MAXBW)
      CALL SOLVE(2,C,FZ,NNP,1HALFB,MAXNP,MAXBW)
      GO TO 300
200    CALL SOLVE(1,C,FZ,NNP,1HALFB,MAXNP,MAXBW)
      CALL SOLVE(2,C,FX,NNP,1HALFB,MAXNP,MAXBW)
300    CONTINUE
      RETURN
      END

```

```

FLUX 255
FLUX 260
FLUX 265
FLUX 270
FLUX 275
FLUX 280
FLUX 285
FLUX 290
FLUX 295
FLUX 300
FLUX 305
FLUX 310
FLUX 315
FLUX 320
FLUX 325
FLUX 330
FLUX 335
FLUX 340
FLUX 345
FLUX 350
FLUX 355
FLUX 360
FLUX 365
FLUX 370
FLUX 375
FLUX 380
FLUX 385
FLUX 390
FLUX 395
FLUX 400
FLUX 405
FLUX 410
FLUX 415
FLUX 420
FLUX 425

```

APPENDIX C (continued)

```

C      SUBROUTINE Q4D(QB,RQ,XQ,ZQ,VXQ,VZQ,CRQ,DO,AL,AT,SNFE,CSFE,IND)
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
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 N(4)
C      DIMENSION QB(4,4),RQ(4),XQ(4),ZQ(4),CRQ(4),VXQ(4),VZQ(4)
C      DIMENSION S(4),T(4),W(4),DNX(4),DNZ(4),DWX(4),DWZ(4)
C      DIMENSION PJAB(2,2),DNSS(4),DNNT(4),DNSS(4),DNNT(4)
C
C      DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00, -
C      > 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /
C
C      INITIALIZE MATRICES QB(IQ,JQ) AND QO(IQ)
C
C      DO 100 IQ=1,4
C        RQ(IQ)=0.0
C        DO 100 JQ=1,4
C          QB(IQ,JQ)=0.0
C
C      100
C
C      SUMMATION OF THE INTEGRAND OVER THE GAUSSIAN POINTS
C
C      DO 400 KG=1,4
C
C      DETERMINE LOCAL COORDINATE (SS,TT) OF
C      GAUSS-INTEGRATION POINT KG
C
C      SS=P*S(KG)
C      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
C      CALL SHAPE(N,W,DNSS,DNNT,DWSS,DWNT,SS,TT)
C      DO 110 I=1,2
C        DO 110 J=1,2
C          PJAB(I,J)=0.0
C
C      110
C      DO 120 I=1,4
C        PJAB(1,1)=PJAB(1,1)+ZQ(I)*DNNT(I)
C        PJAB(1,2)=PJAB(1,2)-ZQ(I)*DNSS(I)
C        PJAB(2,1)=PJAB(2,1)-XQ(I)*DNNT(I)
C        PJAB(2,2)=PJAB(2,2)+XQ(I)*DNSS(I)
C
C      120
C      DJAC=PJAB(2,2)*PJAB(1,1)-PJAB(1,2)*PJAB(2,1)
C      DJACT=1.0/DJAC
C      DO 130 I=1,2
C        DO 130 J=1,2
C          PJAB(I,J)=PJAB(I,J)*DJACT
C
C      130

```

```

040 008
040 010
040 015
040 020
040 025
040 030
040 035
040 040
040 045
040 050
040 055
040 060
040 065
040 070
040 075
040 080
040 085
040 090
040 095
040 100
040 105
040 110
040 115
040 120
040 125
040 130
040 135
040 140
040 145
040 150
040 155
040 160
040 165
040 170
040 175
040 180
040 185
040 190
040 195
040 200
040 205
040 210
040 215
040 220
040 225
040 230
040 235
040 240
040 245
040 250

```

APPENDIX C (continued)

```

DO 140 I=1.4
  DNK(I)=DNSS(I)*PJAB(1.1)+DNTT(I)*PJAB(1.2)
  DNZ(I)=DNSS(I)*PJAB(2.1)+DNTT(I)*PJAB(2.2)
  DNK(I)=DNSS(I)*PJAB(1.1)+DWT(I)*PJAB(1.2)
  DNZ(I)=DNSS(I)*PJAB(2.1)+DWT(I)*PJAB(2.2)
140
C
C INTERPOLATE WITH THE BASIS FUNCTIONS N(IQ) TO OBTAIN VALUES OF
C VELOCITY COMPONENTS VVK AND VZK AT GAUSSIAN POINT
C
  VVK=0.0
  VZK=0.0
  DO 150 IQ=1.4
    VVK=VVK+VXQ(IQ)*N(IQ)
    VZK=VZK+VZQ(IQ)*N(IQ)
150 CONTINUE
C
  VK=DSQRT(VVK*VVK+VZK*VZK)
  VKI=1.0/VK
  AKXK=(AL*VVK*VVK+AT*VZK*VZK)*VKI + DD
  AKZK=(AL*VZK*VZK+AT*VVK*VVK)*VKI + DD
  AKXZK=(AL-AT)*VVK*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 JO=1.4
      QB(IQ,JO)=QB(IQ,JO)+W(IQ)*N(JQ)*DJAC
      IF(IND.EQ.2) GO TO 200
      RQ(IQ)=RQ(IQ)-W(IQ)*CRQ(JQ)*(AKXK*DNK(JQ)+AKXZK*DNZ(JQ))*
        DJAC + W(IQ)*CRQ(JQ)*VVK*N(JQ)*DJAC
      GO TO 300
200 RQ(IQ)=RQ(IQ)-W(IQ)*CRQ(JQ)*(AKXZK*DNK(JQ)+AKZK*DNZ(JQ))*
      DJAC + W(IQ)*CRQ(JQ)*VZK*N(JQ)*DJAC
  300 CONTINUE
400 CONTINUE
RETURN
END

```

```

040 255
040 260
040 265
040 270
040 275
040 280
040 285
040 290
040 295
040 300
040 305
040 310
040 315
040 320
040 325
040 330
040 335
040 340
040 345
040 350
040 355
040 360
040 365
040 370
040 375
040 380
040 385
040 390
040 395
040 400
040 405
040 410
040 415
040 420
040 425
040 430
040 435
040 440
040 445

```


APPENDIX C (continued)

```

SUBROUTINE ASEMEL(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,TH,THP,DH,DTH,
> PROP,W,MAXNP,MAXEL,MAXBW,MAXMAT,MXMPM)

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

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KO,LAMDA

      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,MXMPM)

      DIMENSION QA(4,4),QB(4,4),VXQ(4),VZQ(4),XQ(4),ZQ(4),DHQ(4),
> DTHQ(4),THQ(4)

      COMMON /GEOM/ SNFF,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

      IHALFB=(IBAND-1)/2
      IHBP=IHALFB+1

      DELT I=1./DELT
      W1=W
      W2=1.-W
      IF (KSS.NE.0) GO TO 10
      DELT I=0.
      W1=1.
      W2=0.

      INITIALIZE MATRICES C(NP,IB) AND R(NP)

10 DO 20 NP=1,NNP
      R(NP)=0.
      DO 20 IB=1,IBAND
20      C(NP,IB)=0.0

      COMPUTE MATRICES QA(IQ,JQ) AND QB(IQ,JQ) FOR EACH ELEMENT M

      DO 50 M=1,NEL
          APHA1=WETAB(M,1)
          APHA2=WETAB(M,3)
          BETA1=WETAB(M,2)

```

```

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

```

APPENDIX C (continued)

```

      BETA2=WETAB(M,4)
      MTP=IE(M,5)
      XO=PROP(MTYP,1)
      RHO=PROP(MTYP,2)
      AL=PROP(MTYP,3)
      AT=PROP(MTYP,4)
      LAMDA=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
      CALL Q4(QA,QB,XQ,ZQ,VXQ,VZQ,THQ,DTHQ,DHQ,KD,RHOB,AL,AT,
        > LAMDA,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/DELTA AND FORM THE LOAD VECTOR
      C C(NP) = (A/DELTA - W2*B)*RP. MATRIX C IS ASYMMETRIC DUE TO
      C THE ADVECTION TERM.
      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)*DELTA
          R(NI)=R(NI)+(QA(IQ,JQ)-W2*QB(IQ,JQ))*RP(NJ)
          IB=NJ-NI+IHRP
          C(NI,IB)=C(NI,IB)+QA(IQ,JQ)+W1*QB(IQ,JQ)
          CONTINUE
        40 GO TO 50
      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)*DELTA
          R(NI)=R(NI)+QA(IQ,JQ)*RP(NJ)
          IB=NJ-NI+IHRP
      43

```

```

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

```

APPENDIX C (continued)

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

```
ASEM 505
ASEM 510
ASEM 515
ASEM 520
ASEM 525
```

APPENDIX C (continued)

```

SUBROUTINE Q4(QA,QB,XQ,ZQ,VXQ,VZQ,THQ,      DTHQ,DHQ,KD,RHOB,AL,AT,
>      LAMBDA,POR,ALP,AM,TAU)

FUNCTION GF SUBROUTINE--TO EVALUATE THE MATRIX QUADRATURES QA(IQ,JQ)
AND QB(IQ,JQ) OVER THE AREA OF ONE ELEMENT.

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N(4),NN,KD,LAMBDA

      DIMENSION W(4),DNX(4),DNZ(4),DWX(4),DWZ(4)
      DIMENSION PJAB(2,2),DNSS(4),DNLT(4),DWSS(4),DWLT(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)

      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID

      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 /

      DD=AM*TAU

      INITIALIZE MATRICES QA(IQ,JQ) AND QB(IQ,JQ)

      DO 10 IQ=1,4
      DO 10 JQ=1,4
      QA(IQ,JQ)=0.0
      QB(IQ,JQ)=0.0
10

      AREA=0.0D0

      DO 40 KG=1,4

      DETERMINE THE LOCAL COORDINATES (SS,TT) AND EVALUATE THE JACOBIAN AT
      EACH GAUSS- INTEGRATION POINT KG

      SS = P*S(KG)
      TT = P*T(KG)

      CALCULATE VALUES OF BASIS AND WEIGHTING FUNCTIONS N(IQ) AND W(IQ)
      AND THEIR DERIVATIVES DWX DNX AND DWZ DNZ WITH RESPECT TO X AND Z,
      RESPECTIVELY, AT THE GAUSS POINT KG

      CALL SHAPE(N,W,DNSS,DNLT,DWSS,DWLT,SS,TT)

      COMPUTE THE INVERSE TRANSFORMATION MATRIX I.E. THE INVERSE JACOBIAN:
      PJAB(1,1)={DZ/DT}/DJAC, PJAB(1,2)=-{DZ/DS}/DJAC,

```

Q4	005
Q4	010
Q4	015
Q4	020
Q4	025
Q4	030
Q4	035
Q4	040
Q4	045
Q4	050
Q4	055
Q4	060
Q4	065
Q4	070
Q4	075
Q4	080
Q4	085
Q4	090
Q4	095
Q4	100
Q4	105
Q4	110
Q4	115
Q4	120
Q4	125
Q4	130
Q4	135
Q4	140
Q4	145
Q4	150
Q4	155
Q4	160
Q4	165
Q4	170
Q4	175
Q4	180
Q4	185
Q4	190
Q4	195
Q4	200
Q4	205
Q4	210
Q4	215
Q4	220
Q4	225
Q4	230
Q4	235
Q4	240
Q4	245
Q4	250

APPENDIX C (continued)

```

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)*DN TT(I)
        PJAB(1,2)=PJAB(1,2) - ZQ(I)*DN SS(I)
        PJAB(2,1)=PJAB(2,1) - XQ(I)*DN TT(I)
        PJAB(2,2)=PJAB(2,2) + XQ(I)*DN SS(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=1.000/DJAC
      AREA=AREA + DJAC
C
      DO 17 I=1,2
      DO 17 J=1,2
        PJAB(I,J)=PJAB(I,J)*DJACI
      17 CONTINUE
C
      DO 18 I=1,4
        DN X(I)=PJAB(1,1)*DN SS(I) + PJAB(1,2)*DN TT(I)
        DN Z(I)=PJAB(2,1)*DN SS(I) + PJAB(2,2)*DN TT(I)
        DN W(I)=PJAB(1,1)*DN SS(I) + PJAB(1,2)*DN TT(I)
        DN W(I)=PJAB(2,1)*DN SS(I) + PJAB(2,2)*DN TT(I)
      18 CONTINUE
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)
        VZK=VZK+N(IQ)*VZQ(IQ)
      20 CONTINUE
C
      VK=DSORT(VX, VVK+VZK*VZK)
      VKI=1./VK
      A=DJAC*(THK+RHOB*KD)
      DXX=DJAC*((AL*VVK*VVK+AT*VZK*VZK)*VKI+DD)

```

```

04 255
04 260
04 265
04 270
04 275
04 280
04 285
04 290
04 295
04 300
04 305
04 310
04 315
04 320
04 325
04 330
04 335
04 340
04 345
04 350
04 355
04 360
04 365
04 370
04 375
04 380
04 385
04 390
04 395
04 400
04 405
04 410
04 415
04 420
04 425
04 430
04 435
04 440
04 445
04 450
04 455
04 460
04 465
04 470
04 475
04 480
04 485
04 490
04 495
04 500

```

ORNL-5601

114

Q4	505
Q4	510
Q4	515
Q4	520
Q4	525
Q4	530
Q4	535
Q4	540
Q4	545
Q4	550
Q4	555
Q4	560
Q4	565
Q4	570
Q4	575
Q4	580
Q4	585
Q4	590
Q4	595
Q4	600
Q4	605
Q4	610
Q4	615
Q4	620
Q4	625
Q4	630
Q4	635
Q4	640
Q4	645
Q4	650
Q4	655
Q4	660
Q4	665
Q4	670
Q4	675
Q4	680
Q4	685

```

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
        QB(IQ,JQ)=QB(IQ,JQ)+DWXDNX+DXX+DXZ*(DWXDNZ+DWZDNX)+DZZ*
          DWZDNZ + C*WN - (VXK*DWXN+VZK*DWZN)
30    >
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
62    QA(I,I)=SUM
52    CONTINUE
      RETURN
      END

```

APPENDIX C (continued)

```

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

```

```

SHAP 008
SHAP 010
SHAP 018
SHAP 020
SHAP 028
SHAP 030
SHAP 038
SHAP 040
SHAP 048
SHAP 050
SHAP 058
SHAP 060
SHAP 068
SHAP 070
SHAP 078
SHAP 080
SHAP 088
SHAP 090
SHAP 098
SHAP 100
SHAP 108
SHAP 110
SHAP 118
SHAP 120
SHAP 128
SHAP 130
SHAP 138
SHAP 140
SHAP 148
SHAP 150
SHAP 158
SHAP 160
SHAP 168
SHAP 170
SHAP 178
SHAP 180
SHAP 188
SHAP 190
SHAP 198
SHAP 200
SHAP 208
SHAP 210
SHAP 218
SHAP 220
SHAP 228
SHAP 230
SHAP 238
SHAP 240
SHAP 248
SHAP 250
SHAP 258

```

ORNL-5601

116

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

APPENDIX C (continued)

```

30      DO 40 IB=1, IHALFB
        NJ=NI+IB
        IF (NJ.GT.NNP) GO TO 50
        JB=JHBP-IB
        R(NJ)=R(NJ)-BB(NPP)*C(NJ,JB)
40
ZERO COLUMN NPN(NPP) OF MATRIX C(NP,IB)
50      DO 60 IB=1, IHALFB
        NJ=NI-IB
        IF (NJ.LT.1) GO TO 70
        JB=IHBP+IB
        C(NJ,JB)=0.0
60      DO 80 IB=1, IHALFB
        NJ=NI+IB
        IF (NJ.GT.NNP) GO TO 90
        JB=IHBP-IB
        C(NJ,JB)=0.0
80
90      CONTINUE

100 IF(NRSEL.LE.0) GO TO 140
ENTER SEEPAGE TERMS IN MATRIX C(NP,IB)
      W1=W
      W2=1.-W
      IF (K.S.NE.0) GO TO 110
      W1=1.
      W2=0.
110 DO 130 MP=1,NRSEL
      M=NRSE(MP)
      ALPHA1=WETAB(M,1)
      ALPHA2=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=IF(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
CALL QASP(DFLXQ,KQ,DL(MP),DCOSX(MP),DCOSZ(MP),XQ,ZQ,VXQ, VZQ)
      IB=IHBP

```

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

APPENDIX C (continued)

```

C      SUBROUTINE Q4SP(DFLXQ,KQ,DL,DCOSXQ,DCOSZQ,XQ,ZQ,YXQ,VZQ)
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)
C      REAL*8 N(4)
C
C      DIMENSION W(4),DNX(4),DNZ(4),DWX(4),DWZ(4)
C      DIMENSION PJAB(2,2),DNSS(4),DNLT(4),DWSS(4),DWLT(4)
C      DIMENSION S(4,4),T(4,4),DFLXQ(2,2),KQ(2),SSA(2),TTA(2),
C      > XQ(4),ZQ(4),AKXQ(4),AKZQ(4),VXQ(4),VZQ(4)
C
C      DATA S/0.00,-.5773500,0.00,-1.00,.5773500,0.00,1.00,0.00,0.00,
C      > 1.00,0.00,-.5773500,-1.00,0.00,.5773500,0.00/, T/0.00,-1.00,0.00,
C      > -.5773500,-1.00,0.00,-.5773500,0.00,0.00,.5773500,0.00,1.00,
C      > .5773500,0.00,1.00,0.00/
C
C      INITIALIZE NODAL COMPONENTS OF LINE INTEGRAL
C
C      DO 10 JQ=1,2
C      DO 10 IQ=1,2
C      10      DFLXQ(IQ,JQ)=0.
C
C      DETERMINE LOCAL COORDINATES OF GAUSS-INTEGRATION POINTS KG
C
C      LQ=KQ(1)
C      MQ=KQ(2)
C      SSA(1)=S(LQ,MQ)
C      TTA(1)=T(LQ,MQ)
C      SSA(2)=S(MQ,LQ)
C      TTA(2)=T(MQ,LQ)
C      DO 40 KG=1,2
C      SS = SSA(KG)
C      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
C      CALL SHAPE(N,W,DNSS,DNLT,DWSS,DWLT,SS,TT)
C
C      ( NOTE THE INVERSE TRANSFORMATION MATRIX I.E. THE INVERSE JACOBIAN:
C      F AB(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

```

```

Q4SP 005
Q4SP 010
Q4SP 015
Q4SP 020
Q4SP 025
Q4SP 030
Q4SP 035
Q4SP 040
Q4SP 045
Q4SP 050
Q4SP 055
Q4SP 060
Q4SP 065
Q4SP 070
Q4SP 075
Q4SP 080
Q4SP 085
Q4SP 090
Q4SP 095
Q4SP 100
Q4SP 105
Q4SP 110
Q4SP 115
Q4SP 120
Q4SP 125
Q4SP 130
Q4SP 135
Q4SP 140
Q4SP 145
Q4SP 150
Q4SP 155
Q4SP 160
Q4SP 165
Q4SP 170
Q4SP 175
Q4SP 180
Q4SP 185
Q4SP 190
Q4SP 195
Q4SP 200
Q4SP 205
Q4SP 210
Q4SP 215
Q4SP 220
Q4SP 225
Q4SP 230
Q4SP 235
Q4SP 240
Q4SP 245
Q4SP 250

```

APPENDIX C (continued)

```

      DO 11 I=1,2
      DO 11 J=1,2
      PJAB(I,J)=0.0
      CONTINUE
11
C
      DO 16 I=1,4
      PJAB(1,1)=PJAB(1,1) + ZQ(I)*DNST(I)
      PJAB(1,2)=PJAB(1,2) - ZQ(I)*DNSS(I)
      PJAB(2,1)=PJAB(2,1) - XQ(I)*DNST(I)
      PJAB(2,2)=PJAB(2,2) + XQ(I)*DNSS(I)
16
      CONTINUE
C
C
C
      COMPUTE THE DETERMINANT OF THE JACOBIAN
      DJAC=PJAB(2,2)*PJAB(1,1) - PJAB(1,2)*PJAB(2,1)
      DJACI=1.000/DJAC
C
      DO 17 I=1,2
      DO 17 J=1,2
      PJAB(I,J)=PJAB(I,J)*DJACI
17
C
      DO 18 I=1,4
      DNK(I)=PJAB(1,1)*DNSS(I) + PJAB(1,2)*DNST(I)
      DNZ(I)=PJAB(2,1)*DNSS(I) + PJAB(2,2)*DNST(I)
      DWX(I)=PJAB(1,1)*DWSS(I) + PJAB(1,2)*DWST(I)
      DWZ(I)=PJAB(2,1)*DWSS(I) + PJAB(2,2)*DWST(I)
18
C
C
C
      INTERPOLATE WITH FUNCTIONS N(IQ), DNK(IQ), DNZ(IQ) TO OBTAIN
      VALUES OF DARCY VELOCITIES VXQP AND VZQP
      VVK=0.
      VZK=0.
      DO 20 IQ=1,4
      VVK=VVK+VXQ(IQ)*N(IQ)
      VZK=VZK+VZQ(IQ)*N(IQ)
20
C
C
C
      EVALUATE THE NORMAL DARCY VELOCITY AT THE GAUSS POINT AND ACCUMULATE
      THE INTEGRAL SUMS
      VNK=VVK+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)
      DFLXQ(IQ,JQ)=DFLXQ(IQ,JQ)+W(LQ)*VVK*N(MQ)
30
      CONTINUE
40
      DO 50 JQ=1,2

```

```

04SP 255
04SP 260
04SP 265
04SP 270
04SP 275
04SP 280
04SP 285
04SP 290
04SP 295
04SP 300
04SP 305
04SP 310
04SP 315
04SP 320
04SP 325
04SP 330
04SP 335
04SP 340
04SP 345
04SP 350
04SP 355
04SP 360
04SP 365
04SP 370
04SP 375
04SP 380
04SP 385
04SP 390
04SP 395
04SP 400
04SP 405
04SP 410
04SP 415
04SP 420
04SP 425
04SP 430
04SP 435
04SP 440
04SP 445
04SP 450
04SP 455
04SP 460
04SP 465
04SP 470
04SP 475
04SP 480
04SP 485
04SP 490
04SP 495
04SP 500

```

APPENDIX C (continued)

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

```
Q4SP 505
Q4SP 510
Q4SP 515
Q4SP 520
```

APPENDIX C (continued)

```

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

```

```

SOL V 005
SOL V 010
SOL V 015
SOL V 020
SOL V 025
SOL V 030
SOL V 035
SOL V 040
SOL V 045
SOL V 050
SOL V 055
SOL V 060
SOL V 065
SOL V 070
SOL V 075
SOL V 080
SOL V 085
SOL V 090
SOL V 095
SOL V 100
SOL V 105
SOL V 110
SOL V 115
SOL V 120
SOL V 125
SOL V 130
SOL V 135
SOL V 140
SOL V 145
SOL V 150
SOL V 155
SOL V 160
SOL V 165
SOL V 170
SOL V 175
SOL V 180
SOL V 185
SOL V 190
SOL V 195
SOL V 200
SOL V 205
SOL V 210
SOL V 215
SOL V 220
SOL V 225
SOL V 230
SOL V 235
SOL V 240
SOL V 245
SOL V 250

```

APPENDIX C (continued)

```

      KB=IB+IHALFB
      LB=IHBP-IB
      DO 30 MB=JB,KB
        NB=LB+MB
        C(NL,MB)=C(NL,MB)+A*C(NI,NB)
30      CONTINUE
40      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
          SUM=SUM+C(NI,JB)*R(NJ)
100      R(NI)=(R(NI)-SUM)/C(NI,IHBP)
110      MB=IBAND
      DO 130 IB=NL,NNP
        NI=NU-1+IB
        NJ=NI
        SUM=0.0
        DO 120 JB=NL,MB
          NJ=NJ+1
          SUM=SUM+C(NI,JB)*R(NJ)
120

```

```

SQL V 255
SQL V 260
SQL V 263
SQL V 270
SQL V 275
SQL V 280
SQL V 285
SQL V 290
SQL V 295
SQL V 300
SQL V 305
SQL V 310
SQL V 315
SQL V 320
SQL V 325
SQL V 330
SQL V 335
SQL V 340
SQL V 345
SQL V 350
SQL V 355
SQL V 360
SQL V 365
SQL V 370
SQL V 375
SQL V 380
SQL V 385
SQL V 390
SQL V 395
SQL V 400
SQL V 405
SQL V 410
SQL V 415
SQL V 420
SQL V 425
SQL V 430
SQL V 435
SQL V 440
SQL V 445
SQL V 450
SQL V 455
SQL V 460
SQL V 465
SQL V 470
SQL V 475
SQL V 480
SQL V 485
SQL V 490
SQL V 495
SQL V 500

```

APPENDIX C (continued)

130 R(NI)=(R(NI)-SUM)/C(NI,IHBP)
RETURN
END

SOLV 505
SOLV 510
SOLV 515

ORNL-5601

APPENDIX C (continued)

```

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,OH)
C
C FUNCTION OF SUBROUTINE--TO COMPUTE BOUNDARY FLUXES, FLOW RATES,
C INCREMENTAL FLOWS OCCURRING DURING TIME DELT, TOTAL FLOWS SINCE
C TIME ZERO, AND THE CHANGE IN MOISTURE CONTENT FOR THE ENTIRE
C SYSTEM DURING TIME DELT.
C
C IMPLICIT REAL*8 (A-H,O-Z)
C REAL*8 KO,LAMBDA
C
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
C DIMENSION R(MAXNP),FX(MAXNP),FZ(MAXNP),DH(MAXNP),TH(MAXEL,4)
C DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),BFLX(MAXBNP),
C > BFLXP(MAXBNP),NBE(MAXBEL),ISB(MAXBEL,4),NPB(MAXBNP)
C DIMENSION NPRS(MXRSNP),NPST(MXSTNP),NPN(MAXBCN)
C DIMENSION PROP(MAXMAT,MXMPPM)
C DIMENSION FRATE(10),FLOW(10),TFLOW(10)
C
C DIMENSION XQ(4),ZQ(4),RQ(4),DHQ(4),THQ(4)
C
C COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
C COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN,NST
C COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
C DATA QR,OD,QL,ODH /0.000,0.000,0.000,0.000/
C
C CALCULATE NODAL FLOW RATES
C
C DO 10 NP=1,NBN
C   BFLXP(NP)=BFLX(NP)
C   BFLX(NP)=0.
C
C DO 30 MP=1,NBEL
C   M=NBE(MP)
C   NI=ISB(MP,1)
C   NJ=ISB(MP,2)
C   DO 20 I=1,NBN
C     IJ=NPB(I)
C     IF(IJ,NE,NI) GO TO 20
C     NFI=I
C     GO TO 22
C   CONTINUE
C 20 DO 25 J=1,NBN
C 22   IJ=NPB(J)
C     IF(IJ,NE,NJ) GO TO 25
C     NJJ=J

```

```

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

```

ORNL-5601

SFL0	255
SFL0	260
SFL0	265
SFL0	270
SFL0	275
SFL0	280
SFL0	285
SFL0	290
SFL0	295
SFL0	300
SFL0	305
SFL0	310
SFL0	315
SFL0	320
SFL0	325
SFL0	330
SFL0	335
SFL0	340
SFL0	345
SFL0	350
SFL0	355
SFL0	360
SFL0	365
SFL0	370
SFL0	375
SFL0	380
SFL0	385
SFL0	390
SFL0	395
SFL0	400
SFL0	405
SFL0	410
SFL0	415
SFL0	420
SFL0	425
SFL0	430
SFL0	435
SFL0	440
SFL0	445
SFL0	450
SFL0	455
SFL0	460
SFL0	465
SFL0	470
SFL0	475
SFL0	480
SFL0	485
SFL0	490
SFL0	495
SFL0	500

126

APPENDIX C (continued)

```

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)
    SP=SP+BFLXP(NII)
  70 FRATE(2)=S
    FLOW(2)=.5*(S+SP)*DELT

C
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)
        SP=SP+BFLXP(NII)
    90 FRATE(3)=S
      FLOW(3)=.5*(S+SP)*DELT

C
C C NUMERICAL FLOW THROUGH UNSPECIFIED BOUNDARY NODES
C
  100 S=0.
    SP=0.
    DO 110 I=1,3
      S=S+FRATE(I)
      SP=SP+FLOW(I)
    110 FRATE(4)=FRATE(5)-S
      FLOW(4)=FLOW(5)-SP

C
C C CALCULATE THE INCREASES IN THE INTEGRATED MATERIAL CONTENTS FOR THE
C C C FLUID AND THE SOLID PHASES AND DETERMINE LOSSES DUE TO RADIOACTIVE
C C C DECAY
    QRP=QR

```

```

SFLO 505
SFLO 510
SFLO 515
SFLO 520
SFLO 525
SFLO 530
SFLO 535
SFLO 540
SFLO 545
SFLO 550
SFLO 555
SFLO 560
SFLO 565
SFLO 570
SFLO 575
SFLO 580
SFLO 585
SFLO 590
SFLO 595
SFLO 600
SFLO 605
SFLO 610
SFLO 615
SFLO 620
SFLO 625
SFLO 630
SFLO 635
SFLO 640
SFLO 645
SFLO 650
SFLO 655
SFLO 660
SFLO 665
SFLO 670
SFLO 675
SFLO 680
SFLO 685
SFLO 690
SFLO 695
SFLO 700
SFLO 705
SFLO 710
SFLO 715
SFLO 720
SFLO 725
SFLO 730
SFLO 735
SFLO 740
SFLO 745
SFLO 750

```

APPENDIX C (continued)

```

QDP=QD
QLP=QL
QDHP=QDH
QR=0.
QD=0.
QL=0.
QDH=0.0
DO 130 M=1,NEL
  ALPHA1=WETAB(M,1)
  ALPHA2=WETAB(M,3)
  BETA1=WETAB(M,2)
  BETA2=WETAB(M,4)
  MTYP=IE(M,5)
  KD=PROP(MTYP,1)
  RHOB=PROP(MTYP,2)
  LAMBOA=PROP(MTYP,3)
  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(IQ)=DH(NP)
    THQ(IQ)=TH(M,IQ)
    RQ(IQ)=R(NP)
    THRQ(IQ)=THQ(IQ)*RQ(IQ)
  120
  CALL C4R(QRM,QDM,QDHM,XQ,ZQ,RQ,DHQ,THRQ,THQ,RHOB,KD)
  QR=QR+QRM
  QD=QD+QDM+RHOB*KD
  QLM=QRM+QDM
  QL=QL+LAMBOA*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+QLD)
  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

```

```

SFLO 755
SFLO 760
SFLO 765
SFLO 770
SFLO 775
SFLO 780
SFLO 785
SFLO 790
SFLO 795
SFLO 800
SFLO 805
SFLO 810
SFLO 815
SFLO 820
SFLO 825
SFLO 830
SFLO 835
SFLO 840
SFLO 845
SFLO 850
SFLO 855
SFLO 860
SFLO 865
SFLO 870
SFLO 875
SFLO 880
SFLO 885
SFLO 890
SFLO 895
SFLO 900
SFLO 905
SFLO 910
SFLO 915
SFLO 920
SFLO 925
SFLO 930
SFLO 935
SFLO 940
SFLO 945
SFLO 950
SFLO 955
SFLO 960
SFLO 965
SFLO 970
SFLO 975
SFLO 980
SFLO 985

```

APPENDIX C (continued)

```

C      SUBROUTINE Q4R(QRM,QDM,QDHM,XQ,ZQ,RQ,DHQ,THHQ,THQ,RHQB,KD)
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)
C      REAL*8 N(4),KD
C
C      DIMENSION XQ(4),ZQ(4),RQ(4),DHQ(4),THHQ(4),THQ(4)
C      DIMENSION S(4),T(4)
C      DIMENSION W(4)
C      DIMENSION PJAB(2,2),DNSS(4),DNNT(4),DWSS(4),DWTT(4)
C
C      DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-
C      > 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
C      QRM=0.
C      QDM=0.0
C      QDHM=0.0
C      DO 20 KG=1,4
C
C      DETERMINE LOCAL COORDINATES (S,TT) OF GAUSS-INTEGRATION POINT KG
C
C      SS = P*S(KG)
C      TT = P*T(KG)
C
C      EVALUATE THE JACOBIAN DJAC
C
C      CALL SHAPE(N,W,DNSS,DNNT,DWSS,DWTT,SS,TT)
C
C      DO 11 I=1,2
C      DO 11 J=1,2
C      PJAB(I,J)=0.0
C      CONTINUE
C      11 DO 16 I=1,4
C      PJAB(1,1)=PJAB(1,1) + ZQ(I)*DNNT(I)
C      PJAB(1,2)=PJAB(1,2) - ZQ(I)*DNSS(I)
C      PJAB(2,1)=PJAB(2,1) - XQ(I)*DNNT(I)
C      PJAB(2,2)=PJAB(2,2) + XQ(I)*DNSS(I)
C      CONTINUE
C      16 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
C      RQP=0.

```

```

Q4R 005
Q4R 010
Q4R 015
Q4R 020
Q4R 025
Q4R 030
Q4R 035
Q4R 040
Q4R 045
Q4R 050
Q4R 055
Q4R 060
Q4R 065
Q4R 070
Q4R 075
Q4R 080
Q4R 085
Q4R 090
Q4R 095
Q4R 100
Q4R 105
Q4R 110
Q4R 115
Q4R 120
Q4R 125
Q4R 130
Q4R 135
Q4R 140
Q4R 145
Q4R 150
Q4R 155
Q4R 160
Q4R 165
Q4R 170
Q4R 175
Q4R 180
Q4R 185
Q4R 190
Q4R 195
Q4R 200
Q4R 205
Q4R 210
Q4R 215
Q4R 220
Q4R 225
Q4R 230
Q4R 235
Q4R 240
Q4R 245
Q4R 250

```

ORNL-5601

Q4R	255
Q4R	260
Q4R	265
Q4R	270
Q4R	275
Q4R	280
Q4R	285
Q4R	290
Q4R	295
Q4R	300
Q4R	305
Q4R	310
Q4R	315
Q4R	320
Q4R	325
Q4R	330
Q4R	335

APPENDIX C (continued)

```

SUBROUTINE PRINTT(R,FX,FZ, FRATE, FLOW, TFLOW, TIME, DELT, KPR, KOUT,
> KDIG, MAXNP, NNP, IBAND)

C
C C C 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
C IMPLICIT REAL*8 (A-H,O-Z)
C
C DIMENSION R(MAXNP),FX(MAXNP),FZ(MAXNP)
C DIMENSION FRATE(10),FLOW(10),TFLOW(10)
C IMPLICIT REAL*8 (A-H,O-Z)
C IF (KPR.EQ.0) RETURN
C IF (KOUT.EQ.0) GO TO 10
C
C C C C C
C PRINT DIAGNOSTIC FLOW INFORMATION
C
C KDIG=KDIG+1
C PRINT 10000,KDIG,TIME,DELT
C PRINT 10500, (FRATE(I),FLOW(I),TFLOW(I),I=1,9)
C 10 IF (KPR.EQ.1) RETURN
C
C C C C C
C PRINT CONCENTRATIONS
C
C KOUT=KOUT+1
C PRINT 10000,KOUT,TIME,DELT,IBAND
C KLINE=-1
C DO 20 NI=1,NNP,2
C   NJMN=NI
C   NJMX=MINO(NI+7,NNP)
C   KLINE=KLINE+1
C   IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) PRINT 10000,KOUT,
C   > TIME,DELT,IBAND
C 20 PRINT 10100,NI,(R(NJ),NJ=NJMN,NJMX)
C   IF (KPR.EQ.2) RETURN
C
C C C C C
C PRINT MATERIAL FLUX
C
C KOUT=KOUT+1
C PRINT 10200,KOUT,TIME,DELT,IBAND
C KLINE=-1
C DO 30 NI=1,NNP,4
C   NJMN=NI
C   NJMX=MINO(NI+3,NNP)
C   KLINE=KLINE+1
C   IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) PRINT 10200,KOUT,
C   > TIME,DELT,IBAND
C 30 PRINT 10300, (HJ,FX(NJ),FZ(NJ),NJ=NJMN,NJMX)
C   RETURN

```

```

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

```

APPENDIX C (continued)

```

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(17H1PD15.4)
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,5X,71FZ,6X,4HNODE,9X,2HFX,9X,2HFZ/)
10300 FORMAT(4(15,2E11.3,5X))
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

```

```

PRIN 255
PRIN 260
PRIN 263
PRIN 270
PRIN 275
PRIN 280
PRIN 285
PRIN 290
PRIN 295
PRIN 300
PRIN 305
PRIN 310
PRIN 315
PRIN 320
PRIN 325
PRIN 330
PRIN 335
PRIN 340
PRIN 345
PRIN 350
PRIN 355
PRIN 360
PRIN 365
PRIN 370
PRIN 375

```


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
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      DIMENSION TITLE(9)
C      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),R(MAXNP),FX(MAXNP),
C      > FZ(MAXNP)
C
C      DATA NPPROB/-1/
C
C      IF (NPPROB.EQ.(-1)) REWIND 2
C      IF (NPPROB.EQ.NPROB) GO TO 10
C      WRITE(2) (TITLE(I),I=1,9),NPROB,NNP,NEL,NTI
C      WRITE(2) (X(NP),NP=1,NNP),(Z(NP),NP=1,NNP),((IE(M,I0),M=1,NEL),
C      > I0=1,4)
C      NPPROB=NPROB
10 WRITE(2) TIME,(R(NP),NP=1,NNP),(FX(NP),NP=1,NNP),(FZ(NP),NP=1,NNP)
C      RETURN
C      END

```

```

STOR 005
STOR 010
STOR 015
STOR 020
STOR 025
STOR 030
STOR 035
STOR 040
STOR 045
STOR 050
STOR 055
STOR 060
STOR 065
STOR 070
STOR 075
STOR 080
STOR 085
STOR 090
STOR 095
STOR 100
STOR 105
STOR 110
STOR 115
STOR 120
STOR 125
STOR 130

```