2920

GLOVER

Copy prior to Publication (Approved)

A DUAL-POROSITY MODEL FOR SIMULATING SOLUTE TRANSPORT IN OIL SHALE

U.S. GEOLOGICAL SURVEY

Water-Resources Investigations Report 85-4281

A DUAL-POROSITY MODEL FOR SIMULATING SOLUTE TRANSPORT IN OIL SHALE

By Kent C. Glover

U.S. GEOLOGICAL SURVEY

Water-Resources Investigations Report 85-4281

Cheyenne, Wyoming

1986

UNITED STATES DEPARTMENT OF THE INTERIOR

DONALD PAUL HODEL, Secretary

GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information
write to:

District Chief
U.S. Geological Survey
2120 Capitol Avenue
Box 1125
Cheyenne, Wyoming 82003

Copies of this report can
be purchased from:

Open-File Services Section
Western Distribution Branch
U.S. Geological Survey
Box 25425, Federal Center
Denver, Colorado 80225
(Telephone: (303) 236-7476)

CONTENTS

ILLUSTRATIONS

TABLES

CONVERSION DATUM

For those readers interested in using the metric system, the following table may be used to convert the inch-pound units of measurement used in this report to metric units:

Multiply	By	To obtain
foot	0.3048	meter
foot per day	0.3048	meter per day
foot squared per day	0.09290	meter squared per day

A DUAL-POROSITY MODEL FOR SIMULATING SOLUTE TRANSPORT IN OIL SHALE

By Kent C. Glover

ABSTRACT

A model is described for simulating three-dimensional ground-water flow and solute transport in oil shale and associated hydrogeologic units. The model treats oil shale as a dual-porosity medium by simulating flow and transport within fractures using the finite-element method. Diffusion of solute between fractures and the essentially static water of the shale matrix is simulated by including an analytical solution that acts as a source-sink term to the differential equation of solute transport. While knowledge of fracture orientation and spacing is needed to effectively use the model, it is not necessary to map the locations of individual fractures.

The computer program listed in the report incorporates many of the features of previous dual-porosity models while retaining a practical approach to solving field problems. As a result the report does not extend the theory of solute transport in any appreciable way. The emphasis in the report is on bringing together various aspects of solute-transport theory in a manner that is particularly suited to the unusual ground-water flow and solute-transport characteristics of oil-shale systems.

INTRODUCTION

Digital models of ground-water flow and solute transport have been used to predict possible impacts of oil-shale development on the ground-water resource (Robson and Saulnier, 1981). Unfortunately, modeling techniques used in the past either were not developed for use in rock, such as oil shale, where porosity depends on fracturing, or required data that usually are not available. From 1980 to 1983 the U.S. Geological Survey studied migration of solute from an in situ oil-shale retort near Rock Springs, Wyo. (fig. 1). The emphasis of the study was to identify geo-

---

Figure 1.--(caption on next page) belongs near here.

---

logic, hydraulic, and chemical factors that control the process of solute transport in oil shale. During the study, it became apparent that no existing model was well suited for the unusual flow and transport characteristics of oil-shale systems. Therefore, as part of the study, a model was developed.

This report describes two digital-computer programs that are used to simulate ground-water flow and solute transport in oil shale with interbedded tuff or sandstone. The U.S. Department of Energy experimental, in situ, oil-shale retort near Rock Springs, Wyo., was used as the prototype flow system of the model. The application of modeling techniques to this flow system is described in a report by K. C. Glover, U.S. Geological Survey, written commun. (1985). The model is designed for use in site-specific studies and consequentially includes many features, such as diffusion of solute between fractures and the shale matrix, that may not be important to basin-wide or regional investigations.

Figure 1.--Location of in situ oil-shale retort near Rock Springs, Wyoming.

A wide array of ground-water flow and solute-transport models have become available in recent years. Konikow and Bredehoeft (1978), and INTERCOMP Resource Development and Engineering, Inc. (1976) describe two of the more commonly used programs. Other models that are potentially useful in studies of oil-shale hydrology are described by Bibby (1981), Noorishad and Mehran (1982), and Rasmuson and others (1982). The model described in this report incorporates many of the features of the above models while retaining a practical approach to the solution of field problems. Boundary conditions can be applied easily and mapping of individual fractures is not necessary. As a result of this approach, the present model does not extend the theory of solute transport in fractured media in any appreciable way. Instead, various aspects of this theory are brought together in a manner that is particularly suited to the unusual ground-water flow and solute-transport characteristics of oil-shale systems.

Throughout the remainder of this report it is assumed that the reader is familiar with the basics of solute-transport modeling in porous media, such as sand-and-gravel aquifers, as well as the finite-element method of numerical analysis. It is believed that most hydrologists faced with a problem of solute migration in oil shale will have faced similar problems in unfractured media. For those readers unfamilar with applications of solute transport modeling using the finite-element method, it is suggested that teaching references such as Freeze and Cherry (1979), Konikow and Bredehoeft (1978), and Zienkiewicz (1971) be consulted.

OIL SHALE AS A DUAL-POROSITY MEDIUM

Porosity in oil-shale formations may be classified on the basis of its relationship to hydraulic conductivity. Porosity may be considered effective for increasing hydraulic conductivity if the pores are interconnected or not effective, as in shale, if the pores are relatively isolated. Porosity due to faults, joints, collapse breccia, and solution cavities is the major source of permeability in oil shale with otherwise low permeability (Robson and Saulnier, 1981). Persistent layers of permeable tuff or sandstone, ranging in thickness from less than an inch to several inches, also are common in oil shale (Bradley, 1964) and contribute to the hydraulic conductivity. The porosity of the shale and marlstone matrix, although relatively high, does not contribute significantly to hydraulic conductivity.

Any attempt to simulate ground-water flow and solute transport in oil shale must include consideration of the dual-porosity nature of these sediments. The solute-transport processes of hydrodynamic dispersion and advection are related directly to seepage velocity and therefore are related to the effective porosity of the formation. However, several investigators including Grisak and Pickens (1980) and Bibby (1981) have noted that dispersion and advection alone cannot account for the distribution of non-reacting solute in fractured formations. The extremely low seepage velocity of water within the shale matrix minimizes dispersion and advection, increasing the relative importance of diffusion. The porosity of the shale matrix, although not contributing to hydraulic conductivity, is important to dispersion.

METHODS OF SIMULATION

Ground-Water Flow

Only the part of formation porosity that contributes to hydraulic conductivity needs to be considered in analysing ground-water flow systems. For most practical purposes this porosity in oil shale, whether resulting from faulting, solution channels, or thin beds of permeable tuff and sandstone, can be analysed using the techniques of Snow (1969). He showed that many problems of flow through dual-porosity media can be solved using an anisotropic hydraulic-conductivity tensor in conjunction with standard porous-media techniques. His approach to the problem of ground-water flow is valid if the formation has a fracture density that is high compared to the scale of the problem. In such a case, the hydraulic characteristics of the fractured formation are similar to those of granular media. If fracture spacing is irregular in a given direction, the formation will exhibit heterogeneity. Because mapping of individual fractures is impractical for most field problems, the approach of Snow (1969) is used in this model. From a practical viewpoint the model described in this report may be useful when the hydrologist has some knowledge of the average spacing, aperture size, and directions of fracturing, but accurate mapping of individual fractures is not possible. Through proper alignment of the model cartesian coordinate axes with principal directions of fracturing and use of heterogeneous hydraulic conductivity values, many different fracture geometries can be simulated.

The three-dimensional nature of a fracture system, along with the essentially horizontal bedding plane of most formations, generally results in a three-dimensional flow system in oil-shale strata. Three-dimensional flow systems in oil shale have been observed on both a regional scale by Robson and Saulnier (1981) and on a local scale by the author at the Department of Energy in situ oil-shale retort near Rock Springs, Wyo. Therefore, the program given in this report is intended for use in studies of three-dimensional ground-water flow.

Rocks in the vicinity of an in situ retort are altered due to induced fracturing. As a result fracture patterns in the retort chamber are significantly different than regional fracture patterns. The values and degree of anisotropy of hydraulic conductivity within a retort chamber also differ from regional estimates. Nevertheless the fracture density usually is high, compared to the dimensions of the retort chamber, and standard porous-media techniques can be used to simulate ground-water flow in the immediate vicinity of the retort chamber.

Solute Transport

The extension of solute-transport theory to dual-porosity media has attracted considerable attention in water-resources literature. Much of the literature has been directed toward studies of radionuclide migration through individual fractures and as such is not directly applicable to field problems where mapping of individual fractures is impractical. Grisak and Pickens (1980) modeled fractured media by using separate finite elements to represent fractures and adjacent low permeable rock. Finite elements used to represent the fractures were modeled with material properties that differed from adjacent low-permeability elements. This approach was extended by Noorishad and Mehran (1982), who also introduced the use of an "upstream-weighting" technique to simulate more accurately the essentially advective transport that occurs within the fractures.

Bibby (1981) took a different approach to modeling solute movement through dual-porosity media. Molecular diffusion between fractures and the primarily static water in the shale matrix was incorporated in the model as a source-sink term utilizing an appropriate analytical solution. To apply this two-dimensional model the user needs to know average spacing, aperture diameter, and direction of fractures but does not need to map individual fractures. This approach is analogous to the method of flow analysis discussed by Snow (1969). Although no upstream-weighting technique was used, Bibby (1981) reported no difficulty in applying the model to a field problem in a limestone aquifer.

Rasmuson and others (1982) presented a model for simulating three-dimensional solute transport in fractured rocks that is based on an integrated finite-difference approach. The model produces excellent results if solute transport is dominated by hydrodynamic dispersion. Because advection dominates most solute-transport problems in oil-shale strata (Robson and Saulnier, 1981), the approach of Rasmuson and others (1982) is not considered practical for use with oil shale.

The approach to modeling solute transport presented in this report is basically the model described by Bibby (1981) and extended to three dimensions. This approach has been selected because it strikes a balance between theoretical accuracy and practical applicability. Accurate mapping of fractures is not needed to use the model, although some knowledge of the fracture system is required. The ease with which boundary conditions can be applied also supports this approach. One disadvantage of the approach is the difficulty of extending the analytical solution for diffusion between fractures and the shale matrix to problems of multiple-species transport.

A model that can be extended to problems of multiple-species transport is described by Huyakorn and others (1983). Unfortunately, fracture locations must be accurately mapped to use the model successfully. Therefore the model of Huyakorn and others (1983), although theoretically superior to the model of Bibby (1981), cannot be applied in most practical field problems.

APPLICATION OF THE FINITE-ELEMENT METHOD

Ground-Water Flow

The basic governing equation for three-dimensional flow, when Cartesian coordinate axes are aligned with the principal components of the hydraulic-conductivity tensor, is as follows:

$$\frac{\partial}{\partial x_i}\left(K_i \frac{\partial h}{\partial x_i}\right) + W = S_s \frac{\partial h}{\partial t} \qquad i = 1,2,3 \qquad (1)$$

where K_i = the hydraulic conductivity tensor $[LT^{-1}]$;

W = the source-sink function (positive for a source) $[T^{-1}]$;

S_s = specific storage $[L^{-1}]$;

h = the hydraulic head [L];

x_i = Cartesian coordinate [L], and use of a repeated subscript indicates summation on that subscript; and

t = time [T].

The source-sink term may be distributed areally or may represent a well. Boundary conditions that may be applied on the periphery of the problem area include known specific discharge normal to the boundary or known hydraulic head. Parameters K_i, W, and S_s are approximated by subdividing the region of interest into discrete zones. Parameters are assumed to be constant within each zone which gives rise to internal boundary conditions at zonal discontinuities. Along these internal boundaries both hydraulic head and normal specific discharge must remain unchanged as the boundary is crossed.

Equation 1 may be solved by using the finite-element method based on the Galerkin criteria. Details of this method are described in Zienkiewicz (1971) and Pinder and Gray (1977). Within this report, first-order basis functions are used to describe each cubic element. While use of these basis functions requires more elements than if higher-order functions were used, the reduced oscillatory behavior and integration time associated with linear functions make this simpler approach preferable.

The matrix equation resulting from the finite-element discretization of equation 1 is as follows:

$$(\frac{1}{\Delta t_m}\bar{\bar{C}} + \theta\bar{\bar{K}})\,\bar{h}^m = \bar{q} + (\frac{1}{\Delta t_m}\bar{\bar{C}} - (1-\theta)\bar{\bar{K}})\bar{h}^{\,m-1} \qquad (2)$$

where $\bar{\bar{C}}$ = a coefficient matrix involving specific storage (S_s) and calculated for a single finite element as $\iiint_V S_s\,\bar{n}\,(\bar{n})^t\,dV$;

$\bar{\bar{K}}$ = a coefficient matrix, with off-diagonal components equal to zero, involving hydraulic conductivity and calculated for a single finite element as

$$\iiint_V (K_i \frac{\partial \bar{n}}{\partial x_i}(\frac{\partial \bar{n}}{\partial x_i})^t)dV, \qquad i = 1,2,3;$$

$\bar{h}^m$ = the vector of hydraulic head at time m;

$\bar{q}$ = the known vector involving source-sink terms and specified flux boundary conditions;

Δt_m = the length of the time step;

θ = a number ranging from 0 for an explicit solution to 1 for an implicit solution;

V = the volume of the finite element;

$\bar{n}$ = the vector of finite-element shape functions; and

$(\,)^t$ = the transpose of the enclosed vector.

Equation 2 is solved by Gaussian elimination for banded symmetric matrices. Although other direct-solution techniques, such as Cholesky decompositon, can be used, the procedure used in this report has proven adequate. Gupta and Tanji (1976) has suggested the use of matrix solvers that take full advantage of the sparse nature of coefficient matrices in attempts to reduce core storage requirements. The trade-offs in these matrix solvers are increased disk access and increased computational time. With the introduction of virtual-memory operating systems used by most computers, such detailed attention to minimizing core storage is not required.

Solute Transport

The transport of a conservative solute in ground water is described by the following equation:

$$\frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial c}{\partial x_j}\right) - \frac{\partial}{\partial x_i}(cq_i) = \phi\frac{\partial c}{\partial t} - Wc^* \qquad i,j = 1,2,3 \qquad (3)$$

where D_{ij} = the hydrodynamic-dispersion coefficient $[L^2T^{-1}]$;

c = solute concentration $[ML^{-3}]$; and

ϕ = porosity [dimensionless];

$q_i = -K_i\frac{\partial h}{\partial x_i}$ = the Darcian fluid velocity $[LT^{-1}]$;

W = a source-sink function $[T^{-1}]$;

c^* = solute concentration in a fluid source $[ML^{-3}]$.

When applied to dual-porosity media such as oil shale, the terms are defined relative to the joint or fracture system, with the shale matrix considered impermeable. The hydrodynamic-dispersion coefficient (D_{ij}) is related to Darcian fluid velocity in the fracture system by

$$D_{ij} = a_{ijmn}\frac{q_m q_n}{/q/} + D^* \qquad i,j,m,n, = 1,2,3$$

where a_{ijmn} = dispersivity of the fractured media [L],

q_m and q_n = Darcian fluid velocity $[LT^{-1}]$,

$/q/$ = the magnitude of the Darcian velocity vector $[LT^{-1}]$, and

D^* = the molecular-diffusion coefficient $[L^2T^{-1}]$.

for isotropic media,

$a_{iiii} = \alpha_L$ = the longitudinal dispersivity for i = 1,2,3;

$a_{iijj} = \alpha_T$ = the transverse dispersivity for i,j = 1,2,3; and

$a_{ijij} = \frac{1}{2}(\alpha_L - \alpha_T)$ for i,j = 1,2,3.

There is no corresponding theory describing hydrodynamic dispersion in anisotropic media.

The form of the transport equation used in this report also is used by Pinder and Gray (1977), and Bibby (1981). Konikow and Bredehoeft (1978) and most studies of solute transport done by the U.S. Geological Survey use a form of the equation that is obtained by dividing equation 3 by porosity. The hydrodynamic-dispersion coefficient then is defined in terms of seepage velocity. Either form of the equation can be used successfully. However, when comparing results of studies using different forms of the transport equation, it should be recognized that the dispersion coefficients will differ by the magnitude of porosity.

Galerkin's method of weighted residuals, when applied to equation 3 with isoparametric finite elements, gives

$$\left(\frac{1}{\Delta t_m}\bar{\bar{P}} + \theta\bar{\bar{D}}\right)\bar{c}^{m} = qc^{*} + \left(\frac{1}{\Delta t_m}\bar{\bar{P}} - (1-\theta)\bar{\bar{D}}\right)\bar{c}^{m-1} \qquad (4)$$

where $\bar{\bar{P}}$ = a coefficient matrix involving porosity and is calculated for a single finite element as $\iiint_V \phi\, \bar{n}\, (\bar{n})^t\, dV$;

$\bar{\bar{D}}$ = a coefficient matrix involving hydrodynamic dispersion and advection and is calculated for a single finite element as $\iiint_V (D_{ij} \frac{\partial \bar{n}}{\partial x_i}(\frac{\partial \bar{n}}{\partial x_j})^t - q_i \frac{\partial n}{\partial x_i}(\bar{n})^t)dV \quad i,j, = 1,2,3;$

$\bar{c}^{m}$ = the vector of solute concentration in fractures at time m; and

qc^{*} = a known vector involving specified flux boundary conditions and source-sink terms (W), solute concentration in fluid sources (c*), and diffusion between fractures and the rock.

Detailed information on formulating this equation can be obtained from Bibby (1981) or Pinder and Gray (1977, p. 144-148). Equation 4 is solved by the Gauss-Doolittle method for banded, nonsymmetric matrices.

Diffusion within the Oil-Shale Matrix

The exchange of solute between water in a system of parallel fractures and the essentially static water in the adjacent oil-shale matrix can occur by molecular diffusion and can be simulated by the addition of a source-sink term to equation 4. Bibby (1981) gives this source-sink term as a convolution integral that is expressed here in discrete form at time t_m.

$$Wc^* = - \frac{\phi_B b}{\Delta t_m (f+b)} \left[c_B(t_m) - c_B(t_{m-1}) \right] \qquad (5)$$

where Wc^* = the mass flux of solute entering the fracture [$M/L^3/T$];

ϕ_B = the porosity of the oil-shale matrix [dimensionless];

b = the average distance between parallel fractures [L];

Δt_m = $[t_m - t_{m-1}]$, the length of the time step [T];

f = the average fracture width [L];

c_B = the average solute concentration in the oil-shale matrix [M/L^3];

$$c_B(t_m) = c(t_0) \left[1 - \frac{8}{\pi^2} \sum_{n=0}^{\infty} \frac{1}{E_n} \exp\left(-B_n t_m\right) \right] + \sum_{i=1}^{m} \{ c(t_m) - c(t_{i-1}) \} \left\{ 1 - \frac{8}{\pi^2 \Delta t_i} \sum_{n=0}^{\infty} \frac{1}{E_n B_n} \left\{ \exp\left[-B_n (t_m - t_i)\right] - \exp\left[-B_n (t_m - t_i)\right] \right\} \right\} ;$$

$c(t_0)$ = the solute concentration in the fracture at the beginning of the simulation [M/L^3];

n = an index of summation [dimensionless];

E_n = $(2n - 1)^2$;

B_n = $D_d (2n+1) \pi^2 / b^2$;

$c(t_i)$ = the solute concentration in the fracture at time t_i [M/L^3]; and

D_d = molecular diffusion of solute within the oil-shale matrix [L^2/T].

The only unknown variable in equation 5 is $c(t_m)$. Therefore, the coefficient associated with $c(t_m)$ is placed on the main diagonal of the $\bar{D}$ matrix in equation 4, while the remainder of the expression is placed in the right-hand vector of known values. Equation 5 involves summation over the entire period of the simulation. Fortunately, little truncation error is introduced by retaining only a small number of terms in the summations. The optimum number of terms to retain depends on the values of D_d, b and Δt_m; however, from information presented by Bibby (1981), it is apparent that between five and ten terms are adequate for most applications.

Several aquifer properties must be determined to use equation 5 successfully. These proporties include molecular diffusion within the oil-shale matrix (D_d), average distance between parallel fractures (b), average fracture width (f), and can be obtained for each of these properties, model calibration usually must be used to improve upon the estimates. Therefore equation 5 generally is useful only in studies with well distributed and frequent measurements of solute concentration in wells.

Sources of Solute Within the Retort Chamber

Mechanisms for creating and leaching chemicals from burned shale within an in situ retort chamber to the formation water are poorly understood. Knowledge of conditions during the retort process is needed to determine the chemicals that are created, while an understanding of dissolution controls is needed to predict how the created chemicals are transferred from the burned shale to the formation water.

The retort temperature, total time of retorting, permeability of the retort chamber, and initial mineralogy affect the type and amount of chemicals created during a burn. During combustion of a retort chamber, a "flame front" or zone of combustion moves through the fractured oil-shale bed. Hot gases from this combustion move ahead of the flame front and provide energy for pyrolysis of kerogen. In the process, a number of chemical by-products are created. A discussion of how these by-products are created is outside the scope of this report. However, it is important to recognize that the resulting chemicals will vary from retort chamber to retort chamber. In large chambers, the distribution of chemical by-products within the chamber also may be important. Coring of an abandoned retort chamber before it resaturates and lab analysis of the cores by long-term leaching studies can provide some understanding of the chemicals that are created during retorting.

After the retort operation ceases, the chamber resaturates and chemicals created during the burn are leached into the water. During the time that it takes to resaturate the retort chamber a rapid increase in the concentration of solute occurs. For many studies of solute transport, an equation to describe the initial dissolution is not needed because migration of solute from the retort chamber is unlikely until the formation resaturates.

After the retort chamber resaturates, Hall (1982) proposed that diffusion between the pores of the oil-shale matrix and fractures acts as a dominant mechanism for introducing solute to a ground-water system. This mechanism is simulated by equation 5. As discussed previously, equation 5 describes diffusion between the oil-shale matrix and a system of parallel fractures. Fracturing within a retort chamber is much more irregular but can be conceptualized as two systems of parallel fractures, one vertical and one horizontal, resulting in cubes of oil-shale. Matrix diffusion within the retort chamber can be simulated by applying equation 5 once for each fracture orientation.

In addition to rock and fracture characteristics discussed previously, the use of equation 5 to simulate a source of solute requires an estimate for the intial solute concentration in the pores of the oil-shale matrix (C_B). Water-quality sampling as the retort chamber resaturates will provide a measure of solute concentration in fractures but not of solute concentration in the shale-matrix pores. In cases where solute migration outside the chamber does not occur until some time after the inital resaturation the initial solute concentration with the shale-matrix pores may be approximated by the initial solute concentration within fractures. Study by the author and the Department of Energy in situ oil-shale retort near Rock Springs, Wyo. indicate that this approximation may be reasonable in many cases.

In situ retort chambers are sources of solute for extended periods of time after retorting stops. This characteristic has been observed by the author at the U.S. Department of Energy, experimental, in situ, oil-shale retort near Rock Springs, Wyoming and by Hall (1982) in laboratory experiments. The source of solute cannot be explained solely by diffusion from burned shale into fractures.

Hall (1982) has proposed a mechanism to explain the long-term source of solute that is based on the slow dissolution of the mineral matrix from pore walls. Although he was unable to provide a good theoretical basis for this slow dissolution, he proposed the use of an empirical expression to simulate the mechanism. The expression is

$$K^*A(c_e-c) \tag{6}$$

where K^* = the mass transfer coefficient $[L^{-2}T^{-1}]$,

A = the interfacial area $[L^2]$,

c_e = the equilibrium concentration of fluid in contact with shale blocks $[ML^{-3}]$, and

c = the solute concentration $[ML^{-3}]$.

This equation acts as a source-sink term to the solute-transport equation and simulates the dominant mechanism for leaching of burned shale after the passage of two to three pore volumes of water.

Application of the proposed leaching mechanism to a field problem requires that K^*, A, and c_e in equation 6 be determined. While Hall (1982) determined values of the product K^*A for organic solute from column experiments, the applicability of these values to field problems is unknown. Therefore K^*A is effectively a parameter that must be estimated during model calibration. The equilibrium concentration, c_e, can be estimated from long-term leaching tests of burned shale. Unfortunately, these tests have been performed for very few of the chemical species found in oil-shale retort water.

The use of empirical equation 6 to describe mass transfer essentially provides an infinite source of solute to the medium where, in reality, such a source is finite. Therefore the use of equation 6 will over estimate solute concentrations in the trailing part of a plume. The use of equation 6 is justified in studies of solute transport in oil shale where the distribution of solute in the trailing part of the plume is not considered, or in studies where an estimate of maximum probable concentration of solute in the trailing part of a plume is needed.

MODEL CALIBRATION AND SENSITIVITY

The approach to ground-water flow and solute transport used in this report along with the three-dimensional nature of flow through oil shale, introduces an unusually large number of calibration parameters. Considering the amount and distribution of water-level and solute-concentration data that are available in most field studies, it is possible to reach a point where additional detail in the simulation procedure does not significantly improve the model fit. Therefore, it is important to assess the uncertainty associated with the various calibration parameters if the model is to remain a practical one for use in field problems.

Cooley (1977) presents a method for evaluating the reliability of a model within the framework of steady-state flow of ground water in two dimensions. This technique has been extended to three dimensions and is described in a later section. Before presenting the technique, the relationship between measured data in wells and model-calculated hydraulic head is discussed. Observation wells that have been installed in oil-shale strata rarely are piezometers.

Relationship of Measured Data and Calculated Results

Hydraulic Head

The relationship between hydraulic head in the formation and the water level measured in a well open to part or all of the formation is governed by aquifer properties, well-bore characteristics and the vertical-head gradient within the formation. An accurate treatment of the relationship would involve solving the three-dimensional equation of ground-water flow in the close vicinity of the well bore using model estimates of aquifer properties and a boundary condition of uniform head along the well bore. The computed head in the well could be compared to measured water-level data. This approach is not practical in most field problems. Instead, a relationship is used in the model that is based on the steady-state conservation of water within the well bore and Darcy's law. The result is a simple weighted average of hydraulic head.

$$h_{int} = \frac{\int_{z_1}^{z_2} K_{ii} h \, dz}{\int_{z_1}^{z_2} K_{ii} \, dz} \qquad i = 1,2 \qquad (7)$$

where h_{int} = the depth-integrated head in the well [L],

h = hydraulic head obtained from the solution to equation 2 [L],

z_2 = the altitude of the top of the well bore open to the formation [L],

z_1 = the altitude of the bottom of the well bore open to the formation [L], and

K_{ii} = hydraulic conductivity [LT^{-1}].

Equation 7 is derived by writing Darcy's law for radial flow and integrating along the open interval of the well. Applying Darcy's law for radial flow from an circular boundary with no drawdown due to well-bore effects and assuming a well radius of 1 foot gives

$$q = \frac{2\pi K(h_b - h_{int})}{\ln(r)}$$

where q = the Darcian flux per unit length of well bore $[L^2T^{-1}]$,

K = the hydraulic conductivity $[LT^{-1}]$,

h_b = the head at the boundary [L],

h_{int} = the head in the well [L], and

r = the distance to the boundary [L].

The distance r may be thought of as the radius of influence by the well bore. By assuming steady-state flow along the well bore, the amount of water entering the well along part of the open interval must equal the amount leaving along the remainder of the open section. The integral expression for this conservation of mass is

$$\int_{z_1}^{z_2} q \, dz = 0.$$

Substitution of Darcy's law for radial flow into the expression for conservation of mass gives

$$\int_{z_1}^{z_2} \frac{K}{\ln(r)} h_b \, dz = h_{int} \int_{z_1}^{z_2} \frac{K}{\ln(r)} \, dz.$$

Treating distance to the boundary as a constant, ln(r), may be moved outside the integral and equation 7 obtained.

The model in this report uses information on the open intervals of wells, estimates of hydraulic conductivity and calculated hydraulic head to determine the depth-integrated head at each observation well. The result may be compared to measured water-level data as a guide during model calibration when piezometers are not available.

Solute Concentration

The relationship between solute concentration in the formation and that in a well is governed by the three-dimensional equation of solute transport. As with hydraulic head, it is not practical to solve this problem at each well bore. By assuming steady-state conservation of solute and using Darcy's law, one can obtain an expression similar to equation 7.

$$c_{int} = \frac{\int_{z_3}^{z_4} cq_i \, dz}{\int_{z_3}^{z_4} q_i \, dz} \qquad i = 1,2 \qquad (8)$$

where c_{int} = the depth-integrated solute concentration [M/L^3],

c = solute concentration obtained from the solution to equation 4 [M/L^3],

z_3 to z_4 = the part of the well bore where water enters [L], and

q_i = $K_{ii}(h-h_{int})$.

Equation 7 is used in the model to infer sections of the well bore where water enters. The derivation of equation 8 essentially is identical to the derivation of equation 7 and will not be given here.

Estimating Parameters for Models of Steady-State Ground-Water Flow

Direct measurement of all hydrogeologic parameters needed to construct models of ground-water flow and solute transport is rarely possible. As a result, values of unmeasured parameters usually are adjusted until measured and calculated water levels and solute concentrations match in some acceptable manner. Cooley (1977) proposed a regression method for estimating an optimal set of hydrogeologic parameters and assessing parameter reliability within the framework of two-dimensional steady-state flow. An extension of this method to three dimensions is included in the model described in this report.

The following method is used to estimate parameters and assess reliability for a three-dimensional model of steady-state ground-water flow. Throughout the following development, deviations from the two-dimensional development of Cooley (1977) are noted. The method is an iterative technique that minimizes the squared difference between measured and calculated water levels in wells. To set up the iterative technique it is necessary to linearize the finite-element form of the equation of ground-water flow with respect to unknown model parameters, differentiate the linearized equation with respect to these parameters, and, setting each derivative equal to zero, solve the system of equations.

The derivatives of unknown model parameters are obtained in the following manner. A truncated Taylor series expansion of equation 2 is written for $(\Delta t)^{-1}$ equal to zero and θ equal to one. The resulting iterative equation is identical to the one given by Cooley (1977). Solving this Taylor series for hydraulic head gives

$$\bar{h}^{r+1} = \bar{h}^{r} + \bar{\Delta}^{r} + \bar{\bar{S}}^{r}\bar{b}^{r+1} \qquad (9)$$

where $\bar{h}^{r+1}$ = the vector of hydraulic head at the r+1 iteration;

$\bar{h}$ = the vector of hydraulic head at the r iteration,

$\bar{\Delta}^{r} = -(\bar{\bar{K}}^{r})^{-1}\,\bar{f}^{r}$;

$\bar{\bar{K}}^{r}$ = the matrix involving estimates of hydraulic conductivity used in equation 2 at the r iteration;

$\bar{f}^{r}$ = the functional representation of equation 2 at the r iteration,

$\bar{\bar{S}}^{r} = -(\bar{\bar{K}}^{r})^{-1}\left(\frac{\partial \bar{f}^{r}}{\partial \bar{a}^{r}}\,\bar{a}^{r}\right)$;

$\bar{a}^{r}$ = the vector of model-parameter estimates at the r iteration, and

$\bar{b}^{r+1} = \frac{a^{r+1} - \bar{a}^{r}}{\bar{a}^{r}}$.

This equation is identical to the one used by Cooley (1977) to calculate hydraulic heads for each new iteration. The definitions of the variables are given in three dimensions instead of two.

Hydraulic head calculated by equation 9 cannot be compared directly to water-level measurements in wells unless the wells are constructed as piezometers. Because wells usually are open to some interval of aquifer, water levels represent a depth-integrated value of hydraulic head. Therefore, the least squares criterion that must be satisfied differs slightly from the criterion used by Cooley (1977), where vertical-head variations were not considered.

$$\frac{\partial}{\partial \bar{b}^{r+1}} \{\bar{w}(\bar{h}_m - \bar{h}_{int}^{r+1})^2\} = 0 \qquad (10)$$

where $\bar{w}$ = a vector of weights between 0 and 1 describing the reliability of $\bar{h}_m$,

$\bar{h}_m$ = the measured head, and

$\bar{h}_{int}^{r+1}$ = the vector of depth-integrated head at the r+1 interation.

Depth-integrated head is calculated by using the finite-element approximation to equation 7 in conjunction with equation 9. Substitution of the result in equation 10 gives the following:

$$\bar{w}(\bar{N}^r\bar{S}^r)^T \bar{N}^r\bar{S}^r\bar{b}^{r+1} = (\bar{N}^r\bar{S})^T \bar{w}[\bar{h}_m - \bar{N}^r(\bar{h}^r + \bar{\Delta}^r)] \qquad (11)$$

where $\bar{N}^r$ = the matrix of finite-element approximations of equation 7 involving estimated hydraulic conductivity at the r iteration, and,

$( )^t$ = the matrix-transpose operation.

With the exception of the $\bar{N}^r$ term, equation 11 is identical to the least squares equation used by Cooley (1977). Equation 11 can be solved for $\bar{b}^{r+1}$ which in turn can be used to calculate estimates of model parameters. Because the Taylor series expansion used to linearize the equations of ground-water flow is an approximation, the solution of equation 11 may not give global optimum values of $\bar{b}^{r+1}$ unless $\bar{h}^{r+1} - \bar{h}^r$ and $\bar{a}^{r+1} - \bar{a}^r$ are small. Therefore iteration must be used.

The iteration procedure is identical to the one used by Cooley (1977). An initial estimate of aquifer properties and other unknown model parameters is used to solve equation 2 for hydraulic head. The coefficients $\partial \bar{f}^r / \partial \bar{a}^r$ are calculated and equation 11 is used to solve for $\bar{b}^{r+1}$ and new estimates of model parameters. The iteration number is advanced by one and the Taylor series expansion of equation 2 is used to compute $\bar{f}^r$. An iterative cycle is begun by resolving equation 2 with the latest estimates of aquifer properties and boundary flux rates until values for $\bar{f}^{r+1}$ are less than some acceptable error.

Sensitivity Analysis of Transient Ground-Water Flow and Solute Transport

Development of a statistical procedure for estimating model parameters under transient conditions is more difficult than under steady-state conditions. Several approaches to solving this problem may be possible, but all have proven overly expensive in terms of disc storage, core storage or computational time. Therefore, calibration of transient ground-water flow and solute-transport models remains a trial-and-error process in this report.

A review of the differential equation of solute transport (eq. 3) shows that ground-water velocity acts as a calibration parameter. Therefore, it usually is necessary to develop a ground-water flow model concurrently with the development of a solute-transport model. Darcy's law is used to compute velocity from estimates of hydraulic conductivity and model-calculated head. The linkage between ground-water flow and solute transport often forces the hydrologist to iterate between the flow and transport models until a distribution of hydraulic conductivity is obtained that reproduces both historical water-level and solute-concentration data. Bibby (1981) observed that the solute-transport model can have a strong influence in determining optimal estimates of hydraulic conductivity.

EVALUATION OF MODEL

A number of tests of the model have been made including comparisons of model results to analytical solutions and to one field application. Because analytical solutions rarely exist for three-dimensional problems, tests have been designed to verify the accuracy of the model in one and two dimensions. Repeated applications of the test problems in each of the three directions were used to isolate any programming or logic errors. Comparisons of model results with analytical solutions are discussed below. Application of the model to a field problem, solute transport from an in situ oil-shale retort near Rock Springs, Wyo., will be discussed in a later report (K. C. Glover, U.S. Geological Survey, written commun., 1985).

Ground-Water Flow

An evaluation of the basic model of ground-water flow (eq. 2) has not uncovered any unusual characteristics. For problems with relatively regular boundaries, the evaluation showed that the finite-element method did not improve upon head distribution calculated by finite-difference methods. In problems with complex and irregular aquifer geometry, the finite-element method can be used to model the flow system with fewer nodes.

Model results were compared to Theis's analytical solution for drawdown in the vicinity of a pumping well (Lohman, 1979). The Theis solution is for a fully penetrating well in an infinite confined aquifer with no vertical movement of water. These conditions were simulated in the flow model by using variable node spacing, a zero-drawdown boundary 20,000 ft from the well, and uniform aquifer properties. The aquifer was assumed to be 100 ft thick with a hydraulic conductivity of 12.5 ft/d and specific storage of 10^{-6} ft^{-1}. Well discharge, distributed uniformly along the well bore, was simulated at a rate of 2.0 ft^3/sec. Model results and the analytical solution are plotted in figure 2 for three distances from the

---

Figure 2.--(on following page) belongs near here

---

pumping well.

Figure 2.--Comparison of flow-model results with Theis's analytical solution.

The remainder of this section is a discussion of the regression technique for estimating aquifer properties and boundary-flow rates in steady-state systems. Most of the major conclusions of Cooley (1977 and 1979) seem to be appropriate for flow in three as well as two dimensions. Specific points of interest to three-dimensional problems are discussed below.

The number of aquifer properties and flux rates that are treated as regression parameters can have a strong influence on the convergence characteristics of the model. The vertical anisotropy that is common in most three-dimensional systems increases the number of regression parameters and can cause convergence problems more easily than in two-dimensional flow. The number of regression parameters in three dimensions also increases because horizontal hydraulic conductivity often varies with depth. With the large number of parameters that can occur in three-dimensional models it may be difficult to obtain adequate water-level data for all strata and convergence can be very slow. In cases where the number of regression parameters approaches the number of measured water levels, no solution may be possible. These characteristics can limit the use of the regression technique in three-dimensional problems.

Boundary-flux rates usually are measured with greatest accuracy when the flow occurs at or near the land surface. This fact has a number of implications in using the regression technique. If significant recharge and discharge occurs at depth as underflow and the hydraulic conductivity is not known exactly, the least-squares matrix usually is ill-conditioned and no solution is possible. On the other hand, if boundaries are identified at depth such that no flow occurs across them, a solution may be possible. However, the standard errors of estimate for hydraulic conductivity usually will be very large. Large standard errors can be common especially if measured water-level data do not accurately describe vertical head gradients throughout the study area.

Cooley (1977) found that models of ground-water flow are characterized by having large standard errors for the parameters. These errors are caused by anomalous measured water levels as well as errors in specified boundary conditions. If errors that usually occur in boundary-flux estimates are not considered during model applications, the resulting standard errors for the parameters usually will be artificially small. Test problems using the three-dimensional regression procedure supported these conclusions.

The regression model in this respect can be used to assess reliability of computed parameters and predicted values of head if the model is approximately linear with respect to the parameters. Cooley (1979) provides a test for linearity that also is applicable to three dimensions. Because the model is nonlinear with respect to hydraulic conductivity, the large number of hydraulic-conductivity parameters in most three-dimensional systems makes it more difficult to pass this test of linearity. Therefore, the use of confidence regions and test of hypothesis may not be appropriate for many three-dimensional problems. Additional testing of the regression method in three dimensions is needed to evaluate this conclusion.

Solute Transport

An analytical solution for one-dimensional solute transport from a specified concentration boundary into a fractured aquifer of uniform thickness is given by Bibby (1981, p. 1078). Fractures are assumed to be horizontal and separated by sufficient impermeable strata to be considered of infinite extent. The solute is considered to be non-reactive. The analytical solution was obtained by analogy to a corresponding solution for conduction and convection of heat into an aquifer.

Model results were compared to this analytical solution using representative values for aquifer characteristics. Seepage velocity of water in the fissures was set equal to 0.5 ft/d. Longitudinal and transverse dispersivities were set to 100 ft/d, and the diffusion coefficient of solute in water in the unfractured formation blocks was 5×10^{-5} ft^2/d. The block porosity was set to unity to permit comparison with the analytical solution thickness of the blocks made sufficiently large so as not to violate assumptions in the analytical solution. Several simulations with various block dimensions were made to insure the assumptions were reproduced.

Model and analytical results after 100 days of solute transport are presented in figure 3. The close comparison verifies the model theory

---

Figure 3.--(on following page) belongs near here

---

and program logic. The model slightly overestimates solute concentrations in the leading part of the profile and slightly underestimates concentrations in the trailing part. This characteristic probably is due to the effects of numerical dispersion associated with nodal spacing.

Figure 3.--Comparison of results from solute-transport model with analytical solution at 100 days.

Additional simulations were made with the dual-porosity equation to evaluate the sensitivity of the model to variations in the dual-porosity coefficients. The coefficients that had the greatest effect on the distribution of solute appeared to be shale-matrix porosity and fracture width. The matrix-diffusion coefficient and fracture density also were important in determining the distribution of solute. These observations are in agreement with those of Grisak and Pickens (1980) and show that in fractured rock, matrix diffusion can be an important mechanism for solute transport. Factors reducing the need to consider matrix diffusion are small matrix porosities and large fracture width and density.

A review of the dual-porosity equation (eq. 5) shows that a large number of coefficients must be known even for relatively simple fracture geometries. Because in most studies these coefficients must be estimated during model calibration, the amount of water-quality data needed is far greater than the amount needed to use a continuum model. Within the framework of a three-dimensional system, sufficient data may be available only for a preliminary calibration of the model. Nevertheless in rock where matrix diffusion is dominant, a scarcity of data is not sole justification for ignoring the more complex model.

COMPUTER PROGRAM

A listing of the FORTRAN program that solves the three-dimensional equations of flow and solute transport in dual-porosity media is given in table 1 (at end of report). Data-input formats are described in table 2 (at end of report). Although data entry into a finite-element program typically is more cumbersome than for finite-difference prgrams, the increased data-entry time usually is compensated by increased flexibility in locating nodes. With a finite-element model, nodes can be accurately located at observation or pumping wells. In general, fewer nodes are needed to accurately model aquifer geometry when using a finite-element model.

Data entry into a finite-element program is more cumbersome because of the need to identify the relationships among all nodes and elements. As a result, all nodes and elements must be numbered, the Cartesian coordinates of all nodes must be coded, and the nodes associated with each element must be designated.

The system used to number aquifer nodes and elements has a significant impact on the efficiency and size of the computer program. The global coefficient matrices developed in equations 2 and 4 represent the largest block of computer storage used by the program. The solution technique is more efficient, in terms of time and storage requirements, if the size of the global coefficient matrices is minimized. Storage requirements of the global coefficient matrices are directly related to the largest difference between two node numbers in an aquifer element. Therefore, efficient nodal ordering minimizes this difference and improves the efficiency of the solution. The program calculates and prints the band width. If the dimensions of arrays are not sufficient, the simulation will stop.

SAMPLE SIMULATION

Use of the model is illustrated by simulating solute transport from a buried source in a three-layer aquifer system. The bottom layer represents an aquifer where horizontal joints are common within relatively thick impermeable rock. The middle layer represents an aquifer where vertical fractures are common. The top layer represents a homogeneous isotropic sandstone. Boundary conditons and nodal locations are shown horizontally in figure 4 and vertically in figure 5. Aquifer properties

---

Figures 4 and 5.--(on following pages) belong near here

---

for each layer are given in table 3. Pumping by two wells, one from the bottom layer and one from the top, also is simulated. Steady-state flow conditions are assumed.

The input data used in the model run and results for one time step are listed in tables 4 and 5 (at end of report). Results present total time into the simulation, and the hydraulic head and solute concentration for each node in the finite-element grid. Principal components of Darcian velocity and dispersion coefficient also are printed for each element.

Figure 4.--Boundary conditions and node locations along the bottom of the aquifer system used in the sample simulation.

Figure 5.--Boundary conditions and node locations along a vertical section of the aquifer system used in the sample simulation.

Table 3.--*Aquifer properties used in sample simulation*

Aquifer property	Bottom layer	Middle layer	Top layer
Hydraulic conductivity			
x-direction (feet/day)	20.0	10.0	40.0
Hydraulic conductivity			
y-direction (feet/day)	20.0	2.0	40.0
Hydraulic conductivity			
z-direction (feet/day)	5.0	10.0	40.0
Longitudinal dispersivity (feet)	10.0	1.0	100.0
Transverse dispersivity (feet)	3.0	.3	30.0
Molecular diffusion in low permeability blocks (square feet per day)	.0001	.0001	.0
Fracture width (feet)	.2	.01	.0
Block width (feet)	1.8	.19	.0
Effective porosity (dimensionless)	.1	.05	.3
Block porosity (dimensionless)	.01	.01	.0
Flux boundary (feet/day)	.4	.2	.4
Specified-head boundary (feet)	--	--	500.0
Solute-concentration boundary (milligrams per liter)	100.0	--	--

SUMMARY

The model described in this report can simulate three-dimensional ground-water flow and solute transport in oil shale and associated hydrogeologic units. The model treats oil shale as a dual-porosity medium by simulating flow and transport within fractures using conventional finite-element methods. Diffusion of solute between fractures and the essentially static water of the shale matrix is simulated by including an analytical solution that acts as a source-sink term to the differential equation of solute transport. While knowledge of fracture orientation and spacing is needed to effectively use the model, it is not necessary to map the locations of individual fractures.

The computer program listed in the report incorporates many of the features of previous dual-porosity models while retaining a practical approach to solving field problems. As a result the report does not extend the theory of solute transport in any appreciable way. The emphasis in the report is on bringing together various aspects of solute-transport theory in a manner that is particularly suited to the unusual ground-water flow and solute-transport characteristics of oil-shale systems.

Methods for quantifying the uncertainty in parameter estimates that occur during model development are given in this report. The quasilinear regression method described by Cooley (1977) for estimating parameters and assessing reliability for two-dimensional models of steady-state ground-water flow has been extended to three dimensions. Because the number of model parameters in three-dimensional simulations generally is larger than in two-dimensional simulations, adequate water-level data may not be available for evaluating parameter reliability. When it is possible to evaluate parameter reliability, standard errors for hydraulic-conductivity estimates of buried strata generally are large.

A large number of aquifer properties must be evaluated when simulating solute transport through dual-porosity media. Because properties such as the matrix-diffusion coefficient, matrix porosity, and fracture width and density often are estimated during model calibration, a large amount of water-quality data is needed. Within the framework of a three-dimensional system, sufficient data may be available only for a preliminary calibration of the model. Nevertheless, matrix diffusion can be an important mechanism for solute transport in fractured rock.

REFERENCES CITED

Bibby, Robert, 1981, Mass transport of solutes in dual-porosity media: Water Resources Research, v. 17, no. 4, p. 1075-1081.

Bradley, W. H., 1964, Geology of Green River Formation and associated Eocene rocks in southwestern Wyoming and adjacent parts of Colorado and Utah: U.S. Geological Survey Professional Paper 496-A, p. 31-34.

Cooley, R. L., 1977, A method of estimating parameters and assessing reliability for models of steady state groundwater flow, 1, Theory and numerical properties: Water Resources Research, v. 13, no. 2, p. 318-324.

—— 1979, A method of estimating parameters and assessing reliability for models of steady state groundwater flow, 2, Application of statistical analysis: Water Resources Research, v. 15, no. 3, p. 603-617.

Freeze, R. A., and Cherry, J. A., 1979, Groundwater: Englewood Cliff, New Jersey, Prentice-Hall, Inc., 604 p.

Grisak, G. E., and Pickens, J. F., 1980, Solute transport through fractured media, 1, the effect of matrix diffusion: Water Resources Research, v. 16, no. 4, p. 719-730.

Gupta, S. K., and Tanji, K. K., 1976, A three-dimensional Galerkin finite element solution of flow though multiaquifers in Sutter Basin, California: Water Resources Research, v. 12, no. 2, p. 155-162.

Hall, W. G., 1982, The kinetics of leaching of organic carbon from in situ spent shale: Lawrence Berkley Laboratory Report LBL-14231, 159 p.

Huyakorn, P. S., Lester, B. H., and Mercer, J. W., 1983, An efficient finite-element technique for modeling transport in fractured porous media, 1, Single species transport: Water Resources Research, v. 19, no. 3, p. 841-854.

Huyakorn, P. S., and Nilkuha, K., 1979, Solution of transient transport equation using an upstream finite element scheme: Applied Mathematical Modelling, v. 3, no. 1, p. 7-17.

INTERCOMP Resource Development and Engineering, Inc., 1976, A model for calculating effects of liquid waste disposal in deep saline aquifers, Part I - development, Part II documentation: U.S. Geological Survey Water-Resources Investigations Report 76-61, 253 p.

Konikow, L. F., and Bredehoeft, J. D., 1978, Computer model of two-dimensional solute transport and dispersion in ground water: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 7, chapter C2, 90 p.

Lohman, S. W., 1979, Ground-water hydraulics: U.S. Geological Survey Professional Paper 708, 70 p.

Noorishad, Jahan, and Mehran, Mohsen, 1982, An upstream finite element method for solution of transient transport equation in fractured porous media: Water Resources Research, v. 18, no. 3, p. 588-596.

Pinder, G. F., and Gray, W. G., 1977, Finite element simulation in surface and subsurface hydrology: New York, Academic Press, 295 p.

Rasmuson, A., Narasimham, T. N., and Neretnieks, I., 1982, Chemical transport in a fissured rock: Verification of a numerical model: Water Resources Research, v. 18, no. 5, p. 1479-1492.

Robson, S. G., and Saulnier, G. J., 1981, Hydrogeochemistry and simulated solute transport, Piceance Basin, northwestern Colorado: U.S. Geological Survey Professional Paper 1196, 65 p.

Snow, D. T., 1969, Anisotropic permeability of fractured media: Water Resources Research, v. 5, no. 6, p. 1273-1289.

Zienkiewicz, O. C., 1971, The finite element method in engineering science: New York, McGraw-Hill, 521 p.

SUPPLEMENTAL INFORMATION

Table 1.--*Computer-program listing*

```
C** A FINITE ELEMENT PROGRAM FOR THE SOLUTION OF                          10
C** THREE-DIMENSIONAL TRANSIENT GROUND-WATER FLOW AND SOLUTE              20
C** TRANSPORT IN FRACTURED MATERIAL                                       30
C** OBSERVATIONS MAY BE VALUES OF HEAD AND CONCENTRATION IN WELLS         40
C** OPEN TO SOME OR ALL OF THE AQUIFER SYSTEM                            50
C** ELEMENTS ARE ISOPARAMETRIC CUBES WITH LINEAR SIDES.                  60
C** UPSTREAM FINITE ELEMENTS USED FOR SOLUTE TRANSPORT.                  70
C** GUIDELINES FOR ARRAY DIMENSIONS -- IF NOT SOLVING TRANSPORT PROBLEM  80
C** THE DIMENSIONS OF TRANSPORT ARRAYS MAY BE REDUCED TO 1               90
C**    LET NVARH=NPARH+NQPAR+NBPAR                                      100
C**    SET IDIM.GE.NUMNP, JDIMH.GE.IBH, JDIMC.GE.IBC, KDIM.GE.NUMEL,    110
C**        NVEH.GE.NVARH                                                120
C** CURRENTLY DIMENSIONS ARE FOR A MAXIMUM OF                           130
C** NUMNP= 490 NUMEL=300 NTIME=52 IBH= 90 IBC=180 NTPER=52 NTO=1        140
C** NTRE= 0 NTW= 0 NTS=1 NOBMAX=30 NBPAR= 0 NTB=0                       150
C** NPARH= 8 NQPAR= 0 NUMAT=7 NTFRAC=9 ITFRAC= 10 NLAYER= 3             160
C** MAIN ARRAYS --                                                      170
C** XTXH(NVARH,NVARH),BH(4*NVARH)                                       180
C** PH(NVARH),RKH(NVARH),HINT(NTO*NOBMAX)                              190
C** ZSPACE(NLAYER-1),CINT(NTO*NOBMAX),TITLE(20),HP1(NUMNP*2),          200
C** CP1(NUMNP*2),CO(NTO*NOBMAX),                                       210
C** NXTO(NTO+1),NXTW(NTW+1),NXTS(NTS+1),NXTB(NTB+1)                    220
      DIMENSION XTXH(8,8),BH(32),                                      230
     $          PH(8),RKH(8),HINT(30),CINT(30),                       240
     $          ZSPACE(3),TITL(20),                                   250
     $          CO(30),NXTO(1),                                       260
     $          NXTW(1),NXTS(2),NXTB(1)                               270
C** COFBLK ARRAYS --                                                   280
C** H(NUMNP*2),C(NUMNP*2),QHTMP(NUMNP),WELL(NUMNP*(NTW+1)),           290
C** CFRAC(NTFRAC+1,NUMNP),CINIT(NUMNP),                               300
C** WELLC(NUMNP*(NTW+1)),SFOBSH(NUMNP+NOBMAX),SFOBSC(NUMNP+NOBMAX),   310
C** WTNOD(NUMNP),DM(NUMAT),FP(NUMAT),FQ(NUMAT),                       320
C** RPBR(NUMAT),CBINIT(NUMAT),RX(NUMAT),RY(NUMAT),RZ(NUMAT),          330
C** HO(NTO*NOBMAX),RC(NUMAT),RPR(NUMAT),DL(NUMAT),DT(NUMAT),          340
C** IPRMH(3,NUMAT),MAT(NUMEL),NBP(NUMNP),RMT(NUMAT)                   350
      COMMON/COFBLK/CFRAC(10, 490),QHTMP( 490),CINIT( 490),            360
     $      H( 980),C( 980),WELL( 490),WELLC( 490),                    370
     $      RX(7),RY(7),RZ(7),RC(7),                                   380
     $      RPR(7),DM(7),FP(7),FQ(7),RMT(7),HO(30),                    390
     $      RPBR(7),CBINIT(7),SFOBSH( 520),SFOBSC( 520),WTNOD( 490),   400
     $      DL(7),DT(7),THETAH,DTIME,TIMFAC,TIMBGN,CEQUI,DFW,          410
     $      TIMSUM,IPRMH(3,7),MAT(300),NBP( 490),IELMR                 420
C** SINBLK ARRAYS --                                                   430
C** XORD(NUMNP),YORD(NUMNP),ZORD(NUMNP),QS(NUMNP*(NTS+1)),CS(NUMNP*   440
C** (NTS+1)),QF(NQPAR),CF(NQPAR),QEH(8),QEC(8),NP(NUMEL,8),IBPRM(NUMEL) 450
      COMMON/SINBLK/QEH(8),QEC(8),XORD( 490),YORD( 490),ZORD( 490),    460
     $      QS( 980),CS( 980),QF(1),CF(1),NP(300,8),IBPRM(300)         470
      DOUBLE PRECISION QEH,QEC                                         480
C** SENBLK ARRAY --                                                    490
C** WH(NTO*NOBMAX),WC(NTO*NOBMAX),NPBC(NUMNP,NTB+1),NODOBS(NUMNP)     500
      COMMON/SENBLK/WH(30),                                            510
     $      WC(30),THETAC,NPBC(490,1),NODOBS(490)                      520
C** LDUBLK --                                                          530
```

Table 1.--*Computer-program listing*--Continued

```
C** CPKH(IDIM,IBH),SH(NVARH,NUMNP),QH(NUMNP)                          540
C** CPKC(IDIM,IBC),QC(IBC)                                            550
      COMMON/LDUBLK/CPKC(490,180),CPKH(490, 90),SH(8, 490),           560
     $             QC( 490),QH( 490),IBC,IBH                          570
      REAL*8 CPKC,CPKH,SH,QC,QH                                       580
C** SHABLK,SURBLK AND LINBLK -- ALL DIMENSIONS CONSTANT                590
      COMMON/SHABLK/SF(4,8,8),WF(4,8),WT(8),NUMQPT                    600
      COMMON/SURBLK/AF(3,4),AWT,NQPTA                                 610
      COMMON/LINBLK/RLF(2,2),RLWT,LINOD(4,2)                          620
      DOUBLE PRECISION SF,WT,WF,AF,AWT                                630
C** SET # RECORDS IN FILE 8 TO (NTIME/NTPER+1), # WORDS TO NUMNP       640
C** SET # RECORDS IN FILE 9 TO (NTIME+1), # WORDS TO NUMNP             650
      OPEN (UNIT=8,FILE='HEADS',ACCESS='DIRECT',FORM='UNFORMATTED',    660
     $      RECL=816)                                                 670
      OPEN (UNIT=9,FILE='CONCN',ACCESS='DIRECT',FORM='UNFORMATTED',    680
     $      RECL=816)                                                 690
      OPEN (UNIT=5,FILE='SAMPLE.INPUT',STATUS='OLD')                  700
      OPEN (UNIT=6,FILE='SOLUTION.PRT')                               710
      IDIM=490                                                        720
      JDIMH= 90                                                       730
      KDIM=300                                                        740
      NVEH=8                                                          750
      LDIMH=NVEH*4                                                    760
      JDIMC=180                                                       770
      DO 105 I=1,3                                                    780
      READ(5,1)(TITL(J),J=1,20)                                       790
      WRITE (6,2) (TITL(J),J=1,20)                                    800
105   CONTINUE                                                        810
      WRITE(6,3)                                                      820
      READ (5,4) NUMEL,NUMNP,NUMAT,NTIME,NTPER,NTB,NTO,NOBMAX,NTW,    830
     $           NTS,NPARH,NQPAR,NBPAR                                840
      WRITE(6,5) NUMEL,NUMNP,NUMAT,NTIME,NTPER,NTB,NTO,NOBMAX,NTW,    850
     $           NTS,NPARH,NQPAR,NBPAR                                860
      READ (5,4) NPLAYR,NEPLAY                                        870
      WRITE (6,68) NPLAYR,NEPLAY                                      880
      READ (5,4) ITMAX,IVELPR,IFRAC,NTFRAC,ITFRAC,NLAYER,IBEALE       890
      WRITE (6,6)ITMAX,IVELPR,IFRAC,NTFRAC,ITFRAC,NLAYER,IBEALE       900
      WRITE(6,7)                                                      910
      READ (5,8) NTPRT,IPO,IPRX,AP,AMP,RP,RPF,EVH                     920
      WRITE (6,8) NTPRT,IPO,IPRX,AP,AMP,RP,RPF,EVH                    930
      READ (5,9) DTIME,TIMFAC,THETAH,THETAC,DFW,CEQUI                 940
      WRITE (6,10) DTIME,TIMFAC,THETAH,THETAC,DFW,CEQUI               950
      READ (5,9) ALFLG,ALFAX,ALFAY,ALFAZ                              960
      WRITE(6,11) ALFLG,ALFAX,ALFAY,ALFAZ                             970
C** INITIALIZE                                                        980
      TIMBGN=DTIME                                                    990
      DO 110 I=1,NUMNP                                               1000
      H(I)=0.0                                                       1010
      NBP(I)=0                                                       1020
      NPBC(I,1)=0                                                    1030
      NODOBS(I)=0                                                    1040
110   CONTINUE                                                       1050
      IF (THETAC.LT.0.0) GO TO 130                                   1060
```

Table 1.--*Computer-program listing*--Continued

```
      DO 120 I=1,NUMNP                                                  1070
      C(I)=0.0                                                          1080
      DO 120 J=1,NTFRAC                                                 1090
      CFRAC(J,I)=0.0                                                    1100
120   CONTINUE                                                          1110
130   N=NOBMAX*NTO                                                      1120
      NXTO(1)=999999                                                    1130
      IF (N.LE.0) GO TO 150                                             1140
      DO 140 I=1,N                                                      1150
      HO(I)=0.0                                                         1160
      WH(I)=0.0                                                         1170
      IF (THETAC.LT.0.0) GO TO 140                                      1180
      CO(I)=0.0                                                         1190
      WC(I)=0.0                                                         1200
140   CONTINUE                                                          1210
150   N=NUMNP*(NTW+1)                                                   1220
      NXTW(1)=999999                                                    1230
      DO 156 I=1,N                                                      1240
      WELL(I)=0.0                                                       1250
      IF (THETAC.GE.0.0) WELLC(I)=0.0                                   1260
156   CONTINUE                                                          1270
      N=NUMNP*(NTS+1)                                                   1280
      NXTS(1)=999999                                                    1290
      DO 158 I=1,N                                                      1300
      QS(I)=0.0                                                         1310
      IF (THETAC.GE.0.0) CS(I)=0.0                                      1320
158   CONTINUE                                                          1330
      DO 160 I=1,NUMAT                                                  1340
      IPRMH(1,I)=0                                                      1350
      IPRMH(2,I)=0                                                      1360
      IPRMH(3,I)=0                                                      1370
160   CONTINUE                                                          1380
      NUMM=NUMNP-NLAYER+1                                               1390
      NLAYM2=NLAYER-2                                                   1400
      NLAYM1=NLAYER-1                                                   1410
      READ (5,9) (ZSPACE(I),I=1,NLAYM1)                                 1420
      WRITE(6,9) (ZSPACE(I),I=1,NLAYM1)                                 1430
      WRITE (6,69)                                                      1440
      DO 162 I1=1,NPLAYR                                                1450
      READ (5,12)I,N,XORD(I),YORD(I),                                   1460
     $                ZORD(I)                                          1470
      NM1=N-1                                                           1480
      DO 162 J=1,NM1                                                    1490
      XORD(I+J)=XORD(I)                                                 1500
      YORD(I+J)=YORD(I)                                                 1510
      ZORD(I+J)=ZORD(I)+ZSPACE(J)                                       1520
162   CONTINUE                                                          1530
      READ (5,13) N                                                     1540
      READ (5,14) (I,NPBC(I,1),NODOBS(I),WTNOD(I),H(I),WELL(I),QS(I),   1550
     $             I1=1,N)                                              1560
      WRITE(6,15)(I,NPBC(I,1),NODOBS(I),WTNOD(I),XORD(I),YORD(I),       1570
     $           ZORD(I),H(I),WELL(I),QS(I),I=1,NUMNP)                  1580
      IF (THETAC.LT.0.0) GO TO 164                                      1590
```

Table 1.--*Computer-program listing*--Continued

```
      WRITE (6,16)                                                    1600
      READ (5,17) N                                                   1610
      READ (5,17)(I,C(I),WELLC(I),CS(I),I1=1,N)                       1620
      WRITE(6,67)(I,C(I),WELLC(I),CS(I),I=1,NUMNP)                    1630
      DO 163 I=1,NUMNP                                                1640
      CINIT(I)=C(I)                                                   1650
163   CONTINUE                                                        1660
164   IBH=0                                                           1670
      WRITE(6,18)                                                     1680
      DO 170 I1=1,NEPLAY                                              1690
      READ (5,19) I,N,(NP(I,J),J=1,4)                                 1700
      NM1=N-1                                                         1710
      MAT(I)=1                                                        1720
      DO 166 J=5,8                                                    1730
      NP(I,J)=NP(I,J-4)+1                                             1740
166   CONTINUE                                                        1750
      IF (NLAYER.LE.2) GO TO 170                                      1760
      DO 168 K=1,NM1                                                  1770
      DO 168 J=1,8                                                    1780
      MAT(I+K)=K+1                                                    1790
      NP(I+K,J)=NP(I,J)+K                                             1800
168   CONTINUE                                                        1810
170   CONTINUE                                                        1820
      DO 174 I=1,NUMEL                                                1830
      WRITE (6,20)I,(NP(I,J),J=1,8)                                   1840
      DO 172 J=1,8                                                    1850
      DO 172 K=J,8                                                    1860
      J1=IABS(NP(I,J)-NP(I,K))                                        1870
      IF (J1.GT.IBH) IBH=J1                                           1880
172   CONTINUE                                                        1890
174   CONTINUE                                                        1900
      IBH=IBH+1                                                       1910
      IBC=IBH*2-1                                                     1920
      WRITE (6,72) IBH,IBC                                            1930
      IF (NUMNP.GT.IDIM) STOP                                         1940
      IF (IBH.GT.JDIMH) STOP                                          1950
      IF (IBC.GT.JDIMC.AND.THETAC.GE.0.0) STOP                        1960
      WRITE (6,23)                                                    1970
      DO 188 I=1,NUMEL                                                1980
      IBPRM(I)=0                                                      1990
188   CONTINUE                                                        2000
      READ (5,4) N                                                    2010
      IF (N.GT.0) READ (5,24)(I,MAT(I),IBPRM(I),I1=1,N)               2020
      WRITE (6,13)(I,MAT(I),IBPRM(I),I=1,NUMEL)                       2030
      WRITE (6,25)                                                    2040
      READ (5,26) (I,RX(I),RY(I),RZ(I),RC(I),I1=1,NUMAT)              2050
      WRITE (6,26) (I,RX(I),RY(I),RZ(I),RC(I),I=1,NUMAT)              2060
      ISTEDY=1                                                        2070
      DO 190 I=1,NUMAT                                                2080
      IF (RC(I).NE.0.0) ISTEDY=0                                      2090
190   CONTINUE                                                        2100
      IF (THETAC.LT.0.0) GO TO 196                                    2110
      WRITE (6,28)                                                    2120
```

Table 1.--*Computer-program listing*--Continued

```
      READ (5,27) (DL(I),DT(I),DM(I),RPR(I),FP(I),FQ(I),RMT(I),          2130
     $             RPBR(I),I=1,NUMAT)                                    2140
      WRITE (6,29) (I,DL(I),DT(I),DM(I),RPR(I),FP(I),FQ(I),RMT(I)        2150
     $             ,RPBR(I),I=1,NUMAT)                                   2160
      IF (IFRAC.NE.1) GO TO 196                                          2170
      WRITE (6,32)                                                       2180
      READ (5,27) (CBINIT(I),I=1,NUMAT)                                  2190
      WRITE (6,33) (CBINIT(I),I=1,NUMAT)                                 2200
  196 IF (NPARH.LT.1) GO TO 206                                          2210
      DO 198 I=1,NPARH                                                   2220
      RKH(I)=0.0                                                         2230
  198 CONTINUE                                                           2240
      WRITE (6,34)                                                       2250
      READ (5,35) (I,IPRMH(1,I),IPRMH(2,I),IPRMH(3,I),I1=1,NUMAT)        2260
      WRITE (6,35) (I,IPRMH(1,I),IPRMH(2,I),IPRMH(3,I),I=1,NUMAT)        2270
      IXTRFL=0                                                           2280
      DO 200 I=1,NUMAT                                                   2290
      IF (IPRMH(1,I).GT.0) IXTRFL=1                                      2300
      IF (IPRMH(2,I).GT.0) IXTRFL=1                                      2310
      IF (IPRMH(3,I).GT.0) IXTRFL=1                                      2320
  200 CONTINUE                                                           2330
      WRITE (6,36)                                                       2340
      READ (5,37) (I,RKH(I),I1=1,NPARH)                                  2350
      WRITE (6,37) (I,RKH(I),I=1,NPARH)                                  2360
  206 IF (NQPAR.LE.0) GO TO 208                                          2370
      WRITE (6,40)                                                       2380
      READ (5,41) (I,QF(I),RKH(I+NPARH),I1=1,NQPAR)                      2390
      WRITE (6,41) (I,QF(I),RKH(I+NPARH),I=1,NQPAR)                      2400
      IXTRFL=1                                                           2410
      IF (THETAC.LT.0.0) GO TO 208                                       2420
      WRITE (6,42)                                                       2430
      READ (5,37) (I,CF(I),I1=1,NQPAR)                                   2440
      WRITE (6,37) (I,CF(I),I=1,NQPAR)                                   2450
  208 IF (NBPAR.LE.0) GO TO 212                                          2460
      READ (5,43) (NBP(I),I=1,NUMNP)                                     2470
      WRITE(6,44) (NBP(I),I=1,NUMNP)                                     2480
      WRITE (6,45)                                                       2490
      DO 210 I1=1,NBPAR                                                  2500
      READ (5,37) I,RKH(I+NPARH+NQPAR)                                   2510
      WRITE(6,37) I,RKH(I+NPARH+NQPAR)                                   2520
  210 CONTINUE                                                           2530
  212 NTF1=NTFRAC+1                                                      2540
      DO 216 I=1,NUMNP                                                   2550
      H(I+NUMNP)=H(I)                                                    2560
      IF (THETAC.LT.0.0) GO TO 216                                       2570
      C(I+NUMNP)=C(I)                                                    2580
      IF (IFRAC.EQ.0) GO TO 216                                          2590
      DO 214 J=1,NTF1                                                    2600
  214 CFRAC(J,I)=C(I)                                                    2610
  216 CONTINUE                                                           2620
C** READ TRANSIENT DATA                                                  2630
      IF (NTB.LE.0.AND.ITMAX.EQ.0) GO TO 228                             2640
      NXTB(1)=999999                                                     2650
```

Table 1.--*Computer-program listing*--Continued

```
      IBCNT=1                                                          2660
      IHR=1                                                            2670
      ICR=1                                                            2680
      WRITE (8'IHR) (H(J),J=1,NUMNP)                                   2690
      WRITE (9'ICR) (C(J),J=1,NUMNP)                                   2700
      IHR=IHR+1                                                        2710
      ICR=ICR+1                                                        2720
      NTH=NTIME/NTPER                                                  2730
      IF (NTB.GT.0) READ(5,46) I1,I2                                   2740
      DO 226 I=1,NTH                                                   2750
      IF (NTB.LE.0) GO TO 222                                          2760
      IF ((I-1)*NTPER.LT.I1) GO TO 222                                 2770
      IBM1=IBCNT                                                       2780
      IBCNT=IBCNT+1                                                    2790
      NXTB(IBM1)=I1                                                    2800
      DO 218 J=1,NUMNP                                                 2810
      NPBC(J,IBCNT)=NPBC(J,IBM1)                                       2820
218   CONTINUE                                                         2830
      WRITE (6,47) I1                                                  2840
      DO 220 K=1,I2                                                    2850
      READ (5,48) J,NPBC(J,IBCNT),H(J),X1                              2860
      IF (THETAC.GE.0.0) C(J)=X1                                       2870
220   CONTINUE                                                         2880
      I1=999999                                                        2890
      IF (IBM1.LT.NTB) READ (5,46) I1,I2                               2900
222   WRITE (8'IHR) (H(J),J=1,NUMNP)                                   2910
      IHR=IHR+1                                                        2920
      IF (THETAC.LT.0.0) GO TO 226                                     2930
      DO 224 K=1,NTPER                                                 2940
      WRITE (9'ICR) (C(J),J=1,NUMNP)                                   2950
      ICR=ICR+1                                                        2960
224   CONTINUE                                                         2970
226   CONTINUE                                                         2980
      NXTB(NTB+1)=999999                                               2990
228   IF (NTO.LE.0) GO TO 236                                          3000
      N=NTO*NOBMAX                                                     3010
      DO 230 I=1,N                                                     3020
      HINT(I)=0.0                                                      3030
      IF (THETAC.GE.0.0) CINT(I)=0.0                                   3040
230   CONTINUE                                                         3050
      WRITE (6,49)                                                     3060
      DO 234 I=1,NTO                                                   3070
      READ (5, 50) I1                                                  3080
      WRITE (6, 51) I1                                                 3090
      NXTO(I)=I1                                                       3100
      DO 234 J=1,NOBMAX                                                3110
      READ (5, 52) K,X1,X2,X3,X4                                       3120
      WRITE(6, 52) K,X1,X2,X3,X4                                       3130
      L=NOBMAX*(I-1)+K                                                 3140
      HO(L)=X1                                                         3150
      WH(L)=X2                                                         3160
232   IF (THETAC.LT.0.0) GO TO 234                                     3170
      CO(L)=X3                                                         3180
```

Table 1.--*Computer-program listing*--Continued

```
      WC(L)=X4                                                          3190
234   CONTINUE                                                          3200
236   NXTO(NTO+1)=999999                                                3210
      IF (NTW.LE.0) GO TO 242                                           3220
      WRITE (6,53)                                                      3230
      DO 240 I=1,NTW                                                    3240
      DO 238 J=1,NUMNP                                                  3250
      J1=NUMNP*I+J                                                      3260
      WELL(J1)=WELL(J1-NUMNP)                                           3270
      IF (THETAC.GE.0.0) WELLC(J1)=WELLC(J1-NUMNP)                      3280
238   CONTINUE                                                          3290
      READ (5, 50) I1,I2                                                3300
      WRITE (6, 54) I1                                                  3310
      NXTW(I)=I1                                                        3320
      ITMP=MOD(I1,NTPER)                                                3330
      I1=NUMNP*(I)                                                      3340
      IF (ITMP.NE.1) GO TO 248                                          3350
      DO 240 J=1,I2                                                     3360
      READ (5,52) K,X1,X2                                               3370
      WRITE (6,52) K,X1,X2                                              3380
      L=I1+K                                                            3390
      IF (ITMP.EQ.1) WELL(L)=X1                                         3400
      IF (THETAC.GE.0.0) WELLC(L)=X2                                    3410
240   CONTINUE                                                          3420
242   NXTW(NTW+1)=999999                                                3430
      IF (NTS.LE.0) GO TO 250                                           3440
      WRITE (6,55)                                                      3450
      DO 246 I=1,NTS                                                    3460
      DO 244 J=1,NUMNP                                                  3470
      J1=NUMNP*I+J                                                      3480
      QS(J1)=QS(J1-NUMNP)                                               3490
      IF (THETAC.GE.0.0) CS(J1)=CS(J1-NUMNP)                            3500
244   CONTINUE                                                          3510
      READ (5, 50) I1,I2                                                3520
      WRITE (6, 56) I1                                                  3530
      NXTS(I)=I1                                                        3540
      ITMP=MOD(I1,NTPER)                                                3550
      I1=NUMNP*(I)                                                      3560
      IF (ITMP.NE.1) GO TO 248                                          3570
      DO 246 J=1,I2                                                     3580
      READ (5, 52) K,X1,X2                                              3590
      WRITE (6, 52) K,X1,X2                                             3600
      L=I1+K                                                            3610
      QS(L)=X1                                                          3620
      IF (THETAC.GE.0.0) CS(I)=X2                                       3630
246   CONTINUE                                                          3640
      GO TO 250                                                         3650
248   WRITE(6,80)                                                       3660
      STOP                                                              3670
250   NXTS(NTS+1)=999999                                                3680
1     FORMAT (20A4)                                                     3690
2     FORMAT (1X,20A4)                                                  3700
3     FORMAT ('  NUMEL  NUMNP  NUMAT  NTIME',                          3710
```

Table 1.--*Computer-program listing*--Continued

```
     $         '  NTPER    NTB    NTO NOBMAX    NTW',                          3720
     $         '    NTS  NPARH  NQPAR  NBPAR')                                 3730
4     FORMAT (14I5)                                                           3740
5     FORMAT (14I7)                                                           3750
6     FORMAT ('0ITMAX=',I5,' IVELPR=',I5,' IFRAC=',I5,                        3760
     $        ' NTFRAC=',I5,' ITFRAC=',I5,' NLAYER=',I5,' IBEALE=',I5)       3770
7     FORMAT (' NTPRT IPO IPRX        AP       AMP        RP       RPF',     3780
     $        '       EVH')                                                   3790
8     FORMAT (3I5,5E10.3)                                                     3800
9     FORMAT (7E10.3)                                                         3810
10    FORMAT ('0DTIME = ',E12.5,' TIMFAC = ',F5.2,' THETAH = ',F5.2,          3820
     &        ' THETAC = ',F5.2/' DFW = ',E12.5,                              3830
     $        ' CEQUI = ',E12.5)                                              3840
11    FORMAT (' ALFLG = ',E12.5,' ALFAX = ',E12.5,' ALFAY = ',E12.5,          3850
     $        ' ALFAZ = ',E12.5)                                              3860
12    FORMAT (2I10,3E10.3)                                                    3870
13    FORMAT (12I10)                                                          3880
14    FORMAT (3I10,4E10.3)                                                    3890
15    FORMAT (3I5,7E10.3)                                                     3900
16    FORMAT ('      NODE          C      WELLC         CS',                 3910
     $        '      NODE          C      WELLC         CS',                 3920
     $        '      NODE          C      WELLC         CS')                 3930
17    FORMAT (I10,3E10.3)                                                     3940
18    FORMAT (' ELEMENT    NP ARRAY')                                         3950
19    FORMAT (8I5)                                                            3960
20    FORMAT (I5,5X,8I5)                                                      3970
23    FORMAT ('   ELEMENT        MAT      IBPRM',                            3980
     $        '   ELEMENT        MAT      IBPRM',                            3990
     $        '   ELEMENT        MAT      IBPRM',                            4000
     $        '   ELEMENT        MAT      IBPRM')                            4010
24    FORMAT (3I10)                                                           4020
25    FORMAT ('      ZONE         RX         RY         RZ         RC')      4030
26    FORMAT (I10,4E10.3)                                                     4040
27    FORMAT (8E10.3)                                                         4050
28    FORMAT ('      ZONE         DL         DT         DM         RP',      4060
     $        '        FP         FQ        RMT       RPBR')                 4070
29    FORMAT (I10,8E10.3)                                                     4080
32    FORMAT ('0CBINIT VALUES')                                               4090
33    FORMAT (1X,8F10.3)                                                      4100
34    FORMAT (' ZONE      IPRMH ARRAY')                                       4110
35    FORMAT (5I5)                                                            4120
36    FORMAT (' PARAMETER RKH')                                               4130
37    FORMAT (I10,E10.3)                                                      4140
40    FORMAT (' PARAMETER         QF        RKH')                           4150
41    FORMAT (I10,2E10.3)                                                     4160
42    FORMAT (' PARAMETER         CF')                                        4170
43    FORMAT (15I5)                                                           4180
44    FORMAT (' NBP ARRAY'/15I5)                                              4190
45    FORMAT (' BOUNDRY PARM    RKH')                                         4200
46    FORMAT (2I5)                                                            4210
47    FORMAT ('0TIME VARYING BOUNDRY CONDITIONS --  KT = ',I5/                4220
     $        ' NODE NPBC',9X,'H',9X,'C')                                     4230
48    FORMAT (2I5,F10.3,F10.3)                                                4240
```

Table 1.--*Computer-program listing*--Continued

```
49    FORMAT ('0OBSERVED HEAD AND CONCENTRATION DATA')                          4250
50    FORMAT (2I10)                                                             4260
51    FORMAT (' TIME STEP ',I5/'       OBS        HO        WH',               4270
     &        '        CO        WC')                                          4280
52    FORMAT (I10,4E10.3)                                                       4290
53    FORMAT ('0TIME VARYING POINT SOURCE-SINK DATA')                           4300
54    FORMAT (' TIME STEP ',I5/'      NODE      WELL     WELLC')                4310
55    FORMAT ('0TIME VARYING LINE SOURCE-SINK DATA')                            4320
56    FORMAT (' TIME STEP',I5/'       NODE        QS        CS')               4330
57    FORMAT (' TIME STEP ',I5,' TOTAL TIME OF SIMULATION',F12.2/               4340
     $                        '        NODE        XORD        YORD',          4350
     $        '        ZORD        H           C')                            4360
58    FORMAT (I10,5E12.5)                                                       4370
59    FORMAT (I10,3E12.5,12X,E12.5)                                             4380
60    FORMAT ('0 OBS           HO         HINT',                                4390
     $             '           CO         CINT')                               4400
61    FORMAT (I5,4(1X,E12.5))                                                   4410
62    FORMAT (I5,26X,2(1X,E12.5))                                               4420
63    FORMAT ('0STATISTICS OF INITIAL HEAD SOLUTION')                           4430
64    FORMAT ('0STATISTICS OF INITIAL CONCENTRATION SOLUTION')                  4440
65    FORMAT ('0NUMBER OF OBSERVATIONS = ',I5/' ESTIMATED SUM OF ',             4450
     $        'SQUARED ERRORS FOR INITIAL SOLUTION = ',E12.5/                   4460
     $        ' ERROR VARIANCE FOR INITIAL SOLUTION = ',E12.5)                  4470
66    FORMAT ('0INITIAL SOLUTION')                                              4480
67    FORMAT (I10,3E10.3,I10,3E10.3,I10,3E10.3)                                 4490
68    FORMAT (' NPLAYR=',I5,' NEPLAY=',I5)                                      4500
69    FORMAT (' NODE NPBC NODOBS   WTNOD      XORD      YORD      ZORD',       4510
     $        '          H      WELL        QS')                              4520
70    FORMAT (' FLOW PROBLEM')                                                  4530
71    FORMAT (' SOLUTE TRANSPORT PROBLEM')                                      4540
72    FORMAT ('0FLOW PROBLEM BAND WIDTH = ',I5/' TRANSPORT PROBLEM ',           4550
     $        'BAND WIDTH = ',I5)                                               4560
73    FORMAT (' UPDATED PARAMETERS'/' MATI          RX',                        4570
     $        '          RY          RZ          ')                           4580
74    FORMAT (I5,3(1X,E11.5))                                                   4590
75    FORMAT (' PARAMETER          QF  ')                                       4600
76    FORMAT (I10,1X,E11.5)                                                     4610
77    FORMAT ('FLOW PARAMETER ',I5,' EFFECTIVELY ZERO')                         4620
78    FORMAT (' SOLUTION FAILED TO CONVERGE IN ',I5,' ITERATIONS')              4630
79    FORMAT (' SOLUTION CONVERGED IN ',I5,' ITERATIONS')                       4640
80    FORMAT (' TIME STEP IS INVALID FOR A CHANGE OF PARAMETERS'/               4650
     $        ' TIME STEP MUST BE DIVISIBLE BY NTPER WITH A REMAINDER',         4660
     $        ' OF 1')                                                          4670
      CALL SHAFAC(THETAC)                                                      4680
      CALL SURFAC                                                              4690
      CALL LINFAC                                                              4700
      NVARH=NPARH+NQPAR+NBPAR                                                  4710
      NVHX2=NVARH+NVARH                                                        4720
      NVHX3=NVARH+NVHX2                                                        4730
      NODES=8                                                                  4740
C** COMPUTE AND COUNT PRIOR INFORMATION                                        4750
      NPRIRH=0                                                                 4760
      IF (NVARH.LT.1) GO TO 254                                                4770
```

Table 1.--*Computer-program listing*--Continued

```
      DO 252 I=1,NVARH                                                  4780
      PH(I)=1.0                                                         4790
      IF (RKH(I).LE.0.0) GO TO 252                                      4800
      RKH(I)=EVH/(RKH(I)*RKH(I))                                        4810
      NPRIRH=NPRIRH+1                                                   4820
252   CONTINUE                                                          4830
C** COMPUTE INITIAL SOLUTION                                            4840
254   WRITE (6,66)                                                      4850
      IFLOW=0                                                           4860
      KTH=0                                                             4870
      IOBCNT=1                                                          4880
      IWCNT=1                                                           4890
      ISCNT=1                                                           4900
      IBCNT=1                                                           4910
      SKBIG=0.0                                                         4920
      SKBIG2=0.0                                                        4930
      DTIME=TIMBGN/TIMFAC                                               4940
      TIMSUM=0.0                                                        4950
      OBSH=0.0                                                          4960
      YSQH=0.0                                                          4970
      ERVARH=0.0                                                        4980
      OBSC=0.0                                                          4990
      YSQC=0.0                                                          5000
      ERVARC=0.0                                                        5010
      IOBCNT=1                                                          5020
      DO 294 KT=1,NTIME                                                 5030
      DTIME=DTIME*TIMFAC                                                5040
      TIMSUM=TIMSUM+DTIME                                               5050
      LU=0                                                              5060
      IFLOW=IFLOW+1                                                     5070
      IF (IFLOW.GT.NTPER) IFLOW=1                                       5080
      IF (IFLOW.EQ.1) KTH=KTH+1                                         5090
      IF (KT.EQ.1) LU=1                                                 5100
      IF (KT.GT.NXTO(IOBCNT)) IOBCNT=IOBCNT+1                           5110
      IF (KT.EQ.NXTB(IBCNT)) LU=1                                       5120
      IF (TIMFAC.GT.1.01) LU=1                                          5130
      LUH=LU                                                            5140
      LUC=LU                                                            5150
      IF (KT.EQ.NXTW(IWCNT)) IWCNT=IWCNT+1                              5160
      IF (KT.EQ.NXTS(ISCNT)) ISCNT=ISCNT+1                              5170
      IF (KT.EQ.NXTB(IBCNT)) IBCNT=IBCNT+1                              5180
      IF (IFLOW.GT.1) GO TO 260                                         5190
      IF (NTB.LE.0) GO TO 258                                           5200
      IHR=KTH+1                                                         5210
      DO 256 I=1,NUMNP                                                  5220
      J=I+NUMNP                                                         5230
      H(I)=H(J)                                                         5240
256   CONTINUE                                                          5250
      J1=NUMNP+1                                                        5260
      J2=NUMNP*2                                                        5270
      READ (8'IHR) (H(J),J=J1,J2)                                       5280
258   CALL FLOW (IWCNT,ISCNT,IBCNT,KT,KTH,                              5290
     $           NTPER,NUMNP,NVARH,NPARH,LUH,NUMEL,                     5300
```

Table 1.--*Computer-program listing*--Continued

```
     $           ISTEDY,0,IDIM)                                                    5310
      CALL UDU (NUMNP,LUH)                                                       5320
260   ICR=KT+1                                                                   5330
      DO 262 I=1,NUMNP                                                           5340
      J=I+NUMNP                                                                  5350
      IF (IFLOW.EQ.1) H(J)=QH(I)                                                 5360
      IF (THETAC.LT.0.0) GO TO 262                                               5370
      C(I)=C(J)                                                                  5380
262   CONTINUE                                                                   5390
      IF (THETAC.LT.0.0) GO TO 270                                               5400
      IF (NTB.LE.0) GO TO 264                                                    5410
      J1=NUMNP+1                                                                 5420
      J2=NUMNP*2                                                                 5430
      READ(9'ICR) (C(J),J=J1,J2)                                                 5440
264   IVPTMP=0                                                                   5450
      IF (MOD(KT,NTPRT).EQ.0) IVPTMP=1                                           5460
      CALL SALT (IFLOW,IWCNT,ISCNT,IBCNT,KT,KTH,                                 5470
     $           NTPER,NUMNP,LUC,NUMEL,ISTEDY,IVPTMP,IFRAC,1,0,                  5480
     $           IDIM,ALFAX,ALFAY,ALFAZ,ALFLG,NTFRAC,ITFRAC,                     5490
     $           NOBMAX,IOBCNT)                                                  5500
      CALL LDU (NUMNP,LUC)                                                       5510
      DO 268 I=1,NUMNP                                                           5520
      I1=NUMNP+I                                                                 5530
      C(I1)=QC(I)                                                                5540
      IF (IFRAC.EQ.0) GO TO 268                                                  5550
      DO 266 J=2,NTF1                                                            5560
      JM1=J-1                                                                    5570
      CFRAC(JM1,I)=CFRAC(J,I)                                                    5580
266   CONTINUE                                                                   5590
      CFRAC(NTF1,I)=QC(I)                                                        5600
268   CONTINUE                                                                   5610
270   IHR=KTH+1                                                                  5620
      ICR=KT+1                                                                   5630
      J1=NUMNP+1                                                                 5640
      J2=NUMNP*2                                                                 5650
      IF (IFLOW.GT.1.OR.NVARH.EQ.0) GO TO 272                                    5660
      WRITE (8'IHR) (H(J),J=J1,J2)                                               5670
272   IF (THETAC.LT.0.0) GO TO 274                                               5680
      WRITE (9'ICR) (C(J),J=J1,J2)                                               5690
274   IF (MOD(KT,NTPRT).NE.0) GO TO 280                                          5700
      WRITE (6,57) KT,TIMSUM                                                     5710
      DO 278 I=1,NUMNP                                                           5720
      I1=NUMNP+I                                                                 5730
      I2=NUMNP+I                                                                 5740
      IF (THETAC.GE.0.0) GO TO 276                                               5750
      IF (IFLOW.EQ.1) WRITE (6,58) I,XORD(I),YORD(I),ZORD(I),H(I2)               5760
      GO TO 278                                                                  5770
276   IF (IFLOW.EQ.1) WRITE (6,58) I,XORD(I),YORD(I),ZORD(I),H(I2),C(I1)         5780
      IF (IFLOW.NE.1) WRITE (6,59) I,XORD(I),YORD(I),ZORD(I),C(I1)               5790
278   CONTINUE                                                                   5800
280   IF (NOBMAX.LE.0) GO TO 294                                                 5810
      IF (KT.NE.NXTO(IOBCNT)) GO TO 294                                          5820
C** COMPUTE INITIAL ERROR VARIANCE                                               5830
```

Table 1.--*Computer-program listing*--Continued

```
      DO 286 I=1,NUMNP                                                    5840
282   IF (NODOBS(I).LE.0) GO TO 286                                       5850
      I3=NUMNP+I                                                          5860
      I2=NODOBS(I)+NOBMAX*(IOBCNT-1)                                      5870
      IF (IFLOW.GT.1) GO TO 284                                           5880
      HINT(I2)=HINT(I2)+SFOBSH(I)*H(I3)                                   5890
284   IF (THETAC.LT.0.0) GO TO 286                                        5900
      CINT(I2)=CINT(I2)+SFOBSC(I)*C(I3)                                   5910
286   CONTINUE                                                            5920
      WRITE (6,60)                                                        5930
      DO 290 I=1,NOBMAX                                                   5940
      I2=I+NOBMAX*(IOBCNT-1)                                              5950
      IF (IFLOW.EQ.1.AND.THETAC.GE.0.0) WRITE (6,61) I,HO(I2),HINT(I2)    5960
     $                                  ,CO(I2),CINT(I2)                   5970
      IF (IFLOW.EQ.1.AND.THETAC.LT.0.0) WRITE (6,61) I,HO(I2),HINT(I2)    5980
      IF (IFLOW.GT.1.AND.THETAC.GE.0.0) WRITE (6,62) I,CO(I2),CINT(I2)    5990
      IF (IFLOW.GT.1) GO TO 288                                           6000
      IF (WH(I2).LE.0.0) GO TO 288                                        6010
      OBSH=OBSH+1.0                                                       6020
      YSQH=YSQH+WH(I2)*(HO(I2)-HINT(I2))**2                               6030
288   IF (THETAC.LT.0.0) GO TO 290                                        6040
      IF (WC(I2).LE.0.0) GO TO 290                                        6050
      OBSC=OBSC+1.0                                                       6060
      YSQC=YSQC+WC(I2)*(CO(I2)-CINT(I2))**2                               6070
290   CONTINUE                                                            6080
      IOBCNT=IOBCNT+1                                                     6090
292   CONTINUE                                                            6100
294   CONTINUE                                                            6110
      IF (NOBMAX.LE.0) GO TO 640                                          6120
      ERVARH=YSQH/(OBSH-NVARH+NPRIRH)                                     6130
      NTMP=OBSH                                                           6140
      WRITE (6,63)                                                        6150
      WRITE (6,65) NTMP,YSQH,ERVARH                                       6160
296   IF (THETAC.LT.0.0) GO TO 298                                        6170
      ERVARC=YSQC/(OBSC-NVARC+NPRIRC)                                     6180
      NTMP=OBSC                                                           6190
      WRITE (6,64)                                                        6200
      WRITE (6,65) NTMP,YSQC,ERVARC                                       6210
C** BEGIN ITERATIONS                                                      6220
298   INDT=0                                                              6230
      ER=0.01                                                             6240
      ERP=1000.0                                                          6250
      IF (ITMAX.LE.0) GO TO 640                                           6260
      IF (ISTEDY.EQ.0) GO TO 640                                          6270
      DO 372 ITER=1,ITMAX                                                 6280
      IOBCNT=1                                                            6290
      IWCNT=1                                                             6300
      ISCNT=1                                                             6310
      IBCNT=1                                                             6320
      IFLOW=1                                                             6330
      KTH=1                                                               6340
      DTIME=TIMBGN/TIMFAC                                                 6350
      TIMSUM=0.0                                                          6360
```

Table 1.--*Computer-program listing*--Continued

```
      IF (AMP.LT.-.5) GO TO 304                                          6370
      YSQH=0.0                                                           6380
      DO 302 I=1,NVARH                                                   6390
      BH(I)=0.0                                                          6400
      DO 302 J=1,NVARH                                                   6410
      XTXH(I,J)=0.0                                                      6420
302   CONTINUE                                                           6430
304   IHR=1                                                              6440
      J1=NUMNP+1                                                         6450
      J2=NUMNP*2                                                         6460
      READ (8'IHR) (H(J),J=J1,J2)                                        6470
      DTIME=DTIME*TIMFAC                                                 6480
      TIMSUM=TIMSUM+DTIME                                                6490
      LU=0                                                               6500
      DO 316 I=1,NUMNP                                                   6510
      J=I+NUMNP                                                          6520
      H(I)=H(J)                                                          6530
316   CONTINUE                                                           6540
      J1=NUMNP+1                                                         6550
      J2=NUMNP*2                                                         6560
      IHR=KTH+1                                                          6570
      READ (8'IHR)(H(J),J=J1,J2)                                         6580
      CALL FLOW (IWCNT,ISCNT,IBCNT,KT,KTH,                               6590
     $           NTPER,NUMNP,NVARH,NPARH,LU,NUMEL,                       6600
     $           ISTEDY,ITER,IDIM)                                       6610
      CALL SENS (NVARH,NUMNP,IPO,KT,KTH,IOBCNT,                          6620
     $           IXTRFL,NOBMAX,IBCNT)                                    6630
      DO 322 I=1,NUMNP                                                   6640
      QHTMP(I)=QH(I)                                                     6650
322   CONTINUE                                                           6660
      DO 324 I=1,NUMNP                                                   6670
      IF (NPBC(I,IBCNT).GE.0.OR.NPBC(I,IBCNT).EQ.-2.OR.NPBC(I,IBCNT)     6680
     $    .EQ.-4) GO TO 324                                              6690
      IF (NBP(I).LE.0) GO TO 324                                         6700
      N=NBP(I)+NPARH+NQPAR                                               6710
      J=NUMNP+I                                                          6720
      SH(N,I)=H(J)*THETAH+(1.0-THETAH)*H(I)                              6730
324   CONTINUE                                                           6740
330   CONTINUE                                                           6750
C** FORM LEAST SQUARES MATRIX                                            6760
      IF (AMP.LT.-.5) GO TO 344                                          6770
      DO 342 I=1,NOBMAX                                                  6780
      I1=I+(IOBCNT-1)*NOBMAX                                             6790
      HTMP=0.0                                                           6800
      DO 332 J=1,NUMNP                                                   6810
      IF (NODOBS(J).NE.I) GO TO 332                                      6820
      HTMP=HTMP+SFOBSH(J)*QHTMP(J)                                       6830
332   CONTINUE                                                           6840
      IF (WH(I1).LE.0.0) GO TO 342                                       6850
      TEMP=HO(I1)-HTMP                                                   6860
      DO 340 K=1,NVARH                                                   6870
      STMP=0.0                                                           6880
      DO 334 J=1,NUMNP                                                   6890
```

Table 1.--*Computer-program listing*--Continued

```
      IF (NODOBS(J).EQ.I) STMP=STMP+SFOBSH(J)*SH(K,J)                    6900
334   CONTINUE                                                           6910
      TMP=WH(I1)*STMP                                                    6920
      DO 338 L=1,NVARH                                                   6930
      STMP=0.0                                                           6940
      DO 336 J=1,NUMNP                                                   6950
      IF (NODOBS(J).EQ.I) STMP=STMP+SFOBSH(J)*SH(L,J)                    6960
336   CONTINUE                                                           6970
338   XTXH(L,K)=XTXH(L,K)+TMP*STMP                                       6980
340   BH(K+NVHX2)=BH(K+NVHX2)+TMP*TEMP                                   6990
      YSQH=YSQH+TEMP*TEMP*WH(I1)                                         7000
342   CONTINUE                                                           7010
344   CONTINUE                                                           7020
      J1=NUMNP+1                                                         7030
      J2=NUMNP*2                                                         7040
      IHR=KTH+1                                                          7050
      IF (IFLOW.EQ.1) WRITE (8'IHR) (H(J),J=J1,J2)                       7060
      WRITE (6,70)                                                       7070
      CALL LSTSQ (XTXH,BH,RKH,PH,YSQH,AP,AMP,RP,RPF,NVARH,NVHX2,NVHX3,   7080
     $            IPO,INDT,ITER,NVEH,LDIMH)                             7090
      IF (INDT.EQ.1) GO TO 521                                           7100
C** UPDATE HEAD                                                          7110
348   DO 352 I=1,NUMNP                                                  7120
      I1=NUMNP+I                                                         7130
      SUMH=0.0                                                           7140
      IF (NVARH.LE.0) GO TO 352                                          7150
      DO 350 J=1,NVARH                                                   7160
      SUMH=SUMH+BH(J)*SH(J,I)                                            7170
350   CONTINUE                                                           7180
      IF (IFLOW.EQ.1) H(I1)=QHTMP(I)+SUMH                                7190
352   CONTINUE                                                           7200
      IHR=2                                                              7210
      J1=NUMNP+1                                                         7220
      J2=2*NUMNP                                                         7230
      WRITE (8'IHR) (H(J),J=J1,J2)                                       7240
C** UPDATE PARAMETERS                                                    7250
354   IF (IPO.EQ.1) WRITE (6,73)                                         7260
      DO 356 I=1,NUMAT                                                   7270
      IF (NVARH.LE.0) GO TO 356                                          7280
      L=IPRMH(1,I)                                                       7290
      IF (L.GT.0) RX(I)=RX(I)*(BH(L)+1.0)                                7300
      L=IPRMH(2,I)                                                       7310
      IF (L.GT.0) RY(I)=RY(I)*(BH(L)+1.0)                                7320
      L=IPRMH(3,I)                                                       7330
      IF (L.GT.0) RZ(I)=RZ(I)*(BH(L)+1.0)                                7340
      IF (IPO.EQ.1) WRITE (6,74) I,RX(I),RY(I),RZ(I)                     7350
356   CONTINUE                                                           7360
C** UPDATE SURFACE AND POINT SOURCE-SINK PARAMETERS                       7370
      IF (NQPAR.LE.0) GO TO 360                                          7380
      WRITE (6,75)                                                       7390
      DO 358 I=1,NQPAR                                                   7400
      QF(I)=QF(I)*(BH(I+NPARH)+1.0)                                      7410
      IF (IPO.EQ.1) WRITE (6,76) I,QF(I)                                 7420
```