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

By Kent C. Glover

U.S. GEOLOGICAL SURVEY

Water-Resources Investigations Report 86-4047

Cheyenne, Wyoming

1987

DEPARTMENT OF THE INTERIOR

DONALD PAUL HODEL, Secretary

U.S. GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information
write to:

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

Copies of this report can be
purchased from:

U.S. Geological Survey
Books and Open-File Reports Section
Federal Center
Box 25425
Denver, Colorado 80225

CONTENTS

FIGURES

TABLES

CONVERSION FACTORS

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 geologic, 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 by Glover (1986). 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.

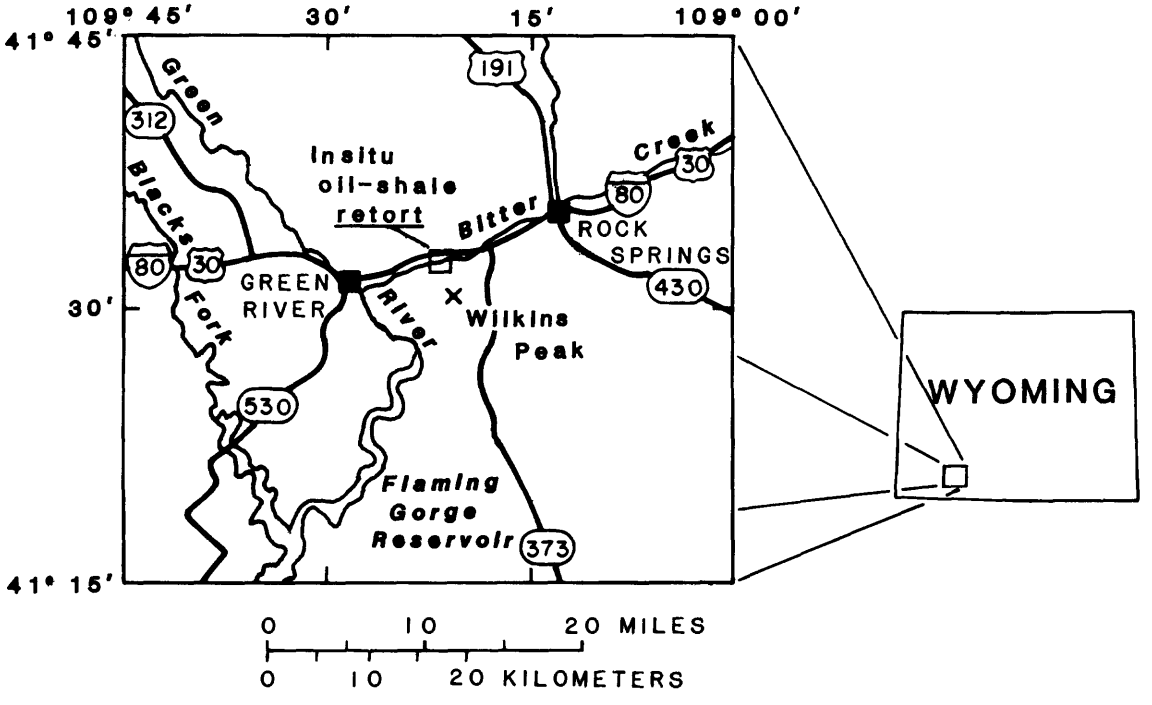

Figure 1.--Location of in situ oil-shale retort near Rock Springs, Wyoming.

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 nonreacting 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, 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:

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

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;

∂t_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 decomposition, 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 tradeoffs 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:

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

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

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

ϕ = porosity [dimensionless];

$q_i = -K_i \frac{\partial h}{\partial xi}$ = 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

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

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.

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

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 righthand 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 properties include molecular diffusion within the oil-shale matrix (D_d), average distance between parallel fractures (b), and average fracture width (f). Because direct estimates of these properties by field tests usually are unreliable, model calibration 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 byproducts are created. A discussion of how these byproducts 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 byproducts 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, as Hall (1982) proposed, 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 initial 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 within the shale-matrix pores may be approximated by the initial solute concentration within the fractures. Study by the author and the Department of Energy of an 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, Wyo., and by Hall (1982) in labortory 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

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

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

A = the interface 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 2 to 3 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 overestimate 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 relation 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.

Relation of Measured Data and Calculated Results

Hydraulic Head

The relation 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 relation 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 relation 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.

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

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 a 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 relation 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;

$\bar{h}^{r}$ = 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 representative of equation 2 at the r iteration,

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

$\bar{a}^{r}$ = 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:

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

where $\bar{\bar{N}}^r$ = 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 $\bar{\bar{N}}^r$ 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 disk 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., is discussed in Glover (1986).

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' 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 pumping well.

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.

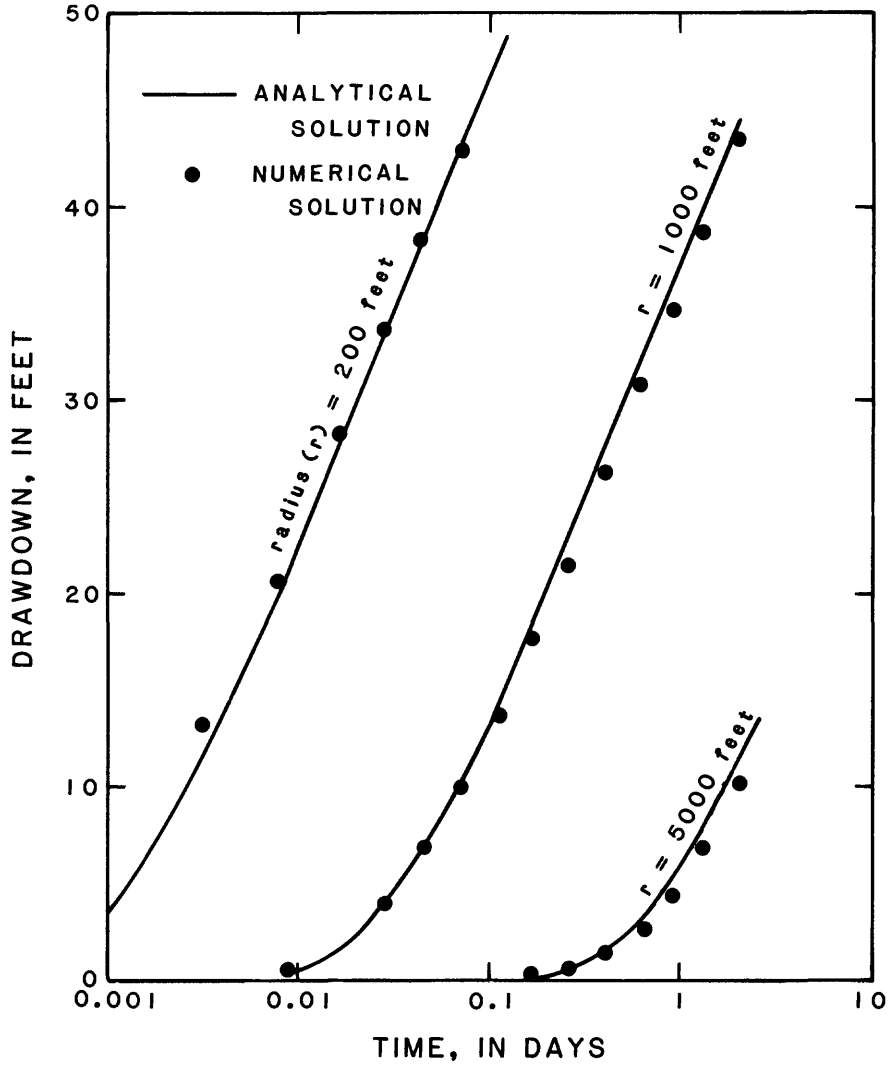

Figure 2.--Comparison of flow-model results with Theis' analytical solution.

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 occur at depth as underflow and the hydraulic conductivity is not known exactly, the least-squares matrix usually is poorly 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 characteristically have large standard errors in estimates of 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 nonreactive. 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. The dimensions of the blocks were made sufficiently large so as not to violate assumptions in the analytical solution. Several simulations with various block dimensions were made to ensure the block dimensions were large enough not to affect the solution.

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

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 2 (in the "Supplemental Information" section at the end of the report). Data-input formats are described in table 3 (in the "Supplemental Information" section at the end of the report). Although data entry into a finite-element program typically is more cumbersome than for finite-difference programs, 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 relation 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.

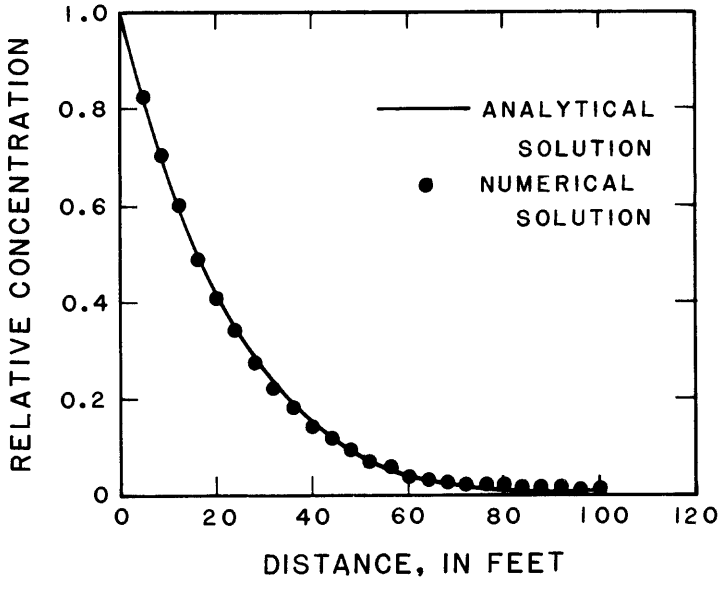

Figure 3.--Comparison of results from solute-transport model with analytical solution at 100 days.

The model procedure used to enter aquifer properties (such as hydraulic conductivity, dispersivity, or distributed recharge due to precipitation) is both flexible and easy to use. Aquifer elements are grouped into a number of user-defined aquifer zones. An aquifer property, hydraulic conductivity, for example, is simulated as the product of an element value and a zonal factor. Because hydraulic conductivity may be anisotropic, a separate zonal multiplication factor is used for each of the three principal directions. By using unique factors for each aquifer zone, it is possible to simulate varying degrees of anisotropy throughout the aquifer.

Steady-state ground-water flow properties in one or more zones can be grouped to form a single parameter of the regression analysis. Computed changes in a regression parameter are applied to the appropriate zonal factors. Element values of an aquifer property are unaffected by the regression analysis.

Boundary conditions for either ground-water flow or solute-transport may be treated as constant throughout a simulation or may be changed at user-specific times. Similarly, measured water levels and solute concentrations may be specified at any time. Input formats for transient boundary conditions and measured data are given in table 3.

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 conditions and nodal locations are shown horizontally in figure 4 and vertically in figure 5. Aquifer properties for each layer are given in table 1. 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 (in the "Supplemental Information" section at the end of the 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 coefficients also are printed for each element.

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.

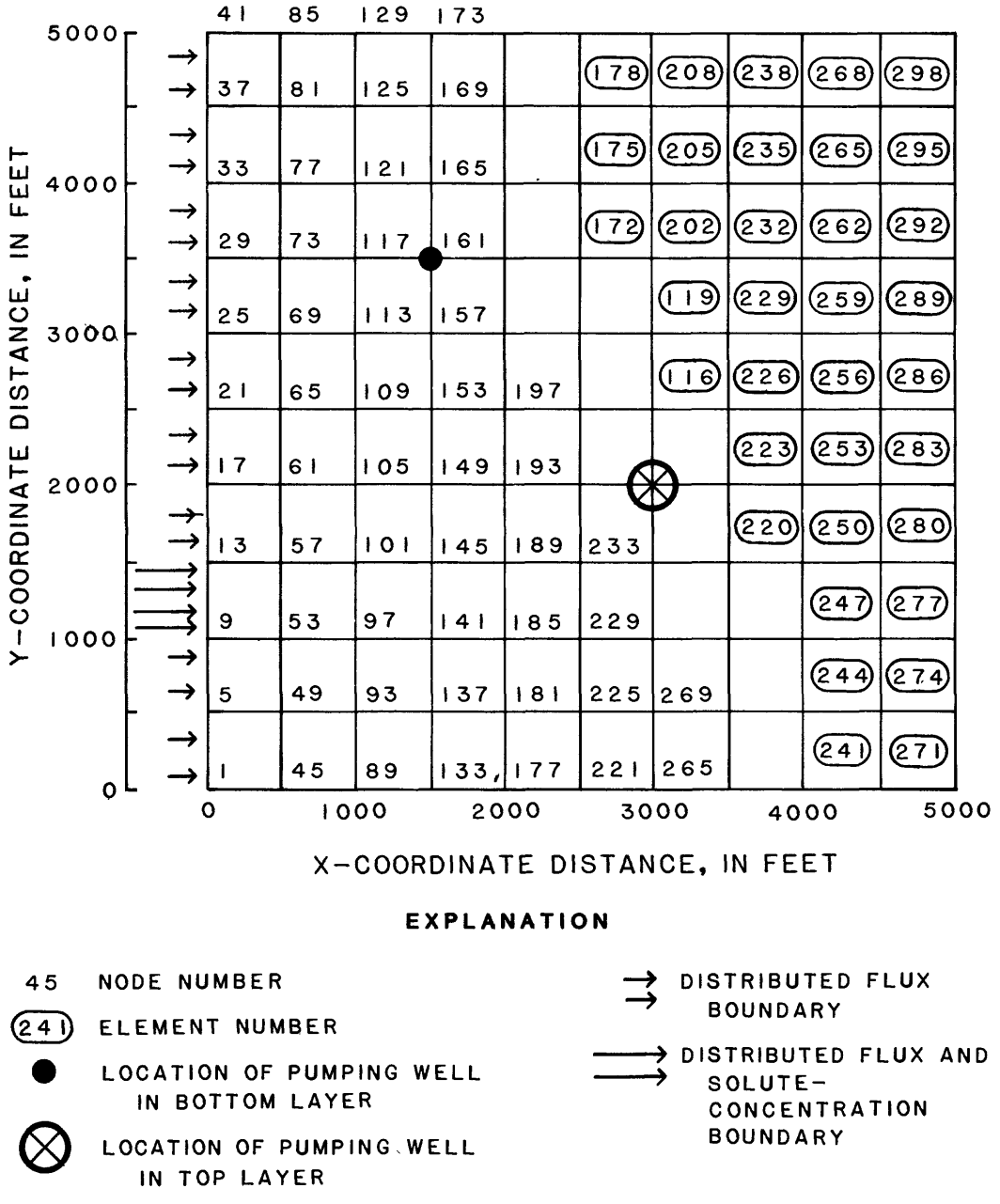

Figure 4.--Boundary conditions and node locations along the bottom of the aquifer system used in the sample simulation.

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.

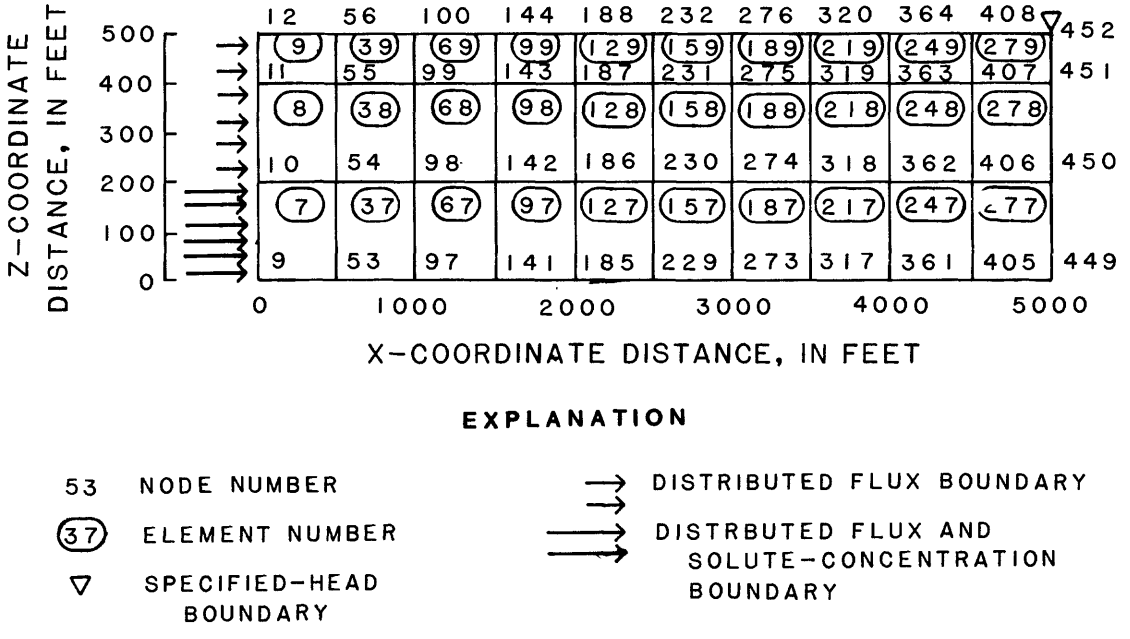

Figure 5.--Boundary conditions and node locations along a vertical section of the aquifer system used in the sample simulation.

Table 1.--*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 (feet squared 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	--	--

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.

Glover, K.C., 1986, Leachate migration from an in situ oil-shale retort near Rock Springs, Wyoming: U.S. Geological Survey Open-File Report 85-575, 68 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., 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.

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.

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 2.--*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
```

Table 2.--*Computer-program listing*--Continued

```
C** LDUBLK --                                                          530
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
```

Table 2.--*Computer-program listing*--Continued

```
110   CONTINUE                                                          1050
      IF (THETAC.LT.0.0) GO TO 130                                      1060
      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
```

Table 2.--*Computer-program listing*--Continued

```
      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
      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
```

Table 2.--*Computer-program listing*--Continued

```
      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
      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
```

Table 2.--*Computer-program listing*--Continued

```
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
      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
```

Table 2.--*Computer-program listing*--Continued

```
      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
      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
```

Table 2.--*Computer-program listing*--Continued

```
      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
     $        '  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
```

Table 2.--*Computer-program listing*--Continued

```
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
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
```

Table 2.--*Computer-program listing*--Continued

```
      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
      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
```

Table 2.--*Computer-program listing*--Continued

```
      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
     $           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
```

Table 2.--*Computer-program listing*--Continued

```
      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
      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
```

Table 2.--*Computer-program listing*--Continued

```
      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
      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
```

Table 2.--*Computer-program listing*--Continued

```
      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
      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
```

Table 2.--*Computer-program listing*--Continued

```
      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
358   CONTINUE                                                        7430
360   CONTINUE                                                        7440
      ITMP=NUMNP*NTIME/NTPER                                          7450
C** CHECK FOR CONVERGENCE                                            7460
362   DO 364 I=1,NVARH                                                7470
      IF (ABS(BH(I)/AP).GT.ER) GO TO 366                              7480
364   CONTINUE                                                        7490
      GO TO 374                                                       7500
C** CALCULATE NEW SCALED PRIOR INFORMATION  PARAMETERS                7510
366   IND=0                                                           7520
      DO 370 I=1,NVARH                                                7530
      TEMP=BH(I)+1.0                                                  7540
      PH(I)=PH(I)/TEMP                                                7550
      IF (ABS(PH(I)).LT.ERP) GO TO 368                                7560
      WRITE (6, 77) I                                                 7570
      IND=1                                                           7580
368   RKH(I)=RKH(I)*TEMP*TEMP                                         7590
370   CONTINUE                                                        7600
C** END ITERATION LOOP                                               7610
372   CONTINUE                                                        7620
      WRITE (6, 78) ITMAX                                             7630
      GO TO 376                                                       7640
374   WRITE (6, 79) ITER                                              7650
376   CONTINUE                                                        7660
C** COMPUTE SUM OF SQUARES AND CORRELATION COEFFICIENT                7670
378   SUMA=0.                                                         7680
      SUMB=0.                                                         7690
      SUMC=0.                                                         7700
      SUMD=0.                                                         7710
      SUM=0.                                                          7720
      DO 382 N=1,NOBMAX                                               7730
      IF (WH(N).LE.0.) GO TO 382                                      7740
      TMP=WH(N)**.5                                                   7750
      WH(N)=TMP                                                       7760
      TEMP=TMP*HO(N)                                                  7770
      HTMP=0.0                                                        7780
      DO 380 J=1,NUMNP                                                7790
      IF (NODOBS(J).EQ.N) HTMP=HTMP+SFOBSH(J)*H(J+NUMNP)              7800
```

Table 2.--*Computer-program listing*--Continued

```
380   CONTINUE                                                          7810
      TMP=TMP*HTMP                                                      7820
      SUMA=SUMA+TEMP                                                    7830
      SUMB=SUMB+TMP                                                     7840
      SUMC=SUMC+TEMP*TEMP                                               7850
      SUMD=SUMD+TMP*TMP                                                 7860
      SUM=SUM+TEMP*TMP                                                  7870
382   CONTINUE                                                          7880
      RCOEF=(OBSH*SUM-SUMA*SUMB)/((OBSH*SUMC-SUMA*SUMA)*                7890
     $  (OBSH*SUMD-SUMB*SUMB))**0.5                                     7900
      TEMP=AP*(2.-AP)                                                   7910
      TMPA=2.*AP*RP/TEMP                                                7920
      SUM=0.0                                                           7930
      DO 384 I=1,NVARH                                                  7940
      TMP=PH(I)-1.0                                                     7950
      SUM=SUM+TEMP*BH(I+NVARH)*(BH(I+NVHX2)-TMPA*BH(I+NVHX3)*           7960
     $        (RPF*PH(I)-1.0))-RKH(I)*TMP*TMP                           7970
384   CONTINUE                                                          7980
      YSQH=YSQH-SUM                                                     7990
C** COMPUTE SCALED VARIANCE-COVARIANCE MATRIX, ERROR VARIANCE, AND      8000
C** OPTIMUM BIAS PARAMETER:                                            8010
C** CORRECT XTXH FOR MARQUARDT PARAMETER                               8020
      IF (NVARH.EQ.1) GO TO 435                                         8030
      IF(AMP.LE.0) GO TO 390                                            8040
      DO 388 I=1,NVARH                                                  8050
      XTXH(I,I)=1.+RP                                                   8060
      DO 386 J=1,NVARH                                                  8070
386   XTXH(J,I)=XTXH(I,J)                                               8080
388   CONTINUE                                                          8090
      AMP=-1.0                                                          8100
      CALL LSTSQ (XTXH,BH,RKH,PH,YSQH,AP,AMP,RP,RPF,NVARH,NVX2,NVX3,    8110
     $            IPO,INDT,ITER,NVEH,LDIMH)                             8120
      IF(INDT.GT.0) GO TO 521                                           8130
C**COMPUTE XTXH-INVERSE                                                 8140
390   XTXH(NVARH,NVARH)=1.0/XTXH(NVARH,NVARH)                           8150
      NM1=NVARH-1                                                       8160
      DO 430 K=1,NM1                                                    8170
      KP1=K+1                                                           8180
      DO 400 I=KP1,NVARH                                                8190
      SUM=0.0                                                           8200
      IM1=I-1                                                           8210
      DO 392 J=K,IM1                                                    8220
392   SUM=SUM+XTXH(I,J)*XTXH(J,K)                                       8230
      XTXH(K,I)=-SUM                                                    8240
400   XTXH(I,K)=-SUM*XTXH(I,I)                                          8250
      DO 420 J=1,K                                                      8260
      SUM=XTXH(K,J)                                                     8270
      DO 410 I=KP1,NVARH                                                8280
410   SUM=SUM+XTXH(I,J)*XTXH(K,I)                                       8290
      XTXH(K,J)=SUM                                                     8300
420   XTXH(J,K)=XTXH(K,J)                                               8310
430   CONTINUE                                                          8320
```

Table 2.--*Computer-program listing*--Continued

```
      GO TO 440                                                          8330
435   XTXH(1,1)=1.0/(1.+RP)                                              8340
440   DO 450 J=1,NVARH                                                   8350
      BH(J)=0.0                                                          8360
      BH(J+NVARH)=AP*BH(J+NVARH)+BH(J+NVHX3)                             8370
      PH(J)=BH(J+NVHX3)*PH(J)                                            8380
450   XTXH(J,NVARH)=XTXH(NVARH,J)                                        8390
C**COMPUTE TR(XTXH-INVERSE**2) AND RPF                                   8400
      TRACE=0.0                                                          8410
      RPF=0.0                                                            8420
      IF(RP.LE.0.0) GO TO 459                                            8430
      DO 457 N=1,NVARH                                                   8440
      DO 452 J=1,NVARH                                                   8450
452   WELL(J)=XTXH(J,N)                                                  8460
      SUMA=0.                                                            8470
      DO 456 J=N,NVARH                                                   8480
      SUM=0.                                                             8490
      DO 454 I=1,NVARH                                                   8500
454   SUM=SUM+WELL(I)*XTXH(I,J)                                          8510
      QHTMP(J)=SUM                                                       8520
      SUMA=SUMA+PH(J)*SUM                                                8530
      BH(J)=BH(J)+PH(N)*SUM                                              8540
456   XTXH(J,N)=XTXH(J,N)-RP*SUM                                         8550
      BH(N)=BH(N)+SUMA-QHTMP(N)*PH(N)                                    8560
457   TRACE=TRACE+QHTMP(N)                                               8570
      IF(IRPF.LE.0) GO TO 459                                            8580
      SUM=0.                                                             8590
      DO 458 I=1,NVARH                                                   8600
      RPF=RPF+BH(I)*BH(I+NVARH)                                          8610
458   SUM=SUM+BH(I)*PH(I)                                                8620
      RPF=RPF/SUM                                                        8630
C** COMPUTE ERROR VARIANCE AND COV(SCALED PARAMETERS)                    8640
459   TEMP=NVARH-NPRIRH                                                  8650
      VARH=YSQH/(OBSH-TEMP+RP*RP*TRACE)                                  8660
      SUM=0.0                                                            8670
      DO 462 J=1,NVARH                                                   8680
      TMP=BH(J+NVARH)-RPF*PH(J)                                          8690
      SUM=SUM+TMP*TMP                                                    8700
      TEMP=BH(J+NVHX3)                                                   8710
      DO 460 I=J,NVARH                                                   8720
      XTXH(I,J)=VARH*XTXH(I,J)/(BH(I+NVHX3)*TEMP)                        8730
460   XTXH(J,I)=XTXH(I,J)                                                8740
462   BH(J)=XTXH(J,J)**0.5                                               8750
      TEMP=NVARH                                                         8760
      SUM=TEMP*VARH/SUM                                                  8770
C** PRINT ERROR VARIANCE, ESTIMATED SUM OF SQUARED ERRORS, CORRELATION   8780
C** COEFFICIENT, AND OPTIMUM RIDGE PARAMETERS                            8790
      WRITE(6,864) VARH,YSQH,RCOEF,SUM,RPF                               8800
C** PRINT SCALED VARIANCE-COVARIANCE MATRIX                              8810
      WRITE(6,874)                                                       8820
      CALL PRTOT(XTXH,NVEH,0,NVARH)                                      8830
C** COMPUTE AND PRINT CORRELATION MATRIX FOR PARAMETERS                  8840
```

Table 2.--*Computer-program listing*--Continued

```
      DO 510 J=1,NVARH                                                  8850
      TEMP=BH(J)                                                        8860
      DO 500 I=J,NVARH                                                  8870
      XTXH(I,J)=XTXH(I,J)/(BH(I)*TEMP)                                  8880
500   XTXH(J,I)=XTXH(I,J)                                               8890
510   CONTINUE                                                          8900
      WRITE(6,876)                                                      8910
      CALL PRTOT(XTXH,NVEH,0,NVARH)                                     8920
521   IF(IPRX.LE.0.AND.ITER.LT.ITMAX.AND.IBEALE.EQ.0) STOP              8930
      IF(NVAR.LT.2.AND.IBEALE.EQ.0) STOP                                8940
C** ORTHOGONALIZE SENSITIVITY MATRIX, COMPRESS SH                       8950
      N=NOBMAX                                                          8960
      DO 536 I=1,NVARH                                                  8970
      DO 532 J=1,NOBMAX                                                 8980
      WH(J)=0.0                                                         8990
532   CONTINUE                                                          9000
      DO 534 J=1,NUMNP                                                  9010
      K=NODOBS(J)                                                       9020
      IF (K.GT.0) WH(K)=WH(K)+SFOBSH(J)*SH(I,J)                         9030
534   CONTINUE                                                          9040
      DO 535 J=1,NOBMAX                                                 9050
      SH(I,J)=WH(J)                                                     9060
535   CONTINUE                                                          9070
536   CONTINUE                                                          9080
      IF(NPRIRH.LE.0) GO TO 539                                         9090
      DO 538 I=1,NVARH                                                  9100
      IF(RKH(I).LT.1.E-10) GO TO 538                                    9110
      N=N+1                                                             9120
      DO 537 J=1,NVARH                                                  9130
537   SH(J,N)=0.0                                                       9140
      SH(I,N)=RKH(I)**0.5                                               9150
538   CONTINUE                                                          9160
C** ORTHOGONALIZE SH                                                   9170
539   IF (IBEALE.EQ.0) GO TO 542                                        9180
      WRITE (6,860)                                                     9190
      DO 540 I=1,NOBMAX                                                 9200
      WRITE (6,862) I,(SH(J,I),J=1,NVARH)                               9210
540   CONTINUE                                                          9220
      IF (IPRX.LE.0.AND.ITER.LT.ITMAX) STOP                             9230
      IF (NVAR.LT.2) STOP                                               9240
542   NTMP=OBSH+NPRIRH                                                  9250
      DO 544 I=1,NTMP                                                   9260
544   WH(I)=SH(1,I)                                                     9270
      DO 600 N=2,NVARH                                                  9280
      NM1=N-1                                                           9290
      SUM=0.0                                                           9300
      DO 550 I=1,NTMP                                                   9310
      SUM=SUM+WH(I)*WH(I)                                               9320
      SH(NM1,I)=WH(I)                                                   9330
550   CONTINUE                                                          9340
      IF(SUM.LT.1.E-20) GO TO 610                                       9350
      BH(NM1)=1.0/SUM                                                   9360
```

Table 2.--*Computer-program listing*--Continued

```
      DO 570 J=1,NM1                                                    9370
      SUM=0.0                                                           9380
      DO 560 K=1,NTMP                                                   9390
560   SUM=SUM+BH(J)*SH(J,K)*SH(N,K)                                     9400
570   WELL(J)=SUM                                                       9410
      DO 590 K=1,NTMP                                                   9420
      SUM=0.0                                                           9430
      DO 580 I=1,NM1                                                    9440
580   SUM=SUM+SH(I,K)*WELL(I)                                           9450
590   WH(K)=SH(N,K)-SUM                                                 9460
600   CONTINUE                                                          9470
610   WRITE(6,886)                                                      9480
      K=1                                                               9490
      L=8                                                               9500
      DO 630 M=1,NVARH,8                                                9510
      IF(L.GT.NVARH) L=NVARH                                            9520
      WRITE(6,888) (I,I=K,L)                                            9530
      DO 620 J=1,NTMP                                                   9540
      SH(NVARH,J)=WH(J)                                                 9550
      WRITE(6,890) J,(SH(I,J),I=K,L)                                    9560
620   CONTINUE                                                          9570
      K=K+8                                                             9580
      L=L+8                                                             9590
630   CONTINUE                                                          9600
640   CONTINUE                                                          9610
      STOP                                                              9620
860   FORMAT ('  OBS',10X,'DEPTH-INTEGRATED SENSITIVITY COEFFICIENTS')  9630
862   FORMAT (I5,9E14.7,1X/5X,9E14.7,1X/5X,9E14.7)                      9640
864   FORMAT ('0ERROR VARIANCE = ',E12.5/' ESTIMATED SUM OF SQUARED ',  9650
     $'ERRORS = ',E12.5/' CORRELATION COEFFICIENT = ',F6.4/' OPTIMUM RID 9660
     $GE PARAMETER = ',E12.5/' OPTIMUM BIAS PARAMETER = ',E12.5)        9670
874   FORMAT ('0SCALED VARIANCE-COVARIANCE MATRIX')                     9680
876   FORMAT ('0CORRELATION MATRIX FOR PARAMETERS')                     9690
882   FORMAT ('0SCALED SENSITIVITY ARRAYS')                             9700
884   FORMAT('0PARAMETER NUMBER ',I5)                                   9710
886   FORMAT(1H0,5X,'ORTHOGONALIZED SENSITIVITY MATRIX'/                9720
     $ '  OBSERVATION NUMBER AND VALUES FOR EACH PARAMETER')            9730
888   FORMAT (15X,'PARAMETER NOS.'/16X,8(I3,9X))                        9740
890   FORMAT (2X,I4,8(1X,G11.5))                                        9750
      END                                                               9760
      SUBROUTINE FLOW (IWCNT,ISCNT,IBCNT,KT,                            9770
     $                 KTH,NTPER,NUMNP,NVARH,NPARH,LU,                  9780
     $                 NUMEL,ISTEDY,ITER,IDIM)                          9790
C** ROUTINE TO FORMULATE GLOBAL STIFFNESS AND                           9800
C** SENSITIVITY MATRICES                                               9810
      COMMON/COFBLK/CFRAC(10, 490),QHTMP( 490),CINIT( 490),             9820
     $      H( 980),C( 980),WELL( 490),WELLC( 490),                     9830
     $      RX(7),RY(7),RZ(7),RC(7),                                    9840
     $      RPR(7),DM(7),FP(7),FQ(7),RMT(7),HO(30),                     9850
     $      RPBR(7),CBINIT(7),SFOBSH( 520),SFOBSC( 520),WTNOD( 490),    9860
     $      DL(7),DT(7),THETAH,DTIME,TIMFAC,TIMBGN,                     9870
     $      CEQUI,DFW,TIMSUM,IPRMH(3,7),MAT(300),                       9880
```

Table 2.--*Computer-program listing*--Continued

```
     $      NBP( 490),IELMR                                                  9890
      COMMON/SINBLK/QEH(8),QEC(8),XORD( 490),YORD( 490),ZORD( 490),          9900
     $      QS( 980),CS( 980),QF(1),CF(1),NP(300,8),IBPRM(300)               9910
      COMMON/SENBLK/WH(30),                                                  9920
     $      WC(30),THETAC,NPBC( 490,1),NODOBS( 490)                          9930
      COMMON/LDUBLK/CPKC(490,180),CPKH(490, 90),SH(8, 490),                  9940
     $              QC( 490),QH( 490),IBC,IBH                                9950
      REAL*8 CPKC,CPKH,SH,QC,QH                                              9960
      COMMON/SHABLK/SF(4,8,8),WF(4,8),WT(8),NUMQPT                           9970
      COMMON/LINBLK/RLF(2,2),RLWT,LINOD(4,2)                                 9980
      DIMENSION KARG(4)                                                      9990
      DIMENSION DH(8,8),EH(8,8),RJAC(8,8),RJACI(8,8),                       10000
     $          DNDX(8),DNDY(8),DNDZ(8)                                     10010
      DOUBLE PRECISION SF,WT,WF,QEH,QEC,DH,EH,RJAC,RJACI,DNDX,DNDY,         10020
     $                 DNDZ,TEMP,TMP,TMP1,TMP2                             10030
      DOUBLE PRECISION DETJ,RN,DXH,DYH,DZH,RNN                              10040
      DO 260 I=1,NUMNP                                                      10050
      N=NPBC(I,IBCNT)                                                       10060
      I1=(IWCNT-1)*NUMNP+I                                                  10070
      QH(I)=WELL(I1)                                                        10080
      IF (N.GT.0) QH(I)=QH(I)*QF(N)                                         10090
      IF (NVARH.LE.0) GO TO 258                                             10100
      DO 255 J=1,NVARH                                                      10110
      SH(J,I)=0.0                                                           10120
255   CONTINUE                                                              10130
258   N1=N+NPARH                                                            10140
      IF (N1.GT.NPARH) SH(N1,I)=SH(N1,I)+QH(I)                              10150
260   CONTINUE                                                              10160
      IF (LU.EQ.0) GO TO 266                                                10170
      N=NUMNP+NOBMAX                                                        10180
      DO 250 I=1,N                                                          10190
      SFOBSH(I)=0.0                                                         10200
250   CONTINUE                                                              10210
      NCORNR=4                                                              10220
      DO 265 I=1,IDIM                                                       10230
      DO 262 J=1,IBH                                                        10240
      CPKH(I,J)=0.0                                                         10250
262   CONTINUE                                                              10260
265   CONTINUE                                                              10270
266   NODES=8                                                               10280
      DO 399 I=1,NUMEL                                                      10290
      MATI=IABS(MAT(I))                                                     10300
      DO 305 J=1,NODES                                                      10310
      QEH(J)=0.0                                                            10320
      DO 305 K=1,NODES                                                      10330
      DH(J,K)=0.0                                                           10340
      EH(J,K)=0.0                                                           10350
305   CONTINUE                                                              10360
C** VOLUME QUADRATURE                                                       10370
      DO 340 J=1,NUMQPT                                                     10380
      RJAC(1,1)=0.0                                                         10390
      RJAC(1,2)=0.0                                                         10400
```

Table 2.--*Computer-program listing*--Continued

```
      RJAC(1,3)=0.0                                                          10410
      RJAC(2,1)=0.0                                                          10420
      RJAC(2,2)=0.0                                                          10430
      RJAC(2,3)=0.0                                                          10440
      RJAC(3,1)=0.0                                                          10450
      RJAC(3,2)=0.0                                                          10460
      RJAC(3,3)=0.0                                                          10470
308   L1=IPRMH(1,MATI)                                                       10480
      L2=IPRMH(2,MATI)                                                       10490
      L3=IPRMH(3,MATI)                                                       10500
      RXJ=RX(MATI)                                                           10510
      RYJ=RY(MATI)                                                           10520
      RZJ=RZ(MATI)                                                           10530
      RCJ=RC(MATI)                                                           10540
      DO 310 K=1,NODES                                                       10550
      NPK=NP(I,K)                                                            10560
      RJAC(1,1)=RJAC(1,1)+SF(2,K,J)*XORD(NPK)                                10570
      RJAC(1,2)=RJAC(1,2)+SF(3,K,J)*XORD(NPK)                                10580
      RJAC(1,3)=RJAC(1,3)+SF(4,K,J)*XORD(NPK)                                10590
      RJAC(2,1)=RJAC(2,1)+SF(2,K,J)*YORD(NPK)                                10600
      RJAC(2,2)=RJAC(2,2)+SF(3,K,J)*YORD(NPK)                                10610
      RJAC(2,3)=RJAC(2,3)+SF(4,K,J)*YORD(NPK)                                10620
      RJAC(3,1)=RJAC(3,1)+SF(2,K,J)*ZORD(NPK)                                10630
      RJAC(3,2)=RJAC(3,2)+SF(3,K,J)*ZORD(NPK)                                10640
      RJAC(3,3)=RJAC(3,3)+SF(4,K,J)*ZORD(NPK)                                10650
310   CONTINUE                                                               10660
      DETJ=RJAC(1,1)*(RJAC(2,2)*RJAC(3,3)-RJAC(3,2)*RJAC(2,3))               10670
      DETJ=-RJAC(1,2)*(RJAC(2,1)*RJAC(3,3)-RJAC(3,1)*RJAC(2,3))+DETJ         10680
      DETJ=RJAC(1,3)*(RJAC(2,1)*RJAC(3,2)-RJAC(3,1)*RJAC(2,2))+DETJ          10690
      IF (DETJ.LE.0.0) GO TO 7001                                            10700
311   RJACI(1,1)=(RJAC(2,2)*RJAC(3,3)-RJAC(3,2)*RJAC(2,3))/DETJ              10710
      RJACI(1,2)=-(RJAC(1,2)*RJAC(3,3)-RJAC(3,2)*RJAC(1,3))/DETJ             10720
      RJACI(1,3)=(RJAC(1,2)*RJAC(2,3)-RJAC(2,2)*RJAC(1,3))/DETJ              10730
      RJACI(2,1)=-(RJAC(2,1)*RJAC(3,3)-RJAC(3,1)*RJAC(2,3))/DETJ             10740
      RJACI(2,2)=(RJAC(1,1)*RJAC(3,3)-RJAC(3,1)*RJAC(1,3))/DETJ              10750
      RJACI(2,3)=-(RJAC(1,1)*RJAC(2,3)-RJAC(2,1)*RJAC(1,3))/DETJ             10760
      RJACI(3,1)=(RJAC(2,1)*RJAC(3,2)-RJAC(3,1)*RJAC(2,2))/DETJ              10770
      RJACI(3,2)=-(RJAC(1,1)*RJAC(3,2)-RJAC(3,1)*RJAC(1,2))/DETJ             10780
      RJACI(3,3)=(RJAC(1,1)*RJAC(2,2)-RJAC(2,1)*RJAC(1,2))/DETJ              10790
      DO 315 K=1,NODES                                                       10800
      DNDX(K)=RJACI(1,1)*SF(2,K,J)+RJACI(2,1)*SF(3,K,J)+RJACI(3,1)           10810
     &        *SF(4,K,J)                                                     10820
      DNDY(K)=RJACI(1,2)*SF(2,K,J)+RJACI(2,2)*SF(3,K,J)+RJACI(3,2)           10830
     &        *SF(4,K,J)                                                     10840
      DNDZ(K)=RJACI(1,3)*SF(2,K,J)+RJACI(2,3)*SF(3,K,J)+RJACI(3,3)           10850
     &        *SF(4,K,J)                                                     10860
315   CONTINUE                                                               10870
C** LINE INTEGRATION TO FILL SFOBSH                                          10880
      IF (NVARH.LE.0.OR.LU.EQ.0) GO TO 314                                   10890
      DO 280 K=1,NCORNR                                                      10900
      NP1=LINOD(K,1)                                                         10910
      NP2=LINOD(K,2)                                                         10920
```

Table 2.--*Computer-program listing*--Continued

```
      NP1=NP(I,NP1)                                                        10930
      NP2=NP(I,NP2)                                                        10940
      IF (NODOBS(NP1).LE.0.OR.NODOBS(NP2).LE.0) GO TO 280                  10950
      IF (NODOBS(NP1).NE.NODOBS(NP2)) GO TO 280                            10960
      N=NUMNP+NODOBS(NP1)                                                  10970
      DETL=0.0                                                             10980
      DO 268 L=1,2                                                         10990
      NPL=LINOD(K,L)                                                       11000
      NPL=NP(I,NPL)                                                        11010
      DETL=DETL+RLF(2,L)*ZORD(NPL)                                         11020
268   CONTINUE                                                             11030
      DETL=ABS(DETL)*WTNOD(NP1)*WTNOD(NP2)                                 11040
      DO 270 L=1,2                                                         11050
      NPL=LINOD(K,L)                                                       11060
      NPL=NP(I,NPL)                                                        11070
      NPK=2                                                                11080
      IF (L.EQ.2) NPK=1                                                    11090
      NPK=LINOD(K,NPK)                                                     11100
      NPK=NP(I,NPK)                                                        11110
      SFOBSH(NPL)=SFOBSH(NPL)+RLWT*RLF(1,L)*DETL*WTNOD(NPL)*               11120
     $            (RXJ+RYJ)                                              11130
      SFOBSH(NPK)=SFOBSH(NPK)+RLWT*RLF(1,L)*DETL*(1.0-WTNOD(NPL))*         11140
     $            (RXJ+RYJ)                                              11150
      SFOBSH(N)=SFOBSH(N)+RLWT*RLF(1,L)*DETL*(RXJ+RYJ)                     11160
270   CONTINUE                                                             11170
280   CONTINUE                                                             11180
C**  END OF LINE INTEGRATION                                              11190
314   DO 330 K=1,NODES                                                     11200
      NPK=NP(I,K)                                                          11210
      I1=NPK+NUMNP                                                         11220
      I2=NPK                                                               11230
      I4=NPK+NUMNP                                                         11240
      I5=I4-NUMNP                                                          11250
229   DO 329 L=1,NODES                                                     11260
317   NPL=NP(I,L)+NUMNP                                                    11270
      NPLM1=NPL-NUMNP                                                      11280
      J1=NP(I,L)+NUMNP                                                     11290
      J2=J1-NUMNP                                                          11300
      RNN=WT(J)*SF(1,K,J)*SF(1,L,J)*DETJ                                   11310
      DXH=WT(J)*DNDX(K)*DNDX(L)*DETJ                                       11320
      DYH=WT(J)*DNDY(K)*DNDY(L)*DETJ                                       11330
      DZH=WT(J)*DNDZ(K)*DNDZ(L)*DETJ                                       11340
      DH(K,L)=DH(K,L)+DXH*RXJ+DYH*RYJ+DZH*RZJ                              11350
      EH(K,L)=EH(K,L)+RNN*RCJ                                              11360
      IF (L1.GT.0) SH(L1,NPK)=SH(L1,NPK)+DXH*(H(J1)*THETAH-H(J2)*(1.0-     11370
     $             THETAH))*RXJ                                            11380
      IF (L2.GT.0) SH(L2,NPK)=SH(L2,NPK)+DYH                               11390
     $             *(H(J1)*THETAH-H(J2)*(1.0-THETAH))*RYJ                  11400
      IF (L3.GT.0) SH(L3,NPK)=SH(L3,NPK)+DZH                               11410
     $             *(H(J1)*THETAH-H(J2)*(1.0-THETAH))*RZJ                  11420
329   CONTINUE                                                             11430
330   CONTINUE                                                             11440
```

Table 2.--*Computer-program listing*--Continued

```
340   CONTINUE                                                    11450
C**  SURFACE QUADRATURE                                          11460
      IF (MAT(I).GE.0) GO TO 382                                  11470
360   DO 369 J=1,6                                                11480
      GO TO (362,364,365,366,367,359),J                           11490
362   K1=NP(I,1)+(ISCNT-1)*NUMNP                                  11500
      K2=NP(I,2)+(ISCNT-1)*NUMNP                                  11510
      K3=NP(I,3)+(ISCNT-1)*NUMNP                                  11520
      K4=NP(I,4)+(ISCNT-1)*NUMNP                                  11530
      NN=4                                                        11540
      GO TO 368                                                   11550
364   K1=NP(I,5)+(ISCNT-1)*NUMNP                                  11560
      K2=NP(I,6)+(ISCNT-1)*NUMNP                                  11570
      K3=NP(I,7)+(ISCNT-1)*NUMNP                                  11580
      K4=NP(I,8)+(ISCNT-1)*NUMNP                                  11590
      NN=4                                                        11600
      GO TO 368                                                   11610
365   K1=NP(I,1)+(ISCNT-1)*NUMNP                                  11620
      K2=NP(I,5)+(ISCNT-1)*NUMNP                                  11630
      K3=NP(I,6)+(ISCNT-1)*NUMNP                                  11640
      K4=NP(I,2)+(ISCNT-1)*NUMNP                                  11650
      NN=4                                                        11660
      GO TO 368                                                   11670
366   K1=NP(I,1)+(ISCNT-1)*NUMNP                                  11680
      K2=NP(I,4)+(ISCNT-1)*NUMNP                                  11690
      K3=NP(I,8)+(ISCNT-1)*NUMNP                                  11700
      K4=NP(I,5)+(ISCNT-1)*NUMNP                                  11710
      NN=4                                                        11720
      GO TO 368                                                   11730
367   K1=NP(I,2)+(ISCNT-1)*NUMNP                                  11740
      K2=NP(I,6)+(ISCNT-1)*NUMNP                                  11750
      K3=NP(I,7)+(ISCNT-1)*NUMNP                                  11760
      K4=NP(I,3)+(ISCNT-1)*NUMNP                                  11770
      NN=4                                                        11780
      GO TO 368                                                   11790
359   K1=NP(I,3)+(ISCNT-1)*NUMNP                                  11800
      K2=NP(I,4)+(ISCNT-1)*NUMNP                                  11810
      K3=NP(I,8)+(ISCNT-1)*NUMNP                                  11820
      K4=NP(I,7)+(ISCNT-1)*NUMNP                                  11830
      NN=4                                                        11840
368   CHKQ=QS(K1)*QS(K2)*QS(K3)*QS(K4)                            11850
      IF (CHKQ.EQ.0.0) GO TO 369                                  11860
      KARG(1)=K1                                                  11870
      KARG(2)=K2                                                  11880
      KARG(3)=K3                                                  11890
      KARG(4)=K4                                                  11900
      CALL SURINT(I,NODES,KARG,NN,NPARH,0,ISCNT,NUMNP)            11910
369   CONTINUE                                                    11920
C**  PLACE IN LARGE CPKH AND Q MATRICES                          11930
382   DO 386 J=1,NODES                                            11940
      NPJ=NP(I,J)                                                 11950
      QH(NPJ)=QH(NPJ)+QEH(J)                                      11960
```

Table 2.--*Computer-program listing*--Continued

```
379    DO 384 K=1,NODES                                                    11970
       NPK=NP(I,K)                                                         11980
       K1=NPK-NPJ+1                                                        11990
390    QH(NPJ)=QH(NPJ)+(EH(J,K)/DTIME/FLOAT(NTPER)-                        12000
      $                (1.0-THETAH)*DH(J,K))*H(I2)                        12010
389    IF (NPBC(NPJ,IBCNT).GE.0.OR.NPBC(NPJ,IBCNT).EQ.-2.OR.               12020
      $    NPBC(NPJ,IBCNT).EQ.-4) GO TO 388                               12030
       IF (NBP(NPJ).LE.0) GO TO 388                                        12040
       IF (NBP(NPJ).EQ.NBP(NPK)) GO TO 388                                 12050
       N=NBP(NPJ)+NQPAR+NPARH                                              12060
       SH(N,NPK)=SH(N,NPK)+(EH(J,K)/DTIME/FLOAT(NTPER)+THETAH*DH(J,K))     12070
      $          *H(NPJ+NUMNP)-(EH(J,K)/DTIME/FLOAT(NTPER)-(1.0-          12080
      $          THETAH)*DH(J,K))*H(NPJ)                                  12090
388    IF (NPK.LT.NPJ) GO TO 385                                           12100
       IF (LU.EQ.1)                                                        12110
      $CPKH(NPJ,K1)=CPKH(NPJ,K1)+EH(J,K)/DTIME/FLOAT(NTPER)+THETAH*        12120
      $             DH(J,K)                                               12130
385    CONTINUE                                                            12140
384    CONTINUE                                                            12150
386    CONTINUE                                                            12160
399    CONTINUE                                                            12170
C** BOUNDRY CONDITIONS                                                     12180
       DO 420 I=1,NUMNP                                                    12190
       IF (NPBC(I,IBCNT).GE.0.OR.NPBC(I,IBCNT).EQ.-4) GO TO 420            12200
       IF (NPBC(I,IBCNT).EQ.-2) GO TO 420                                  12210
       IF (LU.EQ.1) CPKH(I,1)=1.0D+50                                      12220
       I1=I+NUMNP                                                          12230
       QH(I)=H(I1)*1.0D+50                                                 12240
420    CONTINUE                                                            12250
       IF (LU.EQ.0.OR.NVARH.LE.0) RETURN                                   12260
C** FINISH COMPUTATION OF SFOBSH                                           12270
       DO 294 I=1,NUMNP                                                    12280
       N=NUMNP+NODOBS(I)                                                   12290
       IF (N.EQ.NUMNP) GO TO 294                                           12300
       SFOBSH(I)=SFOBSH(I)/SFOBSH(N)                                       12310
294    CONTINUE                                                            12320
       RETURN                                                              12330
7001   WRITE (6,17) I,J                                                    12340
17     FORMAT ('0ELEMENT',I5,' HAS NEGATIVE OR ZERO DETJ AT ',             12350
      $        'QUADRATURE POINT',I5)                                     12360
       STOP                                                                12370
       END                                                                 12380
       SUBROUTINE SALT (IFLOW,IWCNT,ISCNT,IBCNT,KT,KTH,NTPER,NUMNP,        12390
      $                 LU,NUMEL,ISTEDY,IVELPR,IFRAC,IHCFLG,              12400
      $                 ITER,IDIM,ALFAX,ALFAY,ALFAZ,ALFLG,NTFRAC,ITFRAC,  12410
      $                 NOBMAX,IOBCNT)                                    12420
C** ROUTINE TO FORMULATE GLOBAL STIFFNESS AND SENSITIVITY MATRICES          12430
       COMMON/COFBLK/CFRAC(10,490),QHTMP(490),CINIT( 490),                 12440
      $      H( 980),C( 980),WELL( 490),WELLC( 490),                      12450
      $      RX(7),RY(7),RZ(7),RC(7),                                     12460
      $      RPR(7),DM(7),FP(7),FQ(7),RMT(7),HO(30),                      12470
      $      RPBR(7),CBINIT(7),SFOBSH(520),SFOBSC(520),WTNOD( 490),       12480
```

Table 2.--*Computer-program listing*--Continued

```
     $      DL(7),DT(7),THETAH,DTIME,TIMFAC,TIMBGN,                          12490
     $      CEQUI,DFW,TIMSUM,IPRMH(3,7),MAT(300),NBP( 490),IELMR             12500
      COMMON/SINBLK/QEH(8),QEC(8),XORD( 490),YORD( 490),ZORD( 490),           12510
     $      QS( 980),CS( 980),QF(1),CF(1),NP(300,8),IBPRM(300)               12520
      COMMON/SENBLK/WH(30),                                                   12530
     $      WC(30),THETAC,NPBC( 490,1),NODOBS( 490)                          12540
      COMMON/LDUBLK/CPKC(490,180),CPKH(490, 90),SH(8, 490),                   12550
     $             QC( 490),QH( 490),IBC,IBH                                 12560
      REAL*8 CPKC,CPKH,SH,QC,QH                                               12570
      COMMON/SHABLK/SF(4,8,8),WF(4,8),WT(8),NUMQPT                            12580
      COMMON/LINBLK/RLF(2,2),RLWT,LINOD(4,2)                                  12590
      DIMENSION DC(8,8),EC(8,8),RJAC(8,8),RJACI(8,8),DNDX(8),DNDY(8),         12600
     $          DNDZ(8),KARG(4)                                               12610
      DOUBLE PRECISION SF,WT,WF,QEH,QEC,DC,EC,RJAC,RJACI,                     12620
     $                 DNDX,DNDY,DNDZ,TEMP,TMP,TMP1,TMP2,DFDX,                12630
     $                 DFDY,DFDZ                                              12640
      DOUBLE PRECISION RN,DXH,DYH,DZH,RNN,DXYC,DXZC,DYZC,DNXN,DNYN,           12650
     $                 DNZN,RNSUM                                             12660
      IDIAG=(IBC-1)/2+1                                                       12670
      THETMP=(FLOAT(IFLOW-1)+THETAC)/FLOAT(NTPER)                             12680
      IF (ISTEDY.EQ.1) THETMP=1.0                                             12690
      N=NUMNP+NOBMAX                                                          12700
      DO 250 I=1,N                                                            12710
      SFOBSC(I)=0.0                                                           12720
250   CONTINUE                                                                12730
      NCORNR=4                                                                12740
      DO 260 I=1,NUMNP                                                        12750
      N=NPBC(I,IBCNT)                                                         12760
      I1=(IWCNT-1)*NUMNP+I                                                    12770
241   QC(I)=0.0                                                               12780
      IF (WELL(I1).LE.0.0) GO TO 260                                          12790
      QC(I)=WELL(I1)*WELLC(I1)                                                12800
      IF (N.GT.0) QC(I)=QC(I)*CF(N)                                           12810
260   CONTINUE                                                                12820
      IF (LU.EQ.0) GO TO 266                                                  12830
      DO 265 I=1,IDIM                                                         12840
      DO 264 J=1,IBC                                                          12850
      CPKC(I,J)=0.0                                                           12860
264   CONTINUE                                                                12870
265   CONTINUE                                                                12880
266   IF (IVELPR.EQ.1) WRITE (6,18)                                           12890
18    FORMAT ('0ELEMENT   VX',11X,'VY',11X,'VZ',10X,'DISPX',8X,'DISPY',       12900
     $        8X,'DISPZ')                                                     12910
      NODES=8                                                                 12920
      DO 399 I=1,NUMEL                                                        12930
      MATI=IABS(MAT(I))                                                       12940
      DO 305 J=1,NODES                                                        12950
      QEC(J)=0.0                                                              12960
      DO 305 K=1,NODES                                                        12970
261   DC(J,K)=0.0                                                             12980
      EC(J,K)=0.0                                                             12990
305   CONTINUE                                                                13000
```

Table 2.--*Computer-program listing*--Continued

```
C** VOLUME QUADRATURE                                                    13010
      DO 340 J=1,NUMQPT                                                  13020
      XJ=0.0                                                             13030
      YJ=0.0                                                             13040
      ZJ=0.0                                                             13050
      VXJ=0.0                                                            13060
      VYJ=0.0                                                            13070
      VZJ=0.0                                                            13080
      RJAC(1,1)=0.0                                                      13090
      RJAC(1,2)=0.0                                                      13100
      RJAC(1,3)=0.0                                                      13110
      RJAC(2,1)=0.0                                                      13120
      RJAC(2,2)=0.0                                                      13130
      RJAC(2,3)=0.0                                                      13140
      RJAC(3,1)=0.0                                                      13150
      RJAC(3,2)=0.0                                                      13160
      RJAC(3,3)=0.0                                                      13170
      DO 310 K=1,NODES                                                   13180
      NPK=NP(I,K)                                                        13190
      RJAC(1,1)=RJAC(1,1)+SF(2,K,J)*XORD(NPK)                            13200
      RJAC(1,2)=RJAC(1,2)+SF(3,K,J)*XORD(NPK)                            13210
      RJAC(1,3)=RJAC(1,3)+SF(4,K,J)*XORD(NPK)                            13220
      RJAC(2,1)=RJAC(2,1)+SF(2,K,J)*YORD(NPK)                            13230
      RJAC(2,2)=RJAC(2,2)+SF(3,K,J)*YORD(NPK)                            13240
      RJAC(2,3)=RJAC(2,3)+SF(4,K,J)*YORD(NPK)                            13250
      RJAC(3,1)=RJAC(3,1)+SF(2,K,J)*ZORD(NPK)                            13260
      RJAC(3,2)=RJAC(3,2)+SF(3,K,J)*ZORD(NPK)                            13270
      RJAC(3,3)=RJAC(3,3)+SF(4,K,J)*ZORD(NPK)                            13280
310   CONTINUE                                                           13290
      DETJ=RJAC(1,1)*(RJAC(2,2)*RJAC(3,3)-RJAC(3,2)*RJAC(2,3))           13300
      DETJ=-RJAC(1,2)*(RJAC(2,1)*RJAC(3,3)-RJAC(3,1)*RJAC(2,3))+DETJ     13310
      DETJ=RJAC(1,3)*(RJAC(2,1)*RJAC(3,2)-RJAC(3,1)*RJAC(2,2))+DETJ      13320
      IF (DETJ.LE.0.0) GO TO 7001                                        13330
      RJACI(1,1)=(RJAC(2,2)*RJAC(3,3)-RJAC(3,2)*RJAC(2,3))/DETJ          13340
      RJACI(1,2)=-(RJAC(1,2)*RJAC(3,3)-RJAC(3,2)*RJAC(1,3))/DETJ         13350
      RJACI(1,3)=(RJAC(1,2)*RJAC(2,3)-RJAC(2,2)*RJAC(1,3))/DETJ          13360
      RJACI(2,1)=-(RJAC(2,1)*RJAC(3,3)-RJAC(3,1)*RJAC(2,3))/DETJ         13370
      RJACI(2,2)=(RJAC(1,1)*RJAC(3,3)-RJAC(3,1)*RJAC(1,3))/DETJ          13380
      RJACI(2,3)=-(RJAC(1,1)*RJAC(2,3)-RJAC(2,1)*RJAC(1,3))/DETJ         13390
      RJACI(3,1)=(RJAC(2,1)*RJAC(3,2)-RJAC(3,1)*RJAC(2,2))/DETJ          13400
      RJACI(3,2)=-(RJAC(1,1)*RJAC(3,2)-RJAC(3,1)*RJAC(1,2))/DETJ         13410
      RJACI(3,3)=(RJAC(1,1)*RJAC(2,2)-RJAC(2,1)*RJAC(1,2))/DETJ          13420
      DO 315 K=1,NODES                                                   13430
      DNDX(K)=RJACI(1,1)*SF(2,K,J)+RJACI(2,1)*SF(3,K,J)+RJACI(3,1)       13440
     &        *SF(4,K,J)                                                 13450
      DNDY(K)=RJACI(1,2)*SF(2,K,J)+RJACI(2,2)*SF(3,K,J)+RJACI(3,2)       13460
     &        *SF(4,K,J)                                                 13470
      DNDZ(K)=RJACI(1,3)*SF(2,K,J)+RJACI(2,3)*SF(3,K,J)+RJACI(3,3)       13480
     &        *SF(4,K,J)                                                 13490
315   CONTINUE                                                           13500
      DFDX=RJACI(1,1)*WF(2,J)+RJACI(2,1)*WF(3,J)+RJACI(3,1)*WF(4,J)      13510
      DFDY=RJACI(1,2)*WF(2,J)+RJACI(2,2)*WF(3,J)+RJACI(3,2)*WF(4,J)      13520
```

Table 2.--*Computer-program listing*--Continued

```
      DFDZ=RJACI(1,3)*WF(2,J)+RJACI(2,3)*WF(3,J)+RJACI(3,3)*WF(4,J)      13530
C** VELOCITY AND DISPERSION CALCULATIONS                                 13540
308   L1=IPRMH(1,MATI)                                                  13550
      L2=IPRMH(2,MATI)                                                  13560
      L3=IPRMH(3,MATI)                                                  13570
      RXJ=RX(MATI)                                                      13580
      RYJ=RY(MATI)                                                      13590
      RZJ=RZ(MATI)                                                      13600
      RCJ=RC(MATI)                                                      13610
      PORBJ=RPBR(MATI)                                                  13620
      PORJ=RPR(MATI)                                                    13630
      RMASSJ=RMT(MATI)                                                  13640
      DO 311 K=1,NODES                                                  13650
      J1=NP(I,K)+NUMNP                                                  13660
      J2=J1-NUMNP                                                       13670
      VXJ=VXJ-DNDX(K)*RXJ*(THETMP*H(J1)+(1.0-THETMP)*H(J2))             13680
      VYJ=VYJ-DNDY(K)*RYJ*(THETMP*H(J1)+(1.0-THETMP)*H(J2))             13690
      VZJ=VZJ-DNDZ(K)*RZJ*(THETMP*H(J1)+(1.0-THETMP)*H(J2))             13700
311   CONTINUE                                                          13710
      VMAG=(VXJ*VXJ+VYJ*VYJ+VZJ*VZJ)**0.5                               13720
      IF (ABS(VMAG).LT.1.0E-20) GO TO 313                               13730
312   DSPXJ=(DL(MATI)*VXJ*VXJ+DT(MATI)*VYJ*VYJ+DT(MATI)*VZJ*VZJ)/VMAG  13740
     $      +DFW                                                        13750
      DSPYJ=(DT(MATI)*VXJ*VXJ+DL(MATI)*VYJ*VYJ+DT(MATI)*VZJ*VZJ)/VMAG  13760
     $      +DFW                                                        13770
      DSPZJ=(DT(MATI)*VXJ*VXJ+DT(MATI)*VYJ*VYJ+DL(MATI)*VZJ*VZJ)/VMAG  13780
     $      +DFW                                                        13790
301   DSPXYJ=(DL(MATI)-DT(MATI))*VXJ*VYJ/VMAG+DFW                       13800
      DSPXZJ=(DL(MATI)-DT(MATI))*VXJ*VZJ/VMAG+DFW                       13810
      DSPYZJ=(DL(MATI)-DT(MATI))*VYJ*VZJ/VMAG+DFW                       13820
313   IF (IVELPR.EQ.1.AND.J.EQ.1) WRITE (6,19) I,VXJ,VYJ,VZJ,DSPXJ,     13830
     $                                DSPYJ,DSPZJ                       13840
19    FORMAT (I5,6(1X,E12.5))                                           13850
C** LINE INTEGRATION TO FILL SFOBSC                                      13860
      IF (NOBMAX.LE.0) GO TO 314                                        13870
      DO 280 K=1,NCORNR                                                 13880
      NP1=LINOD(K,1)                                                    13890
      NP2=LINOD(K,2)                                                    13900
      NP1=NP(I,NP1)                                                     13910
      NP2=NP(I,NP2)                                                     13920
      IF (NODOBS(NP1).LE.0.OR.NODOBS(NP2).LE.0) GO TO 280               13930
      IF (NODOBS(NP1).NE.NODOBS(NP2)) GO TO 280                         13940
      N=NUMNP+NODOBS(NP1)                                               13950
      DETL=0.0                                                          13960
      DO 268 L=1,2                                                      13970
      NPL=LINOD(K,L)                                                    13980
      NPL=NP(I,NPL)                                                     13990
      DETL=DETL+RLF(2,L)*ZORD(NPL)                                      14000
268   CONTINUE                                                          14010
      DETL=ABS(DETL)*WTNOD(NP1)*WTNOD(NP2)                              14020
C** ADJUST WEIGHTS FOR OBSERVED HEAD .GT. COMPUTED HEAD                  14030
      N1=N-NUMNP+NOBMAX*(IOBCNT-1)                                      14040
```

Table 2.--*Computer-program listing*--Continued

```
      HOBS=HO(N1)                                                          14050
      H1=H(NP1+NUMNP)                                                      14060
      H2=H(NP2+NUMNP)                                                      14070
      HWT1=1.0                                                             14080
      HWT2=1.0                                                             14090
      IF (H1.GE.HOBS.AND.H2.GE.HOBS) GO TO 269                             14100
      IF (H1.GE.HOBS.AND.H2.LT.HOBS) HWT2=(HOBS-H1)/(H2-H1)                14110
      IF (H1.LT.HOBS.AND.H2.GE.HOBS) HWT1=(HOBS-H2)/(H1-H2)                14120
      IF (HWT1.GE.1.0.AND.HWT2.GE.1.0) GO TO 314                           14130
269   DO 270 L=1,2                                                         14140
      NPL=LINOD(K,L)                                                       14150
      NPL=NP(I,NPL)                                                        14160
      NPK=2                                                                14170
      IF (L.EQ.1) GO TO 272                                                14180
      NPK=1                                                                14190
      HWT1=HWT2                                                            14200
      H1=H2                                                                14210
272   NPK=LINOD(K,NPK)                                                     14220
      NPK=NP(I,NPK)                                                        14230
      SFOBSC(NPL)=SFOBSC(NPL)+RLWT*RLF(1,L)*DETL*WTNOD(NPL)*HWT1*          14240
     $            (RXJ+RYJ)*(H1-HOBS)                                      14250
      SFOBSC(NPK)=SFOBSC(NPK)+RLWT*RLF(1,L)*DETL*(1.0-WTNOD(NPL))*HWT1*    14260
     $            (RXJ+RYJ)*(H1-HOBS)                                      14270
      SFOBSC(N)=SFOBSC(N)+RLWT*RLF(1,L)*DETL*HWT1*                         14280
     $            (RXJ+RYJ)*(H1-HOBS)                                      14290
270   CONTINUE                                                             14300
280   CONTINUE                                                             14310
C** END OF LINE INTEGRATION                                                14320
314   DO 330 K=1,NODES                                                     14330
      NPK=NP(I,K)                                                          14340
      I1=NPK+NUMNP                                                         14350
      I2=NPK                                                               14360
      I4=NPK+NUMNP                                                         14370
      I5=I4-NUMNP                                                          14380
319   TMP=WT(J)*SF(1,K,J)*RMASSJ*CEQUI*DETJ                                14390
      QEC(K)=QEC(K)+TMP                                                    14400
322   CONTINUE                                                             14410
C** FRACTURE SOURCE-SINK TERM                                              14420
100   IF (IFRAC.EQ.0.OR.PORBJ.EQ.0.0) GO TO 199                            14430
      PFRACJ=FP(MATI)                                                      14440
      QBLOCJ=FQ(MATI)                                                      14450
      PORJ=RPR(MATI)                                                       14460
      CB=0.0                                                               14470
      DTMP=DTIME                                                           14480
      TMPSUM=TIMSUM                                                        14490
      PI2=9.869604404                                                      14500
      ITI=NTFRAC+2                                                         14510
      I1=NTFRAC                                                            14520
      IF (KT.LT.NTFRAC) I1=KT                                              14530
      IF (I1.LT.2) GO TO 105                                               14540
C     TIME STEPS 1 THROUGH M-1                                             14550
      DO 103 IF1=1,I1                                                      14560
```

Table 2.--*Computer-program listing*--Continued

```
      ITI=ITI-1                                                          14570
      ITIM1=ITI-1                                                        14580
      TMPSUM=TMPSUM-DTMP                                                 14590
      DTMP=DTMP/TIMFAC                                                   14600
      TMP=0.0                                                            14610
      DO 101 IF2=1,ITFRAC                                                14620
      EN=(2.0*(IF2-1)+1)**2                                              14630
      BN=DM(MATI)*EN*PI2/QBLOCJ**2                                       14640
      TMP1=(EXP(-BN*(TIMSUM-TMPSUM))-EXP(-BN*(TIMSUM-TMPSUM+DTMP)))      14650
     $     /EN/BN                                                        14660
      TMP=TMP-TMP1                                                       14670
      TMP1=(EXP(-BN*(TIMSUM-DTIME-TMPSUM))-EXP(-BN*(TIMSUM-DTIME-TMPSUM  14680
     $     +DTMP)))/EN/BN                                                14690
      TMP=TMP+TMP1                                                       14700
101   CONTINUE                                                           14710
      C1=CFRAC(ITI,NPK)                                                  14720
      C0=CFRAC(ITIM1,NPK)                                                14730
      CB=CB+(C1-C0)*8.0*TMP/PI2/DTMP                                     14740
103   CONTINUE                                                           14750
C     INITIAL TERM                                                       14760
105   TMP=0.0                                                            14770
      DO 107 IF2=1,ITFRAC                                                14780
      EN=(2.0*(IF2-1)+1)**2                                              14790
      BN=DM(MATI)*EN*PI2/QBLOCJ**2                                       14800
      TMP1=EXP(-BN*TIMSUM)/EN                                            14810
      TMP=TMP-TMP1                                                       14820
      TMP1=EXP(-BN*(TIMSUM-DTIME))/EN                                    14830
      TMP=TMP+TMP1                                                       14840
107   CONTINUE                                                           14850
      C0=CINIT(NPK)                                                      14860
      CB=CB+C0*8.0*TMP/PI2                                               14870
C      TIME STEP M                                                       14880
110   NTF1=NTFRAC+1                                                      14890
      TMP=0.0                                                            14900
      DO 111 IF2=1,ITFRAC                                                14910
      EN=(2.0*(IF2-1)+1)**2                                              14920
      BN=DM(MATI)*EN*PI2/QBLOCJ**2                                       14930
      TMP=TMP+(1.0-EXP(-BN*DTIME))/EN/BN                                 14940
111   CONTINUE                                                           14950
      CB=CB-(1.0-8.0*TMP/PI2/DTIME)*CFRAC(NTF1,NPK)                      14960
      DCFRA=PORBJ*QBLOCJ*(1.0-8.0*TMP/DTIME/PI2)/DTIME/(PFRACJ+QBLOCJ)   14970
109   QXYFRA=-PORBJ*QBLOCJ*CB/DTIME/(PFRACJ+QBLOCJ)                      14980
190   RN=WT(J)*SF(1,K,J)*DETJ                                            14990
      RNN=WT(J)*SF(1,K,J)*SF(1,K,J)*DETJ                                 15000
      QEC(K)=QEC(K)+RN*QXYFRA                                            15010
      DC(K,K)=DC(K,K)+RNN*NODES*DCFRA                                    15020
199   CONTINUE                                                           15030
C** FILL ELEMENT MATRICES                                                15040
      DO 329 L=1,NODES                                                   15050
317   NPL=NP(I,L)+NUMNP                                                  15060
      NPLM1=NPL-NUMNP                                                    15070
      J1=NP(I,L)+NUMNP                                                   15080
```

Table 2.--*Computer-program listing*--Continued

```
      J2=J1-NUMNP                                                          15090
      RNN=WT(J)*SF(1,K,J)*SF(1,L,J)*DETJ                                   15100
      DXH=WT(J)*DETJ*DNDX(K)*DNDX(L)                                       15110
      DYH=WT(J)*DETJ*DNDY(K)*DNDY(L)                                       15120
      DZH=WT(J)*DETJ*DNDZ(K)*DNDZ(L)                                       15130
      DXYC=WT(J)*DETJ*DNDX(K)*DNDY(L)                                      15140
      DXZC=WT(J)*DETJ*DNDX(K)*DNDZ(L)                                      15150
      DYZC=WT(J)*DETJ*DNDY(K)*DNDZ(L)                                      15160
      BSIGN=1.0                                                            15170
      KB=K                                                                 15180
      LB=L                                                                 15190
      IF (NPBC(K,IBCNT).LT.-4.OR.NPBC(L,IBCNT).LT.-4) GO TO 225            15200
      BSIGN=-1.0                                                           15210
      KB=L                                                                 15220
      LB=K                                                                 15230
225   DNXN=WT(J)*DETJ*DNDX(KB)*SF(1,LB,J)                                  15240
      DNYN=WT(J)*DETJ*DNDY(KB)*SF(1,LB,J)                                  15250
      DNZN=WT(J)*DETJ*DNDZ(KB)*SF(1,LB,J)                                  15260
      RNLB=WT(J)*DETJ*SF(1,LB,J)                                           15270
      RNKB=WT(J)*DETJ*SF(1,KB,J)                                           15280
318   IF (ALFLG.LE.0.0) GO TO 150                                          15290
      ALFAX=ABS(ALFAX)                                                     15300
      IF (VXJ.LT.0.0) ALFAX=-ALFAX                                         15310
      ALFAY=ABS(ALFAY)                                                     15320
      IF (VYJ.LT.0.0) ALFAY=-ALFAY                                         15330
      ALFAZ=ABS(ALFAZ)                                                     15340
      IF (VZJ.LT.0.0) ALFAZ=-ALFAZ                                         15350
      GO TO 197                                                            15360
150   PECNUM=ABS(VXJ*(XORD(NPL)-XORD(NPK))/DSPXJ)                          15370
      ALFAX=0.0                                                            15380
      IF (PECNUM.GE.2.0) ALFAX=1.0/TANH(PECNUM/2.0)-2.0/PECNUM             15390
      IF (VXJ.LT.0.0) ALFAX=-ALFAX                                         15400
      PECNUM=ABS(VYJ*(YORD(NPL)-YORD(NPK))/DSPYJ)                          15410
      ALFAY=0.0                                                            15420
      IF (PECNUM.GE.2.0) ALFAY=1.0/TANH(PECNUM/2.0)-2.0/PECNUM             15430
      IF (VYJ.LT.0.0) ALFAY=-ALFAY                                         15440
      PECNUM=ABS(VZJ*(ZORD(NPL)-ZORD(NPK))/DSPZJ)                          15450
      ALFAZ=0.0                                                            15460
      IF (PECNUM.GE.2.0) ALFAZ=1.0/TANH(PECNUM/2.0)-2.0/PECNUM             15470
      IF (VZJ.LT.0.0) ALFAZ=-ALFAZ                                         15480
197   TMP=(DXH+ALFAX*DFDX*DNDX(L))*DSPXJ+(DYH+ALFAY*DFDY*DNDY(L))*DSPYJ+   15490
     $    (DZH+ALFAZ*DFDZ*DNDZ(L))*DSPZJ+(DXZC+ALFAX*DFDX*DNDZ(L))*        15500
     $    DSPXZJ+(DYZC+ALFAY*DFDY*DNDZ(L))*DSPYZJ+(DXYC+ALFAX*DFDX*        15510
     $    DNDY(L))*DSPXYJ+(DXZC+ALFAZ*DFDZ*DNDX(L))*DSPXZJ+(DYZC+          15520
     $    ALFAZ*DFDZ*DNDY(L))*DSPYZJ+(DXYC+ALFAY*DFDY*DNDX(L))*DSPXYJ      15530
      TEMP=((DNXN+ALFAX*DFDX*RNLB)*VXJ+(DNYN+ALFAY*DFDY*RNLB)*VYJ+         15540
     $     (DNZN+ALFAZ*DFDZ*RNLB)*VZJ)*BSIGN                               15550
      DC(K,L)=DC(K,L)+(TMP-TEMP)                                           15560
      EC(K,L)=EC(K,L)+RNN*PORJ+RNN*RMASSJ                                  15570
329   CONTINUE                                                             15580
330   CONTINUE                                                             15590
340   CONTINUE                                                             15600
```

Table 2.--*Computer-program listing*--Continued

```
C** SURFACE QUADRATURE                                                  15610
      IF (MAT(I).GE.0) GO TO 382                                        15620
360   DO 369 J=1,6                                                      15630
      GO TO (362,364,365,366,367,359),J                                 15640
362   K1=NP(I,1)+(ISCNT-1)*NUMNP                                        15650
      K2=NP(I,2)+(ISCNT-1)*NUMNP                                        15660
      K3=NP(I,3)+(ISCNT-1)*NUMNP                                        15670
      K4=NP(I,4)+(ISCNT-1)*NUMNP                                        15680
      NN=4                                                              15690
      GO TO 368                                                         15700
364   K1=NP(I,5)+(ISCNT-1)*NUMNP                                        15710
      K2=NP(I,6)+(ISCNT-1)*NUMNP                                        15720
      K3=NP(I,7)+(ISCNT-1)*NUMNP                                        15730
      K4=NP(I,8)+(ISCNT-1)*NUMNP                                        15740
      NN=4                                                              15750
      GO TO 368                                                         15760
365   K1=NP(I,1)+(ISCNT-1)*NUMNP                                        15770
      K2=NP(I,5)+(ISCNT-1)*NUMNP                                        15780
      K3=NP(I,6)+(ISCNT-1)*NUMNP                                        15790
      K4=NP(I,2)+(ISCNT-1)*NUMNP                                        15800
      NN=4                                                              15810
      GO TO 368                                                         15820
366   K1=NP(I,1)+(ISCNT-1)*NUMNP                                        15830
      K2=NP(I,4)+(ISCNT-1)*NUMNP                                        15840
      K3=NP(I,8)+(ISCNT-1)*NUMNP                                        15850
      K4=NP(I,5)+(ISCNT-1)*NUMNP                                        15860
      NN=4                                                              15870
      GO TO 368                                                         15880
367   K1=NP(I,2)+(ISCNT-1)*NUMNP                                        15890
      K2=NP(I,6)+(ISCNT-1)*NUMNP                                        15900
      K3=NP(I,7)+(ISCNT-1)*NUMNP                                        15910
      K4=NP(I,3)+(ISCNT-1)*NUMNP                                        15920
      NN=4                                                              15930
      GO TO 368                                                         15940
359   K1=NP(I,3)+(ISCNT-1)*NUMNP                                        15950
      K2=NP(I,4)+(ISCNT-1)*NUMNP                                        15960
      K3=NP(I,8)+(ISCNT-1)*NUMNP                                        15970
      K4=NP(I,7)+(ISCNT-1)*NUMNP                                        15980
      NN=4                                                              15990
368   CHKQ=QS(K1)*QS(K2)*QS(K3)*QS(K4)                                  16000
      IF (CHKQ.EQ.0.0) GO TO 369                                        16010
      KARG(1)=K1                                                        16020
      KARG(2)=K2                                                        16030
      KARG(3)=K3                                                        16040
      KARG(4)=K4                                                        16050
      CALL SURINT(I,NODES,KARG,NN,NPARH,1,ISCNT,NUMNP)                  16060
369   CONTINUE                                                          16070
C** LUMP CAPACITY MATRIX ON DIAGONAL                                    16080
      DO 376 J=1,NODES                                                  16090
      DO 376 K=1,NODES                                                  16100
      IF (J.EQ.K) EC(J,K)=EC(J,K)*8.0                                   16110
      IF (J.NE.K) EC(J,K)=0.0                                           16120
```

Table 2.--*Computer-program listing*--Continued

```
376   CONTINUE                                                          16130
C**  PLACE IN LARGE CPKC AND Q MATRICES                                 16140
382   DO 386 J=1,NODES                                                  16150
      NPJ=NP(I,J)                                                       16160
      QC(NPJ)=QC(NPJ)+QEC(J)                                            16170
      DO 384 K=1,NODES                                                  16180
      NPK=NP(I,K)                                                       16190
      K2=NPK-NPJ+IDIAG                                                  16200
      QC(NPJ)=QC(NPJ)+(EC(J,K)/DTIME-(1.0-THETAC)*DC(J,K))*C(NPK)       16210
383   IF (LU.EQ.1) CPKC(NPJ,K2)=CPKC(NPJ,K2)+EC(J,K)/DTIME              16220
     $             +THETAC*DC(J,K)                                      16230
384   CONTINUE                                                          16240
386   CONTINUE                                                          16250
399   CONTINUE                                                          16260
C**  BOUNDRY CONDITIONS                                                16270
      DO 420 I=1,NUMNP                                                  16280
      IF (NPBC(I,IBCNT).GE.-1.OR.NPBC(I,IBCNT).LE.-4) GO TO 420         16290
      IF (LU.EQ.1) CPKC(I,IDIAG)=1.0D+50                                16300
      I1=I+NUMNP                                                        16310
      QC(I)=C(I1)*1.0D+50                                               16320
420   CONTINUE                                                          16330
      IF (NOBMAX.LE.0) RETURN                                           16340
C**  FINISH COMPUTATION OF SFOBSC                                      16350
      DO 294 I=1,NUMNP                                                  16360
      N=NUMNP+NODOBS(I)                                                 16370
      IF (N.EQ.NUMNP) GO TO 294                                         16380
      SFOBSC(I)=SFOBSC(I)/SFOBSC(N)                                     16390
294   CONTINUE                                                          16400
      RETURN                                                            16410
7001  WRITE (6,17) I,J                                                  16420
17    FORMAT ('0ELEMENT',I5,' HAS NEGATIVE OR ZERO DETJ AT ',           16430
     $        'QUADRATURE POINT',I5)                                    16440
      STOP                                                              16450
      END                                                               16460
C**  PERFORM SURFACE INTEGRATION                                       16470
      SUBROUTINE SURINT(I,N,L,N1,NPARH,IHCFLG,ISCNT,NUMNP)              16480
      COMMON/SINBLK/QEH(8),QEC(8),XORD( 490),YORD( 490),ZORD( 490),     16490
     $      QS( 980),CS( 980),QF(1),CF(1),NP(300,8),IBPRM(300)          16500
      COMMON/SENBLK/WH(30),                                             16510
     $      WC(30),THETAC,NPBC( 490,1),NODOBS( 490)                     16520
      COMMON/SURBLK/AF(3,4),AWT,NQPTA                                   16530
      COMMON/LDUBLK/CPKC(490,180),CPKH(490, 90),SH(8, 490),             16540
     $              QC( 490),QH( 490),IBC,IBH                           16550
      REAL*8 CPKC,CPKH,SH,QC,QH                                         16560
      DOUBLE PRECISION QEH,QEC,AF,AWT,TEMP,RJAC,DETJ,TMP                16570
      DIMENSION RJAC(3,2),L(4)                                          16580
      N3=IBPRM(I)                                                       16590
      N2=N3+NPARH                                                       16600
      RJAC(1,1)=0.0                                                     16610
      RJAC(1,2)=0.0                                                     16620
      RJAC(2,1)=0.0                                                     16630
      RJAC(2,2)=0.0                                                     16640
```

Table 2.--*Computer-program listing*--Continued

```
      RJAC(3,1)=0.0                                                    16650
      RJAC(3,2)=0.0                                                    16660
      QSJ=0.0                                                          16670
      CSJ=0.0                                                          16680
      DO 310 K=1,N1                                                    16690
      N4=L(K)                                                          16700
      NPK=N4-(ISCNT-1)*NUMNP                                           16710
      RJAC(1,1)=RJAC(1,1)+AF(2,K)*XORD(NPK)                            16720
      RJAC(1,2)=RJAC(1,2)+AF(3,K)*XORD(NPK)                            16730
      RJAC(2,1)=RJAC(2,1)+AF(2,K)*YORD(NPK)                            16740
      RJAC(2,2)=RJAC(2,2)+AF(3,K)*YORD(NPK)                            16750
      RJAC(3,1)=RJAC(3,1)+AF(2,K)*ZORD(NPK)                            16760
      RJAC(3,2)=RJAC(3,2)+AF(3,K)*ZORD(NPK)                            16770
      TMP=QS(N4)                                                       16780
      IF (IHCFLG.EQ.1) TEMP=CS(N4)                                     16790
      IF (N3.GT.0) TMP=TMP*QF(N3)                                      16800
      IF (N3.GT.0.AND.IHCFLG.EQ.1) TEMP=TEMP*CF(N3)                    16810
      QSJ=QSJ+AF(1,K)*TMP                                              16820
      IF (IHCFLG.EQ.1) CSJ=CSJ+AF(1,K)*TEMP                            16830
310   CONTINUE                                                         16840
      DETJ=RJAC(1,1)*(RJAC(2,2)-RJAC(3,2))                             16850
     $    -RJAC(2,1)*(RJAC(1,2)-RJAC(3,2))                            16860
     $    +RJAC(3,1)*(RJAC(1,2)-RJAC(2,2))                            16870
      IF (DETJ.LE.0.0) DETJ=-DETJ                                      16880
324   DO 330 K=1,N1                                                    16890
      DO 327 II=1,N                                                    16900
      IF (NP(I,II).NE.L(K)-(ISCNT-1)*NUMNP) GO TO 327                  16910
      K1=II                                                            16920
      GO TO 328                                                        16930
327   CONTINUE                                                         16940
328   NPK=L(K)-(ISCNT-1)*NUMNP                                         16950
      TMP=AWT*AF(1,K)*QSJ*DETJ                                         16960
      IF (IHCFLG.EQ.1) GO TO 326                                       16970
      IF (N2.GT.NPARH)SH(N2,NPK)=SH(N2,NPK)+TMP                        16980
      QEH(K1)=QEH(K1)+TMP                                              16990
      GO TO 330                                                        17000
326   IF (TMP.LT.0.0) GO TO 330                                        17010
      QEC(K1)=QEC(K1)+CSJ*TMP                                          17020
330   CONTINUE                                                         17030
      RETURN                                                           17040
      END                                                              17050
      SUBROUTINE SENS (NVARH,NUMNP,IPO,KT,KTH,IOBCNT,                  17060
     $                 IXTRFL,NOBMAX,IBCNT)                            17070
C** SUBROUTINE TO SETUP AND SOLVE FOR SCALED SENSITIVITIES              17080
      COMMON/SENBLK/WH(30),                                            17090
     $      WC(30),THETAC,NPBC( 490,1),NODOBS( 490)                    17100
      COMMON/LDUBLK/CPKC(490,180),CPKH(490, 90),SH(8, 490),            17110
     $              QC( 490),QH( 490),IBC,IBH                          17120
      REAL*8 CPKC,CPKH,SH,QC,QH                                        17130
      IF (NVARH.LE.0) RETURN                                           17140
C** SOLVE FOR FLOW SENSITIVITIES                                       17150
      DO 224 I=1,NVARH                                                 17160
```

Table 2.--*Computer-program listing*--Continued

```
      DO 224 J=1,NUMNP                                                  17170
      IF (NPBC(J,IBCNT).EQ.-1) SH(I,J)=0.0                              17180
      IF (NPBC(J,IBCNT).EQ.-3) SH(I,J)=0.0                              17190
      IF (NPBC(J,IBCNT).EQ.-5) SH(I,J)=0.0                              17200
224   CONTINUE                                                          17210
      NEM1=NUMNP-1                                                      17220
      NUMEQ=NUMNP                                                       17230
      DO 350 I=1,NEM1                                                   17240
      JEND=NUMEQ-I+1                                                    17250
      IF(JEND.GT.IBH) JEND=IBH                                          17260
      DO 340 J=2,JEND                                                   17270
      J1=I+J-1                                                          17280
      QH(J1)=QH(J1)-QH(I)*CPKH(I,J)                                     17290
      DO 338 K=1,NVARH                                                  17300
      SH(K,J1)=SH(K,J1)-SH(K,I)*CPKH(I,J)                               17310
338   CONTINUE                                                          17320
340   CONTINUE                                                          17330
350   CONTINUE                                                          17340
C** BACK SUBSTITUTION                                                   17350
      QH(NUMEQ)=QH(NUMEQ)/CPKH(NUMEQ,1)                                 17360
      DO 352 K=1,NVARH                                                  17370
      SH(K,NUMEQ)=SH(K,NUMEQ)/CPKH(NUMEQ,1)                             17380
352   CONTINUE                                                          17390
353   DO 370 I=1,NEM1                                                   17400
      I1=NUMEQ-I                                                        17410
      QH(I1)=QH(I1)/CPKH(I1,1)                                          17420
      DO 354 K=1,NVARH                                                  17430
      SH(K,I1)=SH(K,I1)/CPKH(I1,1)                                      17440
354   CONTINUE                                                          17450
355   JEND=NUMEQ-I1+1                                                   17460
      IF(JEND.GT.IBH) JEND=IBH                                          17470
      DO 360 J=2,JEND                                                   17480
      J1=I1+J-1                                                         17490
      QH(I1)=QH(I1)-CPKH(I1,J)*QH(J1)                                   17500
      DO 356 K=1,NVARH                                                  17510
      SH(K,I1)=SH(K,I1)-CPKH(I1,J)*SH(K,J1)                             17520
356   CONTINUE                                                          17530
360   CONTINUE                                                          17540
370   CONTINUE                                                          17550
      IF (IPO.NE.1) GO TO 232                                           17560
      WRITE (6,1) KT                                                    17570
      DO 230 I=1,NUMNP                                                  17580
      J=NODOBS(I)                                                       17590
      IF (J.LE.0) GO TO 230                                             17600
      J=NOBMAX*(IOBCNT-1)+J                                             17610
      WRITE (6, 2)I,(SH(K,I),K=1,NVARH)                                 17620
230   CONTINUE                                                          17630
232   CONTINUE                                                          17640
      RETURN                                                            17650
1     FORMAT (' SCALED FLOW SENSITIVITIES - TIME STEP ',I5/'  NODE',    17660
     $        20X,'PARAMETER')                                          17670
2     FORMAT (1X,I5,7(1X,G11.5)/6X,7(1X,G11.5))                         17680
```

Table 2.--*Computer-program listing*--Continued

```
      END                                                                    17690
      SUBROUTINE LSTSQ (C,B,RK,P,YSQ,AP,AMP,RP,RPF,NVAR,NVX2,NVX3,IPO,       17700
     $                  INDT,KOUNT,NVE,LDIM)                                17710
      DIMENSION P(NVE),RK(NVE),B(LDIM),C(NVE,NVE)                            17720
C** CHECK FOR NONZERO MARQUARDT PARAMETER                                    17730
      IF(AMP.LT.-.5) GO TO 105                                               17740
   74 IF (NVAR.EQ.1) GO TO 190                                               17750
      DO 80 I=1,NVAR                                                         17760
      TEMP=C(I,I)+RK(I)                                                      17770
      IF(TEMP.GT.1.E-10) GO TO 78                                            17780
      WRITE(6,260) I                                                         17790
      INDT=1                                                                 17800
      GO TO 80                                                               17810
   78 C(I,I)=TEMP**.5                                                        17820
   80 CONTINUE                                                               17830
      IF(INDT.GT.0) RETURN                                                   17840
      NM1=NVAR-1                                                             17850
      DO 100 J=1,NM1                                                         17860
      TEMP=C(J,J)                                                            17870
      JP1=J+1                                                                17880
      DO 90 I=JP1,NVAR                                                       17890
      C(I,J)=C(I,J)/(C(I,I)*TEMP)                                            17900
   90 C(J,I)=C(I,J)                                                          17910
      B(J+NVX2)=(B(J+NVX2)+RK(J)*(P(J)-1.))/TEMP+RP*TEMP*(RPF*P(J)-1.)       17920
      B(J)=B(J+NVX2)                                                         17930
      B(J+NVX3)=TEMP                                                         17940
  100 C(J,J)=1.+RP+AMP                                                       17950
      TEMP=C(NVAR,NVAR)                                                      17960
      B(NVX3)=(B(NVX3)+RK(NVAR)*(P(NVAR)-1.))/TEMP                           17970
     1+RP*TEMP*(RPF*P(NVAR)-1.)                                              17980
      B(NVAR)=B(NVX3)                                                        17990
      B(NVAR+NVX3)=TEMP                                                      18000
      C(NVAR,NVAR)=1.+RP+AMP                                                 18010
      IF(IPO.NE.1) GO TO 105                                                 18020
      WRITE(6,250)                                                           18030
      CALL PRTOT(C,NVE,0,NVAR)                                               18040
      WRITE(6,230) (B(I+NVX2),I=1,NVAR)                                      18050
C** SOLVE FOR B USING LDU FACTORIZATION:                                     18060
C** DECOMPOSITION AND FORWARD SUBSTITUTION                                   18070
  105 DET=1.                                                                 18080
      DO 140 K=1,NM1                                                         18090
      PIV=C(K,K)                                                             18100
      DET=DET*PIV                                                            18110
      IF(ABS(PIV).GT.1.E-10) GO TO 110                                       18120
      WRITE(6,210)                                                           18130
      INDT=1                                                                 18140
      RETURN                                                                 18150
  110 PIV=1./PIV                                                             18160
      KP1=K+1                                                                18170
      DO 130 J=KP1,NVAR                                                      18180
      TMP=C(J,K)*PIV                                                         18190
      DO 120 I=J,NVAR                                                        18200
```

Table 2.--*Computer-program listing*--Continued

```
  120 C(I,J)=C(I,J)-TMP*C(I,K)                                          18210
  130 B(J)=B(J)-TMP*B(K)                                                18220
      C(K,K)=PIV                                                        18230
  140 CONTINUE                                                          18240
      DET=DET*C(NVAR,NVAR)                                              18250
      IF(ABS(C(NVAR,NVAR)).GT.1.E-10) GO TO 150                         18260
      WRITE(6,210)                                                      18270
      INDT=1                                                            18280
      RETURN                                                            18290
  150 IF(AMP.LT.-.5) RETURN                                             18300
C** BACK SUBSTITUTION                                                   18310
      B(NVX2)=B(NVAR)/C(NVAR,NVAR)                                      18320
      B(NVAR)=B(NVX2)/B(NVAR+NVX3)                                      18330
      I=NVAR                                                            18340
  160 I=I-1                                                             18350
      IF (I.LE.0) GO TO 200                                             18360
      IP1=I+1                                                           18370
      SUM=0.                                                            18380
      DO 170 J=IP1,NVAR                                                 18390
  170 SUM=SUM+C(J,I)*B(J+NVAR)                                          18400
      B(I+NVAR)=(B(I)-SUM)*C(I,I)                                       18410
      B(I)=B(I+NVAR)/B(I+NVX3)                                          18420
      GO TO 160                                                         18430
C** SOLUTION WHEN NVAR=1                                                18440
  190 TEMP=C(1,1)+RK(1)                                                 18450
      IF(TEMP.GT.1.E-10) GO TO 195                                      18460
      I=1                                                               18470
      WRITE(6,260) I                                                    18480
      INDT=1                                                            18490
      RETURN                                                            18500
  195 B(4)=TEMP**.5                                                     18510
      B(3)=(B(3)+RK(1)*(P(1)-1.))/B(4)+RP*B(4)*(RPF*P(1)-1.)            18520
      C(1,1)=1.+RP+AMP                                                  18530
      DET=C(1,1)                                                        18540
      B(2)=B(3)/C(1,1)                                                  18550
      B(1)=B(2)/B(4)                                                    18560
C** ADJUST AND PRINT REGRESSION COEFFICIENTS                            18570
  200 WRITE(6,220) KOUNT,YSQ,DET                                        18580
      DO 202 J=1,NVAR                                                   18590
  202 B(J)=AP*B(J)                                                      18600
      WRITE(6,230) (B(J),J=1,NVAR)                                      18610
      RETURN                                                            18620
  210 FORMAT (43HOLEAST SQUARES COEFFICIENT MATRIX SINGULAR;/35H SOLUTIO 18630
     1N FOR PARAMETERS NOT UNIQUE)                                      18640
  220 FORMAT (1HO,14HITERATION NO. ,I3/1H ,6HYSQ = ,E12.5               18650
     1,2X,9HDET(C) = ,E12.5/1H ,24HREGRESSION COEFFICIENTS:)            18660
  230 FORMAT ((1H ,8(E12.5,2X)))                                        18670
  250 FORMAT(49HO SCALED LEAST SQUARES MATRIX AND GRADIENT VECTOR)      18680
  260 FORMAT (29HOSENSITIVITIES FOR PARAMETER ,I4,17H EFFECTIVELY ZERO) 18690
      END                                                               18700
      SUBROUTINE PRTOT(C,NO,IT,NUM)                                     18710
C** IF IT=0, PRINT MATRICES DIVIDED VERTICALLY INTO BETWEEN ONE AND      18720
```

Table 2.--*Computer-program listing*--Continued

```
C**          TEN PARTS                                                  18730
C** IF IT=1, PRINT VECTOR IN THREE COLUMNS                              18740
      DIMENSION C(1)                                                    18750
      IF(IT.EQ.1) GO TO 25                                              18760
      ITMP=(NUM-1)/10+1                                                 18770
      IB=1                                                              18780
      DO 20 IBK=1,ITMP                                                  18790
      INC=IBK*10                                                        18800
      IF(NUM.LT.INC) INC=NUM                                            18810
      WRITE(6,30) (I,I=IB,INC)                                          18820
      WRITE(6,50)                                                       18830
      K=-NO                                                             18840
      DO 10 J=1,NUM                                                     18850
      K=K+NO                                                            18860
   10 WRITE(6,40) J,(C(I+K),I=IB,INC)                                   18870
      WRITE(6,60)                                                       18880
      IB=INC+1                                                          18890
   20 CONTINUE                                                          18900
      RETURN                                                            18910
   25 NR=NO/3                                                           18920
      IF((3*NR).NE.NO) NR=NR+1                                          18930
      DO 26 K=1,NR                                                      18940
   26 WRITE(6,80) (L,C(L),L=K,NO,NR)                                    18950
      RETURN                                                            18960
   30 FORMAT (1H0,8X,I3,9(9X,I3))                                       18970
   40 FORMAT (1H ,I3,10(1X,E11.4))                                      18980
   50 FORMAT (1H )                                                      18990
   60 FORMAT (1H0)                                                      19000
   80 FORMAT(1H ,3X,3(I3,5X,E11.4,5X))                                  19010
      END                                                               19020
      SUBROUTINE UDU (NUMEQ,LU)                                         19030
C** AN LDU DECOMPOSITION FOR BANDED SYMETRIC MATRICES                   19040
      COMMON/LDUBLK/CPKC(490,180),CPKH(490, 90),SH(8, 490),             19050
     $              QC( 490),QH( 490),IBC,IBH                           19060
      REAL*8 CPKC,CPKH,SH,QC,QH,FAC                                     19070
      NEM1=NUMEQ-1                                                      19080
      DO 450 I=1,NEM1                                                   19090
      JEND=NUMEQ-I+1                                                    19100
      IF(JEND.GT.IBH) JEND=IBH                                          19110
      DO 440 J=2,JEND                                                   19120
      J1=I+J-1                                                          19130
      IF(LU.EQ.0) GO TO 435                                             19140
      FAC=CPKH(I,J)/CPKH(I,1)                                           19150
429   K1=0                                                              19160
      DO 430 K=J,JEND                                                   19170
      K1=K1+1                                                           19180
      CPKH(J1,K1)=CPKH(J1,K1)-CPKH(I,K)*FAC                             19190
  430 CONTINUE                                                          19200
      CPKH(I,J)=FAC                                                     19210
  435 CONTINUE                                                          19220
      QH(J1)=QH(J1)-QH(I)*CPKH(I,J)                                     19230
  440 CONTINUE                                                          19240
```

Table 2.--*Computer-program listing*--Continued

```
  450 CONTINUE                                                          19250
C     BACK SUBSTITUTION                                                 19260
      QH(NUMEQ)=QH(NUMEQ)/CPKH(NUMEQ,1)                                 19270
      DO 470 I=1,NEM1                                                   19280
      I1=NUMEQ-I                                                        19290
      QH(I1)=QH(I1)/CPKH(I1,1)                                          19300
      JEND=NUMEQ-I1+1                                                   19310
      IF(JEND.GT.IBH) JEND=IBH                                          19320
      DO 460 J=2,JEND                                                   19330
      J1=I1+J-1                                                         19340
      QH(I1)=QH(I1)-CPKH(I1,J)*QH(J1)                                   19350
  460 CONTINUE                                                          19360
  470 CONTINUE                                                          19370
      RETURN                                                            19380
      END                                                               19390
C                                                                       19400
      SUBROUTINE LDU (NUMEQ,LU)                                         19410
C** AN LDU DECOMPOSITION FOR BANDED NONSYMETRIC MATRICES                19420
      COMMON /LDUBLK/ CPKC(490,180),CPKH(490, 90),SH(8, 490),           19430
     $                QC( 490),QH( 490),IBC,IBH                         19440
      REAL*8 CPKC,CPKH,SH,QC,QH,FAC                                     19450
      NEM1=NUMEQ-1                                                      19460
      IBD2=(IBC-1)/2                                                    19470
      IDIAG=IBD2+1                                                      19480
      DO 450 I=1,NEM1                                                   19490
      JEND=NUMEQ-I                                                      19500
      IF(JEND.GT.IBD2) JEND=IBD2                                        19510
      DO 440 J=1,JEND                                                   19520
      JROW=I+J                                                          19530
      JCOL=IBD2-J+1                                                     19540
      IF(LU.EQ.0) GO TO 435                                             19550
      FAC=CPKC(JROW,JCOL)/CPKC(I,IDIAG)                                 19560
      CPKC(JROW,JCOL)=FAC                                               19570
      DO 430 K=1,JEND                                                   19580
      K1=IDIAG+K                                                        19590
      K2=JCOL+K                                                         19600
      CPKC(JROW,K2)=CPKC(JROW,K2)-FAC*CPKC(I,K1)                        19610
  430 CONTINUE                                                          19620
  435 CONTINUE                                                          19630
      QC(JROW)=QC(JROW)-CPKC(JROW,JCOL)*QC(I)                           19640
  440 CONTINUE                                                          19650
  450 CONTINUE                                                          19660
C     BACK SUBSTITUTION                                                 19670
      QC(NUMEQ)=QC(NUMEQ)/CPKC(NUMEQ,IDIAG)                             19680
      JBGN=IDIAG+1                                                      19690
      DO 470 I=1,NEM1                                                   19700
      IROW=NUMEQ-I                                                      19710
      JEND=IDIAG+I                                                      19720
      IF(JEND.GT.IBC) JEND=IBC                                          19730
      JROW=IROW                                                         19740
      DO 460 J=JBGN,JEND                                                19750
      JROW=JROW+1                                                       19760
```

Table 2.--*Computer-program listing*--Continued

```
      QC(IROW)=QC(IROW)-CPKC(IROW,J)*QC(JROW)                          19770
  460 CONTINUE                                                         19780
      QC(IROW)=QC(IROW)/CPKC(IROW,IDIAG)                               19790
  470 CONTINUE                                                         19800
      RETURN                                                           19810
      END                                                              19820
C                                                                      19830
C** SHAPE FACTOR SUBROUTINE                                            19840
C** FOR THREE DIMENSIONAL CUBIC ELEMENTS                              19850
C** SF(FUNCTION,NODAL POINT,QUADRATURE POINT)                          19860
      SUBROUTINE SHAFAC(THETAC)                                        19870
      DIMENSION SF(4,8,8),WF(4,8),WT(8)                                19880
      DIMENSION QPT(8,3)                                               19890
      COMMON /SHABLK/ SF,WF,WT,NUMQPT                                  19900
      DOUBLE PRECISION SF,WF,WT,QPT,A,B,C                              19910
      NUMQPT=8                                                         19920
      WT(1)=1.0/8.0                                                    19930
      WT(2)=1.0/8.0                                                    19940
      WT(3)=WT(2)                                                      19950
      WT(4)=WT(2)                                                      19960
      WT(5)=WT(2)                                                      19970
      WT(6)=WT(2)                                                      19980
      WT(7)=WT(2)                                                      19990
      WT(8)=WT(2)                                                      20000
      QPT(1,1)=0.211324865405187                                       20010
      QPT(1,2)=0.211324865405187                                       20020
      QPT(1,3)=0.211324865405187                                       20030
      QPT(2,1)=0.788675134594813                                       20040
      QPT(2,2)=QPT(1,1)                                                20050
      QPT(2,3)=QPT(1,3)                                                20060
      QPT(3,1)=QPT(1,1)                                                20070
      QPT(3,2)=QPT(2,1)                                                20080
      QPT(3,3)=QPT(1,3)                                                20090
      QPT(4,1)=QPT(2,1)                                                20100
      QPT(4,2)=QPT(2,1)                                                20110
      QPT(4,3)=QPT(1,3)                                                20120
      QPT(5,1)=QPT(1,1)                                                20130
      QPT(5,2)=QPT(1,2)                                                20140
      QPT(5,3)=0.788675134594813                                       20150
      QPT(6,1)=QPT(2,1)                                                20160
      QPT(6,2)=QPT(2,2)                                                20170
      QPT(6,3)=QPT(5,3)                                                20180
      QPT(7,1)=QPT(3,1)                                                20190
      QPT(7,2)=QPT(3,2)                                                20200
      QPT(7,3)=QPT(5,3)                                                20210
      QPT(8,1)=QPT(4,1)                                                20220
      QPT(8,2)=QPT(4,2)                                                20230
      QPT(8,3)=QPT(5,3)                                                20240
      DO 200 I=1,NUMQPT                                                20250
      A=QPT(I,1)                                                       20260
      B=QPT(I,2)                                                       20270
      C=QPT(I,3)                                                       20280
```

Table 2.--*Computer-program listing*--Continued

```
      SF(1,1,I)=(1.0-A)*(B-1.0)*(C-1.0)                                  20290
      SF(1,2,I)=A*(B-1.0)*(C-1.0)                                        20300
      SF(1,3,I)=A*B*(1.0-C)                                              20310
      SF(1,4,I)=B*(A-1.0)*(C-1.0)                                        20320
      SF(1,5,I)=C*(A-1.0)*(B-1.0)                                        20330
      SF(1,6,I)=A*C*(1.0-B)                                              20340
      SF(1,7,I)=A*B*C                                                    20350
      SF(1,8,I)=B*C*(1.0-A)                                              20360
      SF(2,1,I)=(1.0-B)*(C-1.0)                                          20370
      SF(2,2,I)=(B-1.0)*(C-1.0)                                          20380
      SF(2,3,I)=B*(1.0-C)                                                20390
      SF(2,4,I)=B*(C-1.0)                                                20400
      SF(2,5,I)=C*(B-1.0)                                                20410
      SF(2,6,I)=C*(1.0-B)                                                20420
      SF(2,7,I)=B*C                                                      20430
      SF(2,8,I)=(-1.0)*B*C                                               20440
      SF(3,1,I)=(1.0-A)*(C-1.0)                                          20450
      SF(3,2,I)=A*(C-1.0)                                                20460
      SF(3,3,I)=(1.0-C)*A                                                20470
      SF(3,4,I)=(A-1.0)*(C-1.0)                                          20480
      SF(3,5,I)=C*(A-1.0)                                                20490
      SF(3,6,I)=(-1.0)*A*C                                               20500
      SF(3,7,I)=A*C                                                      20510
      SF(3,8,I)=C*(1.0-A)                                                20520
      SF(4,1,I)=(1.0-A)*(B-1.0)                                          20530
      SF(4,2,I)=A*(B-1.0)                                                20540
      SF(4,3,I)=(-1.0)*A*B                                               20550
      SF(4,4,I)=B*(A-1.0)                                                20560
      SF(4,5,I)=(A-1.0)*(B-1.0)                                          20570
      SF(4,6,I)=(-1.0)*A*(B-1.0)                                         20580
      SF(4,7,I)=A*B                                                      20590
      SF(4,8,I)=B*(1.0-A)                                                20600
      IF (THETAC.LT.0.0) GO TO 200                                       20610
      WF(1,I)=-108.0*A*B*C*(A-1.0)*(B-1.0)*(C-1.0)                       20620
      WF(2,I)=-108.0*B*C*(B-1.0)*(C-1.0)*(2.0*A-1.0)                     20630
      WF(3,I)=-108.0*A*C*(A-1.0)*(C-1.0)*(2.0*B-1.0)                     20640
      WF(4,I)=-108.0*A*B*(A-1.0)*(B-1.0)*(2.0*C-1.0)                     20650
200   CONTINUE                                                           20660
      RETURN                                                             20670
      END                                                                20680
C** TWO DIMENSIONAL SHAPE FACTOR ROUTINE                                 20690
      SUBROUTINE SURFAC                                                  20700
      COMMON/SURBLK/ AF(3,4),AWT,NQPTA                                   20710
      DOUBLE PRECISION AF,AWT                                            20720
      NQPTA=1                                                            20730
      AF(1,1)=0.25                                                       20740
      AF(1,2)=0.25                                                       20750
      AF(1,3)=0.25                                                       20760
      AF(1,4)=0.25                                                       20770
      AF(2,1)=-0.5                                                       20780
      AF(2,2)=+0.5                                                       20790
      AF(2,3)=+0.5                                                       20800
```

Table 2.--*Computer-program listing*--Continued

```
AF(2,4)=-0.5                                                    20810
AF(3,1)=-0.5                                                    20820
AF(3,2)=-0.5                                                    20830
AF(3,3)=+0.5                                                    20840
AF(3,4)=+0.5                                                    20850
AWT=1.0                                                         20860
RETURN                                                          20870
END                                                             20880
SUBROUTINE LINFAC                                               20890
COMMON/LINBLK/RLF(2,2),RLWT,LINOD(4,2)                          20900
LINOD(1,1)=1                                                    20910
LINOD(1,2)=5                                                    20920
LINOD(2,1)=2                                                    20930
LINOD(2,2)=6                                                    20940
LINOD(3,1)=3                                                    20950
LINOD(3,2)=7                                                    20960
LINOD(4,1)=4                                                    20970
LINOD(4,2)=8                                                    20980
RLWT=1.0                                                        20990
RLF(1,1)=0.5                                                    21000
RLF(2,1)=-1.0                                                   21010
RLF(1,2)=0.5                                                    21020
RLF(2,2)=1.0                                                    21030
RETURN                                                          21040
END                                                             21050
```

Table 3.--*Instructions for preparing model data*

Group 1: Title and problem setup

Card	Column	Format	Variable	Description
1-3	1-80	20A4	TITLE	Description of problem.
4	1-5	I5	NUMEL	Number of elements.
	6-10	I5	NUMNP	Number of nodes.
	11-15	I5	NUMAT	Number of material zones.
	16-20	I5	NTIME	Number of times steps in transport problem.
	21-25	I5	NTPER	Number of transport time steps per flow time step.
	26-30	I5	NTB	Number of times when boundary conditions change.
	31-35	I5	NTO	Number of times with observed water-level and concentration data.
	36-40	I5	NOBMAX	Number of observations for a single time step.
	41-45	I5	NTW	Number of times when point source-sinks change.
	46-50	I5	NTS	Number of times when distributed source-sinks change.
	51-55	I5	NPARH	Number of flow-calibration parameters.
	56-50	I5	NQPAR	Number of point and distributed source-sink parameters considered in calibration.
	66-70	I5	NBPAR	Number of specified head and concentration boundaries considered as calibration parameters.
5	1-5	I5	NPLAYR	Maximum number of nodes per horizontal layer.
	6-10	I5	NEPLAY	Maximum number of elements per horizontal layer.
6	1-5	I5	ITMAX	Maximum number of regression interations for steady-state flow problem. If not using regression technique code 0.
	6-10	I5	IVELPR	Code 1 for printout of velocity and dispersion coefficients.
	11-15	I5	IFRAC	Code 1 to use dual-porosity option in transport problems (equation 5).
	16-20	I5	NTFRAC	Number of time steps used in equation 5 summation.
	21-25	I5	ITFRAC	Number of iterations used in equation 5 summation.
	26-30	I5	NLAYER	Maximum number of nodes in any vertical section.
	31-35	I5	IBEALE	Code 1 to compute coefficients for Beale's measure of nonlinearity.

Table 3.--*Instructions for preparing model data*--Continued

Group 1: Title and problem setup--Continued

Card	Column	Fortmat	Variable	Description
7	1-5	I5	NTPRT	Number of transport time steps between printouts.
	6-10	I5	IPO	Code 1 to print out sensitivity coefficients.
	11-15	I5	IPRX	Code 1 for additional printout of regression results.
	16-25	E10.3	AP	Accleration parameter for regression analysis (1.0 if not used).
	26-35	E10.3	AMP	Marquardt parameter for regression analysis (0.0 if not used).
	36-45	E10.3	RP	Ridge parameter for regression analysis (0.0 if not used).
	46-55	E10.3	RPF	Bias parameter for regression analysis (0.0 if not used).
	56-65	E10.3	EVH	Estimated error variance for flow problem (if not known code 0.0).
8	1-10	E10.3	DTIME	Initial time step for transport problem.
	11-20	E10.3	TIMFAC	Time step expansion factor.
	21-30	E10.3	THETAH	θ in equation 2. Code 1.0 if simulating steady state.
	31-40	E10.3	THETAC	θ in equation 4. Code -1.0 if not modeling transport.
	41-50	E10.3	DFW	Molecular-dispersion coefficient within fractures.
	51-60	E10.3	CEQUI	Equilibrium solute concentration in equation 6.
9	1-10	E10.3	ALFLG	Code 0.0 if values for this card are to be computed by the program; otherwise code 1.0.
	11-20	E10.3	ALFAX	Upstream weighting factor in x direction.
	21-30	E10.3	ALFAY	Upstream weighting factor in y direction.
	31-40	E10.3	ALFAZ	Upstream weighting factor in z direction. Upstream weighting is useful in advection-dominated transport problems. For a complete discussion of upstream weights refer to Noorishad and Mehran (1982), and Huyakorn and Nilkuha (1979). For no upstream weighting code ALFAX = ALFAY = ALFAZ = 0.0.
10	1-70	7E10.3	ZSPACE (I), I=1,(NYLAYER-1)	Nodal spacing in vertical direction measured from bottom layer.

Table 3.--*Instructions for preparing model data*--Continued

Group 2: Nodal locations

Card	Column	Format	Variable	Description
--	1-10	I10	I	Node number.
	11-20	I10	N	Number of nodes directly overlying node I (include node I).
	21-30	E10.3	XORD	Distance in x-direction
	31-40	E10.3	YORD	Distance in y-direction.
	41-50	E10.3	ZORD	Distance in z-direction.

Note:
Only nodes in the bottom layer are coded. The location of nodes in other layers is calculated by the program. The x and y coordinates of overlying nodes are the same as those of the bottom node. Vertical spacing is determined by ZSPACE. A total of NPLAYR nodal-location cards are coded.

Group 3: Nodal information for ground-water flow problem

Card	Column	Format	Variable	Description
1	1-10	I10	N	Number of nodal flow cards to follow.
--	1-10	I10	I	Node number.
	11-20	I10	NPBC	Nodal-point boundary condition. Code 0 for no boundary condition, - 1 for specified head-boundary nodes, - 2 for specified concentration-boundary nodes, - 3 for specified head and concentration nodes, - 4 for boundary nodes without dispersive solute flux, or - 5 for boundary nodes with specified head and without dispersive solute flux.
	21-30	I10	NODOBS	Observation number. If node is not located along a well bore code 0.
	31-40	E10.3	WTNOD	Weight between 0.0 and 1.0 used to indicate partial penetration of an observation well. Code 1.0 if the well reaches the node, 0.0 if the well only reaches the overlying or underlying node.
	41-50	E10.3	H	Initial hydraulic head.
	51-60	E10.3	WELL	Initial point source-sink. If WELL is treated as a calibration parameter NPBC is used to represent the parameter number and must be coded greater than zero.
	61-70	E10.3	QS	Initial distributed source-sink.

Table 3.--*Instructions for preparing model data*--Continued

Group 4: Nodal information for transport problem

Card	Column	Format	Variable	Description
1	1-10	I10	N	Number of cards to follow. Do not code this card if only solving a flow problem.
--	1-10	I10	I	Node number.
	11-20	E10.3	C	Initial concentration.
	21-30	E10.3	WELLC	Initial concentration of point source.
	31-40	E10.3	CS	Initial concentration of distributed source.

Group 5: Nodal configuration of each element

Only the basal NEPLAY elements are coded.

Card	Column	Format	Variable	Description
--	1-5	I5	I	Element number.
	6-10	I5	N	Number of elements directly overlying element I (include basal element).
	11-30	4I5	NP	Nodes associated with element I. Only the four basal nodes of element I are coded in counter-clockwise direction, as viewed from above.

Group 6: Aquifer zonal configuration

Card	Column	Format	Variable	Description
1	1-10	I5	N	Numbers of cards to follow.
--	1-10	I10	I	Element number.
	11-20	I10	MAT	Material-zone number associated with element I. Code as the negative of the zone number if a distributed source is defined over the element. If no card is coded for an element, MAT is set equal to the element layer, numbered from the base of the problem.
	21-30	I10	IBPRM	Calibration-parameter number for distributed source-sinks.

Table 3.--*Instructions for preparing model data*--Continued

Group 7: Aquifer parameters for ground-water flow

One card must be coded for each zone.

Card	Column	Format	Variable	Description
--	1-10	I10	I	Material-zone number.
	11-20	E10.3	RX	Hydraulic conductivity in the x-direction.
	21-30	E10.3	RY	Hydraulic conductivity in the y-direction.
	31-40	E10.3	RZ	Hydraulic conductivity in the z-direction.
	41-50	E10.3	RC	Specific storage.

Group 8: Aquifer parameters for solute transport

One card must be coded for each zone, ordered by the zone number. If not solving a transport problem this group is not coded.

Card	Column	Format	Variable	Description
--	1-10	E10.3	DL	Longitudinal dispersivity.
	11-20	E10.3	DT	Transverse dispersivity.
	21-30	E10.3	DM	Molecular-diffusion coefficient within the media blocks. Only used if simulating dual-porosity media.
	31-40	E10.3	RPR	Porosity. If simulating dual-porosity media, porosity of fractures.
	41-50	E10.3	FP	Fracture width.
	51-60	E10.3	FQ	Fracture spacing or block width.
	61-70	E10.3	RMT	Mass-transfer coefficient K^*A in equation 6. Code 0.0 if not used.
	71-80	E10.3	RPBR	Block porosity in dual-porosity media.

Group 9: Intial solute concentration of water within the media blocks

Code only if simulating dual-porosity media. One value is coded per material zone, ordered by zone number.

Card	Column	Format	Variable	Description
--	1-80	8E10.3	CBINIT	Initial concentration of water within the media blocks.

Group 10: Calibration-parameter numbers for ground-water flow

If NPARH = 0 do not code these cards.

Card	Column	Format	Variable	Description
--	1-5	I5	I	Material-zone number.
	6-10	I5	IPRMH(1,I)	Parameter number for RX. Code 0 if RX is not a calibration parameter.

Table 3.--*Instructions for preparing model data*--Continued

Group 10: Calibration-parameter numbers for ground-water flow--Continued

If NPARH = 0 do not code these cards.

Card	Column	Format	Variable	Description
	11-15	I5	IPRMH(2,I)	Parameter number for RY.
	16-20	I5	IPRMH(3,I)	Parameter number for RZ.

Group 11: Standard errors of calibration parameters for ground-water flow

One card per calibration parameter is coded. If NPARH is coded as 0 do not code these cards.

Card	Column	Format	Variable	Description
--	1-10	I10	I	Parameter number.
	11-20	E10.3	RKH	Normalized standard error of estimate for parameter I. Code 0.0 if no prior information exists for the parameter.

Group 12: Information for source-sink calibration parameters in the ground-water flow problem

Code one card per source-sink parameter.

Card	Column	Format	Variable	Description
--	1-10	I10	I	Source-sink parameter number.
	11-20	E10.3	QF	Multiplication factor for source-sink (either WELL or QS).
	21-30	E10.3	RKH	Normalized standard error for parameter.

Group 13: Boundary-parameter number associated with each node

Coded 15 values per card ordered by node number. If NBPAR is coded as 0 do not code these cards.

Card	Column	Format	Variable	Description
--	1-75	15I5	NBP	Boundary parameter number for each node.
Boundary				

Table 3.--*Instructions for preparing model data*--Continued

Group 14: Standard errors of boundary parameters for ground-water flow

Code NBPAR cards. If NBPAR is coded as 0 do not code these cards.

Card	Column	Format	Variable	Description
--	1-10	I10	I	Boundary-parameter number.
		E10.3	RKH	Normalized standard error of head-boundary parameter I.

Group 15: Transient boundary condition cards

NTB sets of these cards are needed.

Card	Column	Format	Variable	Description
1	1-5	I5	NXTB	Transport time step when boundary conditions change.
	6-10	I5	I2	Number of cards that follow.
--	1-5	I5	J	Node number.
	6-10	I5	NPBC	New nodal-point boundary condition.
	11-20	E10.3	H	New head.
	21-30	E10.3	C	New concentration.

Group 16: Observation cards

NTO sets of these cards are needed.

Card	Column	Format	Variable	Description
1	1-10	I10	NXTO	Transport time step corresponding to the following observations.
--	1-10	I10	K	Observation number corresponding to NODOBS. NOBMAX nodal observation cards must be coded for each set.
	11-20	E10.3	HO	Observed water level.
	21-30	E10.3	WH	Weighting value for HO.
	31-40	E10.3	CO	Observed concentration.
	41-50	E10.3	WC	Weighting value for CO.

Group 17: Transient point source-sink cards

NTW sets of these cards are needed.

Card	Column	Format	Variable	Description
1	1-10	I10	NXTW	Transport time step when source-sink changes.
	11-20	I10	I2	Number of cards that follow.
--	1-10	I10	K	Node number.
	11-20	E10.0	WELL	New point source-sink.
	21-30	E10.0	WELLC	New point-source concentration.

Table 3.--*Instructions for preparing model data*--Continued

Group 18: Transient distributed source-sink cards

NTS sets of these cards are needed.

Card	Column	Format	Variable	Description
1	1-10	I10	NXTS	Transport time step when source-sink changes.
	11-20	I10	I2	Number of cards that follow.
--	1-10	I10	K	Node number.
	11-20	E10.3	QS	New distributed source-sink.
	21-30	E10.3	CS	New distributed source concentration.

Table 4.--*Input data for sample simulation*

```
300  484    3   05   05    0    0    0    0    0    0    0    0
121  100
  0    1    1    8   10    4    0
  1    0    0        1.0        0.0        0.0        0.0        0.0
   100.0         1.0         1.0        +1.0         0.0         0.0
    +1.0         0.0         0.0         0.0
   200.0       400.0       500.0
       1           4        0.00        0.00         0.0
       5           4        0.00      500.00         0.0
       9           4        0.00     1000.00         0.0
      13           4        0.00     1500.00         0.0
      17           4        0.00     2000.00         0.0
      21           4        0.00     2500.00         0.0
      25           4        0.00     3000.00         0.0
      29           4        0.00     3500.00         0.0
      33           4        0.00     4000.00         0.0
      37           4        0.00     4500.00         0.0
      41           4        0.00     5000.00         0.0
      45           4      500.00        0.00         0.0
      49           4      500.00      500.00         0.0
      53           4      500.00     1000.00         0.0
      57           4      500.00     1500.00         0.0
      61           4      500.00     2000.00         0.0
      65           4      500.00     2500.00         0.0
      69           4      500.00     3000.00         0.0
      73           4      500.00     3500.00         0.0
      77           4      500.00     4000.00         0.0
      81           4      500.00     4500.00         0.0
      85           4      500.00     5000.00         0.0
      89           4     1000.00        0.00         0.0
      93           4     1000.00      500.00         0.0
      97           4     1000.00     1000.00         0.0
     101           4     1000.00     1500.00         0.0
     105           4     1000.00     2000.00         0.0
     109           4     1000.00     2500.00         0.0
     113           4     1000.00     3000.00         0.0
     117           4     1000.00     3500.00         0.0
     121           4     1000.00     4000.00         0.0
     125           4     1000.00     4500.00         0.0
     129           4     1000.00     5000.00         0.0
     133           4     1500.00        0.00         0.0
     137           4     1500.00      500.00         0.0
     141           4     1500.00     1000.00         0.0
     145           4     1500.00     1500.00         0.0
     149           4     1500.00     2000.00         0.0
     153           4     1500.00     2500.00         0.0
     157           4     1500.00     3000.00         0.0
     161           4     1500.00     3500.00         0.0
     165           4     1500.00     4000.00         0.0
     169           4     1500.00     4500.00         0.0
     173           4     1500.00     5000.00         0.0
     177           4     2000.00        0.00         0.0
```

Table 4.--*Input data for sample simulation*--Continued

181	4	2000.00	500.00	0.0
185	4	2000.00	1000.00	0.0
189	4	2000.00	1500.00	0.0
193	4	2000.00	2000.00	0.0
197	4	2000.00	2500.00	0.0
201	4	2000.00	3000.00	0.0
205	4	2000.00	3500.00	0.0
209	4	2000.00	4000.00	0.0
213	4	2000.00	4500.00	0.0
217	4	2000.00	5000.00	0.0
221	4	2500.00	0.00	0.0
225	4	2500.00	500.00	0.0
229	4	2500.00	1000.00	0.0
233	4	2500.00	1500.00	0.0
237	4	2500.00	2000.00	0.0
241	4	2500.00	2500.00	0.0
245	4	2500.00	3000.00	0.0
249	4	2500.00	3500.00	0.0
253	4	2500.00	4000.00	0.0
257	4	2500.00	4500.00	0.0
261	4	2500.00	5000.00	0.0
265	4	3000.00	0.00	0.0
269	4	3000.00	500.00	0.0
273	4	3000.00	1000.00	0.0
277	4	3000.00	1500.00	0.0
281	4	3000.00	2000.00	0.0
285	4	3000.00	2500.00	0.0
289	4	3000.00	3000.00	0.0
293	4	3000.00	3500.00	0.0
297	4	3000.00	4000.00	0.0
301	4	3000.00	4500.00	0.0
305	4	3000.00	5000.00	0.0
309	4	3500.00	0.00	0.0
313	4	3500.00	500.00	0.0
317	4	3500.00	1000.00	0.0
321	4	3500.00	1500.00	0.0
325	4	3500.00	2000.00	0.0
329	4	3500.00	2500.00	0.0
333	4	3500.00	3000.00	0.0
337	4	3500.00	3500.00	0.0
341	4	3500.00	4000.00	0.0
345	4	3500.00	4500.00	0.0
349	4	3500.00	5000.00	0.0
353	4	4000.00	0.00	0.0
357	4	4000.00	500.00	0.0
361	4	4000.00	1000.00	0.0
365	4	4000.00	1500.00	0.0
369	4	4000.00	2000.00	0.0
373	4	4000.00	2500.00	0.0
377	4	4000.00	3000.00	0.0
381	4	4000.00	3500.00	0.0
385	4	4000.00	4000.00	0.0

Table 4.--*Input data for sample simulation*--Continued

389	4	4000.00	4500.00	0.0		
393	4	4000.00	5000.00	0.0		
397	4	4500.00	0.00	0.0		
401	4	4500.00	500.00	0.0		
405	4	4500.00	1000.00	0.0		
409	4	4500.00	1500.00	0.0		
413	4	4500.00	2000.00	0.0		
417	4	4500.00	2500.00	0.0		
421	4	4500.00	3000.00	0.0		
425	4	4500.00	3500.00	0.0		
429	4	4500.00	4000.00	0.0		
433	4	4500.00	4500.00	0.0		
437	4	4500.00	5000.00	0.0		
441	4	5000.00	0.00	0.0		
445	4	5000.00	500.00	0.0		
449	4	5000.00	1000.00	0.0		
453	4	5000.00	1500.00	0.0		
457	4	5000.00	2000.00	0.0		
461	4	5000.00	2500.00	0.0		
465	4	5000.00	3000.00	0.0		
469	4	5000.00	3500.00	0.0		
473	4	5000.00	4000.00	0.0		
477	4	5000.00	4500.00	0.0		
481	4	5000.00	5000.00	0.0		
59						
1	0	0	0.0	0.0	0.0	6.0E-01
2	0	0	0.0	0.0	0.0	2.0E-01
3	0	0	0.0	0.0	0.0	2.0E-01
4	0	0	0.0	0.0	0.0	6.0E-01
5	0	0	0.0	0.0	0.0	6.0E-01
6	0	0	0.0	0.0	0.0	2.0E-01
7	0	0	0.0	0.0	0.0	2.0E-01
8	0	0	0.0	0.0	0.0	6.0E-01
9	0	0	0.0	0.0	0.0	6.0E-01
10	0	0	0.0	0.0	0.0	2.0E-01
11	0	0	0.0	0.0	0.0	2.0E-01
12	0	0	0.0	0.0	0.0	6.0E-01
13	0	0	0.0	0.0	0.0	6.0E-01
14	0	0	0.0	0.0	0.0	2.0E-01
15	0	0	0.0	0.0	0.0	2.0E-01
16	0	0	0.0	0.0	0.0	6.0E-01
17	0	0	0.0	0.0	0.0	6.0E-01
18	0	0	0.0	0.0	0.0	2.0E-01
19	0	0	0.0	0.0	0.0	2.0E-01
20	0	0	0.0	0.0	0.0	6.0E-01
21	0	0	0.0	0.0	0.0	6.0E-01
22	0	0	0.0	0.0	0.0	2.0E-01
23	0	0	0.0	0.0	0.0	2.0E-01
24	0	0	0.0	0.0	0.0	6.0E-01
25	0	0	0.0	0.0	0.0	6.0E-01
26	0	0	0.0	0.0	0.0	2.0E-01
27	0	0	0.0	0.0	0.0	2.0E-01

Table 4.--*Input data for sample simulation*--Continued

```
      28          0          0        0.0        0.0        0.0   6.0E-01
      29          0          0        0.0        0.0        0.0   6.0E-01
      30          0          0        0.0        0.0        0.0   2.0E-01
      31          0          0        0.0        0.0        0.0   2.0E-01
      32          0          0        0.0        0.0        0.0   6.0E-01
      33          0          0        0.0        0.0        0.0   6.0E-01
      34          0          0        0.0        0.0        0.0   2.0E-01
      35          0          0        0.0        0.0        0.0   2.0E-01
      36          0          0        0.0        0.0        0.0   6.0E-01
      37          0          0        0.0        0.0        0.0   6.0E-01
      38          0          0        0.0        0.0        0.0   2.0E-01
      39          0          0        0.0        0.0        0.0   2.0E-01
      40          0          0        0.0        0.0        0.0   6.0E-01
      41          0          0        0.0        0.0        0.0   6.0E-01
      42          0          0        0.0        0.0        0.0   2.0E-01
      43          0          0        0.0        0.0        0.0   2.0E-01
      44          0          0        0.0        0.0        0.0   6.0E-01
     444         -5          0        0.0      500.0        0.0       0.0
     448         -5          0        0.0      500.0        0.0       0.0
     452         -5          0        0.0      500.0        0.0       0.0
     456         -5          0        0.0      500.0        0.0       0.0
     460         -5          0        0.0      500.0        0.0       0.0
     464         -5          0        0.0      500.0        0.0       0.0
     468         -5          0        0.0      500.0        0.0       0.0
     472         -5          0        0.0      500.0        0.0       0.0
     476         -5          0        0.0      500.0        0.0       0.0
     480         -5          0        0.0      500.0        0.0       0.0
     484         -5          0        0.0      500.0        0.0       0.0
     161          0          0        0.0        0.0   -50000.0       0.0
     162          0          0        0.0        0.0   -50000.0       0.0
     323          0          0        0.0        0.0   -25000.0       0.0
     324          0          0        0.0        0.0   -25000.0       0.0
       4
       9        0.0        0.0      100.0
      10        0.0        0.0      100.0
      13        0.0        0.0      100.0
      14        0.0        0.0      100.0
  1   3   1   45   49    5
  4   3   5   49   53    9
  7   3   9   53   57   13
 10   3  13   57   61   17
 13   3  17   61   65   21
 16   3  21   65   69   25
 19   3  25   69   73   29
 22   3  29   73   77   33
 25   3  33   77   81   37
 28   3  37   81   85   41
 31   3  45   89   93   49
 34   3  49   93   97   53
 37   3  53   97  101   57
 40   3  57  101  105   61
 43   3  61  105  109   65
```

Table 4.--*Input data for sample simulation*--Continued

46	3	65	109	113	69
49	3	69	113	117	73
52	3	73	117	121	77
55	3	77	121	125	81
58	3	81	125	129	85
61	3	89	133	137	93
64	3	93	137	141	97
67	3	97	141	145	101
70	3	101	145	149	105
73	3	105	149	153	109
76	3	109	153	157	113
79	3	113	157	161	117
82	3	117	161	165	121
85	3	121	165	169	125
88	3	125	169	173	129
91	3	133	177	181	137
94	3	137	181	185	141
97	3	141	185	189	145
100	3	145	189	193	149
103	3	149	193	197	153
106	3	153	197	201	157
109	3	157	201	205	161
112	3	161	205	209	165
115	3	165	209	213	169
118	3	169	213	217	173
121	3	177	221	225	181
124	3	181	225	229	185
127	3	185	229	233	189
130	3	189	233	237	193
133	3	193	237	241	197
136	3	197	241	245	201
139	3	201	245	249	205
142	3	205	249	253	209
145	3	209	253	257	213
148	3	213	257	261	217
151	3	221	265	269	225
154	3	225	269	273	229
157	3	229	273	277	233
160	3	233	277	281	237
163	3	237	281	285	241
166	3	241	285	289	245
169	3	245	289	293	249
172	3	249	293	297	253
175	3	253	297	301	257
178	3	257	301	305	261
181	3	265	309	313	269
184	3	269	313	317	273
187	3	273	317	321	277
190	3	277	321	325	281
193	3	281	325	329	285
196	3	285	329	333	289
199	3	289	333	337	293

Table 4.--*Input data for sample simulation*--Continued

```
202     3  293  337  341  297
205     3  297  341  345  301
208     3  301  345  349  305
211     3  309  353  357  313
214     3  313  357  361  317
217     3  317  361  365  321
220     3  321  365  369  325
223     3  325  369  373  329
226     3  329  373  377  333
229     3  333  377  381  337
232     3  337  381  385  341
235     3  341  385  389  345
238     3  345  389  393  349
241     3  353  397  401  357
244     3  357  401  405  361
247     3  361  405  409  365
250     3  365  409  413  369
253     3  369  413  417  373
256     3  373  417  421  377
259     3  377  421  425  381
262     3  381  425  429  385
265     3  385  429  433  389
268     3  389  433  437  393
271     3  397  441  445  401
274     3  401  445  449  405
277     3  405  449  453  409
280     3  409  453  457  413
283     3  413  457  461  417
286     3  417  461  465  421
289     3  421  465  469  425
292     3  425  469  473  429
295     3  429  473  477  433
298     3  433  477  481  437
 15
        1        -1         0
        2        -2         0
        3        -3         0
        4        -1         0
        5        -2         0
        6        -3         0
        7        -1         0
        8        -2         0
        9        -3         0
       10        -1         0
       11        -2         0
       12        -3         0
       13        -1         0
       14        -2         0
       15        -3         0
        1      25.0      25.0       5.0       0.0
        2      10.0       2.0      10.0       0.0
        3      40.0      40.0      40.0       0.0
```

Table 4.--*Input data for sample simulation*--Continued

10.0	3.0	1.0E-04	0.1	0.2	1.8	0.0	0.01
1.0	0.3	1.0E-04	0.05	0.01	0.19	0.0	0.01
100.0	30.0	0.0	0.3	0.0	0.0	0.0	0.0
0.0	0.0	0.0					

Table 5.--*Selected output for sample simulation*

OELEMENT	VX	VY	VZ	DISPX	DISPY	DISPZ
1	0.36793E+00	0.22678E-01	0.41290E-02	0.36764E+01	0.11157E+01	0.11063E+01
2	0.14245E+00	0.18094E-02	0.37103E-01	0.14065E+00	0.44179E-01	0.50709E-01
3	0.50986E+00	0.36154E-01	0.34060E-01	0.50890E+02	0.15547E+02	0.15527E+02
4	0.36669E+00	0.70220E-01	0.40384E-02	0.36410E+01	0.12126E+01	0.11204E+01
5	0.14196E+00	0.55819E-02	0.37090E-01	0.14013E+00	0.44199E-01	0.50609E-01
6	0.50792E+00	0.11171E+00	0.34075E-01	0.50285E+02	0.17311E+02	0.15791E+02
7	0.36299E+00	0.12519E+00	0.38113E-02	0.35540E+01	0.14377E+01	0.11522E+01
8	0.14050E+00	0.10006E-01	0.36993E-01	0.13858E+00	0.44172E-01	0.50269E-01
9	0.50217E+00	0.19927E+00	0.34304E-01	0.48848E+02	0.21375E+02	0.16393E+02
10	0.35497E+00	0.19852E+00	0.30204E-02	0.33888E+01	0.18985E+01	0.12203E+01
11	0.13749E+00	0.15444E-01	0.36990E-01	0.13536E+00	0.44131E-01	0.49653E-01
12	0.48985E+00	0.31212E+00	0.34831E-01	0.46323E+02	0.29176E+02	0.17602E+02
13	0.31312E+00	0.36325E+00	0.59700E-03	0.28698E+01	0.33647E+01	0.14387E+01
14	0.12257E+00	0.29051E-01	0.34528E-01	0.11970E+00	0.43707E-01	0.45574E-01
15	0.43271E+00	0.55446E+00	0.36048E-01	0.39739E+02	0.51684E+02	0.21257E+02
16	0.16265E+00	0.36199E+00	0.32516E-02	0.16572E+01	0.35019E+01	0.11908E+01
17	0.62525E-01	0.28962E-01	0.14318E-01	0.59997E-01	0.29457E-01	0.23153E-01
18	0.22703E+00	0.55271E+00	0.11141E-01	0.23966E+02	0.53711E+02	0.17943E+02
19	0.41392E-01	0.19312E+00	0.46153E-02	0.65340E+00	0.19142E+01	0.59345E+00
20	0.15033E-01	0.15020E-01	-0.11152E-02	0.13818E-01	0.13805E-01	0.64249E-02
21	0.63782E-01	0.30366E+00	-0.68293E-02	0.10228E+02	0.30108E+02	0.93213E+01
22	0.31771E-01	0.11402E+00	0.16639E-02	0.41482E+00	0.11239E+01	0.35529E+00
23	0.12650E-01	0.91000E-02	0.58970E-03	0.11862E-01	0.83955E-02	0.46940E-02
24	0.48325E-01	0.18118E+00	-0.96020E-03	0.64973E+01	0.17880E+02	0.56258E+01
25	0.21817E-01	0.58068E-01	0.14778E-02	0.23985E+00	0.56655E+00	0.18639E+00
26	0.86806E-02	0.45777E-02	0.10892E-03	0.83188E-02	0.44389E-02	0.29451E-02
27	0.35092E-01	0.91639E-01	-0.15176E-02	0.38226E+01	0.89340E+01	0.29459E+01
28	0.16481E-01	0.17336E-01	0.10127E-02	0.15124E+00	0.15969E+00	0.72122E-01
29	0.63071E-02	0.13643E-02	0.13207E-03	0.62506E-02	0.21382E-02	0.19382E-02
30	0.24624E-01	0.27225E-01	-0.92102E-03	0.22575E+01	0.25145E+01	0.11032E+01
31	0.31226E+00	0.21581E-01	0.21722E-02	0.31196E+01	0.94945E+00	0.93914E+00
32	0.12219E+00	0.17202E-02	0.82262E-03	0.12218E+00	0.36678E-01	0.36664E-01
33	0.48846E+00	0.34404E-01	-0.25443E-02	0.48798E+02	0.14860E+02	0.14691E+02
34	0.30894E+00	0.66361E-01	0.21089E-02	0.30623E+01	0.10455E+01	0.94808E+00
35	0.12088E+00	0.52933E-02	0.80482E-03	0.12084E+00	0.36463E-01	0.36305E-01
36	0.48326E+00	0.10579E+00	-0.24507E-02	0.47886E+02	0.16425E+02	0.14842E+02
37	0.29959E+00	0.11644E+00	0.19215E-02	0.29189E+01	0.12596E+01	0.96436E+00
38	0.11722E+00	0.92597E-02	0.76553E-03	0.11708E+00	0.35787E-01	0.35280E-01
39	0.46864E+00	0.18529E+00	-0.23205E-02	0.45625E+02	0.19887E+02	0.15119E+02
40	0.27795E+00	0.17957E+00	0.15094E-02	0.26270E+01	0.16748E+01	0.99278E+00
41	0.10889E+00	0.14315E-01	0.61683E-03	0.10852E+00	0.34255E-01	0.32951E-01
42	0.43542E+00	0.28514E+00	-0.19919E-02	0.41113E+02	0.26549E+02	0.15615E+02
43	0.23097E+00	0.23083E+00	0.12852E-02	0.21233E+01	0.21218E+01	0.97967E+00
44	0.90602E-01	0.18471E-01	-0.12718E-04	0.89883E-01	0.30323E-01	0.27740E-01
45	0.36330E+00	0.36977E+00	-0.26759E-02	0.33374E+02	0.34015E+02	0.15553E+02
46	0.17144E+00	0.23511E+00	0.12712E-02	0.15800E+01	0.22027E+01	0.87297E+00
47	0.66391E-01	0.18782E-01	0.48994E-05	0.65417E-01	0.24278E-01	0.20699E-01
48	0.26362E+00	0.37462E+00	-0.17797E-02	0.24362E+02	0.35188E+02	0.13743E+02
49	0.13266E+00	0.17537E+00	0.13104E-02	0.12199E+01	0.16387E+01	0.65975E+00
50	0.49120E-01	0.14109E-01	-0.53863E-03	0.48378E-01	0.18059E-01	0.15337E-01

Table 5.--*Selected output for sample simulation*--Continued

TIME STEP 1 TOTAL TIME OF SIMULATION 100.00

NODE	XORD	YORD	ZORD	H	C
1	0.00000E+00	0.00000E+00	0.00000E+00	0.54688E+03	0.13656E+01
2	0.00000E+00	0.00000E+00	0.20000E+03	0.54670E+03	0.13071E+01
3	0.00000E+00	0.00000E+00	0.40000E+03	0.54577E+03	0.19450E+00
4	0.00000E+00	0.00000E+00	0.50000E+03	0.54566E+03	0.22148E+00
5	0.00000E+00	0.50000E+03	0.00000E+00	0.54642E+03	0.21257E+02
6	0.00000E+00	0.50000E+03	0.20000E+03	0.54625E+03	0.17615E+02
7	0.00000E+00	0.50000E+03	0.40000E+03	0.54531E+03	0.25413E+01
8	0.00000E+00	0.50000E+03	0.50000E+03	0.54520E+03	0.73561E+00
9	0.00000E+00	0.10000E+04	0.00000E+00	0.54500E+03	0.61517E+02
10	0.00000E+00	0.10000E+04	0.20000E+03	0.54484E+03	0.50354E+02
11	0.00000E+00	0.10000E+04	0.40000E+03	0.54390E+03	0.71874E+01
12	0.00000E+00	0.10000E+04	0.50000E+03	0.54379E+03	0.17069E+01
13	0.00000E+00	0.15000E+04	0.00000E+00	0.54247E+03	0.64242E+02
14	0.00000E+00	0.15000E+04	0.20000E+03	0.54230E+03	0.52123E+02
15	0.00000E+00	0.15000E+04	0.40000E+03	0.54138E+03	0.72752E+01
16	0.00000E+00	0.15000E+04	0.50000E+03	0.54127E+03	0.18243E+01
17	0.00000E+00	0.20000E+04	0.00000E+00	0.53841E+03	0.26126E+02
18	0.00000E+00	0.20000E+04	0.20000E+03	0.53843E+03	0.20814E+02
19	0.00000E+00	0.20000E+04	0.40000E+03	0.53744E+03	0.27289E+01
20	0.00000E+00	0.20000E+04	0.50000E+03	0.53731E+03	0.95652E+00
21	0.00000E+00	0.25000E+04	0.00000E+00	0.53054E+03	0.53993E+01
22	0.00000E+00	0.25000E+04	0.20000E+03	0.53042E+03	0.37372E+01
23	0.00000E+00	0.25000E+04	0.40000E+03	0.52995E+03	0.31057E+00
24	0.00000E+00	0.25000E+04	0.50000E+03	0.52989E+03	0.34697E+00
25	0.00000E+00	0.30000E+04	0.00000E+00	0.52268E+03	0.16408E+01
26	0.00000E+00	0.30000E+04	0.20000E+03	0.52244E+03	0.51514E+00
27	0.00000E+00	0.30000E+04	0.40000E+03	0.52249E+03	-0.22495E-01
28	0.00000E+00	0.30000E+04	0.50000E+03	0.52251E+03	0.76086E-02
29	0.00000E+00	0.35000E+04	0.00000E+00	0.51874E+03	0.95041E-01
30	0.00000E+00	0.35000E+04	0.20000E+03	0.51868E+03	0.17359E-03
31	0.00000E+00	0.35000E+04	0.40000E+03	0.51865E+03	-0.29428E-02
32	0.00000E+00	0.35000E+04	0.50000E+03	0.51865E+03	-0.36870E-02
33	0.00000E+00	0.40000E+04	0.00000E+00	0.51642E+03	-0.58869E-02
34	0.00000E+00	0.40000E+04	0.20000E+03	0.51635E+03	0.37361E-03
35	0.00000E+00	0.40000E+04	0.40000E+03	0.51635E+03	0.25604E-03
36	0.00000E+00	0.40000E+04	0.50000E+03	0.51636E+03	0.53348E-03
37	0.00000E+00	0.45000E+04	0.00000E+00	0.51523E+03	0.94322E-03
38	0.00000E+00	0.45000E+04	0.20000E+03	0.51518E+03	-0.45226E-04
39	0.00000E+00	0.45000E+04	0.40000E+03	0.51517E+03	-0.31195E-04
40	0.00000E+00	0.45000E+04	0.50000E+03	0.51518E+03	-0.12459E-03
41	0.00000E+00	0.50000E+04	0.00000E+00	0.51486E+03	-0.37950E-03
42	0.00000E+00	0.50000E+04	0.20000E+03	0.51482E+03	0.85539E-05
43	0.00000E+00	0.50000E+04	0.40000E+03	0.51482E+03	0.95106E-05
44	0.00000E+00	0.50000E+04	0.50000E+03	0.51482E+03	0.59893E-04
45	0.50000E+03	0.00000E+00	0.00000E+00	0.53951E+03	0.12931E+01
46	0.50000E+03	0.00000E+00	0.20000E+03	0.53938E+03	0.10619E+01
47	0.50000E+03	0.00000E+00	0.40000E+03	0.53936E+03	0.13798E+00
48	0.50000E+03	0.00000E+00	0.50000E+03	0.53937E+03	0.12620E+00
49	0.50000E+03	0.50000E+03	0.00000E+00	0.53906E+03	0.46206E+01
50	0.50000E+03	0.50000E+03	0.20000E+03	0.53894E+03	0.36959E+01

Table 5.--*Selected output for sample simulation*--Continued

51	0.50000E+03	0.50000E+03	0.40000E+03	0.53892E+03	0.44603E+00
52	0.50000E+03	0.50000E+03	0.50000E+03	0.53893E+03	0.42936E+00
53	0.50000E+03	0.10000E+04	0.00000E+00	0.53770E+03	0.11358E+02
54	0.50000E+03	0.10000E+04	0.20000E+03	0.53759E+03	0.88935E+01
55	0.50000E+03	0.10000E+04	0.40000E+03	0.53757E+03	0.10297E+01
56	0.50000E+03	0.10000E+04	0.50000E+03	0.53758E+03	0.10160E+01
57	0.50000E+03	0.15000E+04	0.00000E+00	0.53530E+03	0.13174E+02
58	0.50000E+03	0.15000E+04	0.20000E+03	0.53521E+03	0.10009E+02
59	0.50000E+03	0.15000E+04	0.40000E+03	0.53519E+03	0.11104E+01
60	0.50000E+03	0.15000E+04	0.50000E+03	0.53520E+03	0.11133E+01
61	0.50000E+03	0.20000E+04	0.00000E+00	0.53153E+03	0.77139E+01
62	0.50000E+03	0.20000E+04	0.20000E+03	0.53145E+03	0.56143E+01
63	0.50000E+03	0.20000E+04	0.40000E+03	0.53146E+03	0.59393E+00
64	0.50000E+03	0.20000E+04	0.50000E+03	0.53147E+03	0.59816E+00
65	0.50000E+03	0.25000E+04	0.00000E+00	0.52663E+03	0.34570E+01
66	0.50000E+03	0.25000E+04	0.20000E+03	0.52656E+03	0.21757E+01
67	0.50000E+03	0.25000E+04	0.40000E+03	0.52656E+03	0.17815E+00
68	0.50000E+03	0.25000E+04	0.50000E+03	0.52657E+03	0.19774E+00
69	0.50000E+03	0.30000E+04	0.00000E+00	0.52179E+03	0.70828E+00
70	0.50000E+03	0.30000E+04	0.20000E+03	0.52168E+03	0.17924E+00
71	0.50000E+03	0.30000E+04	0.40000E+03	0.52165E+03	-0.19552E-01
72	0.50000E+03	0.30000E+04	0.50000E+03	0.52165E+03	0.32101E-02
73	0.50000E+03	0.35000E+04	0.00000E+00	0.51806E+03	0.39377E-01
74	0.50000E+03	0.35000E+04	0.20000E+03	0.51799E+03	0.30788E-02
75	0.50000E+03	0.35000E+04	0.40000E+03	0.51801E+03	0.22454E-03
76	0.50000E+03	0.35000E+04	0.50000E+03	0.51802E+03	-0.28177E-02
77	0.50000E+03	0.40000E+04	0.00000E+00	0.51596E+03	-0.34128E-02
78	0.50000E+03	0.40000E+04	0.20000E+03	0.51589E+03	-0.11398E-03
79	0.50000E+03	0.40000E+04	0.40000E+03	0.51588E+03	-0.19108E-03
80	0.50000E+03	0.40000E+04	0.50000E+03	0.51589E+03	0.55248E-03
81	0.50000E+03	0.45000E+04	0.00000E+00	0.51488E+03	0.53509E-03
82	0.50000E+03	0.45000E+04	0.20000E+03	0.51485E+03	0.11777E-04
83	0.50000E+03	0.45000E+04	0.40000E+03	0.51486E+03	0.46362E-04
84	0.50000E+03	0.45000E+04	0.50000E+03	0.51486E+03	-0.13666E-03
85	0.50000E+03	0.50000E+04	0.00000E+00	0.51460E+03	-0.21623E-03
86	0.50000E+03	0.50000E+04	0.20000E+03	0.51456E+03	-0.55297E-05
87	0.50000E+03	0.50000E+04	0.40000E+03	0.51456E+03	-0.16452E-04
88	0.50000E+03	0.50000E+04	0.50000E+03	0.51456E+03	0.60816E-04
89	0.10000E+04	0.00000E+00	0.00000E+00	0.53322E+03	0.29068E+00
90	0.10000E+04	0.00000E+00	0.20000E+03	0.53326E+03	0.17762E+00
91	0.10000E+04	0.00000E+00	0.40000E+03	0.53325E+03	0.91995E-02
92	0.10000E+04	0.00000E+00	0.50000E+03	0.53325E+03	0.14266E-01
93	0.10000E+04	0.50000E+03	0.00000E+00	0.53282E+03	0.78351E+00
94	0.10000E+04	0.50000E+03	0.20000E+03	0.53287E+03	0.45562E+00
95	0.10000E+04	0.50000E+03	0.40000E+03	0.53286E+03	0.21103E-01
96	0.10000E+04	0.50000E+03	0.50000E+03	0.53285E+03	0.25960E-01
97	0.10000E+04	0.10000E+04	0.00000E+00	0.53161E+03	0.18427E+01
98	0.10000E+04	0.10000E+04	0.20000E+03	0.53166E+03	0.10197E+01
99	0.10000E+04	0.10000E+04	0.40000E+03	0.53165E+03	0.43592E-01
100	0.10000E+04	0.10000E+04	0.50000E+03	0.53164E+03	0.49508E-01
101	0.10000E+04	0.15000E+04	0.00000E+00	0.52953E+03	0.24778E+01
102	0.10000E+04	0.15000E+04	0.20000E+03	0.52959E+03	0.12279E+01

Table 5.--*Selected output for sample simulation*--Continued

103	0.10000E+04	0.15000E+04	0.40000E+03	0.52958E+03	0.44277E-01
104	0.10000E+04	0.15000E+04	0.50000E+03	0.52957E+03	0.59432E-01
105	0.10000E+04	0.20000E+04	0.00000E+00	0.52661E+03	0.17074E+01
106	0.10000E+04	0.20000E+04	0.20000E+03	0.52665E+03	0.73015E+00
107	0.10000E+04	0.20000E+04	0.40000E+03	0.52664E+03	0.14032E-01
108	0.10000E+04	0.20000E+04	0.50000E+03	0.52664E+03	0.37907E-01
109	0.10000E+04	0.25000E+04	0.00000E+00	0.52303E+03	0.72781E+00
110	0.10000E+04	0.25000E+04	0.20000E+03	0.52306E+03	0.25050E+00
111	0.10000E+04	0.25000E+04	0.40000E+03	0.52304E+03	-0.35839E-02
112	0.10000E+04	0.25000E+04	0.50000E+03	0.52304E+03	0.85501E-02
113	0.10000E+04	0.30000E+04	0.00000E+00	0.51887E+03	0.26513E+00
114	0.10000E+04	0.30000E+04	0.20000E+03	0.51903E+03	0.73789E-01
115	0.10000E+04	0.30000E+04	0.40000E+03	0.51920E+03	0.14296E-02
116	0.10000E+04	0.30000E+04	0.50000E+03	0.51921E+03	0.35718E-02
117	0.10000E+04	0.35000E+04	0.00000E+00	0.51619E+03	0.30831E-01
118	0.10000E+04	0.35000E+04	0.20000E+03	0.51613E+03	0.25067E-02
119	0.10000E+04	0.35000E+04	0.40000E+03	0.51606E+03	-0.15160E-02
120	0.10000E+04	0.35000E+04	0.50000E+03	0.51607E+03	-0.90147E-03
121	0.10000E+04	0.40000E+04	0.00000E+00	0.51425E+03	-0.16723E-02
122	0.10000E+04	0.40000E+04	0.20000E+03	0.51440E+03	-0.14602E-03
123	0.10000E+04	0.40000E+04	0.40000E+03	0.51458E+03	0.19407E-03
124	0.10000E+04	0.40000E+04	0.50000E+03	0.51459E+03	0.10417E-03
125	0.10000E+04	0.45000E+04	0.00000E+00	0.51404E+03	0.49928E-03