A TWO-CONSTITUENT SOLUTE-TRANSPORT MODEL FOR GROUND WATER HAVING VARIABLE DENSITY

by Ward E. Sanford and Leonard F. Konikow

U.S. GEOLOGICAL SURVEY
Water-Resources Investigations Report 85—4279
1985

UNITED STATES DEPARTMENT OF THE INTERIOR

DONALD P. HODEL, Secretary

GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information write to:

Project Chief
U.S. Geological Survey
MS 431 National Center
Reston, Virginia 22092

Copies of this report can be purchase from:

Open-File Services Section
Western Distribution Branch
U.S. Geological Survey
Box 25425, Federal Center
Denver, Colorado 80225

CONTENTS

ILLUSTRATIONS

CONVERSION FACTORS

For use of readers who prefer to use metric units, conversion factors for terms used in this report are listed below:

Multiply	By	To Obtain
foot (ft)	0.3048	meter (m)
square foot (ft^2)	0.0929	square meter (m^2)
square foot per second (ft^2/s)	929.0	square centimeter per second (cm^2/s)
cubic foot per second (ft^3/s)	0.02832	cubic meter per second (m^3/s)
pound (lb)	4.448	newton (N)
pound per square foot (lb/ft^2)	47.88	pascal or newton per meter squared (Pa or N/m^2)
pound per cubic foot (lb/ft^3)	157.1	newton per cubic meter (N/m^3)

PREFACE

This report presents a digital computer model for calculating changes in the concentration of dissolved chemical species in flowing ground water. The computer program represents a basic and general model that may have to be modified by the user for efficient application to his specific field problem. Although this model will produce reliable calculations for a wide variety of field problems, the user is cautioned that in some cases the accuracy and efficiency of the model can be affected significantly by his discretization and his selection of values for certain other user-specified options.

The user is requested to kindly notify the originating office of any errors found in this report or in the computer program. Updates may occasionally be made to both the report and the computer program. Users who wish to be added to the mailing list to receive updates, if any, may send a request to the originating office at the following address:

U.S. Geological Survey
431 National Center
Reston, VA 22092

Copies of the computer program on tape are available at cost of processing from:

U.S. Geological Survey
WATSTORE Program Office
437 National Center
Reston, VA 22092
Telephone: 703/860-6871

A TWO-CONSTITUENT SOLUTE-TRANSPORT MODEL FOR GROUND WATER HAVING VARIABLE DENSITY

By

Ward E. Sanford and Leonard F. Konikow

ABSTRACT

A numerical model has been developed to simulate solute transport and dispersion of either one or two constituents in ground water where there is two-dimensional, density-dependent flow. The model is a modified version of the one documented by Konikow and Bredehoeft (1978), which uses finite-difference methods and the method of characteristics to solve the flow and transport equations. The model was tested on an idealized seawater-intrusion problem for which Henry (1964) developed an analytical solution. The results were nearly identical to those of other numerical models tested on the same problem. A description of the formats for the input data, a sample of input and output for a two-constituent example problem, and a listing of the Fortran program are presented.

INTRODUCTION

Various numerical models have recently been developed that simulate ground-water flow and solute transport for a variety of conditions. Some of the available models are designed to simulate the flow of ground water that has a constant and uniform fluid density, and others can simulate variable density fluids in which the concentration of the solute of interest affects the density of the fluid. The latter models typically have been applied to problems of seawater intrusion in coastal aquifers. However, there are many problems in which contaminants are introduced into an aquifer near the interface or transition zone between freshwater and saltwater. Examples include the injection of waste water into coastal aquifers; Burnham and others (1977), Larson and others (1977), and Rosenshein and Hickey (1977) describe such practices in Hawaii and Florida. In such cases the injection will affect the fluid pressure and flow of both the freshwater and saltwater, but the contaminants being injected are generally in such low concentrations that changes in concentration of the contaminants will not affect the fluid density. Simulation of such problems thus requires the ability to simulate the simultaneous flow of variable-density ground water and the transport and dispersion of at least two solutes or soluble constituents. The fluid density needs to be related to the concentration of one of the constituents, which in practice can be either salinity, dissolved-solids concentration, specific conductance, or chloride concentration. The objective of this report is to document a numerical simulation model that is applicable to these types of problems.

The model described in this report is a modified version of the ground-water flow and solute-transport model of Konikow and Bredehoeft (1978), which was designed to simulate the transport and dispersion of a single solute that does not affect the fluid density. This modified version simulates the flow in a cross-sectional plane rather than in an areal plane. Because the problem of interest involves variable density, the modified model solves for fluid pressure rather than hydraulic head in the flow equation; the solution to the flow equation is still obtained using a finite-difference method. Solute transport is simulated with the method of characteristics as in the original model. Density is considered to be a function of the concentration of one of the constituents. Use of this model depends on assumptions that (1) flow is two-dimensional, with one of the principal axes being parallel to gravity, (2) constituents are conservative (nonreactive), and (3) density and viscosity are a function of concentration and not of other factors such as pressure and temperature. These assumptions are often valid approximations where an aquifer system contains both freshwater and saltwater. This model is applicable in such situations where, in addition to that of the density-controlling species, the movement and concentration of another chemical species, such as a dissolved pollutant, needs to be predicted. The model is also applicable to a two-constituent system with no density-dependence, given that the other assumptions are valid, and to a single-constituent system with variable density. This model documentation needs to be used in conjunction with the original documentation of Konikow and Bredehoeft (1978) because many of the detailed descriptions of theory, numerical methods, and program features and options contained in the original documentation are not repeated in this report.

Acknowledgments

This work was prepared in part under a cooperative research grant awarded by the U.S.-Spain Joint Committee for Scientific and Technological Cooperation (Project Number 83-007). The advice and helpful comments of Dave Pollock, Mike Merritt, and Cliff Voss are greatly appreciated.

THEORY

The model solves an equation which represents the flow of a compressible fluid through a heterogeneous, anisotropic, confined aquifer. By following the developments of Cooper (1966) and of Bredehoeft and Pinder (1973), the general flow equation can be expressed in cartesian tensor notation as:

$$\frac{\partial}{\partial x_i}\left[\frac{\rho g k_{ij}}{\mu}\left(\frac{\partial P}{\partial x_i} + \rho g \frac{\partial z^*}{\partial x_j}\right)\right] = S_s \frac{\partial P}{\partial t} + W^* \rho^* g \qquad i,j=1,2 \qquad (1)$$

where k_{ij} is the intrinsic permeability (a second-order tensor), L^2;

ρ is the fluid density, ML^{-3};

μ is the dynamic viscosity, $ML^{-1}T^{-1}$;

P is the fluid pressure, $ML^{-1}T^{-2}$;

g is the gravitational acceleration constant, LT^{-2};

z^* is the elevation of the reference point above a standard datum, L;

S_s is the specific storage of the aquifer, L^{-1};

$W^*=W^*(x,y,z,t)$ is a source/sink volume flux per unit volume (positive sign for outflow and negative or inflow), T^{-1};

ρ^* is the density of the source/sink fluid, ML^{-3};

x_i are the cartesian coordinates, L; and

t is the time, T.

The source/sink term is handled with the method used by Konikow and Bredehoeft (1978) but is written in terms of pressure, as follows:

$$W^*(x,y,z,t) = \frac{Q^*(x,y,z,t)}{\Delta x \Delta y \Delta z} - \frac{k_z}{\mu m \Delta z}\left(P_s - P + \rho g(z_s - z^*)\right) \qquad (2)$$

where $Q^*(x,y,z,t)$ is the rate of withdrawal (positive sign) or recharge (negative sign), T^{-1};

k_z is the vertical permeability of the confining layer, L^2;

m is the thickness of the confining layer, L;

Δy is the width of the aquifer cross-section, L;

Δx and Δz are the grid dimensions in the x and z directions, respectively, L;

P_s is the fluid pressure in the source bed, $ML^{-1}T^{-2}$; and

z_s is the elevation of the source bed above a standard datum, L.

The second term on the right side of equation two can be used to represent steady leakage through a confining bed, which would only be applicable along a boundary node, or to represent a constant-pressure boundary condition, as explained in more detail below.

The equations that represent solute transport and dispersion are also solved in this model. Based on the work of Pinder and Cooper (1970), Bear (1972), Bredehoeft and Pinder (1973), and Konikow and Grove (1977), the equation for transport and dispersion of solutes in flowing ground water in cartesian tensor notation can be written:

$$\frac{\partial C_n}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial C_n}{\partial x_i}\right) - \frac{\partial}{\partial x_i}(C_n V_i) - \frac{C_n' W^*}{\varepsilon} \tag{3}$$

where D_{ij} is the coefficient of hydrodynamic dispersion (a second-order tensor), L^2T^{-1};

V_i is the seepage velocity in the direction of x_i, LT^{-1};

C_n is the concentration of the n^{th} constituent, ML^{-3};

C_n' is the concentration of the n^{th} constituent in the source or sink fluid, ML^{-3}; and

ε is the effective porosity (dimensionless).

Two constituents are represented in this model, so n=1,2 in equation 3, giving two similar equations, one for each constituent. The density and viscosity are taken to be a linear function of the concentration of the first constituent, C_1, as follows:

$$\rho_{(i,j)} = A_d C_{1(i,j)} + B_d \tag{4}$$

$$\mu(i,j) = A_v C_{1(i,j)} + B_v \tag{5}$$

where A_d and B_d are the slope and intercept, respectively, for the relationship between density and solute concentration, and A_v and B_v are the slope and intercept, respectively for the relationship between viscosity and solute concentration. The user has the option of specifying values for these coefficients, or else using default values built into the model for salinity or dissolved-solids concentration. The default values, based on data from Weast (1981, p. D-229) for seawater at different salinities (in parts per million, or ppm), are $A_d = 4.743 \times 10^{-5}$,

$B_d = 62.43$, $A_v = 3.45 \times 10^{-11}$ and $B_v = 2.089 \times 10^{-5}$ for $C_1 < 20{,}000$ ppm, and $A_v = 4.733 \times 10^{-11}$ and $B_v = 2.063 \times 10^{-5}$ for $C_1 > 20{,}000$ ppm.

The coefficient of hydrodynamic dispersion represents the sum of the mechanical dispersion, which depends upon both the flow of the fluid and the nature of the pore system, and the molecular and ionic diffusion. These terms can be represented as:

$$D_{ij} = \alpha_{ijmm} \frac{V_m V_n}{|V|} + D_m \quad (6)$$

where D_m is the coefficient of molecular and ionic diffusion, L^2T^{-1};

α_{ijmn} is the dispersivity of the aquifer, L;

V_m and V_n are components of velocity in the m and n directions, respectively, LT^{-1}; and

$|V|$ is the magnitude of the velocity, LT^{-1}.

Scheidegger (1961) further shows that for an isotropic aquifer the dispersivity tensor can be defined in terms of two constants. These are the longitudinal and transverse dispersivities of the aquifer (α_L and α_T, respectively). These are related to the longitudinal and transverse dispersion coefficients by:

$$D_L = \alpha_L |V| \quad (7)$$

$$D_T = \alpha_T |V| \quad (8)$$

Based on the work of Scheidegger (1961) and Bear (1972), we may state explicitly the components of the dispersion coefficients for two-dimensional flow in an isotropic aquifer as:

$$D_{xx} = D_L \frac{(V_x)^2}{|V|^2} + D_T \frac{(V_z)^2}{|V|^2} + D_m \quad (9)$$

$$D_{zz} = D_T \frac{(V_x)^2}{|V|^2} + D_L \frac{(V_z)^2}{|V|^2} + D_m \quad (10)$$

$$D_{xz} = D_{zx} = (D_L - D_T) \frac{V_x V_z}{|V|^2} \quad (11)$$

The seepage velocity is calculated from the solution to the flow equation using a form of Darcy's law, as described in Konikow and Grove (1977). The equation can be written in cartesian tensor notation as:

$$V_i = \frac{-k_{ij}}{\varepsilon\mu} \left(\frac{\partial P}{\partial x_i} + \rho g \frac{\partial z^*}{\partial x_j} \right) \quad (12)$$

NUMERICAL METHODS

The numerical methods used to solve the flow and transport equations in this model are similar to those used in the original model by Konikow and Bredehoeft (1978). An iterative finite-difference scheme that uses a strongly implicit procedure (see Trescott and others, 1976) is used to solve for the flow field (equation 1). The method of characteristics is used to solve for the convective (or advective) part of the transport (equation 3), while an explicit finite-difference scheme is used to solve for dispersion. Some of the additions described below were made to improve the performance and applicability of the original 1978 model. Further additions and modifications were made to provide the capability to simulate two constituents and variable density.

The method of characteristics uses particle-tracking to simulate the advection of the solute of interest. In the 1978 version, the user could specify 4, 5, 8, or 9 particles to be generated per node. The first addition made to the original model was to allow 16 particles per node to be specified in order to improve numerical accuracy. As a second addition, molecular diffusion was added to the term describing hydrodynamic dispersion in the solute-transport equation (molecular diffusion had been assumed to be negligible in the original model). This change was necessary to allow a comparison of the model results for a variable-density problem with the analytical solution developed by Henry (1964). Corresponding modifications were then included in the finite-difference scheme that represents dispersion by adding a diffusion constant to the dispersion terms, as shown by equations 9-11. A third minor modification was made that allows the user to specify the maximum number of cells that can be void of tracer particles (NZCRIT). If NZCRIT is exceeded, the particle locations are reinitialized. Fourth, the maximum dimensions of all two-dimensional arrays were increased to (24,20) from the original (20,20). The fifth addition was the introduction of a particle-weighting (or fractional particle) scheme that allows weak sources and sinks to be more accurately represented. In this scheme, particles generated initially at all nodes receive a weight of one. Particles that are later regenerated or removed at sources and sinks are given a weight from zero to one, which represents the fraction of fluid passing through the block that is due to the presence of the source or sink. In effect, particles are weighted in proportion to fluid volumes. This particle-weighting scheme significantly improves the chemical mass balance in the model, especially if the fluid in the source or sink only accounts for a small fraction of the fluid passing through the block.

Additional modifications have been made to the model to allow for cross-sectional, density-dependent flow and the transport and dispersion of two constituents. The general flow equation described in equations 1 and 2 can be written together in the following finite-difference form:

$$\left[\frac{\rho g k_{xx}}{\mu}\right]_{(i-\frac{1}{2},j)} \left[\frac{P_{i-1,j,k} - P_{i,j,k}}{(\Delta x)^2}\right] + \left[\frac{\rho g k_{xx}}{\mu}\right]_{(i+\frac{1}{2},j)} \left[\frac{P_{i+1,j,k} - P_{i,j,k}}{(\Delta x)^2}\right]$$

$$+ \left[\frac{\rho g k_{zz}}{\mu}\right]_{(i,j-\frac{1}{2})} \left[\frac{P_{i,j-1,k} - P_{i,j,k}}{(\Delta z)^2} + \frac{(\rho g)_{i,j-\frac{1}{2}}}{\Delta z}\right]$$

$$+ \left[\frac{\rho g k_{zz}}{\mu}\right]_{(i,j+\frac{1}{2})} \left[\frac{P_{i,j+1,k} - P_{i,j,k}}{(\Delta z)^2} + \frac{(\rho g)_{i,j+\frac{1}{2}}}{\Delta z}\right]$$

$$= S_s \left[\frac{P_{i,j,k} - P_{i,j,k-1}}{\Delta t}\right] + \left[\frac{\rho g q_w}{\Delta x \Delta y \Delta z}\right]_{(i,j)}$$

$$+ \left[\frac{\rho g k_z}{\Delta z \mu m}\right]_{(i,j)} \left[P_{s(i,j)} - P_{i,j} + [\rho g]_{i,j}[z_s - z^*]\right] \qquad (13)$$

where i,j,k are the indices in the x, z, and time dimensions, respectively; and

q_w is the volumetric rate of withdrawal or recharge at the (i,j) node, L^3T^{-1}.

The modifications for the solution of the flow equation thus include (1) solving for fluid pressure rather than hydraulic head, (2) using density, viscosity, and intrinsic permeability rather than hydraulic conductivity, (3) using specific storage rather than storage coefficient to represent storativity, and (4) using a constant aquifer width rather than a variable thickness to represent the third dimension.

Constant-pressure boundaries are treated in the same basic way as in Konikow and Bredehoeft (1978). This approach is based on the principle that as the leakance coefficient of the confining bed (that is, its conductance, defined as the hydraulic conductivity of the confining layer divided by the thickness of the confining layer) increases to a sufficiently high value, the difference in head across the confining bed will decrease to a negligible value so that the heads will be essentially identical on both sides of the confining bed. Thus, given a sufficiently high value for the leakance coefficient and a constant value of head (or pressure in this case) in the source bed, then the head (or pressure) in the aquifer at that location will always remain essentially the same as the specified constant value of head (or pressure) because the confining bed can readily transmit an adequate flux to compensate for any stresses imposed elsewhere in the aquifer. The constant-pressure term in the equation will then take on virtually the same form as the constant-head term in the original model. However, if the leakance coefficient is specified as too high a value relative to the conductance within the aquifer, then although the computed value of head (or pressure) in the aquifer is the desired constant value, the head difference will be so small that numerical truncation errors may induce significant errors in the subsequent calculation of the flux at that constant-head boundary node. An error in the computed flux can have a serious effect on the accuracy of the computed solute concentrations because that flux represents part of the source term in the solute-transport equation (that is, the third term on the right side of equation 3). Therefore, the original model, which required the user to specify the head and the leakance coefficient at a constant-head node, was modified to allow the user to specify only the desired constant-pressure value. The model then automatically calculates a value of the leakance coefficient that is ten orders of magnitude greater than the conductance of the aquifer at that location; this ratio of 10^{10} was found optimal both in providing the desired constant pressure and in eliminating serious numerical truncation error in the calculation of the flux due to leakage.

Modifications were also made to account for an additional constituent. Only one set of particles is tracked, as in the original method of characteristics, but each particle is assigned two independent concentration values. The finite-difference equations that represent the dispersion, which are presented in detail by Konikow and Bredehoeft (1978), are now solved twice, once for each constituent.

The flow equation and the solute-transport equation are coupled by allowing the density and viscosity to be a function of the concentration of the first constituent. In this model, particles are moved in time steps whose lengths are determined by certain stability criteria (see Konikow and Bredehoeft, 1978). After every movement of all particles is completed, the new model checks to see if a concentration change has occurred that would significantly affect the density. If a significant change in concentration has occurred, the pressures are recalculated using the new densities. In this way, the calculated flow field is periodically updated to account for changes in density due to changes in concentration. The criteria defining the amount of change that is considered significant is specified by the user in the input data.

MODEL VERIFICATION

One way to test the accuracy of a numerical model is to compare its results with that of a known analytical solution. Konikow and Bredehoeft (1978) compared the numerical results with analytical solutions for dispersion in one-dimensional steady flow and in plane radial steady flow, in both cases assuming constant fluid density. Their results demonstrated that the model is numerically accurate for these conditions.

For the case of variable-density fluids this model was tested on a problem for which an analytical solution was developed by Henry (1964). The problem was set up with boundary conditions and parameters (as shown in fig. 1) that allowed for the convergence of the infinite series involved in Henry'solution. A constant flux, Q, on the left side of the system balances a tongue of saltwater entering from the right, which gives a steady-state transition zone between freshwater and saltwater. The 0.5 isochlor in figure 1 shows the center of the transition zone.

Pinder and Cooper (1970) were one of the first to use this problem for testing a numerical model. Their parameters of Q = 0.66 cm^2/s, K = 1.0 cm/s, and D_m = 0.066 cm^2/s (7.10 x 10^{-5} ft^2/s) were used in the comparisons between other numerical models and the model being documented in this report. The analyses are further based on C_s = 35,000 mg/L and ρ_s = 1.025 g/cm^3, where C_s is the concentration of salt in seawater and ρ_s is the density of seawater.

A direct comparison was not made with Henry's analytical solution, which requires letting the system run to steady state after beginning with some fixed concentration. Also, the right hand boundary condition in

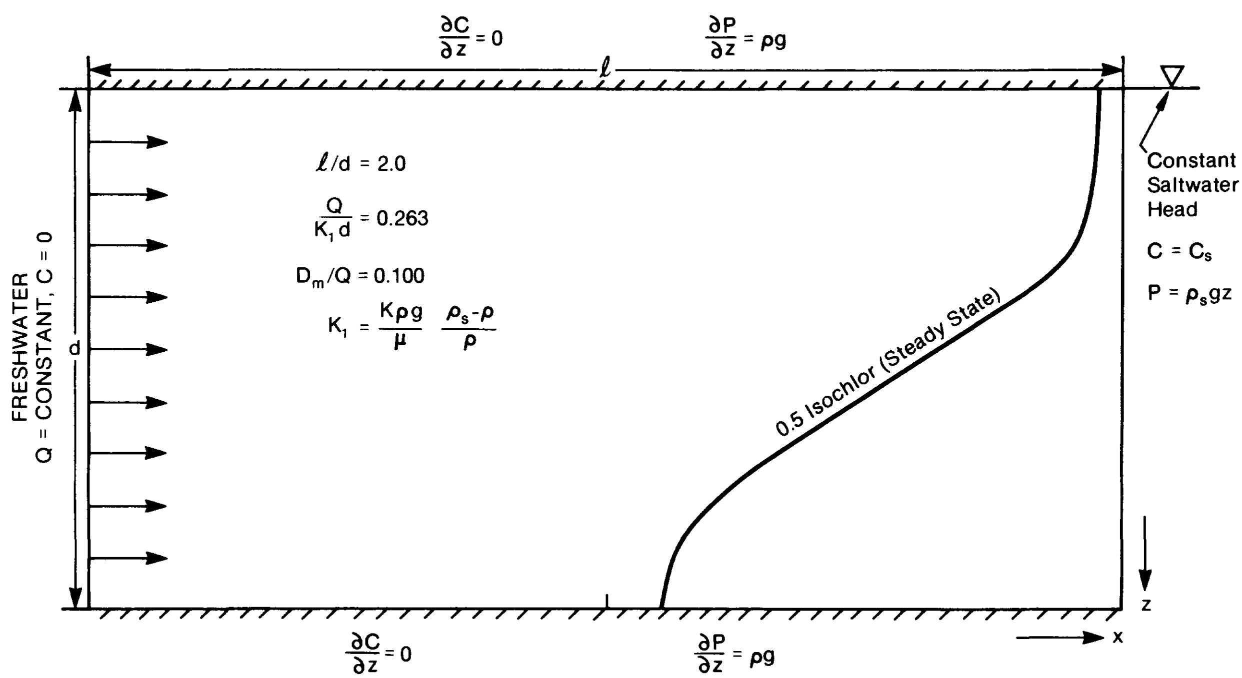

Figure 1. Parameters, Boundary Conditions, and Solution of Henry's Problem (Henry, 1964)

Henry's problem is unrealistic for a system that a numerical model would be simulating. Fluid is exiting the system at the top right, but Henry's solution allows diffusion to occur back across the boundary. For these reasons, the comparison was made with the results from other numerical models that solved Henry's problem for 100 minutes (simulation time) and allowed no backward diffusion across the boundary. The first comparison was with the results from a finite-element model of Segol and others (1975). As seen in figure 2, the 0.5 isochlors from the two models are at virtually the same location. Note that the 0.5 isochlor is equivalent to C = 17,500 mg/L.

The finite-difference model by INTERA (1979) was also used to solve Henry's problem. The comparison in figure 3 shows that the isochlors from the two models are very close. The difference between the positions of the isochlors from the solution of Segol and others (1975) (fig. 2) and the positions of those from the INTERA model (fig. 3) is due to a difference in the value of the diffusion constant used. Segol and others (1975) apparently did not divide the diffusion constant used in Pinder and Cooper (1970) by the porosity, as was done for the simulation using the INTERA model. The smaller value of the diffusion coefficient used by Segol and others (1975) caused the front to move further to the left. Both results show that the model being documented in this report gives results comparable to those of other numerical models used on a problem for which an analytical solution is known. The results obtained with this model for the problems shown in figures 2 and 3 are also essentially identical to those obtained by Voss (1984) using a finite-element model for the same two problems.

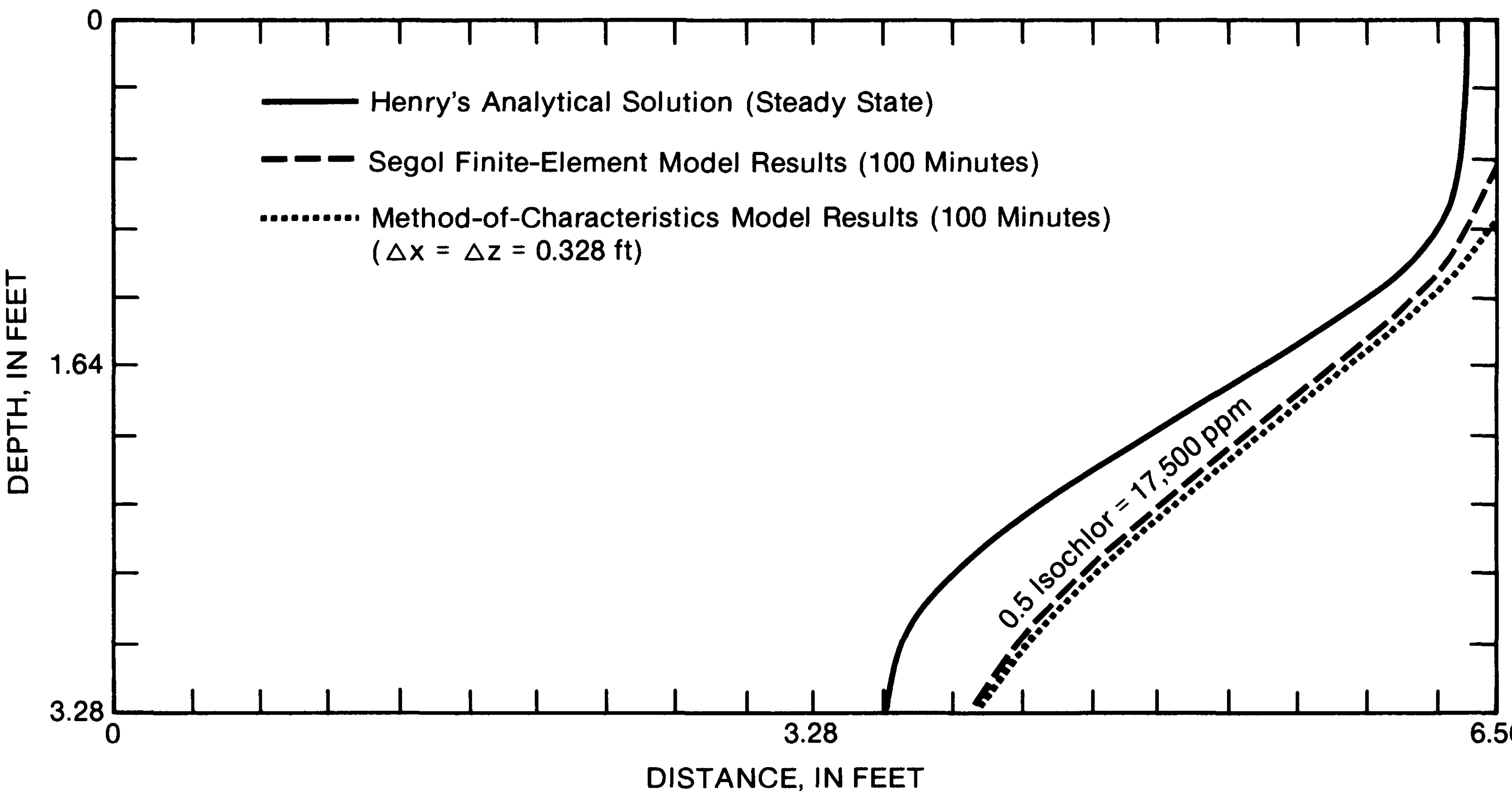

Figure 2. Comparison with Segol's Model for Henry's Problem (D_m = 7.10 x 10^{-5} ft^2/s in Both Numerical Models).

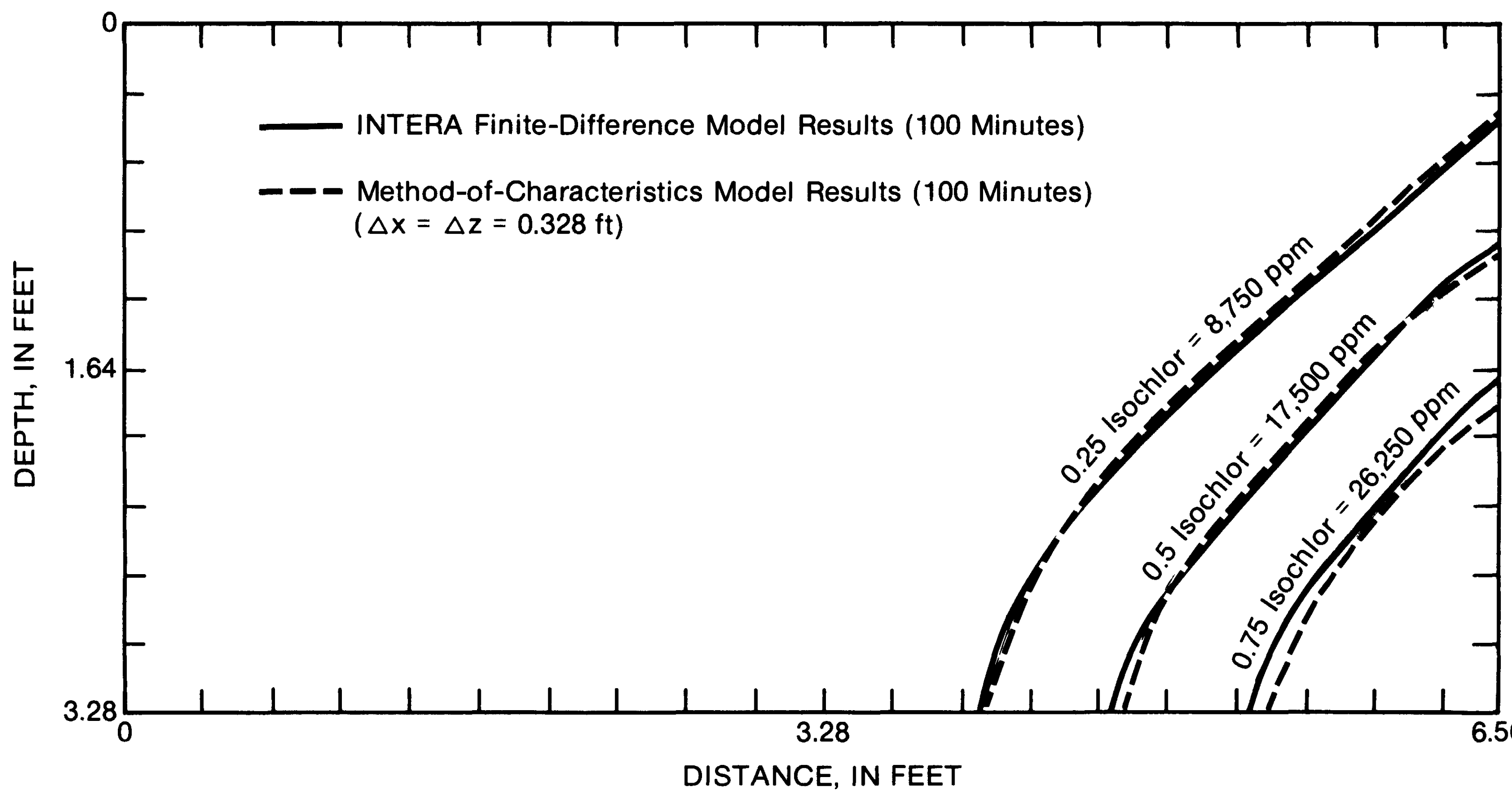

Figure 3. Comparison With the INTERA (1979) Finite-Difference Model for Henry's Problem (D_m = 2.03 x 10^{-4} ft^2/s in Both Numerical Models).

USER'S GUIDE

The input and output formats have been designed for flexibility of use and general compatibility with the analysis of typical problems to which the model is applicable. All input data formats are described in Appendix I. Immediately before the program is run, the input data file must be opened as unit 5 of the computer system and the output file must be opened as unit 6.

The model will allow a unique source/sink rate to be specified at each node, and will allow up to five observation points to be specified for summary printouts of concentration and pressure versus time. The program also includes a node identification array (NODEID), which allows certain nodes or zones to be identified by a unique code number. This feature is used to identify constant-pressure cells. The concentration of the source fluid for each code value is then specified in data set 6. The values of the constant pressures are taken from the initial pressures specified in data set 7. Additional details and general information are presented by Konikow and Bredehoeft (1978).

Note that ε, S, THICK, WIDTH, α_L, α_T, D_m, and ANFCTR are all assumed to be constant and uniform. If it is desired to specify a different value for any of these parameters at different nodes, then these constants must be changed to arrays and the input and output formats and program statements revised accordingly. The user should also change the input/output formats when those specified do not provide enough significant figures.

A labeled listing of the input data for a sample problem is provided in Appendix II to illustrate the use of the data-input formats for the model. The sample problem is a simple approximation of a cross section through a coastal aquifer in which the freshwater part is subject to contamination (see fig. 4). The right side of the grid is specified as a constant-pressure boundary at hydrostatic saltwater pressures. The freshwater contaminant is introduced in two nodes in the upper left corner of the grid, and uncontaminated freshwater recharges through a constant-pressure boundary over the other top-row cells. This data set also illustrates that only a small data file is required to simulate a relatively simple problem.

Selected output from this sample problem is presented in Appendix III. Not all of the output is reproduced, in order to save space, but a sufficient selection is included to illustrate the type and form of the output provided by the model, as well as to allow the user to compare his output with the documented version for verification of the code.

The initial and boundary conditions for the sample problem result in the freshwater contaminant spreading through the aquifer from the upper left and saltwater moving part way into the aquifer from the right-hand boundary, especially in the deeper part of the aquifer. After the first few time steps, the mass-balance errors for both constituents are consistently less than 5 percent, which is generally acceptable. As a general guideline, execution of this sample problem used less than 1.5 minutes of CPU time on a PRIME 850 computer.

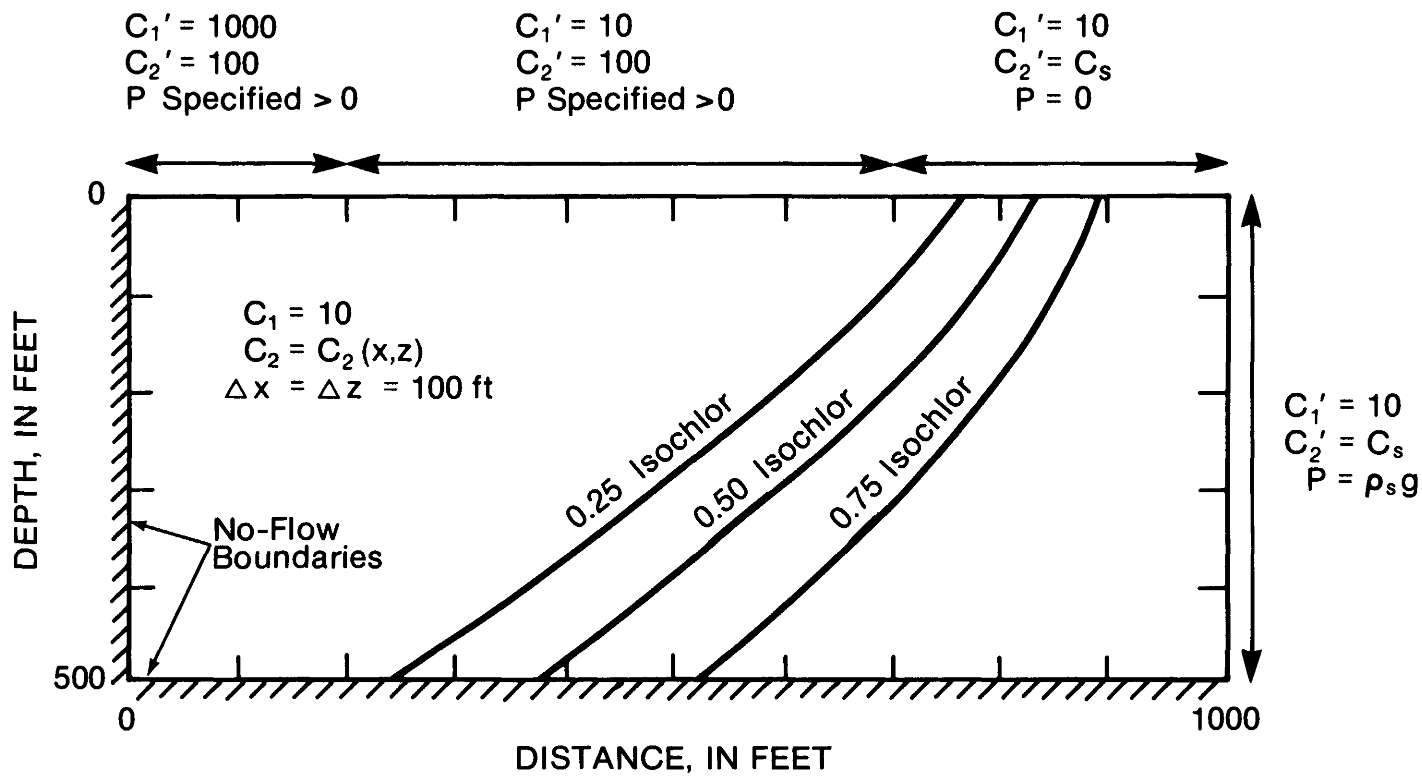

Figure 4. Initial Conditions for Sample Problem

The source code developed for this model was written in Fortran 77. The program is listed in Appendix IV and includes more than 2500 lines. For reference purposes, columns 73-80 of each line contain a label that is numbered sequentially within each subroutine.

SUMMARY AND CONCLUSIONS

For a numerical model to simulate accurately an aquifer system where both freshwater and saltwater exist, the fluid density must be considered as a variable. The introduction of contaminants into such systems requires that the concentrations of at least two dissolved species be represented in a numerical transport simulation model -- one concentration affecting the fluid density and the other representing a contaminant. The model described here will perform such numerical simulations. The model is a modification of the one documented by Konikow and Bredehoeft (1978) in which the equations and numerical methods were modified to represent density-dependent flow and transport of two dissolved constituents in a cross-sectional plane. Other modifications made to the original model include adding a 16-particle-per-cell option, a molecular diffusion option, and the introduction of a particle-weighting scheme.

The variable-density aspects of the model were tested on a problem for which an analytical solution was developed by Henry (1964). The results closely matched the documented results from other numerical models. Therefore, these results, in combination with the previous documentation by Konikow and Bredehoeft (1978) of their model's applicability to constant-density problems, indicate that this new model can be applied to problems involving density-dependent flow and transport of two solutes in a ground-water system.

REFERENCES

Bear, Jacob, 1972, Dynamics of fluids in porous media: New York, Am. Elsevier Publishing Co., 764 p.

Bredehoeft, J. D., and Pinder, G. F., 1973, Mass transport in flowing groundwater: Water Resources Research, v. 9, no. 1, p. 194-210.

Burnham, W. L., Larson, S. P., and Cooper, H. H., Jr., 1977, Distribution of injected wastewater in the saline lava aquifer, Wailuku-Kahului wastewater treatment facility, Kahului, Maui, Hawaii: U.S. Geological Survey Open-File Report 77-469, 58 p.

Cooper, H. H., Jr., 1966, The equation of groundwater flow in fixed and deforming coordinates: Journal of Geophysical Research, v. 71, no. 20, p. 4785-4790.

Henry, H. R., 1964, Effects of dispersion on salt encroachment in coastal aquifers: *In* Seawater in coastal aquifers, U.S. Geological Survey Water-Supply Paper 1613-C, p. C70-C84.

Intera Environmental Consultants, Inc., 1979, Revision of the documentation for a model for calculating effects of liquid waste disposal in deep saline aquifers: U.S. Geological Survey Water-Resources Investigations 79-96, 72 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.

Konikow, L. F., and Grove, D. B., 1977, Derivation of equations describing solute transport in ground water: U.S. Geological Survey Water-Resources Investigations 77-19, 30 p.

Larson, S. P., Papadopulos, S. S., Cooper H. H., Jr., and Burnham, W. L., 1977, Simulation of wastewater injection into a coastal aquifer system near Kahului, Maui, Hawaii: Proceedings of Hydraulics in the Coastal Zone, American Society of Civil Engineers, College Station, Texas, p. 107-116.

Pinder, G. F., and Cooper, H. H., Jr., 1970, A numerical technique for calculating the transient position of the saltwater front: Water Resources Research, v. 6, no. 3, p. 875-882.

Rosenshein, J. S., and Hickey, J. J., 1977, Storage of treated sewage effluent and storm water in a saline aquifer, Pinellas Peninsula, Florida: Ground Water, v. 15, no. 4, p. 285-293.

Scheidegger, A. E., 1961, General theory of dispersion in porous media: Journal of Geophysical Research, v. 66, no. 10, p. 3273-3278.

Segol, G., Pinder, G. F., and Gray, W. G., 1975, A galerkin finite-element technique for calculating the transient position of the saltwater front: Water Resources Research, v. 11, no. 2, p. 343-347.

Trescott, P. C., Pinder, G. F., and Larson, S. P., 1976, Finite-difference model for aquifer simulation in two dimensions with results of numerical experiments: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 7, Chapter C1, 116 p.

Voss, C. I., 1984, A finite-element simulation model for saturated-unsaturated, fluid-density-dependent ground-water flow with energy transport or chemically-reactive single-species solute transport: U.S. Geological Survey Water-Resources Investigations Report 84-4369, 409 p.

Weast, R. C. (ed.), 1981, CRC handbook of chemistry and physics: CRC Press, Inc., Boca Raton, Florida.

APPENDIX I: INPUT DATA SPECIFICATIONS

Line	*Column*	*Format*	*Variable*	*Definition*
1	1-80	10A8	TITLE	Description of problem.
2	1- 4	I4	NTIM	Maximum number of time steps in a pumping period (limit=100)*.
	5- 8	I4	NPMP	Number of pumping periods. Note that if NPMP>1, then data set 11 must be completed.
	9-12	I4	NX	Number of nodes in x direction (limit=24)*.
	13-16	I4	NZ	Number of nodes in z direction (limit=20)*.
	17-20	I4	NPMAX	Maximum number of particles (limit=6400)*.
	21-24	I4	NPNT	Time-step interval for printing hydraulic and chemical output data.
	25-28	I4	NUMOBS	Number of observation points to be specified in a following data set (limit=5)*.
	29-32	I4	ITMAX	Maximum allowable number of iterations in SIP (usually 100≤ITMAX≤200).
	33-36	I4	NREC	Number of pumping or injection wells to be specified in a following data set.
	37-40	I4	NPTPND	Initial number of particles per node (options=4,5,8,9,16).
	41-44	I4	NCODES	Number of node identification codes to be specified in a following data set (limit=10)*.
	45-48	I4	NZCRIT	Maximum number of cells that can be void of particles before particles are redistributed (generally equal

See footnotes at end of table.

Line	*Column*	*Format*	*Variable*	*Definition*
				to 1 to 10 percent of the number of active cells in the grid).
	49-52	I4	NCONST	Number of constituents present (1 or 2).
	53-56	I4	NPNTMV	Particle movement interval (IMOV) for printing chemical output data. (Specify 0 to print only at end of time steps.)
	57-60	I4	NPNTVL	Options for printing computed velocities (0=do not print; 1=print for first time step; 2=print for all time steps).
	61-64	I4	NPNTD	Option for printing computed dispersion equation coefficients (option definition same as for NPNTVL).
	65-68	I4	NPDELC	Option for printing computed changes in concentration (0=do not print; 1=print).
	69-72	I4	NPNCHV	Option to write velocity data in separate file (option definition same as for NPNTVL). When specified, program will write the velocities at nodes on unit 7.
3	1-10	G10.0	PINT	Pumping period in years.
	11-20	G10.0	TOL	Convergence criteria in SIP (usually TOL≤0.01).
	21-30	G10.0	POROS	Effective porosity.
	31-40	G10.0	BETA	Longitudinal dispersivity, in feet.
	41-50	G10.0	S	Storage coefficient (set S=0 for steady-flow problems).
	51-60	G10.0	TIMX	Time increment multiplier for transient flow problems. TIMX is disregarded if S=0.

Line	*Column*	*Format*	*Variable*	*Definition*
	61-70	G10.0	TINIT	Size of initial time step in seconds. TINIT is disregarded if S=0.
4	1-10	G10.0	XDEL	Width of finite-difference cell in x direction, in feet.
	1-20	G10.0	ZDEL	Width in finite-difference cell in z direction, in feet.
	21-30	G10.0	DLTRAT	Ratio of transverse to longitudinal dispersivity.
	31-40	G10.0	CELDIS	Maximum cell distance per particle move (value between 0 and 1.0).
	41-50	G10.0	ANFCTR	Ratio of K_{zz} to K_{xx}.
	51-60	G10.0	WIDTH	Width (third dimension) of the aquifer cross-section, in feet.
	61-70	G10.0	CTOL	Concentration change increment for density-dependent constituent to determine whether pressures are recalculated.
	71-80	G10.0	DMOLEC	Molecular diffusion coefficient, in ft^2/s.

Data Set	*Number of Lines*	*Format*	*Variable*	*Definition*
1	Value of NUMOBS	2I2	IXOBS, IZOBS	x and z coordinates of observation points. This data set is eliminated if NUMOBS is specified as =0.
2	Value of NREC	2I2, 3G10.2	IX,IZ, REC, CNREC, TDSREC	x and z coordinates of pumping (+) or injection (-) wells, rate in ft^3/s, and if an injection well, the concentration in the injected water of the trace constituent, which does not affect density

Data Set	Number of Lines	Format	Variable	Definition
				(CNREC), and of the density-controlling constituent (TDSREC). This data set is eliminated if NREC=0.
3	a. 1	I1,G10.0	INPUT, FCTR	Parameter line† for PERM.
	b. Value of NZ	24G3.0	PERM	The intrinsic permeability of the aquifer, in ft^2.
4	a. 1	I1,G10.0	INPUT, FCTR	Parameter line† for VPRM.
	b. Value of NZ	24G3.0	VPRM	Leakance coefficient, in $ft^{-1}s^{-1}$.
	c. Value of NZ	24G3.0	ELEV	Elevation difference between the source bed and the aquifer node (negative if confining bed is below aquifer), in feet.
5	a. 1	I1,G10.0	INPUT, FCTR	Parameter line† for NODEID.
	b. Value of NZ	24I1	NODEID	Node identification matrix (use a nonzero value to define constant-pressure nodes).
6	Value of NCODES	I2,2G10.2	ICODE, FCTR1, FCTR2	Boundary codes and concentrations of constituents 1 and 2, respectively, in the source fluid where NODEID=ICODE.
7	a. 1	I1,G10.0	INPUT, FCTR	Parameter line† for PI.
	b. Value of NZ (double if NX>12).	12G6.0	PI	Initial fluid pressures in the aquifer, in lb/ft^2.
8	a. 1	I1,G10.0	INPUT, FCTR	Parameter card† for CONC.
	b. Value of NZ (double if NX>12).	12G6.0	CONC	Initial concentration of the trace constitutent in the aquifer. This data set is eliminated if NCONST<2.

See footnotes at end of table.

Data Set	Number of Lines	Format	Variable	Definition
9	a. 1	I1,G10.0	INPUT, FCTR	Parameter line† for TDS.
	b. Value of NZ (double if NX>12).	12G6.0	TDS	Initial concentration of the density-controlling constituent in the aquifer.
10	a. 1	I1	INPUT	Density and viscosity default parameter: 0 = yes, 1 = no.
	b. 1	4G10.3	DEN1, DEN2, VIS1, VIS2	Slopes and intercepts of linear relations between TDS and (1) density and (2) viscosity, respectively. Only read if INPUT=1 previous part a.
11				This data set allows time step parameters, print options, and pumpage data to be revised for each pumping period of the simulation. Data set 11 is only used if NPMP>1. The sequence of lines in data set 11 must be repeated (NPMP-1) times (that is, data set 11 is required for each pumping period after the first).
	a. 1	I1	ICHK	Parameter to check whether any revisions are desired. Set ICHK=1 if data are to be revised, and then complete data set 11b and c. Set ICHK=0 if data are not to be revised for the next pumping period, and skip rest of data set 11.
	b. 1	9I4,3G5.0	NTIM,NPNT, ITMAX,NREC, NPNTMV, NPNTVL, NPNTD, NPDELC, NPNCHV, PINT,TIMX, TINIT	Twelve parameters to be revised for next pumping period; the parameters were previously defined in the description of data lines 2 and 3. Only include this line if ICHK=1 in previous part a.

See footnotes at end of table.

Data Set	*Number of Lines*	*Format*	*Variable*	*Definition*
	c. Value of NREC	2I2,3G10.2	IX,IZ,REC, CNREC, TDSREC	Revision of previously defined data set 2. Include part c only if ICHK=1 in previous part a and if NREC>0 in previous part b.

* These limits can be modified if necessary by changing the corresponding array dimensions in the COMMON statements of the program.

† The parameter line must be the first line of the indicated data sets. It is used to specify whether the parameter is constant and uniform, and can be defined by one value, or whether it varies in space and must be defined at each node. If INPUT=0, the data set has a constant value, which is defined by FCTR. If INPUT=1, the data set is read as described by part b (and part C, if applicable) immediately following the parameter line. Then FCTR is a multiplication factor for the values read in part b of the data set.

Appendix II:

Input Data from a Sample Problem

```
LINE 1:    | CROSS-SECTIONAL PROBLEM WITH VARIABLE DENSITY
LINE 2:    |   1   1  12    74000    1    1 100    0  16    3    5    2    1    0    0    0    0
LINE 3:    |        10.      .000001        0.20        100.        0.0        0.0        0.0
LINE 4:    |       100.       100.           1.0        0.25        1.0        100.      1000.        0.0
DATA SET 1:| 7 4
DATA SET 3:|1 1.338E-11
           |0  0  0  0  0  0  0  0  0  0  0  0
           |0  1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0
           |0  1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0
           |0  1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0
           |0  1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0
           |0  1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0
           |0  0  0  0  0  0  0  0  0  0  0  0
DATA SET 4:|0    0.0
DATA SET 5:|1     1.
           |000000000000
           |033222221110
           |000000000010
           |000000000010
           |000000000010
           |000000000010
           |000000000000
DATA SET 6:| 1   35000.    10.
           | 2     100.    10.
           | 3     100.  1000.
DATA SET 7:|1   10.0
           |     0     0     0     0     0     0     0     0     0     0     0     0
           |     0    65    60    50    40    30    20    10     0     0     0     0
           |     0     0     0     0     0     0     0     0     0     0   641     0
           |     0     0     0     0     0     0     0     0     0     0  1282     0
           |     0     0     0     0     0     0     0     0     0     0  1923     0
           |     0     0     0     0     0     0     0     0     0     0  2564     0
           |     0     0     0     0     0     0     0     0     0     0     0     0
DATA SET 8:|0    10.0
DATA SET 9:|1    10.0
           |     0     0     0     0     0     0     0     0     0     0     0     0
           |     0    14    20    33    56   103   232   481   913  2573  3203     0
           |     0    19    27    44    81   178   504  1051  1822  2847  3237     0
           |     0    30    42    79   187   443  1175  1972  2575  3084  3314     0
           |     0    47    67   135   367   951  1919  2596  2975  3263  3388     0
           |     0    89   129   294   920  1554  2404  2933  3205  3367  3435     0
           |     0     0     0     0     0     0     0     0     0     0     0     0
DATA SET 10:|0
```


page 33 follows

Appendix III:

Selected Output from a Sample Problem

```
U.S.G.S. METHOD-OF-CHARACTERISTICS MODEL FOR SOLUTE TRANSPORT IN GROUND WATER

 CROSS-SECTIONAL PROBLEM WITH VARIABLE DENSITY

             I N P U T    D A T A

               GRID DESCRIPTORS

      NX     (NUMBER OF COLUMNS)  =    12
      NZ     (NUMBER OF ROWS)     =     7
      XDEL   (X-DISTANCE IN FEET) =   100.000
      ZDEL   (Z-DISTANCE IN FEET) =   100.000
      WIDTH  (Y-DISTANCE IN FEET) =   100.000

               TIME  PARAMETERS

      NTIM    (MAX. NO. OF TIME STEPS)        =       1
      NPMP    (NO. OF PUMPING PERIODS)        =       1
      PINT    (PUMPING PERIOD IN YEARS)       =      10.000
      TIMX    (TIME INCREMENT MULTIPLIER)     =       0.00
      TINIT   (INITIAL TIME STEP IN SEC.)     =       0.

        HYDROLOGIC AND CHEMICAL PARAMETERS

      S        (SPECIFIC STORAGE)            =       0.000000
      POROS    (EFFECTIVE POROSITY)          =       0.20
      BETA     (LONGITUDINAL DISPERSIVITY)   =     100.0
      DLTRAT   (RATIO OF TRANSVERSE TO
                LONGITUDINAL DISPERSIVITY)   =       1.00
      ANFCTR   (RATIO OF K-ZZ TO K-XX)       =       1.000000
      DMOLEC   (COEF. OF DIFFUSION)          =    0.00E-01
      NCONST   (NUMBER OF CONSTITUENTS)      =    2

             EXECUTION PARAMETERS

      TOL     (CONVERGENCE CRITERIA - SIP)   = 0.0000010
      ITMAX   (MAX.NO.OF ITERATIONS - SIP)   =  100
      CELDIS  (MAX.CELL DISTANCE PER MOVE
                 OF PARTICLES - M.O.C.)      =    0.250
      NPMAX   (MAX. NO. OF PARTICLES)        = 4000
      NPTPND  (NO. PARTICLES PER NODE)       =    16
      CTOL    (MINIMUM CONC. CHANGE
               FOR PRESSURE RECALCULATION)   = 1000.

               PROGRAM OPTIONS

      NPNT    (TIME STEP INTERVAL FOR
               COMPLETE PRINTOUT)       =     1
      NPNTMV  (MOVE INTERVAL FOR CHEM.
               CONCENTRATION PRINTOUT)  =     1
      NPNTVL  (PRINT OPTION-VELOCITY
               0=NO; 1=FIRST TIME STEP;
               2=ALL TIME STEPS)        =     0
      NPNTD   (PRINT OPTION-DISP.COEF.
               0=NO; 1=FIRST TIME STEP;
               2=ALL TIME STEPS)        =     0
      NUMOBS  (NO. OF OBSERVATION WELLS
               FOR HYDROGRAPH PRINTOUT) =     1
      NREC    (NO. OF RECHARGE CELLS)   =     0
      NCODES  (FOR NODE IDENT.)         =     3
      NPNCHV  (WRITE VELOCITIES-UNIT 7) =     0
      NPDELC  (PRINT OPT.-CONC. CHANGE) =     0
```

```
               STEADY-STATE FLOW

     TIME INTERVAL (IN SEC) FOR SOLUTE-TRANSPORT SIMULATION =  0.31558E+09

           LOCATION OF OBSERVATION WELLS

                 NO.      X      Z

                  1       7      4

           AREA OF ONE CELL =    0.1000E+05

           X-Z SPACING:
              100.00
              100.00
```

```
PERMEABILITY MAP (FT**2)
  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01
  0.00E-01  0.00E-01
  0.00E-01  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11
  1.34E-11  0.00E-01
  0.00E-01  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11
  1.34E-11  0.00E-01
  0.00E-01  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11
  1.34E-11  0.00E-01
  0.00E-01  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11
  1.34E-11  0.00E-01
  0.00E-01  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11  1.34E-11
  1.34E-11  0.00E-01
  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01
  0.00E-01  0.00E-01
```

```
           NO. OF FINITE-DIFFERENCE CELLS IN AQUIFER =    50

           AREA OF AQUIFER IN MODEL  =  0.50000E+06    SQ. FT.


           NZCRIT   (MAX. NO. OF CELLS THAT CAN BE VOID OF
                     PARTICLES;  IF EXCEEDED, PARTICLES ARE REGENERATED)   =     5
```

```
NODE IDENTIFICATION MAP

     0    0    0    0    0    0    0    0    0    0    0    0
     0    3    3    2    2    2    2    2    1    1    1    0
     0    0    0    0    0    0    0    0    0    0    1    0
     0    0    0    0    0    0    0    0    0    0    1    0
     0    0    0    0    0    0    0    0    0    0    1    0
     0    0    0    0    0    0    0    0    0    0    1    0
     0    0    0    0    0    0    0    0    0    0    0    0

       NO. OF NODE IDENT. CODES SPECIFIED =   3

            THE FOLLOWING ASSIGNMENTS HAVE BEEN MADE:
     CODE NO.      TDS CONC.           CONC.
         1         35000.00             10.00
         2           100.00             10.00
         3           100.00           1000.00
```

```
VERTICAL PERMEABILITY FACTOR (1/FT*SEC)
  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01
  0.00E-01  0.00E-01
  0.00E-01  1.34E-03  1.34E-03  1.34E-03  1.34E-03  1.34E-03  1.34E-03  1.34E-03  1.34E-03  1.34E-03
  1.34E-03  0.00E-01
  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01
  1.34E-03  0.00E-01
  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01
  1.34E-03  0.00E-01
  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01
  1.34E-03  0.00E-01
  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01
  1.34E-03  0.00E-01
  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01
  0.00E-01  0.00E-01
```

```
INITIAL PRESSURES (LB/FT**2)

   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
   0.  650.  600.  500.  400.  300.  200.  100.    0.    0.    0.    0.
   0.    0.    0.    0.    0.    0.    0.    0.    0.    0. 6410.    0.
   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.12820.    0.
   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.19230.    0.
   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.25640.    0.
   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.


PRESSURE DISTRIBUTION - ROW
NUMBER OF TIME STEPS =     0
      TIME(SECONDS) =  0.00000
      TIME(DAYS)    =  0.00000E-01
      TIME(YEARS)   =  0.00000E-01

     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000

     0.0000     0.0000

     0.0000   650.0000   600.0000   500.0000   400.0000   300.0000   200.0000   100.0000     0.0000     0.0000

     0.0000     0.0000

     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000

  6410.0000     0.0000

     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000

 12820.0000     0.0000

     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000

 19230.0000     0.0000

     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000

 25640.0000     0.0000

     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000

     0.0000     0.0000


INITIAL CONCENTRATION MAP - TRACE SOLUTE

   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
   0.   10.   10.   10.   10.   10.   10.   10.   10.   10.   10.    0.
   0.   10.   10.   10.   10.   10.   10.   10.   10.   10.   10.    0.
   0.   10.   10.   10.   10.   10.   10.   10.   10.   10.   10.    0.
   0.   10.   10.   10.   10.   10.   10.   10.   10.   10.   10.    0.
   0.   10.   10.   10.   10.   10.   10.   10.   10.   10.   10.    0.
   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.

INITIAL TDS MAP - DENSITY-CONTROLLING SOLUTE

   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
   0.  140.  200.  330.  560. 1030. 2320. 4810. 9130.25730.32030.    0.
   0.  190.  270.  440.  810. 1780. 5040.10510.18220.28470.32370.    0.
   0.  300.  420.  790. 1870. 4430.11750.19720.25750.30840.33140.    0.
   0.  470.  670. 1350. 3670. 9510.19190.25960.29750.32630.33880.    0.
   0.  890. 1290. 2940. 9200.15540.24040.29330.32050.33670.34350.    0.
   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
```

```
INITIAL DENSITIES (LB/FT**3)

 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
 0.00 62.44 62.44 62.45 62.46 62.48 62.54 62.66 62.86 63.65 63.95  0.00
 0.00 62.44 62.44 62.45 62.47 62.51 62.67 62.93 63.29 63.78 63.97  0.00
 0.00 62.44 62.45 62.47 62.52 62.64 62.99 63.37 63.65 63.89 64.00  0.00
 0.00 62.45 62.46 62.49 62.60 62.88 63.34 63.66 63.84 63.98 64.04  0.00
 0.00 62.47 62.49 62.57 62.87 63.17 63.57 63.82 63.95 64.03 64.06  0.00
 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00

INITIAL VISCOSITIES (LB*SEC/FT**2)

  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01
  0.00E-01  0.00E-01
  0.00E-01  2.09E-05  2.09E-05  2.09E-05  2.09E-05  2.09E-05  2.10E-05  2.11E-05  2.12E-05  2.18E-05
  2.21E-05  0.00E-01
  0.00E-01  2.09E-05  2.09E-05  2.09E-05  2.09E-05  2.10E-05  2.11E-05  2.13E-05  2.15E-05  2.20E-05
  2.22E-05  0.00E-01
  0.00E-01  2.09E-05  2.09E-05  2.09E-05  2.10E-05  2.10E-05  2.13E-05  2.16E-05  2.18E-05  2.21E-05
  2.22E-05  0.00E-01
  0.00E-01  2.09E-05  2.09E-05  2.09E-05  2.10E-05  2.12E-05  2.16E-05  2.19E-05  2.20E-05  2.22E-05
  2.22E-05  0.00E-01
  0.00E-01  2.09E-05  2.09E-05  2.10E-05  2.12E-05  2.14E-05  2.18E-05  2.20E-05  2.21E-05  2.22E-05
  2.23E-05  0.00E-01
  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01  0.00E-01
  0.00E-01  0.00E-01


CONCENTRATION

NUMBER OF TIME STEPS =      0
       TIME(SECONDS) =  0.00000
  CHEM.TIME(SECONDS) =  0.00000E-01
  CHEM.TIME(DAYS)    =  0.00000E+00
       TIME(YEARS)   =  0.00000E+00
  CHEM.TIME(YEARS)   =  0.00000E-01
 NO. MOVES COMPLETED =      0

              TRACE SOLUTE

    0    0    0    0    0    0    0    0    0    0    0    0

    0   10   10   10   10   10   10   10   10   10   10    0

    0   10   10   10   10   10   10   10   10   10   10    0

    0   10   10   10   10   10   10   10   10   10   10    0

    0   10   10   10   10   10   10   10   10   10   10    0

    0   10   10   10   10   10   10   10   10   10   10    0

    0    0    0    0    0    0    0    0    0    0    0    0

              DENSITY-CONTROLLING SOLUTE

    0    0    0    0    0    0    0    0    0    0    0    0

    0  140  200  330  560 1030 2320 4810 91302573032030    0

    0  190  270  440  810 1780 50401051018220284703237O    0

    0  300  420  790 1870 44301175019720257503084033140    0

    0  470  670 1350 3670 95101919025960297503263033880    0

    0  890 1290 2940 92001554024040293303205033670343 50    0

    0    0    0    0    0    0    0    0    0    0    0    0
BETA= 1.00
 10 ITERATIONS PARAMETERS:
   0.213163E-13   0.680943E+00   0.898203E+00   0.967521E+00   0.989637E+00   0.213163E-13
   0.680943E+00   0.898203E+00   0.967521E+00   0.989637E+00


  N =     1
NUMBER OF ITERATIONS=  22
```

```
PRESSURE DISTRIBUTION - ROW
NUMBER OF TIME STEPS =      1
      TIME(SECONDS) =  0.31558E+09
      TIME(DAYS)    =  3.65250E+03
      TIME(YEARS)   =  1.00000E+01

     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     0.0000     0.0000
     0.0000   649.9996   599.9997   499.9999   400.0000   300.0001   200.0001   100.0002     0.0006     0.0001
     0.0000     0.0000
     0.0000  6820.6004  6787.5782  6726.6893  6655.0527  6579.3778  6503.9940  6438.2527  6394.2279  6397.1654
  6410.0000     0.0000
     0.0000 13024.5980 13003.5358 12965.1986 12917.2128 12866.4306 12820.5938 12790.1387 12781.3106 12796.6829
 12819.9999     0.0000
     0.0000 19250.3172 19237.7063 19215.4853 19188.8894 19166.7538 19155.2752 19157.1112 19171.7907 19198.2502
 19229.9999     0.0000
     0.0000 25490.0426 25483.5405 25475.2166 25473.4440 25482.7521 25505.8228 25534.1522 25565.5755 25601.3318
 25639.9999     0.0000
     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     0.0000     0.0000


  PRESSURE MAP

     0     0     0     0     0     0     0     0     0     0     0     0
     0   650   600   500   400   300   200   100     0     0     0     0
     0  6821  6788  6727  6655  6579  6504  6438  6394  6397  6410     0
     0 13025 13004 12965 12917 12866 12821 12790 12781 12797 12820     0
     0 19250 19238 19215 19189 19167 19155 19157 19172 19198 19230     0
     0 25490 25484 25475 25473 25483 25506 25534 25566 25601 25640     0
     0     0     0     0     0     0     0     0     0     0     0     0
   CUMULATIVE MASS BALANCE -- (IN FT**3)

    RECHARGE                       =  0.00000E-01
    DISCHARGE                      =  0.00000E-01
    CUMULATIVE  NET  RECHARGE      =  0.00000E-01
    WATER RELEASE FROM STORAGE     =  0.00000E-01
    LEAKAGE INTO AQUIFER           =  4.59854E+02
    LEAKAGE OUT OF AQUIFER         = -4.59854E+02
    CUMULATIVE  NET  LEAKAGE       =  3.03705E-04

        MASS BALANCE RESIDUAL      =  0.30371E-03
        ERROR  (AS PERCENT)        =  0.66044E-04

   RATE MASS BALANCE -- (IN C.F.S.)

    LEAKAGE INTO AQUIFER           =  1.45719E+00
    LEAKAGE OUT OF AQUIFER         = -1.45719E+00
    NET LEAKAGE     (QNET)         =  9.62384E-07
    RECHARGE                       =  0.00000E-01
    DISCHARGE                      =  0.00000E-01
    NET WITHDRAWAL    (TPUM)       =  0.00000E-01
```

```
         STABILITY CRITERIA --- M.O.C.


 VMAX =  3.20E-06      VMAZ =  1.96E-06
 VMXBD=  3.20E-06      VMZBD=  2.70E-06
 TMV (MAX. INJ.) =  0.17021E+08
 TIMV (CELDIS)   =  0.78099E+07

 TIMV =  7.81E+06      NTIMV =    40     NMOV =    41

  TIM (N)   =  0.31558E+09
  TIMEVELO =  0.76970E+07
  TIMEDISP =  0.77580E+07

 TIMV =  7.70E+06      NTIMD =    40     NMOV =    41

         THE LIMITING STABILITY CRITERION IS CELDIS

         NO. OF PARTICLE MOVES REQUIRED TO COMPLETE THIS TIME STEP  =   41


  NP        =           1051           IMOV      =              1
  TIM(N)    =  0.31558E+09             TIMV      =  0.76970E+07            SUMTCH =  0.76970E+07

RECALCULATE PRESSURES DUE TO CONCENTRATION CHANGE


  N =    1
NUMBER OF ITERATIONS=  13




         STABILITY CRITERIA --- M.O.C.


 VMAX =  3.20E-06      VMAZ =  2.02E-06
 VMXBD=  3.20E-06      VMZBD=  2.73E-06
 TMV (MAX. INJ.) =  0.16990E+08
 TIMV (CELDIS)   =  0.78099E+07

 TIMV =  7.81E+06      NTIMV =    40     NMOV =    41

  TIM (N)   =  0.31558E+09
  TIMEVELO =  0.76970E+07
  TIMEDISP =  0.77573E+07

 TIMV =  7.70E+06      NTIMD =    40     NMOV =    41

         THE LIMITING STABILITY CRITERION IS CELDIS
```

```
CONCENTRATION

NUMBER OF TIME STEPS =       1
       DELTA T        =  0.31558E+09
       TIME(SECONDS)  =  0.31558E+09
  CHEM.TIME(SECONDS)  =  7.69698E+06
  CHEM.TIME(DAYS)     =  0.89085E+02
       TIME(YEARS)    =  0.10000E+02
  CHEM.TIME(YEARS)    =  2.43902E-01
 NO. MOVES COMPLETED =       1

               TRACE SOLUTE

    0    0    0    0    0    0    0    0    0    0    0    0

    0  311  270   10   10   10   10   10   10   10   10    0

    0   10   10   10   10   10   10   10   10   10   10    0

    0   10   10   10   10   10   10   10   10   10   10    0

    0   10   10   10   10   10   10   10   10   10   10    0

    0   10   10   10   10   10   10   10   10   10   10    0

    0    0    0    0    0    0    0    0    0    0    0    0

               DENSITY-CONTROLLING SOLUTE

    0    0    0    0    0    0    0    0    0    0    0    0

    0  145  207  336  608 1257 2962 66041170824958318940    0

    0  195  280  467  945 2251 5112 96061617227554322870    0

    0  306  445  874 2017 481210623183302507030667331200    0

    0  477  702 1489 3924 92471852625850297403251333869    0

    0  888 1298 2916 909415648237842902431838335163433 5    0

    0    0    0    0    0    0    0    0    0    0    0    0

CHEMICAL MASS BALANCE
               TRACE SOLUTE

       MASS IN BOUNDARIES      =  1.13861E+08
       MASS OUT BOUNDARIES     = -1.78475E+06
       MASS PUMPED IN          =  0.00000E-01
       MASS PUMPED OUT         =  0.00000E-01
       INFLOW MINUS OUTFLOW    =  1.12076E+08
       INITIAL MASS STORED     =  1.00000E+08
       PRESENT MASS STORED     =  2.12079E+08
       CHANGE MASS STORED      =  1.12079E+08
     COMPARE RESIDUAL WITH NET FLUX AND MASS ACCUMULATION:
       MASS BALANCE RESIDUAL   = -2.70385E+03
       ERROR  (AS PERCENT)     = -2.37469E-03
     COMPARE INITIAL MASS STORED WITH CHANGE IN MASS STORED:
       ERROR  (AS PERCENT)     = -2.23896E-02

CHEMICAL MASS BALANCE

               DENSITY-CONTROLLING SOLUTE

       MASS IN BOUNDARIES      =  1.97450E+09
       MASS OUT BOUNDARIES     = -1.70081E+09
       MASS PUMPED IN          =  0.00000E-01
       MASS PUMPED OUT         =  0.00000E-01
       INFLOW MINUS OUTFLOW    =  2.73695E+08
       INITIAL MASS STORED     =  1.28754E+11
       PRESENT MASS STORED     =  1.28098E+11
       CHANGE MASS STORED      = -6.56057E+08
     COMPARE RESIDUAL WITH NET FLUX AND MASS ACCUMULATION:
       MASS BALANCE RESIDUAL   =  9.29752E+08
       ERROR  (AS PERCENT)     =  4.70879E+01
     COMPARE INITIAL MASS STORED WITH CHANGE IN MASS STORED:
       ERROR  (AS PERCENT)     =  7.23653E-01
```

```
CONCENTRATION

NUMBER OF TIME STEPS =      1
       DELTA T       =  0.31558E+09
       TIME(SECONDS) =  0.31558E+09
  CHEM.TIME(SECONDS) =  3.15576E+08
  CHEM.TIME(DAYS)    =  0.36525E+04
       TIME(YEARS)   =  0.10000E+02
  CHEM.TIME(YEARS)   =  1.00000E+01
 NO. MOVES COMPLETED =     41

               TRACE SOLUTE

    0    0    0    0    0    0    0    0    0    0    0    0

    0  951  902  777  711  618  509  407  308  143   50    0

    0  904  861  769  671  566  457  338  228  101   41    0

    0  807  756  671  556  442  356  231  114   48   24    0

    0  604  536  476  372  296  182   96   41   22   15    0

    0  212  228  216  154   90   53   35   22   15   12    0

    0    0    0    0    0    0    0    0    0    0    0    0

               DENSITY-CONTROLLING SOLUTE

    0    0    0    0    0    0    0    0    0    0    0    0

    0  216  376  774 1617 3083 5536 915813107218342882 4    0

    0  311  480  945 1906 3686 67231120816999251783022 5    0

    0  449  669 1276 2671 5156 89051542223760296363241 7    0

    0  603  929 1811 4126 818415586233382959532353365 0    0

    0  806 1169 2493 88141783224377285463144933208341 20    0

    0    0    0    0    0    0    0    0    0    0    0    0
CHEMICAL MASS BALANCE
               TRACE SOLUTE

       MASS IN BOUNDARIES      =  4.63830E+09
       MASS OUT BOUNDARIES     = -1.27448E+09
       MASS PUMPED IN          =  0.00000E-01
       MASS PUMPED OUT         =  0.00000E-01
       INFLOW MINUS OUTFLOW    =  3.36383E+09
       INITIAL MASS STORED     =  1.00000E+08
       PRESENT MASS STORED     =  3.59895E+09
       CHANGE MASS STORED      =  3.49895E+09
     COMPARE RESIDUAL WITH NET FLUX AND MASS ACCUMULATION:
       MASS BALANCE RESIDUAL   = -1.35126E+08
       ERROR  (AS PERCENT)     = -2.91327E+00
     COMPARE INITIAL MASS STORED WITH CHANGE IN MASS STORED:
       ERROR  (AS PERCENT)     = -4.14012E+00

CHEMICAL MASS BALANCE

               DENSITY-CONTROLLING SOLUTE

       MASS IN BOUNDARIES      =  9.17920E+10
       MASS OUT BOUNDARIES     = -9.34962E+10
       MASS PUMPED IN          =  0.00000E-01
       MASS PUMPED OUT         =  0.00000E-01
       INFLOW MINUS OUTFLOW    = -1.70416E+09
       INITIAL MASS STORED     =  1.28754E+11
       PRESENT MASS STORED     =  1.27107E+11
       CHANGE MASS STORED      = -1.64682E+09
     COMPARE RESIDUAL WITH NET FLUX AND MASS ACCUMULATION:
       MASS BALANCE RESIDUAL   = -5.73390E+07
       ERROR  (AS PERCENT)     = -6.24662E-02
     COMPARE INITIAL MASS STORED WITH CHANGE IN MASS STORED:
       ERROR  (AS PERCENT)     = -4.39520E-02
```

CROSS-SECTIONAL PROBLEM WITH VARIABLE DENSITY

TIME VERSUS HEAD AND CONCENTRATION AT SELECTED OBSERVATION POINTS

PUMPING PERIOD NO. 1

STEADY-STATE SOLUTION

OBS.WELL NO.	X	Z	N	PRESSURE (LB/FT**2)	CONC.(MG/L)	TDS (MG/L)	TIME (YEARS)
1	7	4					
			0	0.0	10.0	11750.0	0.000
			1	12821.0	10.0	10623.1	0.244
			2	12825.2	10.0	9589.7	0.488
			3	12825.7	10.0	10883.2	0.732
			4	12826.3	10.0	9930.0	0.976
			5	12827.8	10.3	9590.4	1.220
			6	12827.8	11.4	10155.3	1.463
			7	12827.9	13.6	9756.9	1.707
			8	12827.9	17.4	9240.1	1.951
			9	12827.9	21.8	10335.0	2.195
			10	12827.9	28.5	10244.6	2.439
			11	12828.1	37.0	9789.1	2.683
			12	12827.7	46.4	9839.2	2.927
			13	12829.3	57.2	9446.7	3.171
			14	12829.3	66.2	10047.3	3.415
			15	12827.4	74.7	9241.7	3.659
			16	12827.4	80.5	9904.2	3.902
			17	12827.2	97.5	9458.6	4.146
			18	12828.5	108.5	9306.3	4.390
			19	12829.0	115.5	9185.0	4.634
			20	12829.0	133.6	9080.0	4.878
			21	12829.5	139.4	9111.4	5.122
			22	12829.5	155.9	8903.1	5.366
			23	12827.4	147.9	9098.9	5.610
			24	12827.4	152.1	9566.0	5.854
			25	12827.4	180.7	9222.6	6.098
			26	12827.2	200.0	9218.4	6.341
			27	12827.2	201.1	9149.8	6.585
			28	12827.2	213.8	8792.0	6.829
			29	12827.9	231.9	8636.0	7.073
			30	12827.9	248.0	8616.8	7.317
			31	12828.2	238.9	9132.1	7.561
			32	12828.2	256.3	8849.4	7.805
			33	12828.2	275.1	8618.6	8.049
			34	12826.8	280.2	8831.9	8.293
			35	12826.8	278.2	9347.3	8.537
			36	12826.8	274.1	9406.1	8.780
			37	12828.0	297.8	9426.6	9.024
			38	12828.0	310.2	9497.1	9.268
			39	12828.0	309.0	9902.6	9.512
			40	12828.0	339.2	9392.2	9.756
			41	12829.1	356.0	8905.4	10.000

```
PRESSURE DISTRIBUTION - ROW
NUMBER OF TIME STEPS =      1
       TIME(SECONDS) =  0.31558E+09
       TIME(DAYS)    =  3.65250E+03
       TIME(YEARS)   =  1.00000E+01


   0.0000      0.0000      0.0000      0.0000      0.0000      0.0000      0.0000     0.0000      0.0000      0.0000

   0.0000      0.0000

   0.0000    649.9996    599.9997    499.9999    400.0000    300.0001    200.0001   100.0002      0.0006      0.0001

   0.0001      0.0000

   0.0000   6821.9500   6789.1432   6728.8278   6658.4667   6585.2778   6513.8730  6448.9856   6400.5963   6396.5393

6409.9999      0.0000

   0.0000  13027.2510  13006.4717  12968.8480  12922.1782  12873.5568  12829.0524 12796.2058  12782.2901  12794.2695

12819.9999     0.0000

   0.0000  19254.0165  19241.6942  19219.9270  19193.0369  19168.3022  19153.5298 19153.2745  19168.1763  19195.4638

19229.9999     0.0000

   0.0000  25493.9469  25487.5349  25478.9805  25476.2633  25483.1227  25500.2442 25526.4975  25560.1024  25597.9893

25639.9999     0.0000

   0.0000      0.0000      0.0000      0.0000      0.0000      0.0000      0.0000     0.0000      0.0000      0.0000

   0.0000      0.0000



PRESSURE MAP

     0     0     0     0     0     0     0     0     0     0     0     0

     0   650   600   500   400   300   200   100     0     0     0     0

     0  6822  6789  6729  6658  6585  6514  6449  6401  6397  6410     0

     0 13027 13006 12969 12922 12874 12829 12796 12782 12794 12820     0

     0 19254 19242 19220 19193 19168 19154 19153 19168 19195 19230     0

     0 25494 25488 25479 25476 25483 25500 25526 25560 25598 25640     0

     0     0     0     0     0     0     0     0     0     0     0     0
 CUMULATIVE MASS BALANCE -- (IN FT**3)

  RECHARGE                     =  0.00000E-01
  DISCHARGE                    =  0.00000E-01
  CUMULATIVE  NET  RECHARGE    =  0.00000E-01
  WATER RELEASE FROM STORAGE   =  0.00000E-01
  LEAKAGE INTO AQUIFER         =  1.09362E+04
  LEAKAGE OUT OF AQUIFER       = -1.09362E+04
  CUMULATIVE  NET  LEAKAGE     =  8.09610E-03

      MASS BALANCE RESIDUAL    =  0.80961E-02
      ERROR  (AS PERCENT)      =  0.74031E-04


 RATE MASS BALANCE -- (IN C.F.S.)

  LEAKAGE INTO AQUIFER         =  1.53639E+00
  LEAKAGE OUT OF AQUIFER       = -1.53639E+00
  NET LEAKAGE     (QNET)       =  5.70576E-07
  RECHARGE                     =  0.00000E-01
  DISCHARGE                    =  0.00000E-01
  NET WITHDRAWAL    (TPUM)     =  0.00000E-01
```

Appendix IV:

Program Listing

```
C     *************************************************************
C     *                                                           *
C     *     SOLUTE TRANSPORT AND DISPERSION IN A POROUS MEDIUM    *
C     *      NUMERICAL SOLUTION --- METHOD OF CHARACTERISTICS     *
C     *      PROGRAMMED BY L. F. KONIKOW AND J. D. BREDEHOEFT     *
C     *       MODIFIED BY W. E. SANFORD FOR CROSS-SECTIONAL       *
C     *          2-DIMENSIONAL FLOW WITH VARIABLE DENSITY         *
C     *                AND TWO DISSOLVED CONSTITUENTS             *
C     *                 AUGUST 1984 - NOVEMBER 1985               *
C     *                                                           *
C     *                                                           *
C     *************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /PRMJ/ NTIM,NPMP,NPNT,NITP,N,NX,NZ,NP,NREC,INT,NNX,NNZ,NUMO
     1BS,NMOV,IMOV,NPMAX,ITMAX,NZCRIT,IPRNT,NPTPND,NPNTMV,NPNTVL,NPNTD,N
     2PNCHV,NPDELC,ICHK,NCONST
      COMMON /PRMC/ NODEID(24,20),NPCELL(24,20),NPOLD(24,20),LIMBO(500),
     1IXOBS(5),IZOBS(5)
      COMMON /PRESS/ PERM(24,20),PMRX(24,20,4),PI(24,20),PR(24,20),PC(24
     1,20),PK(24,20),REC(24,20),DENS(24,20),GTERM(24,20),VISC(24,20),VPR
     2M(24,20),TMWL(5,50),TMOBS(50),TIM(100),AOPT(20),TITLE(10),ANFCTR,X
     3DEL,ZDEL,WIDTH,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,VOL
      COMMON /CHMA/ PART(4,6400),CONC(24,20),TDS(24,20),VX(24,20),VZ(24,
     120),CONINT(24,20),TDSINT(24,20),CNREC(24,20),TDSREC(24,20),TMCN(5,
     250),TMTDS(5,50),POROS,SUMTCH,BETA,TIMV,STORM,STORT,STORMI,STORTI,C
     3MSIN,TDSIN,CMSOUT,TDSOUT,FLMIN,FLTIN,FLMOT,FLTOT,SUMIO,TDSIO,CELDI
     4S,DLTRAT,CSTORM,CSTORT,DMOLEC
      COMMON /CHMC/ SUMC(24,20),VXBDY(24,20),VZBDY(24,20),SUMTDS(24,20),
     1WTFCTR(24,20),SUMWT(24,20),PTQ(24,20),PTWT(6400),ELEV(24,20)
C     ************************************************************
C      ---LOAD DATA---
      INT=0
      TMSUM=0.0
      CALL PARLOD
      CALL GENPT
C     ************************************************************
C     ---START COMPUTATIONS---
C        ---COMPUTE ONE PUMPING PERIOD---
      DO 160 INT=1,NPMP
      IF (INT.GT.1) TMSUM=TMSUM+PYR
      IF (INT.GT.1) CALL PARLOD
      IPCK=0
C        ---COMPUTE ONE TIME STEP---
      DO 140 N=1,NTIM
      IPRNT=0
C        ---LOAD NEW DELTA T---
      TINT=SUMT-TMSUM
      TDEL=DMIN1(TIM(N),PYR-TINT)
      SUMT=SUMT+TDEL
      IF (TDEL.EQ.(PYR-TINT)) IPCK=1
      TIM(N)=TDEL
      REMN=MOD(N,NPNT)
C     ************************************************************
```

```
      IF (S.EQ.0.0.AND.ICHK.EQ.0.AND.(N.GT.1.OR.INT.GT.1)) GO TO 110
      CALL ITERAT
      IF (REMN.EQ.0.0.OR.N.EQ.NTIM.OR.IPCK.EQ.1) CALL OUTPT
      CALL VELO
  110 CALL MVPT
C     ***************************************************************
C     ---STORE OBS. WELL DATA FOR TRANSIENT FLOW PROBLEMS---
      IF (S.EQ.0.0) GO TO 130
      IF (NUMOBS.LE.0) GO TO 130
      J=MOD(N,50)
      IF (J.EQ.0) J=50
      TMOBS(J)=SUMT
      DO 120 I=1,NUMOBS
      TMWL(I,J)=PK(IXOBS(I),IZOBS(I))
      TMCN(I,J)=CONC(IXOBS(I),IZOBS(I))
  120 CONTINUE
C     ***************************************************************
C     ---OUTPUT ROUTINES---
  130 IF (REMN.EQ.0.0.OR.N.EQ.NTIM.OR.MOD(N,50).EQ.0.OR.IPCK.EQ.1) CALL
     1CHMOT
      IF (SUMT.GE.(PYR+TMSUM)) GO TO 150
  140 CONTINUE
C     ***************************************************************
C     ---SUMMARY OUTPUT---
  150 CONTINUE
      IPRNT=1
      CALL CHMOT
  160 CONTINUE
      CALL OUTPT
C     ***************************************************************
      ENDFILE(6)
      IF (NPNCHV.EQ.0) GO TO 170
      ENDFILE(7)
  170 CONTINUE
      STOP
C     ***************************************************************
      END
      SUBROUTINE PARLOD
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER OVERRD
      COMMON /PRMJ/ NTIM,NPMP,NPNT,NITP,N,NX,NZ,NP,NREC,INT,NNX,NNZ,NUMO
     1BS,NMOV,IMOV,NPMAX,ITMAX,NZCRIT,IPRNT,NPTPND,NPNTMV,NPNTVL,NPNTD,N
     2PNCHV,NPDELC,ICHK,NCONST
      COMMON /PRMC/ NODEID(24,20),NPCELL(24,20),NPOLD(24,20),LIMBO(500),
     1IXOBS(5),IZOBS(5)
      COMMON /PRESS/ PERM(24,20),PMRX(24,20,4),PI(24,20),PR(24,20),PC(24
     1,20),PK(24,20),REC(24,20),DENS(24,20),GTERM(24,20),VISC(24,20),VPR
     2M(24,20),TMWL(5,50),TMOBS(50),TIM(100),AOPT(20),TITLE(10),ANFCTR,X
     3DEL,ZDEL,WIDTH,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,VOL
      COMMON /DENVIS/ DEN1,DEN2,VIS1,VIS2
      COMMON /CHMA/ PART(4,6400),CONC(24,20),TDS(24,20),VX(24,20),VZ(24,
     120),CONINT(24,20),TDSINT(24,20),CNREC(24,20),TDSREC(24,20),TMCN(5,
     250),TMTDS(5,50),POROS,SUMTCH,BETA,TIMV,STORM,STORT,STORMI,STORTI,C
     3MSIN,TDSIN,CMSOUT,TDSOUT,FLMIN,FLTIN,FLMOT,FLTOT,SUMIO,TDSIO,CELDI
     4S,DLTRAT,CSTORM,CSTORT,DMOLEC
      COMMON /BALM/ TOTLQ,TOTLQI,TPIN,TPOUT
      COMMON /XINV/ DXINV,DZINV,ARINV,PORINV
```

```
      COMMON /CHMC/ SUMC(24,20),VXBDY(24,20),VZBDY(24,20),SUMTDS(24,20),
     1WTFCTR(24,20),SUMWT(24,20),PTQ(24,20),PTWT(6400),ELEV(24,20)
      COMMON /CNCHNG/ CNCHCK(24,20),CTOL
C     ***************************************************************
      IF (INT.GT.1) GO TO 10
      WRITE (6,760)
      READ (5,730) TITLE
      WRITE (6,740) TITLE
C     ***************************************************************
C     ---INITIALIZE TEST AND CONTROL VARIABLES---
      STORMI=0.0
      STORTI=0.0
      TEST=0.0
      TOTLQ=0.0
      TOTLQI=0.0
      TPIN=0.0
      TPOUT=0.0
      SUMT=0.0
      SUMTCH=0.0
      INT=0
      IPRNT=0
      NCA=0
      N=0
      IMOV=0
      NMOV=0
      ICHK=0
C     ***************************************************************
C     ---LOAD CONTROL PARAMETERS---
      READ (5,750) NTIM,NPMP,NX,NZ,NPMAX,NPNT,NUMOBS,ITMAX,NREC,NPTPND,N
     1CODES,NZCRIT,NCONST,NPNTMV,NPNTVL,NPNTD,NPDELC,NPNCHV
      READ (5,810) PINT,TOL,POROS,BETA,S,TIMX,TINIT
      READ (5,820) XDEL,ZDEL,DLTRAT,CELDIS,ANFCTR,WIDTH,CTOL,DMOLEC
      NNX=NX-1
      NNZ=NZ-1
      NP=NPMAX
      NITP=10
      A1=1.0D0
      A2=2.0D0
      A3=86400.0D0
      A4=365.25D0
      A5=3.1415927D0
      DXINV=A1/XDEL
      DZINV=A1/ZDEL
      ARINV=DXINV*DZINV
      PORINV=A1/POROS
C     ---PRINT CONTROL PARAMETERS---
      WRITE (6,770)
      WRITE (6,780) NX,NZ,XDEL,ZDEL,WIDTH
      WRITE (6,790) NTIM,NPMP,PINT,TIMX,TINIT
      WRITE (6,800) S,POROS,BETA,DLTRAT,ANFCTR,DMOLEC,NCONST
      WRITE (6,880) TOL,ITMAX,CELDIS,NPMAX,NPTPND,CTOL
      IF (NPTPND.NE.4.AND.NPTPND.NE.5.AND.NPTPND.NE.8.AND.NPTPND.NE.9..A
     1ND.NPTPND.NE.16) WRITE (6,890)
      WRITE (6,900) NPNT,NPNTMV,NPNTVL,NPNTD,NUMOBS,NREC,NCODES,NPNCHV,N
     1PDELC
      GO TO 20
C     ***************************************************************
```

```
C     ---READ DATA TO REVISE TIME STEPS AND STRESSES FOR SUBSEQUENT
C        PUMPING PERIODS---
   10 READ (5,980) ICHK
      IF (ICHK.LE.0) WRITE (6,1020) INT
      IF (ICHK.LE.0) GO TO 20
      READ (5,990) NTIM,NPNT,ITMAX,NREC,NPNTMV,NPNTVL,NPNTD,NPDELC,NPNCH
     1V,PINT,TIMX,TINIT
      WRITE (6,1000) INT
      WRITE (6,1010) NTIM,NPNT,NITP,ITMAX,NREC,NPNTMV,NPNTVL,NPNTD,NPDEL
     1C,NPNCHV,PINT,TIMX,TINIT
C     ****************************************************************
C     ---LIST TIME INCREMENTS---
   20 DO 30 J=1,100
      TIM(J)=0.0
   30 CONTINUE
      PYR=PINT*A3*A4
      TIM(1)=TINIT
      IF (NPNTMV.EQ.0) NPNTMV=999
      IF (S.EQ.0.0) GO TO 50
      DO 40 K=2,NTIM
   40 TIM(K)=TIMX*TIM(K-1)
      WRITE (6,520)
      WRITE (6,550) TIM
      IF (TINIT.GT.PYR) WRITE (6,530)
      GO TO 70
   50 ANTIM=NTIM
      DO 60 K=1,NTIM
   60 TIM(K)=PYR/ANTIM
      WRITE (6,540) TIM(1)
C     ****************************************************************
C     ---INITIALIZE MATRICES---
   70 IF (INT.GT.1) GO TO 110
      DO 80 IZ=1,NZ
      DO 80 IX=1,NX
      VPRM(IX,IZ)=0.0
      ELEV(IX,IZ)=0.0
      PERM(IX,IZ)=0.0
      NODEID(IX,IZ)=0
      PMRX(IX,IZ,1)=0.0
      PMRX(IX,IZ,2)=0.0
      PMRX(IX,IZ,3)=0.0
      PMRX(IX,IZ,4)=0.0
      PI(IX,IZ)=0.0
      PR(IX,IZ)=0.0
      PC(IX,IZ)=0.0
      PK(IX,IZ)=0.0
      DENS(IX,IZ)=0.0
      VISC(IX,IZ)=0.0
      GTERM(IX,IZ)=0.0
      VX(IX,IZ)=0.0
      VZ(IX,IZ)=0.0
      VXBDY(IX,IZ)=0.0
      VZBDY(IX,IZ)=0.0
      CONC(IX,IZ)=0.0
      CONINT(IX,IZ)=0.0
      SUMC(IX,IZ)=0.0
      TDS(IX,IZ)=0.0
```

```
      TDSINT(IX,IZ)=0.0
      SUMTDS(IX,IZ)=0.0
      CNREC(IX,IZ)=0.0
      REC(IX,IZ)=0.0
      SUMWT(IX,IZ)=0.0
      PTQ(IX,IZ)=0.0
   80 CONTINUE
C     ****************************************************************
C     ---READ OBSERVATION WELL LOCATIONS---
      IF (NUMOBS.LE.0) GO TO 110
      WRITE (6,910)
      DO 90 J=1,NUMOBS
      READ (5,710) IX,IZ
      WRITE (6,830) J,IX,IZ
      IXOBS(J)=IX
   90 IZOBS(J)=IZ
      DO 100 I=1,NUMOBS
      DO 100 J=1,50
      TMWL(I,J)=0.0
      TMCN(I,J)=0.0
  100 TMTDS(I,J)=0.0
C     ****************************************************************
C             --- READ SPECIFIED RECHARGE AND/OR DISCHARGE ---
C                   (X-Z COORDINATES AND RATE IN CFS)
C       ---SIGNS:  RECHARGE = NEGATIVE,  DISCHARGE = POSITIVE---
C       ---IF RECHARGE, ALSO READ CONCENTRATION OF RECHARGE WATER---
  110 IF (NREC.LE.0) GO TO 140
      IF (INT.GT.1.AND.ICHK.LE.0) RETURN
      WRITE (6,920)
      DO 130 I=1,NREC
      READ (5,720) IX,IZ,FCTR,CNREC1,CNREC2
      IF (FCTR.GT.0.0) GO TO 120
      CNREC(IX,IZ)=CNREC1
      TDSREC(IX,IZ)=CNREC2
  120 REC(IX,IZ)=FCTR
  130 WRITE (6,840) IX,IZ,REC(IX,IZ),CNREC(IX,IZ),TDSREC(IX,IZ)
C     ****************************************************************
  140 IF (INT.GT.1) RETURN
      AREA=XDEL*ZDEL
      VOL=AREA*WIDTH
      WRITE (6,700) AREA
      WRITE (6,620)
      WRITE (6,630) XDEL
      WRITE (6,630) ZDEL
C     ****************************************************************
C                   ---READ PERMEABILITY IN FT**2---
      WRITE (6,560)
      READ (5,570) INPUT,FCTR
      DO 180 IZ=1,NZ
      IF (INPUT.EQ.1) READ (5,580) (PERM(IX,IZ),IX=1,NX)
      DO 170 IX=1,NX
      IF (INPUT.NE.1) GO TO 150
      PERM(IX,IZ)=PERM(IX,IZ)*FCTR
      GO TO 160
  150 PERM(IX,IZ)=FCTR
  160 IF (PERM(IX,IZ).NE.0.0) NCA=NCA+1
  170 CONTINUE
```

```
  180 WRITE (6,850) (PERM(IX,IZ),IX=1,NX)
C     *****************************************************************
C     --- READ VERTICAL PERMEABILITY FACTOR ---
      READ (5,570) INPUT,FCTR
      IF (INPUT.EQ.0) GO TO 220
      DO 200 IZ=1,NZ
      READ (5,580) (VPRM(IX,IZ),IX=1,NX)
      DO 190 IX=1,NX
      VPRM(IX,IZ)=VPRM(IX,IZ)*FCTR
  190 CONTINUE
  200 CONTINUE
      DO 210 IZ=1,NZ
      READ (5,580) (ELEV(IX,IZ),IX=1,NX)
  210 CONTINUE
C     *****************************************************************
  220 AAQ=NCA*AREA
      WRITE (6,650) NCA,AAQ,NZCRIT
C     *****************************************************************
C     ---READ NODE IDENTIFICATION CARDS---
C        ---SET VERT. PERM., SOURCE CONC., AND DIFFUSE RECHARGE---
C           ---SPECIFY CODES TO FIT YOUR NEEDS---
      WRITE (6,590)
      READ (5,570) INPUT,FCTR
      DO 240 IZ=1,NZ
      IF (INPUT.EQ.1) READ (5,660) (NODEID(IX,IZ),IX=1,NX)
      DO 230 IX=1,NX
  230 IF (INPUT.NE.1.AND.PERM(IX,IZ).NE.0.0) NODEID(IX,IZ)=FCTR
  240 WRITE (6,600) (NODEID(IX,IZ),IX=1,NX)
      WRITE (6,930) NCODES
      IF (NCODES.LE.0) GO TO 280
      WRITE (6,940)
      DO 270 IJ=1,NCODES
      READ (5,860) ICODE,FCTR1,FCTR2
      DO 260 IX=1,NX
      DO 260 IZ=1,NZ
      IF (NODEID(IX,IZ).NE.ICODE) GO TO 260
      IF (ELEV(IX,IZ).NE.0.0) GO TO 250
      VPRM(IX,IZ)=10000000000.*PERM(IX,IZ)/((XDEL+ZDEL)*0.5)
  250 TDSREC(IX,IZ)=FCTR1
      CNREC(IX,IZ)=FCTR2
  260 CONTINUE
  270 WRITE (6,870) ICODE,FCTR1,FCTR2
  280 WRITE (6,610)
      DO 290 IZ=1,NZ
  290 WRITE (6,850) (VPRM(IX,IZ),IX=1,NX)
      IWRITE=0
      DO 300 IZ=1,NZ
      DO 300 IX=1,NX
  300 IF (ELEV(IX,IZ).NE.0.0) IWRITE=1
      IF (IWRITE.EQ.1) WRITE (6,640)
      DO 310 IZ=1,NZ
  310 IF (IWRITE.EQ.1) WRITE (6,850) (ELEV(IX,IZ),IX=1,NX)
C     *****************************************************************
C     ---READ INITIAL PRESSURES---
      WRITE (6,680)
      READ (5,570) INPUT,FCTR
      DO 340 IZ=1,NZ
```

```
      IF (INPUT.EQ.1) READ (5,670) (PI(IX,IZ),IX=1,NX)
      DO 330 IX=1,NX
      IF (INPUT.NE.1) GO TO 320
      PI(IX,IZ)=PI(IX,IZ)*FCTR
      GO TO 330
  320 IF (PERM(IX,IZ).NE.0.0) PI(IX,IZ)=FCTR
  330 CONTINUE
  340 WRITE (6,690) (PI(IX,IZ),IX=1,NX)
C     ****************************************************************
C     ---SET INITIAL PRESSURES---
      DO 350 IX=1,NX
      DO 350 IZ=1,NZ
      PC(IX,IZ)=PI(IX,IZ)
      PR(IX,IZ)=PI(IX,IZ)
  350 PK(IX,IZ)=PI(IX,IZ)
C
      CALL OUTPT
C     ***************************************************************
C     ---READ INITIAL CONCENTRATIONS AND COMPUTE INITIAL MASS STORED---
      IF (NCONST.LT.2) GO TO 400
      WRITE (6,510)
      READ (5,570) INPUT,FCTR
      DO 390 IZ=1,NZ
      IF (INPUT.EQ.1) READ (5,670) (CONC(IX,IZ),IX=1,NX)
      DO 380 IX=1,NX
      IF (INPUT.NE.1) GO TO 360
      CONC(IX,IZ)=CONC(IX,IZ)*FCTR
      GO TO 370
  360 IF (PERM(IX,IZ).NE.0.0) CONC(IX,IZ)=FCTR
  370 CONINT(IX,IZ)=CONC(IX,IZ)
  380 STORMI=STORMI+CONINT(IX,IZ)*VOL*POROS
  390 WRITE (6,690) (CONC(IX,IZ),IX=1,NX)
  400 WRITE (6,1070)
      READ (5,570) INPUT,FCTR
      DO 440 IZ=1,NZ
      IF (INPUT.EQ.1) READ (5,670) (TDS(IX,IZ),IX=1,NX)
      DO 430 IX=1,NX
      IF (INPUT.NE.1) GO TO 410
      TDS(IX,IZ)=TDS(IX,IZ)*FCTR
      GO TO 420
  410 IF (PERM(IX,IZ).NE.0.0) TDS(IX,IZ)=FCTR
  420 TDSINT(IX,IZ)=TDS(IX,IZ)
      STORTI=STORTI+TDSINT(IX,IZ)*VOL*POROS
  430 CNCHCK(IX,IZ)=TDS(IX,IZ)
  440 WRITE (6,690) (TDS(IX,IZ),IX=1,NX)
C     ****************************************************************
C       --- CALCULATE INITIAL DENSITIES AND VISCOSITIES ---
      VIS1=0.0
      VIS2=0.0
      DEN1=4.743E-05
      DEN2=62.43
      READ (5,980) INPUT
      IF (INPUT.GT.0) READ (5,1060) DEN1,DEN2,VIS1,VIS2
      DO 470 IZ=2,NNZ
      DO 470 IX=2,NNX
      DENS(IX,IZ)=DEN1*TDS(IX,IZ)+DEN2
      DENSE=DENS(IX,IZ)
```

```
      IF (REC(IX,IZ).LT.0.0) DENSE=DEN1*TDSREC(IX,IZ)+DEN2
      REC(IX,IZ)=(REC(IX,IZ)*DENSE)/VOL
      IF (INPUT.GT.0) GO TO 460
      IF (TDS(IX,IZ).GT.20000.) GO TO 450
      VISC(IX,IZ)=3.45E-11*TDS(IX,IZ)+2.089E-05
      GO TO 470
  450 VISC(IX,IZ)=4.733E-11*TDS(IX,IZ)+2.063E-05
      GO TO 470
  460 VISC(IX,IZ)=VIS1*TDS(IX,IZ)+VIS2
  470 CONTINUE
      WRITE (6,1030)
      DO 480 IZ=1,NZ
  480 WRITE (6,1040) (DENS(IX,IZ),IX=1,NX)
      WRITE (6,1050)
      DO 490 IZ=1,NZ
  490 WRITE (6,850) (VISC(IX,IZ),IX=1,NX)
C     ****************************************************************
C     ---CHECK DATA SETS FOR INTERNAL CONSISTENCY---
      DO 500 IX=1,NX
      DO 500 IZ=1,NZ
      IF (PERM(IX,IZ).GT.0.0) GO TO 500
      IF (NODEID(IX,IZ).GT.0.0) WRITE (6,950) IX,IZ
      IF (PI(IX,IZ).NE.0.0) WRITE (6,960) IX,IZ
      IF (REC(IX,IZ).NE.0.0) WRITE (6,970) IX,IZ
  500 CONTINUE
C     ****************************************************************
      RETURN
C     ****************************************************************
C
C
C
  510 FORMAT (1H1,40HINITIAL CONCENTRATION MAP - TRACE SOLUTE/)
  520 FORMAT (1H1,27HTIME INTERVALS (IN SECONDS))
  530 FORMAT (1H0,5X,65H*** WARNING ***   INITIAL TIME STEP IS LONGER TH
     1AN PUMPING PERIOD/25X,34H***ADJUST EITHER TINIT OR PINT.***/)
  540 FORMAT (1H1,15X,17HSTEADY-STATE FLOW//5X,57HTIME INTERVAL (IN SEC)
     1 FOR SOLUTE-TRANSPORT SIMULATION = ,G12.5)
  550 FORMAT (3H   ,10G12.5)
  560 FORMAT (1H1,24HPERMEABILITY MAP (FT**2))
  570 FORMAT (I1,G10.0)
  580 FORMAT (24G3.0)
  590 FORMAT (1H1,23HNODE IDENTIFICATION MAP//)
  600 FORMAT (1H ,24I5)
  610 FORMAT (1H1,39HVERTICAL PERMEABILITY FACTOR (1/FT*SEC))
  620 FORMAT (1H0,10X,12HX-Z SPACING:)
  630 FORMAT (1H ,12X,10G12.5)
  640 FORMAT (1H1,27HCONFINING BED THICKNESS MAP//)
  650 FORMAT (1H0,////10X,44HNO. OF FINITE-DIFFERENCE CELLS IN AQUIFER =
     1 ,I4//10X,28HAREA OF AQUIFER IN MODEL  = ,G12.5,10H   SQ. FT.////1
     20X,47HNZCRIT   (MAX. NO. OF CELLS THAT CAN BE VOID OF/20X,56HPARTI
     3CLES;  IF EXCEEDED, PARTICLES ARE REGENERATED)   = ,I4/)
  660 FORMAT (24I1)
  670 FORMAT (12G6.0)
  680 FORMAT (1H1,28HINITIAL PRESSURES (LB/FT**2)/)
  690 FORMAT (1H ,20F6.0)
  700 FORMAT (1H0,10X,19HAREA OF ONE CELL = ,G12.4)
  710 FORMAT (2I2)
```

```
720 FORMAT (2I2,3G10.2)
730 FORMAT (10A8)
740 FORMAT (1H0,10A8)
750 FORMAT (20I4)
760 FORMAT (1H1,77HU.S.G.S. METHOD-OF-CHARACTERISTICS MODEL FOR SOLUTE
   1 TRANSPORT IN GROUND WATER)
770 FORMAT (1H0,21X,21HI N P U T     D A T A)
780 FORMAT (1H0,23X,16HGRID DESCRIPTORS//13X,30HNX    (NUMBER OF COLUM
   1NS)  =   ,I4/13X,28HNZ     (NUMBER OF ROWS)      =,I6/13X,29HXDEL   (X
   2-DISTANCE IN FEET) = ,F9.3/13X,29HZDEL  (Z-DISTANCE IN FEET) = ,F9
   3.3/13X,29HWIDTH (Y-DISTANCE IN FEET) = ,F9.3)
790 FORMAT (1H0,23X,16HTIME  PARAMETERS//13X,40HNTIM    (MAX. NO. OF TI
   1ME STEPS)         = ,I6/13X,40HNPMP    (NO. OF PUMPING PERIODS)
   2 = ,I6/13X,39HPINT    (PUMPING PERIOD IN YEARS)        =,F11.3/13X,39
   3HTIMX    (TIME INCREMENT MULTIPLIER)    =,F10.2/13X,39HTINIT   (INIT
   4IAL TIME STEP IN SEC.)     =,F8.0)
800 FORMAT (1H0,14X,34HHYDROLOGIC AND CHEMICAL PARAMETERS//13X,1HS,7X,
   129H(SPECIFIC STORAGE)          =,5X,F9.6/13X,28HPOROS   (EFFECTIVE
   2 POROSITY),8X,3H=   ,F8.2/13X,39HBETA    (LONGITUDINAL DISPERSIVITY
   3) =   ,F7.1/13X,31HDLTRAT  (RATIO OF TRANSVERSE TO/21X,30H LONGITUD
   4INAL DISPERSIVITY) = ,F9.2/13X,39HANFCTR  (RATIO OF K-ZZ TO K-XX)
   5    =   ,F12.6/13X,28HDMOLEC  (COEF. OF DIFFUSION),8X,3H=   ,1PE9.2/
   613X,39HNCONST  (NUMBER OF CONSTITUENTS)     =   ,I2)
810 FORMAT (7G10.0)
820 FORMAT (8G10.0)
830 FORMAT (1H ,16X,I2,5X,I2,4X,I2)
840 FORMAT (1H ,7X,2I4,3X,E12.5,2(3X,F8.2))
850 FORMAT (1H ,1P10E10.2)
860 FORMAT (I2,2G10.2)
870 FORMAT (1H ,7X,I2,2(7X,F9.2))
880 FORMAT (1H0,21X,20HEXECUTION PARAMETERS//13X,39HTOL     (CONVERGENC
   1E CRITERIA - SIP)  = ,F9.7/13X,39HITMAX  (MAX.NO.OF ITERATIONS - S
   2IP)  = ,I4/13X,34HCELDIS (MAX.CELL DISTANCE PER MOVE/24X,28HOF PAR
   3TICLES - M.O.C.)     = ,F8.3/13X,30HNPMAX  (MAX. NO. OF PARTICLES),
   47X,2H= ,I4/12X,32H NPTPND (NO. PARTICLES PER NODE),6X,3H=   ,I4/13X
   5,33HCTOL    (MINIMUM CONC. CHANGE       /21X,31HFOR PRESSURE RECALCUL
   6ATION)  = ,F5.0)
890 FORMAT (1H0,5X,47H*** WARNING *** NPTPND MUST EQUAL 4,5,8,9,OR 16)
900 FORMAT (1H0,23X,15HPROGRAM OPTIONS//13X,30HNPNT    (TIME STEP INTER
   1VAL FOR/21X,18HCOMPLETE PRINTOUT),7X,3H=   ,I4/13X,31HNPNTMV (MOVE
   2INTERVAL FOR CHEM./21X,28HCONCENTRATION PRINTOUT)  =   ,I4/13X,29HN
   3PNTVL (PRINT OPTION-VELOCITY/21X,24H0=NO; 1=FIRST TIME STEP;/21X,1
   47H2=ALL TIME STEPS),8X,3H=   ,I4/13X,31HNPNTD  (PRINT OPTION-DISP.C
   5OEF./21X,24H0=NO; 1=FIRST TIME STEP;/21X,17H2=ALL TIME STEPS),8X,3
   6H=   ,I4/13X,32HNUMOBS (NO. OF OBSERVATION WELLS/21X,28HFOR HYDROGR
   7APH PRINTOUT) =   ,I4/13X,35HNREC    (NO. OF RECHARGE CELLS)    = ,I5
   8/13X,24HNCODES (FOR NODE IDENT.),9X,2H= ,I5/13X,32HNPNCHV (WRITE V
   9ELOCITIES-UNIT 7),1X,2H= ,I5/13X,36HNPDELC (PRINT OPT.-CONC. CHANG
   $E) =   ,I4)
910 FORMAT (1H0,10X,29HLOCATION OF OBSERVATION WELLS//17X,3HNO.,5X,1HX
   1,5X,1HZ/)
920 FORMAT (1H0,10X,28HLOCATION  OF  RECHARGE CELLS//11X,37HX    Z    RA
   1TE(IN CFS)  CONC. TDS CONC.)
930 FORMAT (1H0,5X,37HNO. OF NODE IDENT. CODES SPECIFIED = ,I2)
940 FORMAT (1H0,10X,41HTHE FOLLOWING ASSIGNMENTS HAVE BEEN MADE:/5X,40
   1HCODE NO.     TDS CONC.          CONC.     )
950 FORMAT (1H ,5X,61H*** WARNING ***    PERM.EQ.0.0 AND NODEID.GT.0.0
```

```
     1 AT NODE IX =,I4,6H, IZ =,I4)
  960 FORMAT (1H ,5X,56H*** WARNING ***   PERM.EQ.0.0 AND PI.NE.0.0 AT N
     1ODE IX =,I4,6H, IZ =,I4)
  970 FORMAT (1H ,5X,58H*** WARNING ***   PERM.EQ.0.0 AND RECH.NE.0.0 AT
     1 NODE IX =,I4,6H, IZ =,I4)
  980 FORMAT (I1)
  990 FORMAT (9I4,3G5.0)
 1000 FORMAT (1H1,5X,25HSTART PUMPING PERIOD NO. ,I2//2X,75HTHE FOLLOWIN
     1G TIME STEP, PUMPAGE, AND PRINT PARAMETERS HAVE BEEN REDEFINED:/)
 1010 FORMAT (1H0,14X,9HNTIM    = ,I4/15X,9HNPNT    = ,I4/15X,9HNITP    = ,
     1I4/15X,9HITMAX   = ,I4/15X,9HNREC    = ,I4/15X,9HNPNTMV = ,I4/15X,9H
     2NPNTVL = ,I4/15X,9HNPNTD   = ,I4/15X,9HNPDELC = ,I4/15X,9HNPNCHV =
     3,I4/15X,9HPINT    = ,F10.3/15X,9HTIMX    = ,F10.3/15X,9HTINIT   = ,F1
     40.3/)
 1020 FORMAT (1H1,5X,25HSTART PUMPING PERIOD NO. ,I2//2X,23HNO PARAMETER
     1S REDEFINED/)
 1030 FORMAT (1H0,28HINITIAL DENSITIES (LB/FT**3)/)
 1040 FORMAT (20F6.2)
 1050 FORMAT (1H0,34HINITIAL VISCOSITIES (LB*SEC/FT**2)/)
 1060 FORMAT (4G10.3)
 1070 FORMAT (1H0,44HINITIAL TDS MAP - DENSITY-CONTROLLING SOLUTE/)
      END
      SUBROUTINE ITERAT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /PRMJ/ NTIM,NPMP,NPNT,NITP,N,NX,NZ,NP,NREC,INT,NNX,NNZ,NUMO
     1BS,NMOV,IMOV,NPMAX,ITMAX,NZCRIT,IPRNT,NPTPND,NPNTMV,NPNTVL,NPNTD,N
     2PNCHV,NPDELC,ICHK,NCONST
      COMMON /PRMC/ NODEID(24,20),NPCELL(24,20),NPOLD(24,20),LIMBO(500),
     1IXOBS(5),IZOBS(5)
      COMMON /PRESS/ PERM(24,20),PMRX(24,20,4),PI(24,20),PR(24,20),PC(24
     1,20),PK(24,20),REC(24,20),DENS(24,20),GTERM(24,20),VISC(24,20),VPR
     2M(24,20),TMWL(5,50),TMOBS(50),TIM(100),AOPT(20),TITLE(10),ANFCTR,X
     3DEL,ZDEL,WIDTH,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,VOL
      COMMON /CHMA/ PART(4,6400),CONC(24,20),TDS(24,20),VX(24,20),VZ(24,
     120),CONINT(24,20),TDSINT(24,20),CNREC(24,20),TDSREC(24,20),TMCN(5,
     250),TMTDS(5,50),POROS,SUMTCH,BETA,TIMV,STORM,STORT,STORMI,STORTI,C
     3MSIN,TDSIN,CMSOUT,TDSOUT,FLMIN,FLTIN,FLMOT,FLTOT,SUMIO,TDSIO,CELDI
     4S,DLTRAT,CSTORM,CSTORT,DMOLEC
      COMMON /CHMC/ SUMC(24,20),VXBDY(24,20),VYBDY(24,20),SUMTDS(24,20),
     1WTFCTR(24,20),SUMWT(24,20),PTQ(24,20),PTWT(6400),ELEV(24,20)
      COMMON /BALM/ TOTLQ,TOTLQI,TPIN,TPOUT
      COMMON /XINV/ DXINV,DZINV,ARINV,PORINV
      COMMON /DENVIS/ DEN1,DEN2,VIS1,VIS2
      COMMON /CNCHNG/ CNCHCK(24,20),CTOL
      DIMENSION DEL(24,20), ETA(24,20), V(24,20), XI(24,20), IORDER(21),
     1 RHOP(20), TEMP(20), TEST3(201)
      DATA IORDER/1,2,3,4,5,1,2,3,4,5,11*1/
      HMAX=1.0
      DO 30 IX=1,NX
      DO 30 IZ=1,NZ
      IF (PERM(IX,IZ).EQ.0.0) GO TO 30
      CNCHCK(IX,IZ)=TDS(IX,IZ)
C       --- REASSIGN DENSITIES AND VISCOSITIES ---
      DTEMP=DENS(IX,IZ)
      DENS(IX,IZ)=DEN1*TDS(IX,IZ)+DEN2
      IF (REC(IX,IZ).GT.0.0) REC(IX,IZ)=REC(IX,IZ)*DENS(IX,IZ)/DTEMP
      IF (VIS1.NE.0.0) GO TO 20
```

```
      IF (TDS(IX,IZ).GT.20000) GO TO 10
      VISC(IX,IZ)=3.45E-11*TDS(IX,IZ)+2.089E-05
      GO TO 30
   10 VISC(IX,IZ)=4.733E-11*TDS(IX,IZ)+2.063E-05
      GO TO 30
   20 VISC(IX,IZ)=VIS1*TDS(IX,IZ)+VIS2
   30 CONTINUE
      DO 80 IZ=2,NNZ
      DO 80 IX=2,NNX
      IF (PERM(IX,IZ).EQ.0.0) GO TO 80
      FF1=PERM(IX,IZ)/VISC(IX,IZ)
      F1=FF1*DENS(IX,IZ)
      IF (PERM(IX+1,IZ).EQ.0.0) GO TO 40
      FF2=PERM(IX+1,IZ)/VISC(IX+1,IZ)
      F2=FF2*DENS(IX+1,IZ)
      GO TO 50
   40 FF2=0.0
      F2=0.0
   50 IF (PERM(IX,IZ+1).EQ.0.0) GO TO 60
      FF3=PERM(IX,IZ+1)/VISC(IX,IZ+1)
      F3=FF3*DENS(IX,IZ+1)
      GO TO 70
   60 FF3=0.0
      F3=0.0
   70 PMRX(IX,IZ,1)=2.0*F1*F2/((F1+F2)*XDEL)
      PMRX(IX,IZ,2)=2.0*F1*F3/((F1+F3)*ZDEL)
      PMRX(IX,IZ,2)=PMRX(IX,IZ,2)*ANFCTR
      PMRX(IX,IZ,3)=2.0*FF1*FF2/(FF1+FF2)
      PMRX(IX,IZ,4)=2.0*FF1*FF3/(FF1+FF3)
      PMRX(IX,IZ,4)=PMRX(IX,IZ,4)*ANFCTR
      DENS1=(DENS(IX,IZ+1)+DENS(IX,IZ))*0.500
      DENS2=(DENS(IX,IZ)+DENS(IX,IZ-1))*0.500
      GTERM(IX,IZ)=PMRX(IX,IZ,2)*DENS1-PMRX(IX,IZ-1,2)*DENS2
   80 CONTINUE
C
C      COMPUTE AND PRINT ITERATION PARAMETERS
C
      PQIN=0.0
      PQOUT=0.0
      KOUNT=-1
      DO 90 I=1,NX
      DO 90 J=1,NZ
      PR(I,J)=PK(I,J)
   90 CONTINUE
      IF (INT.NE.1) GO TO 120
C
C      ---INITIALIZE ORDER OF ITERATION PARAMETERS
C
      INO1=NX-1
      JNO1=NZ-1
      I2=INO1-1
      J2=JNO1-1
      L2=NITP/2
      PL2=L2-1
C
C      COMPUTE MAXIMUM PARAMETER FOR PROBLEM
C
```

```
      DX=(1./NX)**2
      DZ=(1./NZ)**2
      W=1.0D0-DMIN1(2.0D0*DX/(1.0D0+ANFCTR*DX/DZ),2.0D0*DZ/(1.0D0+DZ/(AN
     1FCTR*DX)))
C
C       --- COMPUTE PARAMETERS IN GEOMETRIC SEQUENCE ---
C
      PJ=-1.
      DO 100 I=1,L2
      PJ=PJ+1
  100 TEMP(I)=1.0D0-(1.0D0-W)**(PJ/PL2)
C
C       --- ORDER SEQUENCE OF PARAMETERS ---
C
      DO 110 J=1,NITP
  110 RHOP(J)=TEMP(IORDER(J))
      IF (IMOV.EQ.0) WRITE (6,250) HMAX,NITP,(RHOP(J),J=1,NITP)
C
C       INITIALIZE DATA FOR A NEW ITERATION
C
  120 KOUNT=KOUNT+1
      IF (KOUNT.LE.ITMAX) GO TO 130
      WRITE (6,290)
      CALL OUTPT
      WRITE (6,260) (TEST3(I),I=1,KOUNT)
      STOP
  130 IF (MOD(KOUNT,NITP)) 140,140,150
C
C       INITIALIZE DATA FOR A NEW ITERATION
C
  140 NTH=0
  150 NTH=NTH+1
      W=RHOP(NTH)
      TEST3(KOUNT+1)=0
      TEST=0
      DO 160 I=1,NX
      DO 160 J=1,NZ
      DEL(I,J)=0
      ETA(I,J)=0
      V(I,J)=0
  160 XI(I,J)=0
      BIGI=0
      RHO=S/TIM(N)
C
C       CHOOSE SIP NORMAL OR REVERSE ALGORITHM
C
      IF (MOD(KOUNT,2)) 170,220,170
C
C       ...............................................................
C       ---ORDER EQUATIONS WITH ROW 1 FIRST- 3X3 EXAMPLE:
C                             1 2 3
C                             4 5 6
C                             7 8 9
C       ...............................................................
  170 DO 180 J=2,JN01
      DO 180 I=2,IN01
C
```

```
C       --- SKIP COMPUTATIONS IF NODE IS OUTSIDE AQUIFER BOUNDRY---
      IF (PERM(I,J).EQ.0.) GO TO 180
C
C       --- COMPUTE COEFFICIENTS---
C
      D=PMRX(I-1,J,1)/XDEL
      F=PMRX(I,J,1)/XDEL
      B=PMRX(I,J-1,2)/ZDEL
      H=PMRX(I,J,2)/ZDEL
      CH=DEL(I,J-1)*B/(1.+W*DEL(I,J-1))
      GH=ETA(I-1,J)*D/(1.+W*ETA(I-1,J))
C
C       ---SIP 'NORMAL' ALGORITHM ---
C       ---FOWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---
C
      E=-B-D-F-H-RHO-VPRM(I,J)
      BH=B-W*CH
      DH=D-W*GH
      EH=E+W*(CH+GH)
      FH=F-W*CH
      HH=H-W*GH
      ALFA=BH
      BEDA=DH
      GAMA=EH-ALFA*ETA(I,J-1)-BEDA*DEL(I-1,J)
      DEL(I,J)=FH/GAMA
      ETA(I,J)=HH/GAMA
      QL=-VPRM(I,J)*(PI(I,J)+DENS(I,J)*ELEV(I,J))
      RES=-D*PK(I-1,J)-F*PK(I+1,J)-H*PK(I,J+1)-B*PK(I,J-1)-E*PK(I,J)-RHO
     1*PR(I,J)+QL+REC(I,J)+GTERM(I,J)
      V(I,J)=(HMAX*RES-ALFA*V(I,J-1)-BEDA*V(I-1,J))/GAMA
  180 CONTINUE
C
C       ---BACK SUBSTITUTE FOR VECTOR XI ---
C
      DO 190 J=1,J2
      J3=NZ-J
      DO 190 I=1,I2
      I3=NX-I
      IF (PERM(I3,J3).EQ.0.) GO TO 190
      XI(I3,J3)=V(I3,J3)-DEL(I3,J3)*XI(I3+1,J3)-ETA(I3,J3)*XI(I3,J3+1)
C
C       --- COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERION---
C
      TCHK=DABS(XI(I3,J3))
      IF (TCHK.GT.BIGI) BIGI=TCHK
      PK(I3,J3)=PK(I3,J3)+XI(I3,J3)
  190 CONTINUE
  200 IF (BIGI.GT.TOL) TEST=1
      TEST3(KOUNT+1)=BIGI
      IF (TEST.EQ.1.) GO TO 120
      DO 210 IZ=1,NZ
      DO 210 IX=1,NX
      IF (PERM(IX,IZ).EQ.0.0) GO TO 210
      PTQ(IX,IZ)=REC(IX,IZ)
      IF (REC(IX,IZ).LT.0.0) PQIN=PQIN+REC(IX,IZ)*VOL
      IF (REC(IX,IZ).GT.0.0) PQOUT=PQOUT+REC(IX,IZ)*VOL
C     ---COMPUTE LEAKAGE FOR MASS BALANCE---
```

```
      IF (VPRM(IX,IZ).EQ.0.0) GO TO 210
      DELQ=VPRM(IX,IZ)*(PI(IX,IZ)-PK(IX,IZ)+DENS(IX,IZ)*ELEV(IX,IZ))
      PTQ(IX,IZ)=PTQ(IX,IZ)-DELQ
      IF (DELQ.GT.0.0) TOTLQI=TOTLQI+DELQ*TIM(N)
      IF (DELQ.LT.0.0) TOTLQ=TOTLQ+DELQ*TIM(N)
  210 CONTINUE
      TPIN=PQIN*TIM(N)+TPIN
      TPOUT=PQOUT*TIM(N)+TPOUT
C
      WRITE (6,270) N
      WRITE (6,280) KOUNT
C     ****************************************************************
      RETURN
C     ****************************************************************
C
C
C      ..................................................................
C      ---ORDER EQUATIONS WITH THE LAST ROW FIRST- 3X3 EXAMPLE:
C               7 8 9
C               4 5 6
C               1 2 3
C      ..................................................................
  220 DO 230 JJ=1,J2
      J=NZ-JJ
      DO 230 I=2,INO1
C
C      --- SKIP COMPUTATIONS IF NODE IS OUTSIDE OF AQUIFER BOUNDRY---
      IF (PERM(I,J).EQ.0.) GO TO 230
C
C      ---COMPUTE COEFFICIENTS---
      D=PMRX(I-1,J,1)/XDEL
      F=PMRX(I,J,1)/XDEL
      B=PMRX(I,J-1,2)/ZDEL
      H=PMRX(I,J,2)/ZDEL
C
C      --- SIP 'REVERSE' ALGORITHM---
C      --- FOWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---
C
      E=-B-D-F-H-RHO-VPRM(I,J)
      CH=DEL(I,J+1)*H/(1.+W*DEL(I,J+1))
      GH=ETA(I-1,J)*D/(1.+W*ETA(I-1,J))
      BH=H-W*CH
      DH=D-W*GH
      EH=E+W*(CH+GH)
      FH=F-W*CH
      HH=B-W*GH
      ALFA=BH
      BEDA=DH
      GAMA=EH-ALFA*ETA(I,J+1)-BEDA*DEL(I-1,J)
      DEL(I,J)=FH/GAMA
      ETA(I,J)=HH/GAMA
      QL=-VPRM(I,J)*(PI(I,J)+DENS(I,J)*ELEV(I,J))
      RES=-D*PK(I-1,J)-F*PK(I+1,J)-H*PK(I,J+1)-B*PK(I,J-1)-E*PK(I,J)-RHO
     1*PR(I,J)+QL+REC(I,J)+GTERM(I,J)
      V(I,J)=(HMAX*RES-ALFA*V(I,J+1)-BEDA*V(I-1,J))/GAMA
  230 CONTINUE
C
```

```
C       --- BACK SUBSTITUTE FOR VECTOR XI ---
      DO 240 J=2,JN01
      DO 240 I3=1,I2
      I=NX-I3
      IF (PERM(I,J).EQ.0.) GO TO 240
      XI(I,J)=V(I,J)-DEL(I,J)*XI(I+1,J)-ETA(I,J)*XI(I,J-1)
C
C       --- COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERION ---
C
      TCHK=DABS(XI(I,J))
      IF (TCHK.GT.BIGI) BIGI=TCHK
      PK(I,J)=PK(I,J)+XI(I,J)
  240 CONTINUE
      GO TO 200
C
  250 FORMAT (1X,6HBETA= ,F4.2,/,1X,I3,23H ITERATIONS PARAMETERS:,6(/1X,
     16E15.6))
  260 FORMAT (1X,39HMAXIMUM HEAD CHANGE FOR EACH ITERATION:,20(/,1X,10(F
     112.5)))
  270 FORMAT (1H0//3X,4HN = ,1I4)
  280 FORMAT (1X,22HNUMBER OF ITERATIONS= ,I3)
  290 FORMAT (1H0,5X,53H*** EXECUTION TERMINATED -- MAX # ITERATIONS EXC
     1EEDED/26X,21HFINAL OUTPUT FOLLOWS:)
      END
      SUBROUTINE GENPT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER *2PTID
      COMMON /PRMJ/ NTIM,NPMP,NPNT,NITP,N,NX,NZ,NP,NREC,INT,NNX,NNZ,NUMO
     1BS,NMOV,IMOV,NPMAX,ITMAX,NZCRIT,IPRNT,NPTPND,NPNTMV,NPNTVL,NPNTD,N
     2PNCHV,NPDELC,ICHK,NCONST
      COMMON /PRMC/ NODEID(24,20),NPCELL(24,20),NPOLD(24,20),LIMBO(500),
     1IXOBS(5),IZOBS(5)
      COMMON /PRESS/ PERM(24,20),PMRX(24,20,4),PI(24,20),PR(24,20),PC(24
     1,20),PK(24,20),REC(24,20),DENS(24,20),GTERM(24,20),VISC(24,20),VPR
     2M(24,20),TMWL(5,50),TMOBS(50),TIM(100),AOPT(20),TITLE(10),ANFCTR,X
     3DEL,ZDEL,WIDTH,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,VOL
      COMMON /CHMA/ PART(4,6400),CONC(24,20),TDS(24,20),VX(24,20),VZ(24,
     120),CONINT(24,20),TDSINT(24,20),CNREC(24,20),TDSREC(24,20),TMCN(5,
     250),TMTDS(5,50),POROS,SUMTCH,BETA,TIMV,STORM,STORT,STORMI,STORTI,C
     3MSIN,TDSIN,CMSOUT,TDSOUT,FLMIN,FLTIN,FLMOT,FLTOT,SUMIO,TDSIO,CELDI
     4S,DLTRAT,CSTORM,CSTORT,DMOLEC
      COMMON /CHMC/ SUMC(24,20),VXBDY(24,20),VYBDY(24,20),SUMTDS(24,20),
     1WTFCTR(24,20),SUMWT(24,20),PTQ(24,20),PTWT(6400),ELEV(24,20)
      COMMON /CHMP/ PTID(6400)
      DIMENSION RPT(16), RNT(16), RP(16), RN(16), IPT(16), RPT2(16), RNT
     12(16), RP2(16), RN2(16)
C     ****************************************************************
      F1=0.30
      F2=1.0/3.0
      IF (NPTPND.EQ.4) F1=0.25
      IF (NPTPND.EQ.9) F1=1.0/3.0
      IF (NPTPND.EQ.8) F2=0.25
      IF (NPTPND.EQ.16) F1=0.25
      IF (NPTPND.EQ.16) F2=0.125
      NCHK=NPTPND
      IF (NPTPND.EQ.5.OR.NPTPND.EQ.9) NCHK=NPTPND-1
      IF (TEST.GT.98.) GO TO 10
```

```
C     ******************************************************************
C     ---INITIALIZE VALUES---
      STORM=0.0
      STORT=0.0
      FLTOT=0.0
      FLTIN=0.0
      CMSIN=0.0
      CMSOUT=0.0
      FLMIN=0.0
      FLMOT=0.0
      SUMIO=0.0
C     ******************************************************************
   10 DO 20 IN=1,NPMAX
      PTID(IN)=0
      PTWT(IN)=1.0
      DO 20 ID=1,4
   20 PART(ID,IN)=0.0
      DO 30 IA=1,16
      RP(IA)=0.0
      RN(IA)=0.0
      RPT(IA)=0.0
      RNT(IA)=0.0
      RP2(IA)=0.0
      RN2(IA)=0.0
      RPT2(IA)=0.0
      RNT2(IA)=0.0
   30 IPT(IA)=0
C     ---SET UP LIMBO ARRAY---
      DO 40 IN=1,500
   40 LIMBO(IN)=0.0
      IND=1
      DO 50 IL=1,500,2
      LIMBO(IL)=IND
   50 IND=IND+1
C     ******************************************************************
C     ---INSERT PARTICLES---
      DO 720 IX=2,NNX
      DO 720 IZ=2,NNZ
      IF (PERM(IX,IZ).EQ.0.0) GO TO 720
      KR=0
      KR2=0
      TEST2=0.0
      METH=1
      NPCELL(IX,IZ)=0
      NPOLD(IX,IZ)=NPTPND
      C1=CONC(IX,IZ)
      C2=TDS(IX,IZ)
      IF (C1.LE.1.0E-05) TEST2=1.0
      IF (VPRM(IX,IZ).GT.0.00) TEST2=1.0
      IF (PERM(IX+1,IZ+1).EQ.0.0.OR.PERM(IX+1,IZ-1).EQ.0.0.OR.PERM(IX-1,
     1IZ+1).EQ.0.0.OR.PERM(IX-1,IZ-1).EQ.0.0) TEST2=1.0
      IF ((PERM(IX,IZ+1).EQ.0.0.OR.PERM(IX,IZ-1).EQ.0.0.OR.PERM(IX+1,IZ)
     1.EQ.0.0.OR.PERM(IX-1,IZ).EQ.0.0).AND.NPTPND.GT.5) TEST2=1.0
      CNODE=C1*(1.0-F1)
      CNODE2=C2*(1.0-F1)
      IF (TEST.LT.98.0.OR.TEST2.GT.0.0) GO TO 70
      SUM=CONC(IX+1,IZ)+CONC(IX-1,IZ)+CONC(IX,IZ+1)+CONC(IX,IZ-1)
```

```
      IF (NCHK.EQ.4) GO TO 60
      SUM=SUM+CONC(IX+1,IZ+1)+CONC(IX+1,IZ-1)+CONC(IX-1,IZ+1)+CONC(IX-1,
     1IZ-1)
   60 AVC=SUM/NCHK
      IF (AVC.GT.C1) METH=2
C
C        ---PUT 4 PARTICLES ON CELL DIAGONALS---
   70 DO 250 IT=1,2
      EVET=(-1.0)**IT
      DO 250 IS=1,2
      EVES=(-1.0)**IS
      IF (NPTPND.EQ.16) GO TO 80
      PART(1,IND)=IX+F1*EVET
      PART(2,IND)=IZ+F1*EVES
      PART(2,IND)=-PART(2,IND)
      PART(3,IND)=C2
      PART(4,IND)=C1
      KR=KR+1
      IPT(KR)=IND
      PTID(IND)=KR
      GO TO 90
   80 IF (TEST.LT.98.0.OR.TEST2.GT.0.0) GO TO 200
   90 IF (TEST.LT.98.0.OR.TEST2.GT.0.0) GO TO 240
      IXD=IX+EVET
      IZD=IZ+EVES
      IF (METH.EQ.2) GO TO 100
      PARTC=CNODE+CONC(IXD,IZD)*F1
      PARTC2=CNODE2+TDS(IX,IZ)*F1
      GO TO 110
  100 PARTC=2.0*C1*CONC(IXD,IZD)/(C1+CONC(IXD,IZD))
      PARTC2=2.0*C2*TDS(IXD,IZD)/(C2+TDS(IXD,IZD))
  110 IF (C1-CONC(IXD,IZD)) 120,130,140
  120 RPT(KR)=CONC(IXD,IZD)-PARTC
      RNT(KR)=C1-PARTC
      GO TO 150
  130 RPT(KR)=0.0
      RNT(KR)=0.0
      GO TO 150
  140 RPT(KR)=C1-PARTC
      RNT(KR)=CONC(IXD,IZD)-PARTC
  150 IF (C2-TDS(IXD,IZD)) 160,170,180
  160 RPT2(KR)=TDS(IXD,IZD)-PARTC2
      RNT2(KR)=C2-PARTC2
      GO TO 190
  170 RPT2(KR)=0.0
      RNT2(KR)=0.0
      GO TO 190
  180 RPT2(KR)=C2-PARTC2
      RNT2(KR)=TDS(IXD,IZD)-PARTC2
  190 IF (NPTPND.EQ.16) GO TO 200
      PART(3,IND)=PARTC2
      PART(4,IND)=PARTC
      RP2(KR)=RPT2(KR)
      RN2(KR)=RNT2(KR)
      RP(KR)=RPT(KR)
      RN(KR)=RNT(KR)
      GO TO 240
```

```
  200 DO 230 ITT=1,2
      EVET2=(-1.0)**ITT
      DO 230 ISS=1,2
      EVES2=(-1.0)**ISS
      PART(1,IND)=(IX+F1*EVET)+F2*EVET2
      PART(2,IND)=(IZ+F1*EVES)+F2*EVES2
      PART(2,IND)=-PART(2,IND)
      KR2=KR2+1
      IF (TEST.LT.98.0.OR.TEST2.GT.0.0) GO TO 210
      PART(3,IND)=PARTC2
      PART(4,IND)=PARTC
      RP(KR2)=RPT(KR)
      RN(KR2)=RNT(KR)
      RP2(KR2)=RPT2(KR)
      RN2(KR2)=RNT2(KR)
      IPT(KR2)=IND
      GO TO 220
  210 PART(3,IND)=C2
      PART(4,IND)=C1
  220 PTID(IND)=KR2
      IND=IND+1
  230 CONTINUE
      GO TO 250
  240 IND=IND+1
  250 CONTINUE
      IF (NPTPND.EQ.16) GO TO 480
      IF (NPTPND.EQ.5.OR.NPTPND.EQ.9) GO TO 260
      GO TO 270
C        ---PUT ONE PARTICLE AT CENTER OF CELL---
  260 PART(1,IND)=IX
      PART(2,IND)=-IZ
      PART(3,IND)=C2
      PART(4,IND)=C1
      PTID(IND)=5
      IND=IND+1
C        ---PLACE NORTH, SOUTH, EAST, AND WEST PARTICLES---
  270 IF (NPTPND.LT.8) GO TO 480
      CNODE=C1*(1.0-F2)
      CNODE2=C2*(1.0-F2)
      DO 470 IT=1,2
      EVET=(-1.0)**IT
      PART(1,IND)=IX+F2*EVET
      PART(2,IND)=-IZ
      PART(3,IND)=C2
      PART(4,IND)=C1
      IF (EVET.LT.0) PTID(IND)=6
      IF (EVET.GT.0) PTID(IND)=8
      IF (TEST.LT.98.0.OR.TEST2.GT.0.0) GO TO 370
      IXD=IX+EVET
      KR=KR+1
      IPT(KR)=IND
      IF (METH.EQ.2) GO TO 280
      PART(4,IND)=CNODE+CONC(IXD,IZ)*F2
      PART(3,IND)=CNODE2+TDS(IXD,IZ)*F2
      GO TO 290
  280 PART(4,IND)=2.0*C1*CONC(IXD,IZ)/(C1+CONC(IXD,IZ))
      PART(3,IND)=2.0*C2*TDS(IXD,IZ)/(C2+TDS(IXD,IZ))
```

```
  290 IF (C1-CONC(IXD,IZ)) 300,310,320
  300 RP(KR)=CONC(IXD,IZ)-PART(4,IND)
      RN(KR)=C1-PART(4,IND)
      GO TO 330
  310 RP(KR)=0.0
      RN(KR)=0.0
      GO TO 330
  320 RP(KR)=C1-PART(4,IND)
      RN(KR)=CONC(IXD,IZ)-PART(4,IND)
  330 IF (C2-TDS(IXD,IZ)) 340,350,360
  340 RP2(KR)=TDS(IXD,IZ)-PART(3,IND)
      RN2(KR)=C2-PART(3,IND)
      GO TO 370
  350 RP2(KR)=0.0
      RN2(KR)=0.0
      GO TO 370
  360 RP2(KR)=C2-PART(3,IND)
      RN2(KR)=TDS(IXD,IZ)-PART(3,IND)
  370 IND=IND+1
      PART(1,IND)=IX
      PART(2,IND)=IZ+F2*EVET
      PART(2,IND)=-PART(2,IND)
      PART(3,IND)=C2
      PART(4,IND)=C1
      IF (EVET.LT.0) PTID(IND)=7
      IF (EVET.GT.0) PTID(IND)=9
      IF (TEST.LT.98.0.OR.TEST2.GT.0.0) GO TO 470
      IZD=IZ+EVET
      KR=KR+1
      IPT(KR)=IND
      IF (METH.EQ.2) GO TO 380
      PART(4,IND)=CNODE+CONC(IX,IZD)*F2
      PART(3,IND)=CNODE2+TDS(IX,IZD)*F2
      GO TO 390
  380 PART(4,IND)=2.0*C1*CONC(IX,IZD)/(C1+CONC(IX,IZD))
      PART(3,IND)=2.0*C2*TDS(IX,IZD)/(C2+TDS(IX,IZD))
  390 IF (C1-CONC(IX,IZD)) 400,410,420
  400 RP(KR)=CONC(IX,IZD)-PART(4,IND)
      RN(KR)=C1-PART(4,IND)
      GO TO 430
  410 RP(KR)=0.0
      RN(KR)=0.0
      GO TO 430
  420 RP(KR)=C1-PART(4,IND)
      RN(KR)=CONC(IX,IZD)-PART(4,IND)
  430 IF (C2-TDS(IX,IZD)) 440,450,460
  440 RP2(KR)=TDS(IX,IZD)-PART(3,IND)
      RN2(KR)=C2-PART(3,IND)
      GO TO 470
  450 RP2(KR)=0.0
      RN2(KR)=0.0
      GO TO 470
  460 RP2(KR)=C2-PART(3,IND)
      RN2(KR)=TDS(IX,IZD)-PART(3,IND)
  470 IND=IND+1
C
  480 IF (TEST.LT.98.0.OR.TEST2.GT.0.0) GO TO 720
```

```
      SUMPT=0.0
C        ---COMPUTE CONC. GRADIENT WITHIN CELL---
      IF (NCONST.LT.2) GO TO 600
      DO 490 KPT=1,NCHK
      IK=IPT(KPT)
  490 SUMPT=PART(4,IK)+SUMPT
      CBAR=SUMPT/NCHK
C        ---CHECK MASS BALANCE WITHIN CELL AND ADJUST PT. CONCS.---
      SUMPT=0.0
      IF (CBAR-C1) 500,600,520
  500 CRCT=1.0-(CBAR/C1)
      IF (METH.EQ.1) CRCT=CBAR/C1
      DO 510 KPT=1,NCHK
      IK=IPT(KPT)
      PART(4,IK)=PART(4,IK)+RP(KPT)*CRCT
  510 SUMPT=SUMPT+PART(4,IK)
      CBARN=SUMPT/NCHK
      GO TO 540
  520 CRCT=1.0-(C1/CBAR)
      IF (METH.EQ.1) CRCT=C1/CBAR
      DO 530 KPT=1,NCHK
      IK=IPT(KPT)
      PART(4,IK)=PART(4,IK)+RN(KPT)*CRCT
  530 SUMPT=SUMPT+PART(4,IK)
      CBARN=SUMPT/NCHK
  540 IF (CBARN.EQ.C1) GO TO 600
C        ---CORRECT FOR OVERCOMPENSATION---
      CRCT=C1/CBARN
      DO 570 KPT=1,NCHK
      IK=IPT(KPT)
      PART(4,IK)=PART(4,IK)*CRCT
C        ---CHECK CONSTRAINTS---
      IF (PART(4,IK)-C1) 550,570,560
  550 CLIM=C1-RP(KPT)+RN(KPT)
      IF (PART(4,IK).LT.CLIM) GO TO 580
      GO TO 570
  560 CLIM=C1+RP(KPT)-RN(KPT)
      IF (PART(4,IK).GT.CLIM) GO TO 580
  570 CONTINUE
      GO TO 600
  580 TEST2=1.0
      DO 590 KPT=1,NCHK
      IK=IPT(KPT)
  590 PART(4,IK)=C1
  600 DO 610 KPT=1,NCHK
      IK=IPT(KPT)
  610 SUMPT=PART(3,IK)+SUMPT
      CBAR=SUMPT/NCHK
C        ---CHECK MASS BALANCE WITHIN CELL AND ADJUST PT. CONCS.---
      SUMPT=0.0
      IF (CBAR-C2) 620,720,640
  620 CRCT=1.0-(CBAR/C2)
      IF (METH.EQ.1) CRCT=CBAR/C2
      DO 630 KPT=1,NCHK
      IK=IPT(KPT)
      PART(3,IK)=PART(3,IK)+RP2(KPT)*CRCT
  630 SUMPT=SUMPT+PART(3,IK)
```

```
      CBARN=SUMPT/NCHK
      GO TO 660
  640 CRCT=1.0-(C2/CBAR)
      IF (METH.EQ.1) CRCT=C2/CBAR
      DO 650 KPT=1,NCHK
      IK=IPT(KPT)
      PART(3,IK)=PART(3,IK)+RN2(KPT)*CRCT
  650 SUMPT=SUMPT+PART(3,IK)
      CBARN=SUMPT/NCHK
  660 IF (CBARN.EQ.C2) GO TO 720
C         ---CORRECT FOR OVERCOMPENSATION---
      CRCT=C2/CBARN
      DO 690 KPT=1,NCHK
      IK=IPT(KPT)
      PART(3,IK)=PART(3,IK)*CRCT
C         ---CHECK CONSTRAINTS---
      IF (PART(3,IK)-C2) 670,690,680
  670 CLIM=C2-RP2(KPT)+RN2(KPT)
      IF (PART(3,IK).LT.CLIM) GO TO 700
      GO TO 690
  680 CLIM=C2+RP2(KPT)-RN2(KPT)
      IF (PART(3,IK).GT.CLIM) GO TO 700
  690 CONTINUE
      GO TO 720
  700 TEST2=1.0
      DO 710 KPT=1,NCHK
      IK=IPT(KPT)
  710 PART(3,IK)=C2
  720 CONTINUE
      NP=IND
      IF (INT.EQ.0) CALL CHMOT
C     ****************************************************************
      RETURN
C     ****************************************************************
      END
      SUBROUTINE VELO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /PRMJ/ NTIM,NPMP,NPNT,NITP,N,NX,NZ,NP,NREC,INT,NNX,NNZ,NUMO
     1BS,NMOV,IMOV,NPMAX,ITMAX,NZCRIT,IPRNT,NPTPND,NPNTMV,NPNTVL,NPNTD,N
     2PNCHV,NPDELC,ICHK,NCONST
      COMMON /PRMC/ NODEID(24,20),NPCELL(24,20),NPOLD(24,20),LIMBO(500),
     1IXOBS(5),IZOBS(5)
      COMMON /PRESS/ PERM(24,20),PMRX(24,20,4),PI(24,20),PR(24,20),PC(24
     1,20),PK(24,20),REC(24,20),DENS(24,20),GTERM(24,20),VISC(24,20),VPR
     2M(24,20),TMWL(5,50),TMOBS(50),TIM(100),AOPT(20),TITLE(10),ANFCTR,X
     3DEL,ZDEL,WIDTH,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,VOL
      COMMON /DENVIS/ DEN1,DEN2,VIS1,VIS2
      COMMON /XINV/ DXINV,DZINV,ARINV,PORINV
      COMMON /CHMA/ PART(4,6400),CONC(24,20),TDS(24,20),VX(24,20),VZ(24,
     120),CONINT(24,20),TDSINT(24,20),CNREC(24,20),TDSREC(24,20),TMCN(5,
     250),TMTDS(5,50),POROS,SUMTCH,BETA,TIMV,STORM,STORT,STORMI,STORTI,C
     3MSIN,TDSIN,CMSOUT,TDSOUT,FLMIN,FLTIN,FLMOT,FLTOT,SUMIO,TDSIO,CELDI
     4S,DLTRAT,CSTORM,CSTORT,DMOLEC
      COMMON /CHMC/ SUMC(24,20),VXBDY(24,20),VZBDY(24,20),SUMTDS(24,20),
     1WTFCTR(24,20),SUMWT(24,20),PTQ(24,20),PTWT(6400),ELEV(24,20)
      COMMON /DIFUS/ DISP(24,20,4)
C     ***************************************************************
```

```
C     ---COMPUTE VELOCITIES AND STORE---
      VMAX=1.0E-10
      VMAZ=1.0E-10
      VMXBD=1.0E-10
      VMZBD=1.0E-10
      TMV=TIM(N)*1.0E5
      LIM=0
      MAXX=0
      MAXZ=0
C
      DO 40 IX=1,NX
      DO 40 IZ=1,NZ
      WTFCTR(IX,IZ)=1.0
      DO 10 IY=1,4
   10 DISP(IX,IZ,IY)=0.0
C
      IF (PERM(IX,IZ).EQ.0.0) GO TO 40
      DENSE=DENS(IX,IZ)
      IF (VPRM(IX,IZ).GT.0.0.AND.(PI(IX,IZ).GT.PK(IX,IZ))) DENSE=DEN1*TD
     1SREC(IX,IZ)+DEN2
      SLEAK=VPRM(IX,IZ)/DENS(IX,IZ)
      SLEAK=SLEAK*(PI(IX,IZ)-PK(IX,IZ)+DENS(IX,IZ)*ELEV(IX,IZ))
      DENSE=DENS(IX,IZ)
      IF (REC(IX,IZ).LT.0.0) DENSE=DEN1*TDSREC(IX,IZ)+DEN2
      DIV=SLEAK+REC(IX,IZ)/DENSE
C
C        ---VELOCITIES AT NODES---
C           ---X-DIRECTION---
      DPX=PK(IX-1,IZ)-PK(IX+1,IZ)
      IF (PERM(IX-1,IZ).EQ.0.0) DPX=PK(IX,IZ)-PK(IX+1,IZ)
      IF (PERM(IX+1,IZ).EQ.0.0) DPX=PK(IX-1,IZ)-PK(IX,IZ)
      IF (PERM(IX-1,IZ).EQ.0.0.AND.PERM(IX+1,IZ).EQ.0.0) DPX=0.0
      GRDX=DPX*DXINV*0.50
      VX(IX,IZ)=PERM(IX,IZ)*GRDX*PORINV/VISC(IX,IZ)
      ABVX=ABS(VX(IX,IZ))
      IF (ABVX.GT.VMAX) VMAX=ABVX
C           ---Z-DIRECTION---
      DPZ=PK(IX,IZ-1)-PK(IX,IZ+1)
      DENSE=0.25*DENS(IX,IZ-1)+0.25*DENS(IX,IZ+1)+0.500*DENS(IX,IZ)
      GRDZ=DPZ*DZINV*0.500+DENSE
      IF (PERM(IX,IZ-1).EQ.0.0.AND.PERM(IX,IZ+1).EQ.0.0) GO TO 20
      IF (PERM(IX,IZ-1).EQ.0.0) DPZ=(PK(IX,IZ)-PK(IX,IZ+1))
      IF (PERM(IX,IZ-1).EQ.0.0) DENSE=(DENS(IX,IZ)+DENS(IX,IZ+1))/2.0
      IF (PERM(IX,IZ-1).EQ.0.0) GRDZ=(DPZ*DZINV+DENSE)*0.500
      IF (PERM(IX,IZ+1).EQ.0.0) DPZ=(PK(IX,IZ-1)-PK(IX,IZ))
      IF (PERM(IX,IZ+1).EQ.0.0) DENSE=(DENS(IX,IZ)+DENS(IX,IZ-1))/2.0
      IF (PERM(IX,IZ+1).EQ.0.0) GRDZ=(DPZ*DZINV+DENSE)*0.500
      GO TO 30
   20 GRDZ=0.0
   30 VZ(IX,IZ)=PERM(IX,IZ)*GRDZ*PORINV*ANFCTR/VISC(IX,IZ)
      ABVZ=ABS(VZ(IX,IZ))
      IF (ABVZ.GT.VMAZ) VMAZ=ABVZ
C
C        ---VELOCITIES AT CELL BOUNDARIES---
      GRDX=(PK(IX,IZ)-PK(IX+1,IZ))*DXINV
      VXBDY(IX,IZ)=PMRX(IX,IZ,3)*GRDX*PORINV
      DENSE=(DENS(IX,IZ)+DENS(IX,IZ+1))*0.500
```

```
      GRDZ=(PK(IX,IZ)-PK(IX,IZ+1))*DZINV+DENSE
      VZBDY(IX,IZ)=PMRX(IX,IZ,4)*GRDZ*PORINV
      ABVX=ABS(VXBDY(IX,IZ))
      ABVZ=ABS(VZBDY(IX,IZ))
      IF (ABVX.GT.VMXBD) VMXBD=ABVX
      IF (ABVZ.GT.VMZBD) VMZBD=ABVZ
C
      IF (DIV.GE.0.0) GO TO 40
      TDIV=POROS/DABS(DIV)
      IF (TDIV.GE.TMV) GO TO 40
      TMV=TDIV
      MAXX=IX
      MAXZ=IZ
   40 CONTINUE
C     *****************************************************************
C     ---PRINT VELOCITIES---
      IF (NPNTVL.EQ.0) GO TO 100
      IF (NPNTVL.EQ.2) GO TO 50
      IF (NPNTVL.EQ.1.AND.N.EQ.1.AND.IMOV.EQ.0) GO TO 50
      GO TO 100
   50 WRITE (6,400)
      WRITE (6,410)
      DO 60 IZ=1,NZ
   60 WRITE (6,430) (VX(IX,IZ),IX=1,NX)
      WRITE (6,420)
      DO 70 IZ=1,NZ
   70 WRITE (6,430) (VXBDY(IX,IZ),IX=1,NX)
      WRITE (6,440)
      WRITE (6,410)
      DO 80 IZ=1,NZ
   80 WRITE (6,430) (VZ(IX,IZ),IX=1,NX)
      WRITE (6,420)
      DO 90 IZ=1,NZ
   90 WRITE (6,430) (VZBDY(IX,IZ),IX=1,NX)
C     ---WRITE VELOCITIES TO UNIT 7---
  100 IF (NPNCHV.EQ.0) GO TO 130
      IF (NPNCHV.EQ.2) GO TO 110
      IF (NPNCHV.EQ.1.AND.N.EQ.1) GO TO 110
      GO TO 130
  110 WRITE (7,590) NX,NZ,XDEL,ZDEL,VMAX,VMAZ
      DO 120 IZ=1,NZ
      WRITE (7,600) (VX(IX,IZ),IX=1,NX)
  120 WRITE (7,600) (VZ(IX,IZ),IX=1,NX)
C     ******************************************************************
C     ---COMPUTE NEXT TIME STEP---
  130 WRITE (6,470)
      WRITE (6,480) VMAX,VMAZ
      WRITE (6,490) VMXBD,VMZBD
      TDELX=CELDIS*XDEL/VMAX
      TDELZ=CELDIS*ZDEL/VMAZ
      TDELXB=CELDIS*XDEL/VMXBD
      TDELZB=CELDIS*ZDEL/VMZBD
      TIMV=DMIN1(TDELX,TDELZ,TDELXB,TDELZB)
      IF (DMAX1(VMAX,VMAZ,VMXBD,VMZBD).LE.1.0E-10) WRITE (6,650)
      WRITE (6,390) TMV,TIMV
      IF (TMV.LT.TIMV) GO TO 140
      LIM=-1
```

```
      GO TO 150
  140 TIMV=TMV
      LIM=1
  150 NTIMV=TIM(N)/TIMV
      NMOV=NTIMV+1
      WRITE (6,500) TIMV,NTIMV,NMOV
      TIMV=TIM(N)/NMOV
      WRITE (6,450) TIM(N)
      WRITE (6,460) TIMV
C     ******************************************************************
C     --- CALCULATE THE WEIGHTING FACTORS FOR SOURCES AND SINKS ---
C
      DO 190 IX=1,NX
      DO 190 IZ=1,NZ
      IF (PERM(IX,IZ).EQ.0.0) GO TO 190
      IF (PTQ(IX,IZ).EQ.0.0) GO TO 190
      SUMQ=0.0
      SUMQIN=0.0
      QSNK=0.0
      QSRC=0.0
      Q=VXBDY(IX-1,IZ)*POROS*ZDEL*WIDTH
      IF (Q.LT.0.0) SUMQ=SUMQ-Q
      IF (Q.GT.0.0) SUMQIN=SUMQIN+Q
      Q=VXBDY(IX,IZ)*POROS*ZDEL*WIDTH
      IF (Q.GT.0.0) SUMQ=SUMQ+Q
      IF (Q.LT.0.0) SUMQIN=SUMQIN-Q
      Q=VZBDY(IX,IZ-1)*POROS*XDEL*WIDTH
      IF (Q.LT.0.0) SUMQ=SUMQ-Q
      IF (Q.GT.0.0) SUMQIN=SUMQIN+Q
      Q=VZBDY(IX,IZ)*POROS*XDEL*WIDTH
      IF (Q.GT.0.0) SUMQ=SUMQ+Q
      IF (Q.LT.0.0) SUMQIN=SUMQIN-Q
      IF (REC(IX,IZ).GT.0.0) GO TO 180
  160 QSRC=QSRC-REC(IX,IZ)
      IF (VPRM(IX,IZ).LE.0.0) GO TO 170
      QSRC=QSRC+VPRM(IX,IZ)*VOL*(PI(IX,IZ)-PK(IX,IZ)+DENS(IX,IZ)*ELEV(IX
     1,IZ))/DENS(IX,IZ)
      IF (QSRC.LT.0.0) GO TO 180
  170 WTFCTR(IX,IZ)=QSRC/SUMQ
      IF (WTFCTR(IX,IZ).GT.0.999) WTFCTR(IX,IZ)=1.0
      GO TO 190
  180 QSNK=QSNK+REC(IX,IZ)
      QSNK=QSNK-VOL*VPRM(IX,IZ)*(PI(IX,IZ)-PK(IX,IZ)+DENS(IX,IZ)*ELEV(IX
     1,IZ))/DENS(IX,IZ)
      IF (QSNK.LT.0.0) GO TO 160
      WTFCTR(IX,IZ)=1.0-(QSNK/SUMQIN)
      IF (WTFCTR(IX,IZ).LT.0.001) WTFCTR(IX,IZ)=0.0
  190 CONTINUE
C     WRITE (6,680)
C     DO 136 IZ=1,NZ
C 136 WRITE (6,690)  (WTFCTR(IX,IZ),IX=1,NX)
      IF (BETA.EQ.0.0.AND.DMOLEC.EQ.0.0) GO TO 270
C     ******************************************************************
C     ---COMPUTE DISPERSION COEFFICIENTS---
      ALPHA=BETA
      ALNG=ALPHA
      TRAN=DLTRAT*ALPHA
```

```
      XX2=XDEL*XDEL
      ZZ2=ZDEL*ZDEL
      XZ2=4.0*XDEL*ZDEL
      DO 210 IX=2,NNX
      DO 210 IZ=2,NNZ
      IF (PERM(IX,IZ).EQ.0.0) GO TO 210
      VXE=VXBDY(IX,IZ)
      VZS=VZBDY(IX,IZ)
      IF (PERM(IX+1,IZ).EQ.0.0) GO TO 200
C        ---FORWARD COEFFICIENTS: X-DIRECTION---
      VZE=(VZBDY(IX,IZ-1)+VZBDY(IX+1,IZ-1)+VZS+VZBDY(IX+1,IZ))/4.0
      VXE2=VXE*VXE
      VZE2=VZE*VZE
      VMGE=SQRT(VXE2+VZE2)
      IF (VMGE.LT.1.0E-20) GO TO 200
      DALN=ALNG*VMGE
      DTRN=TRAN*VMGE
      VMGE2=VMGE*VMGE
C           ---XX COEFFICIENT---
      DISP(IX,IZ,1)=(DALN*VXE2+DTRN*VZE2)/(VMGE2*XX2)
C           ---XZ COEFFICIENT---
      DISP(IX,IZ,3)=(DALN-DTRN)*VXE*VZE/(VMGE2*XZ2)
C        ---FORWARD COEFFICIENTS: Z-DIRECTION---
  200 IF (PERM(IX,IZ+1).EQ.0.0) GO TO 210
      VXS=(VXBDY(IX-1,IZ)+VXE+VXBDY(IX-1,IZ+1)+VXBDY(IX,IZ+1))/4.0
      VZS2=VZS*VZS
      VXS2=VXS*VXS
      VMGS=SQRT(VXS2+VZS2)
      IF (VMGS.LT.1.0E-20) GO TO 210
      DALN=ALNG*VMGS
      DTRN=TRAN*VMGS
      VMGS2=VMGS*VMGS
C           ---ZZ COEFFICIENT---
      DISP(IX,IZ,2)=(DALN*VZS2+DTRN*VXS2)/(VMGS2*ZZ2)
C           ---ZX COEFFICIENT---
      DISP(IX,IZ,4)=(DALN-DTRN)*VXS*VZS/(VMGS2*XZ2)
  210 CONTINUE
C     ******************************************************************
C     ---ADJUST CROSS-PRODUCT TERMS FOR ZERO THICKNESS---
      DO 240 IX=2,NNX
      DO 240 IZ=2,NNZ
      IF (PERM(IX,IZ).EQ.0.0) GO TO 230
      IF (PERM(IX+1,IZ).EQ.0.0) GO TO 220
      DISP(IX,IZ,1)=DISP(IX,IZ,1)+DMOLEC/XX2
  220 IF (PERM(IX,IZ+1).EQ.0.0) GO TO 230
      DISP(IX,IZ,2)=DISP(IX,IZ,2)+DMOLEC/ZZ2
  230 IF (PERM(IX,IZ+1).EQ.0.0.OR.PERM(IX+1,IZ+1).EQ.0.0.OR.PERM(IX,IZ-1
     1).EQ.0.0.OR.PERM(IX+1,IZ-1).EQ.0.0) DISP(IX,IZ,3)=0.0
      IF (PERM(IX+1,IZ).EQ.0.0.OR.PERM(IX+1,IZ+1).EQ.0.0.OR.PERM(IX-1,IZ
     1).EQ.0.0.OR.PERM(IX-1,IZ+1).EQ.0.0) DISP(IX,IZ,4)=0.0
  240 CONTINUE
C     ******************************************************************
C     ---CHECK FOR STABILITY OF EXPLICIT METHOD---
      TIMDIS=0.0
      DO 250 IX=2,NNX
      DO 250 IZ=2,NNZ
      TDCO=DISP(IX,IZ,1)+DISP(IX,IZ,2)
```

```
  250 IF (TDCO.GT.TIMDIS) TIMDIS=TDCO
      TIMDC=0.5/TIMDIS
      WRITE (6,520) TIMDC
      NTIMD=TIM(N)/TIMDC
      NDISP=NTIMD+1
      IF (NDISP.LE.NMOV) GO TO 260
      NMOV=NDISP
      TIMV=TIM(N)/NMOV
      LIM=0
  260 IF ((TIM(N)-SUMTCH).LT.TIMV) TIMV=TIM(N)-SUMTCH
      WRITE (6,510) TIMV,NTIMD,NMOV
C     ****************************************************************
  270 IF (NMOV.EQ.1) GO TO 310
      IF (LIM) 280,290,300
  280 WRITE (6,610)
      GO TO 320
  290 WRITE (6,620)
      GO TO 320
  300 WRITE (6,630)
      WRITE (6,640) MAXX,MAXZ
      GO TO 320
  310 WRITE (6,660)
C     ****************************************************************
C     ---PRINT DISPERSION EQUATION COEFFICIENTS---
  320 IF (NPNTD.EQ.0) GO TO 380
      IF (NPNTD.EQ.2) GO TO 330
      IF (NPNTD.EQ.1.AND.N.EQ.1.AND.IMOV.EQ.0) GO TO 330
      GO TO 380
  330 WRITE (6,530)
      WRITE (6,540)
      DO 340 IZ=1,NZ
  340 WRITE (6,580) (DISP(IX,IZ,1),IX=1,NX)
      WRITE (6,550)
      DO 350 IZ=1,NZ
  350 WRITE (6,580) (DISP(IX,IZ,2),IX=1,NX)
      WRITE (6,560)
      DO 360 IZ=1,NZ
  360 WRITE (6,580) (DISP(IX,IZ,3),IX=1,NX)
      WRITE (6,570)
      DO 370 IZ=1,NZ
  370 WRITE (6,580) (DISP(IX,IZ,4),IX=1,NX)
C     ****************************************************************
  380 RETURN
C     ****************************************************************
C
C
C
  390 FORMAT (1H ,19H TMV (MAX. INJ.) = ,G12.5/20H  TIMV (CELDIS)   = ,G
     112.5)
  400 FORMAT (1H1,12HX VELOCITIES)
  410 FORMAT (1H ,25X,8HAT NODES/)
  420 FORMAT (1H0,25X,13HON BOUNDARIES/)
  430 FORMAT (1H ,10G12.3)
  440 FORMAT (1H1,12HZ VELOCITIES)
  450 FORMAT (3H   ,11HTIM (N)   = ,1G12.5)
  460 FORMAT (3H   ,11HTIMEVELO = ,1G12.5)
  470 FORMAT (1H1,10X,29HSTABILITY CRITERIA --- M.O.C.//)
```

```
480 FORMAT (1H0,8H VMAX = ,1PE9.2,5X,7HVMAZ = ,1PE9.2)
490 FORMAT (1H ,8H VMXBD= ,1PE9.2,5X,7HVMZBD= ,1PE9.2)
500 FORMAT (1H0,8H TIMV = ,1PE9.2,5X,8HNTIMV = ,I5,5X,7HNMOV = ,I5/)
510 FORMAT (1H0,8H TIMV = ,1PE9.2,5X,8HNTIMD = ,I5,5X,7HNMOV = ,I5)
520 FORMAT (3H   ,11HTIMEDISP = ,1E12.5)
530 FORMAT (1H1,32HDISPERSION EQUATION COEFFICIENTS,10X,25H=(D-IJ)*(B)
   1/(GRID FACTOR))
540 FORMAT (1H ,35X,14HXX COEFFICIENT/)
550 FORMAT (1H ,35X,14HZZ COEFFICIENT/)
560 FORMAT (1H ,35X,14HXZ COEFFICIENT/)
570 FORMAT (1H ,35X,14HZX COEFFICIENT/)
580 FORMAT (1H ,1P10E8.1)
590 FORMAT (2I4,2F10.1,2F10.7)
600 FORMAT (8F10.7)
610 FORMAT (1H0,10X,42HTHE LIMITING STABILITY CRITERION IS CELDIS)
620 FORMAT (1H0,10X,40HTHE LIMITING STABILITY CRITERION IS BETA)
630 FORMAT (1H0,10X,58HTHE LIMITING STABILITY CRITERION IS MAXIMUM INJ
   1ECTION RATE)
640 FORMAT (1H ,15X,35HMAX. INJECTION OCCURS AT CELL IX = ,I3,7H  IZ =
   1 ,I3)
650 FORMAT (1H0,5X,47H*** WARNING ***  DECREASE CRITERIA IN E 230-260)
660 FORMAT (1H0,10X,63H*TIME INCREMENT FOR SOLUTE TRANSPORT EQUALS TIM
   1E STEP FOR FLOW*)
680 FORMAT (1H ,17HWEIGHTING FACTORS)
690 FORMAT (1H ,20F5.2)
    END
    SUBROUTINE MVPT
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    INTEGER *2PTID
    COMMON /PRMJ/ NTIM,NPMP,NPNT,NITP,N,NX,NZ,NP,NREC,INT,NNX,NNZ,NUMO
   1BS,NMOV,IMOV,NPMAX,ITMAX,NZCRIT,IPRNT,NPTPND,NPNTMV,NPNTVL,NPNTD,N
   2PNCHV,NPDELC,ICHK,NCONST
    COMMON /PRMC/ NODEID(24,20),NPCELL(24,20),NPOLD(24,20),LIMBO(500),
   1IXOBS(5),IZOBS(5)
    COMMON /PRESS/ PERM(24,20),PMRX(24,20,4),PI(24,20),PR(24,20),PC(24
   1,20),PK(24,20),REC(24,20),DENS(24,20),GTERM(24,20),VISC(24,20),VPR
   2M(24,20),TMWL(5,50),TMOBS(50),TIM(100),AOPT(20),TITLE(10),ANFCTR,X
   3DEL,ZDEL,WIDTH,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,VOL
    COMMON /XINV/ DXINV,DZINV,ARINV,PORINV
    COMMON /CHMA/ PART(4,6400),CONC(24,20),TDS(24,20),VX(24,20),VZ(24,
   120),CONINT(24,20),TDSINT(24,20),CNREC(24,20),TDSREC(24,20),TMCN(5,
   250),TMTDS(5,50),POROS,SUMTCH,BETA,TIMV,STORM,STORT,STORMI,STORTI,C
   3MSIN,TDSIN,CMSOUT,TDSOUT,FLMIN,FLTIN,FLMOT,FLTOT,SUMIO,TDSIO,CELDI
   4S,DLTRAT,CSTORM,CSTORT,DMOLEC
    COMMON /CHMC/ SUMC(24,20),VXBDY(24,20),VZBDY(24,20),SUMTDS(24,20),
   1WTFCTR(24,20),SUMWT(24,20),PTQ(24,20),PTWT(6400),ELEV(24,20)
    COMMON /CHMP/ PTID(6400)
C   ***************************************************************
    WRITE (6,970) NMOV
    SUMTCH=SUMT-TIM(N)
    F1=0.30
    F2=1.0/3.0
    IF (NPTPND.EQ.4) F1=0.25
    IF (NPTPND.EQ.9) F1=F2
    IF (NPTPND.EQ.8) F2=0.25
    IF (NPTPND.EQ.16) F1=0.25
    IF (NPTPND.EQ.16) F2=0.125
```

```
C     ---MOVE PARTICLES 'NMOV' TIMES---
      IMOV=0
   10 IMOV=IMOV+1
      CONST1=TIMV*DXINV
      CONST2=TIMV*DZINV
   20 NPTM=NP
C         ---MOVE EACH PARTICLE---
      DO 880 IN=1,NP
      IF (PART(1,IN).EQ.0.0) GO TO 880
C     ****************************************************************
C           ---COMPUTE OLD LOCATION---
      XOLD=PART(1,IN)
      IX=XOLD+0.5
      IFLAG=1
      IF (PART(2,IN).GE.0.0) GO TO 30
      IFLAG=-1
      PART(2,IN)=-PART(2,IN)
   30 ZOLD=PART(2,IN)
      IZ=ZOLD+0.5
      IF (PERM(IX,IZ).EQ.0.0) GO TO 880
C     ****************************************************************
C           ---COMPUTE NEW LOCATION AND LOCATE CLOSEST NODE---
C           ---LOCATE NORTHWEST CORNER---
      IVX=XOLD
      IVZ=ZOLD
      IXE=IVX+1
      IZS=IVZ+1
C     ****************************************************************
C        ---LOCATE QUADRANT, VEL. AT 4 CORNERS, CHECK FOR BOUNDARIES---
      CELDX=XOLD-IX
      CELDZ=ZOLD-IZ
      ICD=9
      IF (CELDX.EQ.0.0.AND.CELDZ.EQ.0.0) GO TO 450
      IF (CELDX.GE.0.0.OR.CELDZ.GE.0.0) GO TO 70
C              ---PT. IN NW QUADRANT---
      VXNW=VXBDY(IVX,IVZ)
      VXNE=VX(IXE,IVZ)
      VXSW=VXBDY(IVX,IZS)
      VXSE=VX(IXE,IZS)
      VZNW=VZBDY(IVX,IVZ)
      VZNE=VZBDY(IXE,IVZ)
      VZSW=VZ(IVX,IZS)
      VZSE=VZ(IXE,IZS)
      ICD=1
      IF (PERM(IVX,IVZ).EQ.0.0) GO TO 50
      IF (PTQ(IXE,IVZ).EQ.0.0) GO TO 40
      VXNE=VXNW
   40 IF (PTQ(IVX,IZS).EQ.0.0) GO TO 50
      VZSW=VZNW
   50 IF (PTQ(IXE,IZS).EQ.0.0) GO TO 270
      IF (PERM(IVX,IZS).EQ.0.0) GO TO 60
      IF (PERM(IXE+1,IZS).GT.0.0) VXSE=VXSW
   60 IF (PERM(IXE,IVZ).EQ.0.0) GO TO 270
      IF (PERM(IXE,IZS+1).GT.0.0) VZSE=VZNE
      GO TO 270
C
   70 IF (CELDX.LE.0.0.OR.CELDZ.GE.0.0) GO TO 130
```

```
C              ---PT. IN NE QUADRANT---
   80 VXNW=VX(IVX,IVZ)
      VXNE=VXBDY(IVX,IVZ)
      VXSW=VX(IVX,IZS)
      VXSE=VXBDY(IVX,IZS)
      VZNW=VZBDY(IVX,IVZ)
      VZNE=VZBDY(IXE,IVZ)
      VZSW=VZ(IVX,IZS)
      VZSE=VZ(IXE,IZS)
      ICD=2
      IF (CELDX.EQ.0.0) GO TO 120
      IF (PERM(IXE,IVZ).EQ.0.0) GO TO 100
      IF (PTQ(IXE,IZS).EQ.0.0) GO TO 90
      VXNW=VXNE
   90 IF (PTQ(IXE,IZS).EQ.0.0) GO TO 100
      VZSE=VZNE
  100 IF (PTQ(IVX,IZS).EQ.0.0) GO TO 270
      IF (PERM(IXE,IZS).EQ.0.0) GO TO 110
      IF (PERM(IVX-1,IZS).GT.0.0) VXSW=VXSE
  110 IF (PERM(IVX,IVZ).EQ.0.0) GO TO 270
      IF (PERM(IVX,IZS+1).GT.0.0) VZSW=VZNW
      GO TO 270
  120 IF (PTQ(IVX,IZS).EQ.0.0) GO TO 270
      IF (PERM(IVX,IVZ).EQ.0.0) GO TO 270
      IF (PERM(IVX,IZS+1).GT.0.0) VZSW=VZNW
      GO TO 270
C
  130 IF (CELDZ.LE.0.0.OR.CELDX.GE.0.0) GO TO 190
C              ---PT. IN SW QUADRANT---
  140 VXNW=VXBDY(IVX,IVZ)
      VXNE=VX(IXE,IVZ)
      VXSW=VXBDY(IVX,IZS)
      VXSE=VX(IXE,IZS)
      VZNW=VZ(IVX,IVZ)
      VZNE=VZ(IXE,IVZ)
      VZSW=VZBDY(IVX,IVZ)
      VZSE=VZBDY(IXE,IVZ)
      ICD=3
      IF (CELDZ.EQ.0.0) GO TO 180
      IF (PERM(IVX,IZS).EQ.0.0) GO TO 160
      IF (PTQ(IVX,IVZ).EQ.0.0) GO TO 150
      VZNW=VZSW
  150 IF (PTQ(IXE,IZS).EQ.0.0) GO TO 160
      VXSE=VXSW
  160 IF (PTQ(IXE,IVZ).EQ.0.0) GO TO 270
      IF (PERM(IVX,IVZ).EQ.0.0) GO TO 170
      IF (PERM(IXE+1,IVZ).GT.0.0) VXNE=VXNW
  170 IF (PERM(IXE,IZS).EQ.0.0) GO TO 270
      IF (PERM(IXE,IVZ-1).GT.0.0) VZNE=VZSE
      GO TO 270
  180 IF (PTQ(IXE,IVZ).EQ.0.0) GO TO 270
      IF (PERM(IVX,IVZ).EQ.0.0) GO TO 270
      IF (PERM(IXE+1,IVZ).GT.0.0) VXNE=VXNW
      GO TO 270
C
  190 IF (CELDZ.LE.0.0.OR.CELDX.LE.0.0) GO TO 260
C              ---PT. IN SE QUADRANT---
```

```
  200 VXNW=VX(IVX,IVZ)
      VXNE=VXBDY(IVX,IVZ)
      VXSW=VX(IVX,IZS)
      VXSE=VXBDY(IVX,IZS)
      VZNW=VZ(IVX,IVZ)
      VZNE=VZ(IXE,IVZ)
      VZSW=VZBDY(IVX,IVZ)
      VZSE=VZBDY(IXE,IVZ)
      ICD=4
      IF (CELDZ.EQ.0.0) GO TO 240
      IF (CELDX.EQ.0.0) GO TO 250
      IF (PERM(IXE,IZS).EQ.0.0) GO TO 220
      IF (PTQ(IXE,IVZ).EQ.0.0) GO TO 210
      VZNE=VZSE
  210 IF (PTQ(IVX,IZS).EQ.0.0) GO TO 220
      VXSW=VXSE
  220 IF (PTQ(IVX,IVZ).EQ.0.0) GO TO 270
      IF (PERM(IXE,IVZ).EQ.0.0) GO TO 230
      IF (PERM(IVX-1,IVZ).GT.0.0) VXNW=VXNE
  230 IF (PERM(IVX,IZS).EQ.0.0) GO TO 270
      IF (PERM(IVX,IVZ-1).GT.0.0) VZNW=VZSW
      GO TO 270
  240 IF (PTQ(IVX,IVZ).EQ.0.0) GO TO 270
      IF (PERM(IXE,IVZ).EQ.0.0) GO TO 270
      IF (PERM(IVX-1,IVZ).GT.0.0) VXNW=VXNE
      GO TO 270
  250 IF (PTQ(IVX,IVZ).EQ.0.0) GO TO 270
      IF (PERM(IVX,IZS).EQ.0.0) GO TO 270
      IF (PERM(IVX,IVZ-1).GT.0.0) VZNW=VZSW
      GO TO 270
C
  260 IF (CELDX.EQ.0.0.AND.CELDZ.LT.0.0) GO TO 80
      IF (CELDX.LT.0.0.AND.CELDZ.EQ.0.0) GO TO 140
      IF (CELDX.GT.0.0.AND.CELDZ.EQ.0.0) GO TO 200
      IF (CELDX.EQ.0.0.AND.CELDZ.GT.0.0) GO TO 200
      WRITE (6,980) IN,IX,IZ
  270 CONTINUE
C                --- CHECK FOR ADJACENT NO-FLOW BOUNDARIES---
      GO TO (280,320,360,400,440), ICD
      GO TO 440
  280 IF (PERM(IXE,IVZ).EQ.0.0) GO TO 290
      IF (PERM(IVX,IZS).EQ.0.0) GO TO 300
      IF (PERM(IVX,IVZ).EQ.0.0) GO TO 310
      GO TO 440
  290 VXNE=VXSE
      IF (PERM(IVX,IZS).GT.0.0) GO TO 310
  300 VZSW=VZSE
  310 VXNW=VXSW
      VZNW=VZNE
      GO TO 440
  320 IF (PERM(IVX,IVZ).EQ.0.0) GO TO 330
      IF (PERM(IXE,IZS).EQ.0.0) GO TO 340
      IF (PERM(IXE,IVZ).EQ.0.0) GO TO 350
      GO TO 440
  330 VXNW=VXSW
      IF (PERM(IXE,IZS).GT.0.0) GO TO 350
  340 VZSE=VZSW
```

```
  350 VXNE=VXSE
      VZNE=VZNW
      GO TO 440
  360 IF (PERM(IXE,IZS).EQ.0.0) GO TO 370
      IF (PERM(IVX,IVZ).EQ.0.0) GO TO 380
      IF (PERM(IVX,IZS).EQ.0.0) GO TO 390
      GO TO 440
  370 VXSE=VXNE
      IF (PERM(IVX,IVZ).GT.0.0) GO TO 390
  380 VZNW=VZNE
  390 VXSW=VXNW
      VZSW=VZSE
      GO TO 440
  400 IF (PERM(IVX,IZS).EQ.0.0) GO TO 410
      IF (PERM(IXE,IVZ).EQ.0.0) GO TO 420
      IF (PERM(IXE,IZS).EQ.0.0) GO TO 430
      GO TO 440
  410 VXSW=VXNW
      IF (PERM(IXE,IVZ).GT.0.0) GO TO 430
  420 VZNE=VZNW
  430 VZSE=VZSW
      VXSE=VXNE
  440 CONTINUE
C     ***************************************************************
C        ---BILINEAR INTERPOLATION---
      CELXD=XOLD-IVX
      CELDXH=DMOD(CELXD,0.50D0)
      CELDX=CELDXH*2.0
      CELDZ=ZOLD-IVZ
C     ***************************************************************
C           ---X VELOCITY---
      VXN=VXNW*(1.0-CELDX)+VXNE*CELDX
      VXS=VXSW*(1.0-CELDX)+VXSE*CELDX
      XVEL=VXN*(1.0-CELDZ)+VXS*CELDZ
C           ---Z VELOCITY---
      CELDZH=DMOD(CELDZ,0.50D0)
      CELZD=CELDZH*2.0
      VZW=VZNW*(1.0-CELZD)+VZSW*CELZD
      VZE=VZNE*(1.0-CELZD)+VZSE*CELZD
      ZVEL=VZW*(1.0-CELXD)+VZE*CELXD
C
      GO TO 460
  450 XVEL=VX(IX,IZ)
      ZVEL=VZ(IX,IZ)
  460 DISTX=XVEL*CONST1
      DISTZ=ZVEL*CONST2
C     ***************************************************************
C        ---BOUNDARY CONDITIONS---
      TEMPX=XOLD+DISTX
      TEMPZ=ZOLD+DISTZ
      INX=TEMPX+0.5
      INZ=TEMPZ+0.5
      IF (PERM(INX,INZ).GT.0.0) GO TO 500
C     ***************************************************************
C           ---X BOUNDARY---
      IF (PERM(INX,IZ).EQ.0.0) GO TO 470
      PART(1,IN)=TEMPX
```

```
      GO TO 480
  470 BEYON=TEMPX-IX
      IF (BEYON.LT.0.0) BEYON=BEYON+0.5
      IF (BEYON.GT.0.0) BEYON=BEYON-0.5
      PART(1,IN)=TEMPX-2.0*BEYON
      INX=PART(1,IN)+0.5
      TEMPX=PART(1,IN)
C     ***************************************************************
C           ---Z BOUNDARY---
  480 IF (PERM(INX,INZ).EQ.0.0) GO TO 490
      PART(2,IN)=TEMPZ
      GO TO 510
C     ***************************************************************
  490 BEYON=TEMPZ-IZ
      IF (BEYON.LT.0.0) BEYON=BEYON+0.5
      IF (BEYON.GT.0.0) BEYON=BEYON-0.5
      PART(2,IN)=TEMPZ-2.0*BEYON
      INZ=PART(2,IN)+0.5
      TEMPZ=PART(2,IN)
      GO TO 510
  500 PART(1,IN)=TEMPX
      PART(2,IN)=TEMPZ
  510 CONTINUE
C     ***************************************************************
C        ---SUM CONCENTRATIONS AND COUNT PARTICLES---
      SUMC(INX,INZ)=SUMC(INX,INZ)+PART(4,IN)*PTWT(IN)
      SUMTDS(INX,INZ)=SUMTDS(INX,INZ)+PART(3,IN)*PTWT(IN)
      NPCELL(INX,INZ)=NPCELL(INX,INZ)+1
      SUMWT(INX,INZ)=SUMWT(INX,INZ)+PTWT(IN)
C     ***************************************************************
C        ---CHECK FOR CHANGE IN CELL LOCATION---
      IF (IX.EQ.INX.AND.IZ.EQ.INZ) GO TO 870
C           ---CHECK FOR CONST.-HEAD BDY. OR SOURCE AT OLD LOCATION---
      IF (PTQ(IX,IZ).LT.0.0) GO TO 520
      GO TO 830
C     ***************************************************************
C           ---CREATE NEW PARTICLES AT BOUNDARIES---
  520 IF (IFLAG.GT.0) GO TO 840
      DO 530 IL=1,500
      IF (LIMBO(IL).EQ.0) GO TO 530
      IP=LIMBO(IL)
      IF (IP.LT.IN) GO TO 540
  530 CONTINUE
C     ***************************************************************
C           ---GENERATE NEW PARTICLE---
      IF (NPTM.EQ.NPMAX) GO TO 890
      NPTM=NPTM+1
      IP=NPTM
      GO TO 550
  540 LIMBO(IL)=0
  550 ITEM=PTID(IN)
      IF (NPTPND.EQ.16) GO TO 560
      GO TO (570,580,590,600,610,620,630,640,650), ITEM
      GO TO 610
  560 GO TO (660,670,680,690,700,710,720,730,740,750,760,770,780,790,800
     1,810), ITEM
      GO TO 610
```

```
570 PART(1,IP)=IX-F1
    PART(2,IP)=IZ-F1
    PTID(IP)=1
    GO TO 820
580 PART(1,IP)=IX-F1
    PART(2,IP)=IZ+F1
    PTID(IP)=2
    GO TO 820
590 PART(1,IP)=IX+F1
    PART(2,IP)=IZ-F1
    PTID(IP)=3
    GO TO 820
600 PART(1,IP)=IX+F1
    PART(2,IP)=IZ+F1
    PTID(IP)=4
    GO TO 820
610 PART(1,IP)=IX
    PART(2,IP)=IZ
    PTID(IP)=5
    GO TO 820
620 PART(1,IP)=IX-F2
    PART(2,IP)=IZ
    PTID(IP)=6
    GO TO 820
630 PART(1,IP)=IX
    PART(2,IP)=IZ-F2
    PTID(IP)=7
    GO TO 820
640 PART(1,IP)=IX+F2
    PART(2,IP)=IZ
    PTID(IP)=8
    GO TO 820
650 PART(1,IP)=IX
    PART(2,IP)=IZ+F2
    PTID(IP)=9
    GO TO 820
660 PART(1,IP)=IX-F1-F2
    PART(2,IP)=IZ-F1-F2
    PTID(IP)=1
    GO TO 820
670 PART(1,IP)=IX-F1-F2
    PART(2,IP)=IZ-F1+F2
    PTID(IP)=2
    GO TO 820
680 PART(1,IP)=IX-F1+F2
    PART(2,IP)=IZ-F1-F2
    PTID(IP)=3
    GO TO 820
690 PART(1,IP)=IX-F1+F2
    PART(2,IP)=IZ-F1+F2
    PTID(IP)=4
    GO TO 820
700 PART(1,IP)=IX-F1-F2
    PART(2,IP)=IZ+F1-F2
    PTID(IP)=5
    GO TO 820
710 PART(1,IP)=IX-F1-F2
```

```
      PART(2,IP)=IZ+F1+F2
      PTID(IP)=6
      GO TO 820
  720 PART(1,IP)=IX-F1+F2
      PART(2,IP)=IZ+F1-F2
      PTID(IP)=7
      GO TO 820
  730 PART(1,IP)=IX-F1+F2
      PART(2,IP)=IZ+F1+F2
      PTID(IP)=8
      GO TO 820
  740 PART(1,IP)=IX+F1-F2
      PART(2,IP)=IZ-F1-F2
      PTID(IP)=9
      GO TO 820
  750 PART(1,IP)=IX+F1-F2
      PART(2,IP)=IZ-F1+F2
      PTID(IP)=10
      GO TO 820
  760 PART(1,IP)=IX+F1+F2
      PART(2,IP)=IZ-F1-F2
      PTID(IP)=11
      GO TO 820
  770 PART(1,IP)=IX+F1+F2
      PART(2,IP)=IZ-F1+F2
      PTID(IP)=12
      GO TO 820
  780 PART(1,IP)=IX+F1-F2
      PART(2,IP)=IZ+F1-F2
      PTID(IP)=13
      GO TO 820
  790 PART(1,IP)=IX+F1-F2
      PART(2,IP)=IZ+F1+F2
      PTID(IP)=14
      GO TO 820
  800 PART(1,IP)=IX+F1+F2
      PART(2,IP)=IZ+F1-F2
      PTID(IP)=15
      GO TO 820
  810 PART(1,IP)=IX+F1+F2
      PART(2,IP)=IZ+F1+F2
      PTID(IP)=16
      GO TO 820
C
  820 PART(2,IP)=-PART(2,IP)
      PART(3,IP)=TDS(IX,IZ)
      PART(4,IP)=CONC(IX,IZ)
      PTWT(IP)=WTFCTR(IX,IZ)
C     ***************************************************************
C           ---CHECK FOR DISCHARGE BOUNDARY AT NEW LOCATION---
  830 IFLAG=1.0
  840 IF (PTQ(INX,INZ).GT.0.0) GO TO 850
      GO TO 880
C     ***************************************************************
C           ---PUT PT. IN LIMBO IF PT. DENSITY NOT INCREASED---
  850 PTWT(IN)=PTWT(IN)*WTFCTR(INX,INZ)
      IF (PTWT(IN).GT.0.001) GO TO 880
```

```
      PART(1,IN)=0.0
      PART(2,IN)=0.0
      PART(3,IN)=0.0
      PART(4,IN)=0.0
      DO 860 ID=1,500
      IF (LIMBO(ID).GT.0) GO TO 860
      LIMBO(ID)=IN
      GO TO 880
  860 CONTINUE
C
  870 IF (IFLAG.LT.0) PART(2,IN)=-TEMPZ
  880 CONTINUE
C     ---END OF LOOP---
C     ****************************************************************
      GO TO 920
C        ---RESTART MOVE IF PT. LIMIT EXCEEDED---
  890 WRITE (6,990) IMOV,IN
      TEST=100.0
      WRITE (6,1000)
      DO 900 IZ=1,NZ
  900 WRITE (6,1010) (NPCELL(IX,IZ),IX=1,NX)
      CALL GENPT
      DO 910 IX=1,NX
      DO 910 IZ=1,NZ
      SUMC(IX,IZ)=0.0
      SUMTDS(IX,IZ)=0.0
      SUMWT(IX,IZ)=0.0
  910 NPCELL(IX,IZ)=0
      TEST=0.0
      GO TO 20
C     ****************************************************************
  920 SUMTCH=SUMTCH+TIMV
C        ---ADJUST NUMBER OF PARTICLES---
      NP=NPTM
      WRITE (6,960) NP,IMOV
C     ****************************************************************
      CALL CNCON
C     ****************************************************************
C        ---STORE OBS. WELL DATA FOR STEADY FLOW PROBLEMS---
      IF (S.GT.0.0) GO TO 940
      IF (NUMOBS.LE.0) GO TO 940
      J=MOD(IMOV,50)
      IF (J.EQ.0) J=50
      TMOBS(J)=SUMTCH
      DO 930 I=1,NUMOBS
      TMWL(I,J)=PK(IXOBS(I),IZOBS(I))
      TMCN(I,J)=CONC(IXOBS(I),IZOBS(I))
  930 TMTDS(I,J)=TDS(IXOBS(I),IZOBS(I))
C        ---PRINT CHEMICAL OUTPUT---
      IF (MOD(IMOV,50).EQ.0) IPRNT=1
  940 IF (MOD(IMOV,NPNTMV).EQ.0) IPRNT=-1
      IF (IPRNT.NE.0) CALL CHMOT
      IF (TIMV.LT.0.1) GO TO 950
      IF ((TIM(N)-SUMTCH).LT.TIMV) TIMV=TIM(N)-SUMTCH
      GO TO 10
C     ****************************************************************
  950 RETURN
```

```
C     *****************************************************************
C
C
C
  960 FORMAT (1H0,2X,2HNP,7X,2H= ,8X,I4,10X,11HIMOV      = ,8X,I4)
  970 FORMAT (1H0,10X,61HNO. OF PARTICLE MOVES REQUIRED TO COMPLETE THIS
     1 TIME STEP  = ,I4//)
  980 FORMAT (1H0,5X,53H*** WARNING ***     QUADRANT NOT LOCATED FOR PT.
     1 NO. ,I5,11H , IN CELL ,2I4)
  990 FORMAT (1H0,5X,17H ***   NOTE   ***,10X,23HNPTM.EQ.NPMAX --- IMOV=
     1,I4,5X,8HPT. NO.=,I4,5X,10HCALL GENPT/)
 1000 FORMAT (1H0,2X,6HNPCELL/)
 1010 FORMAT (1H ,4X,24I3)
      END
      SUBROUTINE CNCON
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /PRMJ/ NTIM,NPMP,NPNT,NITP,N,NX,NZ,NP,NREC,INT,NNX,NNZ,NUMO
     1BS,NMOV,IMOV,NPMAX,ITMAX,NZCRIT,IPRNT,NPTPND,NPNTMV,NPNTVL,NPNTD,N
     2PNCHV,NPDELC,ICHK,NCONST
      COMMON /PRMC/ NODEID(24,20),NPCELL(24,20),NPOLD(24,20),LIMBO(500),
     1IXOBS(5),IZOBS(5)
      COMMON /PRESS/ PERM(24,20),PMRX(24,20,4),PI(24,20),PR(24,20),PC(24
     1,20),PK(24,20),REC(24,20),DENS(24,20),GTERM(24,20),VISC(24,20),VPR
     2M(24,20),TMWL(5,50),TMOBS(50),TIM(100),AOPT(20),TITLE(10),ANFCTR,X
     3DEL,ZDEL,WIDTH,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,VOL
      COMMON /DENVIS/ DEN1,DEN2,VIS1,VIS2
      COMMON /XINV/ DXINV,DZINV,ARINV,PORINV
      COMMON /CHMA/ PART(4,6400),CONC(24,20),TDS(24,20),VX(24,20),VZ(24,
     120),CONINT(24,20),TDSINT(24,20),CNREC(24,20),TDSREC(24,20),TMCN(5,
     250),TMTDS(5,50),POROS,SUMTCH,BETA,TIMV,STORM,STORT,STORMI,STORTI,C
     3MSIN,TDSIN,CMSOUT,TDSOUT,FLMIN,FLTIN,FLMOT,FLTOT,SUMIO,TDSIO,CELDI
     4S,DLTRAT,CSTORM,CSTORT,DMOLEC
      COMMON /DIFUS/ DISP(24,20,4)
      COMMON /CHMC/ SUMC(24,20),VXBDY(24,20),VZBDY(24,20),SUMTDS(24,20),
     1WTFCTR(24,20),SUMWT(24,20),PTQ(24,20),PTWT(6400),ELEV(24,20)
      COMMON /CNCHNG/ CNCHCK(24,20),CTOL
      DIMENSION CNCNC(24,20), CNOLD(24,20), TDSOLD(24,20), CNTDS(24,20)
C     ****************************************************************
      ITEST=0
      DO 10 IX=2,NNX
      DO 10 IZ=2,NNZ
      CNOLD(IX,IZ)=CONC(IX,IZ)
      TDSOLD(IX,IZ)=TDS(IX,IZ)
      CNCNC(IX,IZ)=0.0
   10 CNTDS(IX,IZ)=0.0
      APC=0.0
      NZERO=0
      TVA=AREA*TIMV
      ARPOR=AREA*POROS
C     ****************************************************************
C     ---CONC. CHANGE FOR 0.5*TIMV DUE TO:
C           RECHARGE, LEAKAGE, DIVERGENCE OF VELOCITY...
      CONST=0.5*TIMV
   20 DO 70 IX=2,NNX
      DO 70 IZ=2,NNZ
      IF (PERM(IX,IZ).EQ.0.0) GO TO 70
      EQFCT1=CONST
```

```
      EQFCT2=EQFCT1/POROS
      C1=CONC(IX,IZ)
      C2=TDS(IX,IZ)
      IF (ABS(C1).LT.1.0E-20) C1=0.0
      IF (ABS(C2).LT.1.0E-20) C2=0.0
      CLKCN1=0.0
      CLKCN2=0.0
      SLEAK=(PK(IX,IZ)-PI(IX,IZ)+DENS(IX,IZ)*ELEV(IX,IZ))
      SLEAK=SLEAK*VPRM(IX,IZ)/DENS(IX,IZ)
      DENSE=DENS(IX,IZ)
      IF (PI(IX,IZ).GT.PK(IX,IZ)) DENSE=DEN1*TDSREC(IX,IZ)+DEN2
      IF (SLEAK.GT.0.0) GO TO 30
      CLKCN1=CNREC(IX,IZ)
      CLKCN2=TDSREC(IX,IZ)
      GO TO 40
   30 CLKCN1=C1
      CLKCN2=C2
   40 CNREC1=C1
      CNREC2=C2
      DENSE=DENS(IX,IZ)
      IF (REC(IX,IZ).LT.0.0) DENSE=DEN1*TDSREC(IX,IZ)+DEN2
      RATE=REC(IX,IZ)/DENSE
      IF (REC(IX,IZ).GT.0.0) GO TO 50
      CNREC1=CNREC(IX,IZ)
      CNREC2=TDSREC(IX,IZ)
   50 DIV1=SLEAK+RATE
      DIV2=SLEAK+RATE
      DELC1=EQFCT2*(C1*DIV1-SLEAK*CLKCN1-RATE*CNREC1)
      DELC2=EQFCT2*(C2*DIV2-SLEAK*CLKCN2-RATE*CNREC2)
      CNCNC(IX,IZ)=CNCNC(IX,IZ)+DELC1
      CNTDS(IX,IZ)=CNTDS(IX,IZ)+DELC2
C     ---CONC. CHANGE DUE TO DISPERSION FOR 0.5*TIMV---
C        ---DISPERSION WITH TENSOR COEFFICIENTS---
      IF (BETA.EQ.0.0.AND.DMOLEC.EQ.0.0) GO TO 70
      IF (NCONST.LT.2) GO TO 60
      X1=DISP(IX,IZ,1)*(CONC(IX+1,IZ)-C1)
      X2=DISP(IX-1,IZ,1)*(CONC(IX-1,IZ)-C1)
      Z1=DISP(IX,IZ,2)*(CONC(IX,IZ+1)-C1)
      Z2=DISP(IX,IZ-1,2)*(CONC(IX,IZ-1)-C1)
      XX1=DISP(IX,IZ,3)*(CONC(IX,IZ+1)+CONC(IX+1,IZ+1)-CONC(IX,IZ-1)-CON
     1C(IX+1,IZ-1))
      XX2=DISP(IX-1,IZ,3)*(CONC(IX,IZ+1)+CONC(IX-1,IZ+1)-CONC(IX,IZ-1)-C
     1ONC(IX-1,IZ-1))
      ZZ1=DISP(IX,IZ,4)*(CONC(IX+1,IZ)+CONC(IX+1,IZ+1)-CONC(IX-1,IZ)-CON
     1C(IX-1,IZ+1))
      ZZ2=DISP(IX,IZ-1,4)*(CONC(IX+1,IZ)+CONC(IX+1,IZ-1)-CONC(IX-1,IZ)-C
     1ONC(IX-1,IZ-1))
      CNCNC(IX,IZ)=CNCNC(IX,IZ)+EQFCT1*(X1+X2+Z1+Z2+XX1-XX2+ZZ1-ZZ2)
   60 X1=DISP(IX,IZ,1)*(TDS(IX+1,IZ)-C2)
      X2=DISP(IX-1,IZ,1)*(TDS(IX-1,IZ)-C2)
      Z1=DISP(IX,IZ,2)*(TDS(IX,IZ+1)-C2)
      Z2=DISP(IX,IZ-1,2)*(TDS(IX,IZ-1)-C2)
      XX1=DISP(IX,IZ,3)*(TDS(IX,IZ+1)+TDS(IX+1,IZ+1)-TDS(IX,IZ-1)-TDS(IX
     1+1,IZ-1))
      XX2=DISP(IX-1,IZ,3)*(TDS(IX,IZ+1)+TDS(IX-1,IZ+1)-TDS(IX,IZ-1)-TDS(
     1IX-1,IZ-1))
      ZZ1=DISP(IX,IZ,4)*(TDS(IX+1,IZ)+TDS(IX+1,IZ+1)-TDS(IX-1,IZ)-TDS(IX
```

```
     1-1,IZ+1))
      ZZ2=DISP(IX,IZ-1,4)*(TDS(IX+1,IZ)+TDS(IX+1,IZ-1)-TDS(IX-1,IZ)-TDS(
     1IX-1,IZ-1))
      CNTDS(IX,IZ)=CNTDS(IX,IZ)+EQFCT1*(X1+X2+Z1+Z2+XX1-XX2+ZZ1-ZZ2)
   70 CONTINUE
C     ******************************************************************
      ITEST=ITEST+1
      IF (ITEST.EQ.1) GO TO 80
      GO TO 120
C     ******************************************************************
C     ---CONC. CHANGE AT NODES DUE TO CONVECTION---
   80 DO 100 IX=2,NNX
      DO 100 IZ=2,NNZ
      IF (PERM(IX,IZ).EQ.0.0) GO TO 100
      APC=NPCELL(IX,IZ)
      IF (APC.GT.0.0) GO TO 90
      IF (PTQ(IX,IZ).NE.0.0) GO TO 100
      NZERO=NZERO+1
      GO TO 100
   90 CONC(IX,IZ)=SUMC(IX,IZ)/SUMWT(IX,IZ)
      TDS(IX,IZ)=SUMTDS(IX,IZ)/SUMWT(IX,IZ)
  100 CONTINUE
C        ---CHECK NUMBER OF CELLS VOID OF PTS.---
      IF (NZERO.GT.0) WRITE (6,330) NZERO,IMOV
      IF (NZERO.LE.NZCRIT) GO TO 20
      TEST=99.0
      WRITE (6,340)
      WRITE (6,360)
      DO 110 IZ=1,NZ
  110 WRITE (6,370) (NPCELL(IX,IZ),IX=1,NX)
      GO TO 20
C     ******************************************************************
C     ---CHANGE CONCENTRATIONS AT NODES---
  120 DO 150 IX=2,NNX
      DO 150 IZ=2,NNZ
      IF (PERM(IX,IZ).EQ.0.0) GO TO 140
      CNCPCT=0.0
      TDSPCT=0.0
      IF (CONC(IX,IZ).GT.0.0) CNCPCT=CNCNC(IX,IZ)/CONC(IX,IZ)
      IF (TDS(IX,IZ).GT.0.0)  TDSPCT=CNTDS(IX,IZ)/TDS(IX,IZ)
      CONC(IX,IZ)=CONC(IX,IZ)+CNCNC(IX,IZ)
      TDS(IX,IZ)=TDS(IX,IZ)+CNTDS(IX,IZ)
      SUMC(IX,IZ)=0.0
      SUMTDS(IX,IZ)=0.0
      IF (CNCPCT.LT.0.0) SUMC(IX,IZ)=CNCPCT
      IF (TDSPCT.LT.0.0) SUMTDS(IX,IZ)=TDSPCT
      GO TO 150
  140 IF (CONC(IX,IZ).GT.0.0) WRITE (6,350) IX,IZ,CONC(IX,IZ)
      CONC(IX,IZ)=0.0
  150 CONTINUE
C     ******************************************************************
C     ---CHANGE CONCENTRATION OF PARTICLES---
      DO 220 IN=1,NP
      IF (PART(1,IN).EQ.0.0) GO TO 220
      INX=ABS(PART(1,IN))+0.5
      INZ=ABS(PART(2,IN))+0.5
C     ---UPDATE CONC. OF PTS. IN SINK/SOURCE CELLS---
```

```
      IF (PTQ(INX,INZ).EQ.0.0) GO TO 160
      PART(4,IN)=CONC(INX,INZ)
      PART(3,IN)=TDS(INX,INZ)
      GO TO 220
  160 IF (CNCNC(INX,INZ).LT.0.0) GO TO 180
  170 PART(4,IN)=PART(4,IN)+CNCNC(INX,INZ)
      GO TO 190
  180 IF (CONC(INX,INZ).LE.0.0) GO TO 170
      IF (SUMC(INX,INZ).LT.-1.0) GO TO 170
      PART(4,IN)=PART(4,IN)+PART(4,IN)*SUMC(INX,INZ)
  190 IF (CNTDS(INX,INZ).LT.0.0) GO TO 210
  200 PART(3,IN)=PART(3,IN)+CNTDS(INX,INZ)
      GO TO 220
  210 IF (TDS(INX,INZ).LE.0.0) GO TO 200
      IF (SUMTDS(INX,INZ).LT.-1.0) GO TO 200
      PART(3,IN)=PART(3,IN)+PART(3,IN)*SUMTDS(INX,INZ)
  220 CONTINUE
      WRITE (6,320) TIM(N),TIMV,SUMTCH
C     ****************************************************************
C     ---COMPUTE MASS BALANCE FOR SOLUTE---
      CSTORM=0.0
      STORM=0.0
      CSTORT=0.0
      STORT=0.0
      DO 290 IX=2,NNX
      DO 290 IZ=2,NNZ
      IF (PERM(IX,IZ).EQ.0.0) GO TO 290
      SUMC(IX,IZ)=0.0
      SUMWT(IX,IZ)=0.0
      SUMTDS(IX,IZ)=0.0
      WTFCT=0.0
C        ---COMPUTE MASS OF SOLUTE IN STORAGE---
      STORM=STORM+CONC(IX,IZ)*WIDTH*ARPOR
      STORT=STORT+TDS(IX,IZ)*WIDTH*ARPOR
C        ---ACCOUNT FOR MASS PUMPED IN, OUT, RECHARGED, & DISCHARGED---
      IF (REC(IX,IZ)) 240,250,230
  230 RATE=REC(IX,IZ)*VOL/DENS(IX,IZ)
      TDSOUT=TDSOUT+(RATE*TIMV*0.5*(TDSOLD(IX,IZ)+TDS(IX,IZ)))
      CMSOUT=CMSOUT+(RATE*TIMV*0.5*(CNOLD(IX,IZ)+CONC(IX,IZ)))
      GO TO 250
  240 DENSE=DEN1*TDSREC(IX,IZ)+DEN2
      RATE=REC(IX,IZ)*VOL/DENSE
      TDSIN=TDSIN+RATE*TDSREC(IX,IZ)*TIMV
      CMSIN=CMSIN+RATE*CNREC(IX,IZ)*TIMV
C     ****************************************************************
C        ---ACCOUNT FOR BOUNDARY FLOW---
  250 IF (VPRM(IX,IZ).EQ.0.0) GO TO 280
      FLW=VPRM(IX,IZ)*(PI(IX,IZ)-PK(IX,IZ)+DENS(IX,IZ)*ELEV(IX,IZ))
      FLW=FLW/DENS(IX,IZ)
      IF (FLW.GT.0.0) GO TO 260
      IF (FLW.LT.0.0) GO TO 270
      GO TO 280
C           ---MASS IN BOUNDARY DURING TIME STEP---
  260 FLMIN=FLMIN+FLW*CNREC(IX,IZ)*TVA*WIDTH
      FLTIN=FLTIN+FLW*TDSREC(IX,IZ)*TVA*WIDTH
      GO TO 280
C           ---MASS OUT DURING TIME STEP---
```

```
  270 FLMOT=FLMOT+FLW*TVA*WIDTH*0.5*(CNOLD(IX,IZ)+CONC(IX,IZ))
      FLTOT=FLTOT+FLW*TVA*WIDTH*0.5*(TDSOLD(IX,IZ)+TDS(IX,IZ))
  280 NPOLD(IX,IZ)=NPCELL(IX,IZ)
      NPCELL(IX,IZ)=0
  290 CONTINUE
C     ****************************************************************
C        ---COMPUTE CHANGE IN MASS OF SOLUTE STORED---
      CSTORM=STORM-STORMI
      CSTORT=STORT-STORTI
      SUMIO=FLMIN+FLMOT-CMSIN-CMSOUT
      TDSIO=FLTIN+FLTOT-TDSOUT-TDSIN
C     ****************************************************************
C     ---REGENERATE PARTICLES IF 'NZCRIT' EXCEEDED---
      IF (TEST.GT.98.0) CALL GENPT
      TEST=0.0
C     ****************************************************************
C       --- CHECK FOR A SIGNIFICANT CONCENTRATION CHANGE ---
      TEST3=0.0
      DO 300 IX=1,NX
      DO 300 IZ=1,NZ
      IF (PERM(IX,IZ).EQ.0.0) GO TO 300
      IF (ABS(TDS(IX,IZ)-CNCHCK(IX,IZ)).GT.CTOL) TEST3=1.0
  300 CONTINUE
      IF (TEST3.EQ.0.0) GO TO 310
C       --- RECALCULATE PRESSURES AND VELOCITIES WITH NEW VALUES ---
      WRITE (6,390)
      WRITE (6,380)
      CALL ITERAT
      CALL VELO
      WRITE (6,390)
C     ****************************************************************
  310 RETURN
C     ****************************************************************
C
C
C
  320 FORMAT (3H    ,11HTIM(N)    = ,1G12.5,10X,11HTIMV      = ,1G12.5,10X,
     19HSUMTCH = ,G12.5)
  330 FORMAT (1H0,5X,40HNUMBER OF CELLS WITH ZERO PARTICLES  =  ,I4,5X,9
     1HIMOV   =  ,I4/)
  340 FORMAT (1H0,5X,44H***   NZCRIT EXCEEDED  ---  CALL GENPT   ***/)
  350 FORMAT (1H ,5X,37H***CONC.GT.0.AND.PERM.EQ.0 AT NODE = ,2I4,4X,7HC
     1ONC = ,G10.4,4H ***)
  360 FORMAT (1H0,2X,6HNPCELL/)
  370 FORMAT (1H ,4X,20I3)
  380 FORMAT (1H ,49HRECALCULATE PRESSURES DUE TO CONCENTRATION CHANGE)
  390 FORMAT (1H )
      END
      SUBROUTINE OUTPT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /PRMJ/ NTIM,NPMP,NPNT,NITP,N,NX,NZ,NP,NREC,INT,NNX,NNZ,NUMO
     1BS,NMOV,IMOV,NPMAX,ITMAX,NZCRIT,IPRNT,NPTPND,NPNTMV,NPNTVL,NPNTD,N
     2PNCHV,NPDELC,ICHK,NCONST
      COMMON /PRMC/ NODEID(24,20),NPCELL(24,20),NPOLD(24,20),LIMBO(500),
     1IXOBS(5),IZOBS(5)
      COMMON /PRESS/ PERM(24,20),PMRX(24,20,4),PI(24,20),PR(24,20),PC(24
     1,20),PK(24,20),REC(24,20),DENS(24,20),GTERM(24,20),VISC(24,20),VPR
```

```
     2M(24,20),TMWL(5,50),TMOBS(50),TIM(100),AOPT(20),TITLE(10),ANFCTR,X
     3DEL,ZDEL,WIDTH,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,VOL
      COMMON /CHMA/ PART(4,6400),CONC(24,20),TDS(24,20),VX(24,20),VZ(24,
     120),CONINT(24,20),TDSINT(24,20),CNREC(24,20),TDSREC(24,20),TMCN(5,
     250),TMTDS(5,50),POROS,SUMTCH,BETA,TIMV,STORM,STORT,STORMI,STORTI,C
     3MSIN,TDSIN,CMSOUT,TDSOUT,FLMIN,FLTIN,FLMOT,FLTOT,SUMIO,TDSIO,CELDI
     4S,DLTRAT,CSTORM,CSTORT,DMOLEC
      COMMON /CHMC/ SUMC(24,20),VXBDY(24,20),VZBDY(24,20),SUMTDS(24,20),
     1WTFCTR(24,20),SUMWT(24,20),PTQ(24,20),PTWT(6400),ELEV(24,20)
      COMMON /DENVIS/ DEN1,DEN2,VIS1,VIS2
      COMMON /BALM/ TOTLQ,TOTLQI,TPIN,TPOUT
      DIMENSION IP(24)
C     ***************************************************************
      TIMD=SUMT/86400.
      TIMY=SUMT/(86400.0*365.25)
C     ---PRINT PRESSURE VALUES---
      WRITE (6,90)
      WRITE (6,100) N
      WRITE (6,110) SUMT
      WRITE (6,120) TIMD
      WRITE (6,130) TIMY
      WRITE (6,140)
      DO 10 IZ=1,NZ
   10 WRITE (6,150) (PK(IX,IZ),IX=1,NX)
      IF (N.EQ.0) GO TO 80
C     ***************************************************************
C     ---PRINT PRESSURE MAP---
      WRITE (6,140)
      WRITE (6,300)
      WRITE (6,140)
      DO 30 IZ=1,NZ
      DO 20 IX=1,NX
   20 IP(IX)=PK(IX,IZ)+0.5
   30 WRITE (6,160) (IP(ID),ID=1,NX)
C     ***************************************************************
C     ---COMPUTE WATER BALANCE AND DRAWDOWN---
      QSTR=0.0
      PUMP=0.0
      PQIN=0.0
      PQOUT=0.0
      TPUM=0.0
      QIN=0.0
      QOUT=0.0
      QNET=0.0
      DELQ=0.0
      PCTERR=0.0
C
      DO 60 IZ=1,NZ
      DO 50 IX=1,NX
      IP(IX)=0.0
      IF (PERM(IX,IZ).EQ.0.0) GO TO 50
      IF (REC(IX,IZ).GT.0.0) PQOUT=PQOUT+REC(IX,IZ)*VOL
      IF (REC(IX,IZ).LT.0.0) PQIN=PQIN+REC(IX,IZ)*VOL
      IF (VPRM(IX,IZ).EQ.0.0) GO TO 40
      DELQC=VPRM(IX,IZ)*VOL
      DENSE=DENS(IX,IZ)
      IF (PI(IX,IZ).GT.PK(IX,IZ)) DENSE=DEN1*TDSREC(IX,IZ)+DEN2
```

```
      DELQ=DELQC*(PI(IX,IZ)-PK(IX,IZ)+DENS(IX,IZ)*ELEV(IX,IZ))
      IF (DELQ.GT.0.0) QIN=QIN+DELQ
      IF (DELQ.LT.0.0) QOUT=QOUT+DELQ
      QNET=QNET+DELQ
   40 DELP=PI(IX,IZ)-PK(IX,IZ)
      QSTR=QSTR+DELP*VOL*S
   50 CONTINUE
   60 CONTINUE
      TPUM=PQOUT+PQIN
      PUMP=TPOUT+TPIN
      TOTLQN=TOTLQI+TOTLQ
      SRCS=QSTR-TPIN+TOTLQI
      SINKS=TPOUT-TOTLQ
      ERRMB=SRCS-SINKS
      DENOM=(SRCS+SINKS)*0.5
      IF (DENOM.EQ.0.0) GO TO 70
      PCTERR=ERRMB*100.0/DENOM
C        ---PRINT MASS BALANCE DATA FOR FLOW MODEL---
   70 WRITE (6,250)
      WRITE (6,220) TPIN
      WRITE (6,230) TPOUT
      WRITE (6,260) PUMP
      WRITE (6,240) QSTR
      WRITE (6,180) TOTLQI
      WRITE (6,190) TOTLQ
      WRITE (6,270) TOTLQN
      WRITE (6,280) ERRMB
      WRITE (6,290) PCTERR
      WRITE (6,170)
      WRITE (6,180) QIN
      WRITE (6,190) QOUT
      WRITE (6,200) QNET
      WRITE (6,220) PQIN
      WRITE (6,230) PQOUT
      WRITE (6,210) TPUM
C     ***************************************************************
   80 RETURN
C     ***************************************************************
C
C
C
   90 FORMAT (1H1,27HPRESSURE DISTRIBUTION - ROW)
  100 FORMAT (1X,23HNUMBER OF TIME STEPS = ,1I5)
  110 FORMAT (8X,16HTIME(SECONDS) = ,1G12.5)
  120 FORMAT (8X,16HTIME(DAYS)    = ,1PE12.5)
  130 FORMAT (8X,16HTIME(YEARS)   = ,1PE12.5)
  140 FORMAT (1H )
  150 FORMAT (1H0,10F12.4)
  160 FORMAT (1H0,20I6)
  170 FORMAT (1H0,2X,33HRATE MASS BALANCE -- (IN C.F.S.) //)
  180 FORMAT (4X,29HLEAKAGE INTO AQUIFER        = ,1PE12.5)
  190 FORMAT (4X,29HLEAKAGE OUT OF AQUIFER      = ,1PE12.5)
  200 FORMAT (4X,29HNET LEAKAGE    (QNET)       = ,1PE12.5)
  210 FORMAT (4X,29HNET WITHDRAWAL   (TPUM)     = ,1PE12.5)
  220 FORMAT (4X,29HRECHARGE                    = ,1PE12.5)
  230 FORMAT (4X,29HDISCHARGE                   = ,1PE12.5)
  240 FORMAT (4X,29HWATER RELEASE FROM STORAGE = ,1PE12.5)
```

```
  250 FORMAT (1H0,2X,38HCUMULATIVE MASS BALANCE -- (IN FT**3) //)
  260 FORMAT (4X,29HCUMULATIVE  NET  RECHARGE  = ,1PE12.5)
  270 FORMAT (4X,29HCUMULATIVE  NET  LEAKAGE   = ,1PE12.5)
  280 FORMAT (1H0,7X,25HMASS BALANCE RESIDUAL  = ,G12.5)
  290 FORMAT (1H ,7X,25HERROR  (AS PERCENT)    = ,G12.5/)
  300 FORMAT (1H ,12HPRESSURE MAP)
      END
      SUBROUTINE CHMOT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /PRMJ/ NTIM,NPMP,NPNT,NITP,N,NX,NZ,NP,NREC,INT,NNX,NNZ,NUMO
     1BS,NMOV,IMOV,NPMAX,ITMAX,NZCRIT,IPRNT,NPTPND,NPNTMV,NPNTVL,NPNTD,N
     2PNCHV,NPDELC,ICHK,NCONST
      COMMON /PRMC/ NODEID(24,20),NPCELL(24,20),NPOLD(24,20),LIMBO(500),
     1IXOBS(5),IZOBS(5)
      COMMON /PRESS/ PERM(24,20),PMRX(24,20,4),PI(24,20),PR(24,20),PC(24
     1,20),PK(24,20),REC(24,20),DENS(24,20),GTERM(24,20),VISC(24,20),VPR
     2M(24,20),TMWL(5,50),TMOBS(50),TIM(100),AOPT(20),TITLE(10),ANFCTR,X
     3DEL,ZDEL,WIDTH,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR,VOL
      COMMON /CHMA/ PART(4,6400),CONC(24,20),TDS(24,20),VX(24,20),VZ(24,
     120),CONINT(24,20),TDSINT(24,20),CNREC(24,20),TDSREC(24,20),TMCN(5,
     250),TMTDS(5,50),POROS,SUMTCH,BETA,TIMV,STORM,STORT,STORMI,STORTI,C
     3MSIN,TDSIN,CMSOUT,TDSOUT,FLMIN,FLTIN,FLMOT,FLTOT,SUMIO,TDSIO,CELDI
     4S,DLTRAT,CSTORM,CSTORT,DMOLEC
      COMMON /CHMC/ SUMC(24,20),VXBDY(24,20),VZBDY(24,20),SUMTDS(24,20),
     1WTFCTR(24,20),SUMWT(24,20),PTQ(24,20),PTWT(6400),ELEV(24,20)
      DIMENSION IC(24)
C     ****************************************************************
      TMFY=86400.0*365.25
      TMYR=SUMT/TMFY
      TCHD=SUMTCH/86400.0
      TCHYR=SUMTCH/TMFY
      ERR1=0.0
      ERR3=0.0
      IF (IPRNT.GT.0) GO TO 180
C     ****************************************************************
C     ---PRINT CONCENTRATIONS---
      WRITE (6,240)
      WRITE (6,250) N
      IF (N.GT.0) WRITE (6,260) TIM(N)
      WRITE (6,270) SUMT
      WRITE (6,550) SUMTCH
      WRITE (6,280) TCHD
      WRITE (6,290) TMYR
      WRITE (6,560) TCHYR
      WRITE (6,480) IMOV
      WRITE (6,300)
      IF (NCONST.LT.2) GO TO 30
      WRITE (6,310)
      DO 20 IZ=1,NZ
      DO 10 IX=1,NX
   10 IC(IX)=CONC(IX,IZ)+0.5
   20 WRITE (6,340) (IC(IX),IX=1,NX)
   30 WRITE (6,320)
      DO 50 IZ=1,NZ
      DO 40 IX=1,NX
   40 IC(IX)=TDS(IX,IZ)+0.5
   50 WRITE (6,340) (IC(IX),IX=1,NX)
```

```
C       ******************************************************************
        IF (N.EQ.0) GO TO 230
        IF (NPDELC.EQ.0) GO TO 110
C
C       ---PRINT CHANGES IN CONCENTRATION---
        WRITE (6,330)
        WRITE (6,250) N
        WRITE (6,260) TIM(N)
        WRITE (6,270) SUMT
        WRITE (6,550) SUMTCH
        WRITE (6,280) TCHD
        WRITE (6,290) TMYR
        WRITE (6,560) TCHYR
        WRITE (6,480) IMOV
        WRITE (6,300)
        IF (NCONST.LT.2) GO TO 80
        WRITE (6,310)
        DO 70 IZ=1,NZ
        DO 60 IX=1,NX
        CNG=CONC(IX,IZ)-CONINT(IX,IZ)
   60   IC(IX)=CNG
   70   WRITE (6,340) (IC(IX),IX=1,NX)
   80   WRITE (6,320)
        DO 100 IZ=1,NZ
        DO 90 IX=1,NX
        CNG=TDS(IX,IZ)-TDSINT(IX,IZ)
   90   IC(IX)=CNG
  100   WRITE (6,340) (IC(IX),IX=1,NX)
C       ******************************************************************
C       ---PRINT MASS BALANCE DATA FOR SOLUTE---
  110   IF (NCONST.LT.2) GO TO 140
        RESID=SUMIO-CSTORM
        SUMIN=FLMIN-CMSIN
        IF (SUMIN.EQ.0.0) GO TO 120
        ERR1=RESID*100.0/SUMIN
  120   IF (STORMI.EQ.0.0) GO TO 130
        ERR3=-100.0*RESID/(STORMI-SUMIO)
  130   WRITE (6,300)
        WRITE (6,350)
        WRITE (6,310)
        WRITE (6,300)
        WRITE (6,360) FLMIN
        WRITE (6,370) FLMOT
        RECIN=-CMSIN
        RECOUT=-CMSOUT
        WRITE (6,390) RECIN
        WRITE (6,380) RECOUT
        WRITE (6,400) SUMIO
        WRITE (6,410) STORMI
        WRITE (6,420) STORM
        WRITE (6,430) CSTORM
        WRITE (6,440)
        WRITE (6,450) RESID
        WRITE (6,460) ERR1
        IF (STORMI.EQ.0.0) GO TO 140
        WRITE (6,470)
        WRITE (6,460) ERR3
```

```
C     ****************************************************************
C     --- PRINT MASS BALANCE FOR TDS ---
  140 RESID=TDSIO-CSTORT
      SUMIN=FLTIN-TDSIN
      IF (SUMIN.EQ.0.0) GO TO 150
      ERR1=RESID*100.0/SUMIN
  150 IF (STORTI.EQ.0.0) GO TO 160
      ERR3=100.0*RESID/(STORTI-TDSIO)
  160 WRITE (6,300)
      WRITE (6,350)
      WRITE (6,320)
      WRITE (6,300)
      WRITE (6,360) FLTIN
      WRITE (6,370) FLTOT
      WRITE (6,390) -TDSIN
      WRITE (6,380) -TDSOUT
      WRITE (6,400) TDSIO
      WRITE (6,410) STORTI
      WRITE (6,420) STORT
      WRITE (6,430) CSTORT
      WRITE (6,440)
      WRITE (6,450) RESID
      WRITE (6,460) ERR1
      IF (STORTI.EQ.0.0) GO TO 170
      WRITE (6,470)
      WRITE (6,460) ERR3
C     ****************************************************************
C     ---PRINT HYDROGRAPHS AFTER 50 STEPS OR END OF SIMULATION---
  170 IF (MOD(IMOV,50).EQ.0.AND.S.EQ.0.0) GO TO 180
      IF (MOD(N,50).EQ.0.AND.S.GT.0.0) GO TO 180
      IF (S.EQ.0.0.AND.N.LT.NTIM.AND.INT.GT.0) GO TO 180
      GO TO 230
  180 WRITE (6,490) TITLE
      IF (NUMOBS.LE.0) GO TO 230
      WRITE (6,500) INT
      IF (S.GT.0.0) WRITE (6,510)
      IF (S.EQ.0.0) WRITE (6,520)
C        ---TABULATE HYDROGRAPH DATA---
      MOZ=0
      IF (S.GT.0.0) GO TO 190
      NTO=NMOV
      IF (NMOV.GT.50) NTO=MOD(IMOV,50)
      GO TO 200
  190 NTO=NTIM
      IF (NTIM.GT.50) NTO=MOD(N,50)
  200 IF (NTO.EQ.0) NTO=50
      DO 220 J=1,NUMOBS
      TMYR=0.0
      WRITE (6,530) J,IXOBS(J),IZOBS(J)
      WRITE (6,540) MOZ,PI(IXOBS(J),IZOBS(J)),CONINT(IXOBS(J),IZOBS(J)),
     1TDSINT(IXOBS(J),IZOBS(J)),TMYR
      DO 210 M=1,NTO
      TMYR=TMOBS(M)/TMFY
  210 WRITE (6,540) M,TMWL(J,M),TMCN(J,M),TMTDS(J,M),TMYR
  220 CONTINUE
C     ****************************************************************
  230 IPRNT=0
```

```
      RETURN
C     ****************************************************************
C
C
C
  240 FORMAT (1H1,13HCONCENTRATION/)
  250 FORMAT (1X,23HNUMBER OF TIME STEPS = ,1I5)
  260 FORMAT (8X,16HDELTA T        = ,1G12.5)
  270 FORMAT (8X,16HTIME(SECONDS) = ,1G12.5)
  280 FORMAT (3X,21HCHEM.TIME(DAYS)     = ,1E12.5)
  290 FORMAT (8X,16HTIME(YEARS)    = ,1E12.5)
  300 FORMAT (1H )
  310 FORMAT (1H ,15X,12HTRACE SOLUTE)
  320 FORMAT (1H0,15X,26HDENSITY-CONTROLLING SOLUTE)
  330 FORMAT (1H1,23HCHANGE IN CONCENTRATION/)
  340 FORMAT (1H0,20I5)
  350 FORMAT (1H ,21HCHEMICAL MASS BALANCE)
  360 FORMAT (8X,25HMASS IN BOUNDARIES        = ,1PE12.5)
  370 FORMAT (8X,25HMASS OUT BOUNDARIES       = ,1PE12.5)
  380 FORMAT (8X,25HMASS PUMPED OUT           = ,1PE12.5)
  390 FORMAT (8X,25HMASS PUMPED IN            = ,1PE12.5)
  400 FORMAT (8X,25HINFLOW MINUS OUTFLOW      = ,1PE12.5)
  410 FORMAT (8X,25HINITIAL MASS STORED       = ,1PE12.5)
  420 FORMAT (8X,25HPRESENT MASS STORED       = ,1PE12.5)
  430 FORMAT (8X,25HCHANGE MASS STORED        = ,1PE12.5)
  440 FORMAT (1H ,5X,53HCOMPARE RESIDUAL WITH NET FLUX AND MASS ACCUMULA
     1TION:)
  450 FORMAT (8X,25HMASS BALANCE RESIDUAL     = ,1PE12.5)
  460 FORMAT (8X,25HERROR  (AS PERCENT)       = ,1PE12.5)
  470 FORMAT (1H ,5X,55HCOMPARE INITIAL MASS STORED WITH CHANGE IN MASS
     1STORED:)
  480 FORMAT (1X,23H NO. MOVES COMPLETED = ,1I5)
  490 FORMAT (1H1,10A8//)
  500 FORMAT (1H0,5X,65HTIME VERSUS HEAD AND CONCENTRATION AT SELECTED O
     1BSERVATION POINTS//15X,19HPUMPING PERIOD NO. ,I4////)
  510 FORMAT (1H0,16X,19HTRANSIENT  SOLUTION////)
  520 FORMAT (1H0,15X,21HSTEADY-STATE SOLUTION////)
  530 FORMAT (1H0,20X,22HOBS.WELL NO.    X    Z,17X,1HN,6X,64HPRESSURE (
     1LB/FT**2)   CONC.(MG/L)  TDS (MG/L)     TIME (YEARS)  //24X,I3,9X,
     2I2,3X,I2//)
  540 FORMAT (1H ,58X,I2,12X,F7.1,8X,F7.1,8X,F7.1,8X,F7.3)
  550 FORMAT (1H ,2X,21HCHEM.TIME(SECONDS) = ,1PE12.5)
  560 FORMAT (1H ,2X,21HCHEM.TIME(YEARS)   = ,1PE12.5)
      END
```