Chapter C2

COMPUTER MODEL OF TWO-DIMENSIONAL SOLUTE TRANSPORT AND DISPERSION IN GROUND WATER

By L. F. Konikow and J. D. Bredehoeft

Book 7

AUTOMATED DATA PROCESSING AND COMPUTATIONS
DEPARTMENT OF THE INTERIOR
WILLIAM P. CLARK, Secretary

U.S. GEOLOGICAL SURVEY
Dallas L. Peck, Director

Requests, at cost, for the Card Deck listed in Attachment VII should be directed to:
Ralph N. Eicher, Chief, Office of Teleprocessing, M.S. 805, National Center,

First printing 1978
Second printing 1984

UNITED STATES GOVERNMENT PRINTING OFFICE, WASHINGTON : 1978

For sale by the Distribution Branch, U.S. Geological Survey
604 South Pickett Street, Alexandria, VA 22304
PREFACE

The series of manuals on techniques describes procedures for planning and executing specialized work in water-resources investigations. The material is grouped under major headings called books and further subdivided into sections and chapters; section C of Book 7 is on computer programs.

This chapter presents a digital computer model for calculating changes in the concentration of a 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 selection of values for certain user-specified options.
## CONTENTS

<table>
<thead>
<tr>
<th>Abstract</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Theoretical background</td>
<td>2</td>
</tr>
<tr>
<td>Flow equation</td>
<td>2</td>
</tr>
<tr>
<td>Transport equation</td>
<td>3</td>
</tr>
<tr>
<td>Dispersion coefficient</td>
<td>3</td>
</tr>
<tr>
<td>Review of assumptions</td>
<td>4</td>
</tr>
<tr>
<td>Numerical methods</td>
<td>4</td>
</tr>
<tr>
<td>Flow equation</td>
<td>4</td>
</tr>
<tr>
<td>Transport equation</td>
<td>5</td>
</tr>
<tr>
<td>Method of characteristics</td>
<td>5</td>
</tr>
<tr>
<td>Particle tracking</td>
<td>6</td>
</tr>
<tr>
<td>Finite-difference approximations</td>
<td>7</td>
</tr>
<tr>
<td>Stability criteria</td>
<td>11</td>
</tr>
<tr>
<td>Boundary and initial conditions</td>
<td>13</td>
</tr>
<tr>
<td>Mass balance</td>
<td>14</td>
</tr>
<tr>
<td>Special problems</td>
<td>15</td>
</tr>
<tr>
<td>Computer program</td>
<td>19</td>
</tr>
<tr>
<td>General program features</td>
<td>20</td>
</tr>
<tr>
<td>Program segments</td>
<td>21</td>
</tr>
<tr>
<td>MAIN</td>
<td>21</td>
</tr>
<tr>
<td>Subroutine PARLOD</td>
<td>22</td>
</tr>
<tr>
<td>Subroutine ITERAT</td>
<td>22</td>
</tr>
<tr>
<td>Subroutine GENPT</td>
<td>22</td>
</tr>
<tr>
<td>Subroutine VELO</td>
<td>23</td>
</tr>
<tr>
<td>Subroutine MOVE</td>
<td>23</td>
</tr>
<tr>
<td>Computer program—Continued</td>
<td>25</td>
</tr>
<tr>
<td>Program segments—Continued</td>
<td>25</td>
</tr>
<tr>
<td>Subroutine CNCON</td>
<td>25</td>
</tr>
<tr>
<td>Subroutine OUTPT</td>
<td>25</td>
</tr>
<tr>
<td>Subroutine CHMOT</td>
<td>25</td>
</tr>
<tr>
<td>Evaluation of model</td>
<td>25</td>
</tr>
<tr>
<td>Comparison with analytical solutions</td>
<td>25</td>
</tr>
<tr>
<td>Mass balance tests</td>
<td>28</td>
</tr>
<tr>
<td>Test problem 1—spreading of a tracer slug</td>
<td>28</td>
</tr>
<tr>
<td>Test problem 2—effects of wells</td>
<td>31</td>
</tr>
<tr>
<td>Test problem 3—effects of user options</td>
<td>32</td>
</tr>
<tr>
<td>Possible program modifications</td>
<td>34</td>
</tr>
<tr>
<td>Coordinate system and boundary conditions</td>
<td>35</td>
</tr>
<tr>
<td>Basic equations</td>
<td>36</td>
</tr>
<tr>
<td>Input and output</td>
<td>36</td>
</tr>
<tr>
<td>Conclusions</td>
<td>37</td>
</tr>
<tr>
<td>References cited</td>
<td>37</td>
</tr>
<tr>
<td>Attachment I, Fortran IV program listing</td>
<td>41</td>
</tr>
<tr>
<td>Attachment II, Definition of selected program variables</td>
<td>74</td>
</tr>
<tr>
<td>Attachment III, Data input formats</td>
<td>76</td>
</tr>
<tr>
<td>Attachment IV, Input data for test problem 3</td>
<td>79</td>
</tr>
<tr>
<td>Attachment V, Selected output for test problem 3</td>
<td>80</td>
</tr>
</tbody>
</table>

## FIGURES

1. Part of hypothetical finite-difference grid showing relation of flow field to movement of points... 7
2. Part of hypothetical finite-difference grid showing areas over which bilinear interpolation is used to compute the velocity at a point... 7
3. Representative change in breakthrough curve from time level \( k-1 \) to \( k \) 11
4. Possible movement of particles near an impermeable (no-flow) boundary 15
5. Replacement of points in source cells adjacent to a no-flow boundary 16
6. Replacement of points in source cells not adjacent to a no-flow boundary for negligible regional flow (a) and for relatively strong regional flow (b) 17
7. Relation between possible initial locations of points and indices of adjacent nodes 19
8. Simplified flow chart illustrating the major steps in the calculation procedure 21
9. Parts of finite-difference grids showing the initial geometry of particle distribution for the specification of four (a), five (b), eight (c), and nine (d) particles per cell 23
10. Generalized flow chart of subroutine MOVE 24
11. Comparison between analytical and numerical solutions for dispersion in one-dimensional, steady-state flow 26
FIGURES—Continued

12. Comparison between analytical and numerical solutions for dispersion in plane radial steady-state flow .................................................. 28
13. Grid, boundary conditions, and flow field for test problem 1 ................................................................. 29
15. Grid, boundary conditions, and flow field for test problem 2 ........................................................................ 30
17. Grid, boundary conditions, and flow field for test problem 3 ........................................................................ 32
18. Effect of NPTPND on mass-balance error for test problem 3; CELDIS=0.50 in all cases ............... 33
19. Effect of CELDIS on mass-balance error for test problem 3; NPTPND=9 in all cases ............. 34

TABLES

1. List of subroutines for solute-transport model ................................................................. 20
2. Model parameters for test problem 1 ....................................................................................... 29
3. Model parameters for test problems 2 and 3 .......................................................................... 31
4. Effect of NPTPND on accuracy, precision, and efficiency of solution to test problem 3 ................ 33
5. Effect of CELDIS on accuracy, precision, and efficiency of solution to test problem 3 ............. 33
COMPUTER MODEL OF TWO-DIMENSIONAL SOLUTE TRANSPORT AND DISPERSION IN GROUND WATER

By L. F. Konikow and J. D. Bredehoeft

Abstract

This report presents a model that simulates solute transport in flowing ground water. The model is both general and flexible in that it can be applied to a wide range of problem types. It is applicable to one- or two-dimensional problems involving steady-state or transient flow. The model computes changes in concentration over time caused by the processes of convective transport, hydrodynamic dispersion, and mixing (or dilution) from fluid sources. The model assumes that the solute is non-reactive and that gradients of fluid density, viscosity, and temperature do not affect the velocity distribution. However, the aquifer may be heterogeneous and (or) anisotropic.

The model couples the ground-water flow equation with the solute-transport equation. The digital computer program uses an alternating-direction implicit procedure to solve a finite-difference approximation to the ground-water flow equation, and it uses the method of characteristics to solve the solute-transport equation. The latter uses a particle-tracking procedure to represent convective transport and a two-step explicit procedure to solve a finite-difference equation that describes the effects of hydrodynamic dispersion, fluid sources and sinks, and divergence of velocity. This explicit procedure has several stability criteria, but the consequent time-step limitations are automatically determined by the program.

The report includes a listing of the computer program, which is written in FORTRAN IV and contains about 2,000 lines. The model is based on a rectangular, block-centered, finite-difference grid. It allows the specification of any number of injection or withdrawal wells and of spatially varying diffuse recharge or discharge, saturated thickness, transmissivity, boundary conditions, and initial heads and concentrations. The program also permits the designation of up to five nodes as observation points, for which a summary table of head and concentration versus time is printed at the end of the calculations.

The data input formats for the model require three data cards and from seven to nine data sets to describe the aquifer properties, boundaries, and stresses.

The accuracy of the model was evaluated for two idealized problems for which analytical solutions could be obtained. In the case of one-dimensional flow the agreement was nearly exact, but in the case of plane radial flow a small amount of numerical dispersion occurred. An analysis of several test problems indicates that the error in the mass balance will be generally less than 10 percent. The test problems demonstrated that the accuracy and precision of the numerical solution is sensitive to the initial number of particles placed in each cell and to the size of the time increment, as determined by the stability criteria. Mass balance errors are commonly the greatest during the first several time increments, but tend to decrease and stabilize with time.

Introduction

This report describes and documents a computer model for calculating transient changes in the concentration of a nonreactive solute in flowing ground water. The computer program solves two simultaneous partial differential equations. One equation is the ground-water flow equation, which describes the head distribution in the aquifer. The second is the solute-transport equation, which describes the chemical concentration in the system. By coupling the flow equation with the solute-transport equation, the model can be applied to both steady-state and transient flow problems.

The purpose of the simulation model is to compute the concentration of a dissolved chemical species in an aquifer at any specified place and time. Changes in chemical concentration occur within a dynamic ground-water system primarily due to four
distinct processes: (1) convective transport, in which dissolved chemicals are moving with the flowing ground water; (2) hydrodynamic dispersion, in which molecular and ionic diffusion and small-scale variations in the velocity of flow through the porous media cause the paths of dissolved molecules and ions to diverge or spread from the average direction of ground-water flow; (3) fluid sources, where water of one composition is introduced into water of a different composition; and (4) reactions, in which some amount of a particular dissolved chemical species may be added to or removed from the ground water due to chemical and physical reactions in the water or between the water and the solid aquifer materials. The model presented in this report assumes (1) that no reactions occur that affect the concentration of the species of interest, and (2) that gradients of fluid density, viscosity, and temperature do not affect the velocity distribution.

This model can be applied to a wide variety of field problems. However, the user should first become aware of the assumptions and limitations inherent in the model, as described in this report. The computer program presented in this report is offered as a basic working tool that may have to be modified by the user for efficient application to specific field problems. The program is written in FORTRAN IV and is compatible with most high-speed computers. The data requirements, input format specifications, program options, and output formats are all structured in a general manner that should be readily adaptable to many field problems.

This report includes a detailed description of the numerical method used to solve the solute-transport equation. The reader is assumed to have (or can obtain elsewhere) a moderate familiarity with finite-difference methods and ground-water flow models.

Theoretical Background
Flow equation

By following the derivation of Pinder and Bredehoeft (1968), the equation describing the transient two-dimensional areal flow of a homogeneous compressible fluid through a nonhomogeneous anisotropic aquifer can be written in Cartesian tensor notation as

$$\frac{\partial}{\partial x_i} (T_{ij} \frac{\partial h}{\partial x_j}) = S \frac{\partial h}{\partial t} + W \quad i,j = 1,2 \quad (1)$$

where

- $T_{ij}$ is the transmissivity tensor, $L^2/T$;
- $h$ is the hydraulic head, $L$;
- $S$ is the storage coefficient, (dimensionless);
- $t$ is the time, $T$;
- $W - W(x,y,t)$ is the volume flux per unit area (positive sign for outflow and negative for inflow), $L/T$; and
- $x_i$ and $x_j$ are the Cartesian coordinates, $L$.

If we only consider fluxes of (1) direct withdrawal or recharge, such as well pumpage, well injection, or evapotranspiration, and (2) steady leakage into or out of the aquifer through a confining layer, streambed, or lakebed, then $W(x,y,t)$ may be expressed as

$$W(x,y,t) = Q(x,y,t) - \frac{K_s}{m} (H_s - h) \quad (2)$$

where

- $Q$ is the rate of withdrawal (positive sign) or recharge (negative sign), $L/T$;
- $K_s$ is the vertical hydraulic conductivity of the confining layer, streambed, or lakebed, $L/T$;
- $m$ is the thickness of the confining layer, streambed, or lakebed, $L$; and
- $H_s$ is the hydraulic head in the source bed, stream, or lake, $L$.

Lohman (1972) shows that an expression for the average seepage velocity of ground water can be derived from Darcy's law. This expression can be written in Cartesian tensor notation as

$$V_i = - \frac{K_{ij}}{\epsilon} \frac{\partial h}{\partial x_j} \quad (3)$$
MODEL OF SOLUTE TRANSPORT IN GROUND WATER

where

\[ V_i \] is the seepage velocity in the direction of \( x_i , L/T \);  
\[ K_{ij} \] is the hydraulic conductivity tensor, \( L/T \); and  
\[ \epsilon \] is the effective porosity of the aquifer, (dimensionless).

Transport equation

The equation used to describe the two-dimensional areal transport and dispersion of a given nonreactive dissolved chemical species in flowing ground water was derived by Reddell and Sunada (1970), Bear (1972), Bredehoeft and Pinder (1973), and Konikow and Grove (1977): The equation may be written as

\[
\frac{\partial (Cb)}{\partial t} - \frac{\partial}{\partial x_i} (bD_{ij} \frac{\partial C}{\partial x_j}) - \frac{\partial}{\partial x_i} (bCV_i) - \frac{C'W}{\epsilon} i,j=1,2 \tag{4}
\]

where

\[ C \] is the concentration of the dissolved chemical species, \( M/L^3 \);  
\[ D_{ij} \] is the coefficient of hydrodynamic dispersion (a second-order tensor), \( L^3/T \);  
\[ b \] is the saturated thickness of the aquifer, \( L \); and  
\[ C' \] is the concentration of the dissolved chemical in a source or sink fluid, \( M/L^3 \).

The first term on the right side of equation 4 represents the change in concentration due to hydrodynamic dispersion. The second term describes the effects of convective transport, while the third term represents a fluid source or sink.

Dispersion coefficient

Bear (1972, p. 580–581) states that hydrodynamic dispersion is the macroscopic outcome of the actual movements of individual tracer particles through the pores and that it includes two processes. One process is mechanical dispersion, which depends upon both the flow of the fluid and the nature of the poro system through which the flow takes place. The second process is molecular and ionic diffusion, which because it depends on time, is more significant at low flow velocities. Bear (1972) further states that the separation between the two processes is artificial. In developing our model we assume for flowing ground-water systems that the definable contribution of molecular and ionic diffusion to hydrodynamic dispersion is negligible.

The dispersion coefficient may be related to the velocity of ground-water flow and to the nature of the aquifer using Scheidegger's (1961) equation:

\[ D_{ij} = \alpha_{ijmn} \frac{V_m V_n}{|V|} \tag{5} \]

where

\[ \alpha_{ijmn} \] is the dispersivity of the aquifer, \( L \);  
\[ V_m \] and \( V_n \) are components of velocity in the \( m \) and \( n \) directions, respectively, \( L/T \); and  
\[ |V| \] is the magnitude of the velocity, \( L/T \).

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 (\( \alpha_L \) and \( \alpha_T \), respectively). These are related to the longitudinal and transverse dispersion coefficients by

\[ D_L = \alpha_L |V| \tag{6} \]

and

\[ D_T = \alpha_T |V| . \tag{7} \]

After expanding equation 5, substituting Scheidegger's identities, and eliminating terms with coefficients that equal zero, the components of the dispersion coefficient for two-dimensional flow in an isotropic aquifer may be stated explicitly as

\[ D_{xx} = D_L \frac{(V_x)^2}{|V|^2} + D_T \frac{(V_x)^2}{|V|^2} \tag{8} \]

\[ D_{yy} = D_T \frac{(V_y)^2}{|V|^2} + D_L \frac{(V_y)^2}{|V|^2} \tag{9} \]
\[ D_{xy} = D_{yx} = (D_L - D_T) \frac{V_x V_y}{|V|^2}. \quad (10) \]

Note that while \( D_{xx} \) and \( D_{yy} \) must have positive values, it is possible for the cross-product terms (eq 10) to have negative values if \( V_x \) and \( V_y \) have opposite signs.

**Review of assumptions**

A number of assumptions have been made in the development of the previous equations. Following is a list of the main assumptions that must be carefully evaluated before applying the model to a field problem.

1. Darcy’s law is valid and hydraulic-head gradients are the only significant driving mechanism for fluid flow.
2. The porosity and hydraulic conductivity of the aquifer are constant with time, and porosity is uniform in space.
3. Gradients of fluid density, viscosity, and temperature do not affect the velocity distribution.
4. No chemical reactions occur that affect the concentration of the solute, the fluid properties, or the aquifer properties.
5. Ionic and molecular diffusion are negligible contributors to the total dispersive flux.
6. Vertical variations in head and concentration are negligible.
7. The aquifer is homogeneous and isotropic with respect to the coefficients of longitudinal and transverse dispersivity.

The nature of a specific field problem may be such that not all of these underlying assumptions are completely valid. The degree to which field conditions deviate from these assumptions will affect the applicability and reliability of the model for that problem. If the deviation from a particular assumption is significant, the governing equations will have to be modified to account for the appropriate processes or factors.

**Numerical Methods**

Because aquifers have variable properties and complex boundary conditions, exact analytical solutions to the partial differential equations of flow (eq 1) and solute transport (eq 4) cannot be obtained directly. Therefore, approximate numerical methods must be employed.

The numerical methods require that the area of interest be subdivided by a grid into a number of smaller subareas. The model developed here utilizes a rectangular, uniformly spaced, block-centered, finite-difference grid, in which nodes are defined at the centers of the rectangular cells.

**Flow equation**

Pinder and Bredehoeft (1968) show that if the coordinate axes are aligned with the principal directions of the transmissivity tensor, equation 1 may be approximated by the following implicit finite-difference equation:

\[
T_{x[i\pm\frac{1}{2}, j]} \left[ h_{i-1,j,k} - h_{i,j,k} \right] \\
+ T_{y[i, j\pm\frac{1}{2}]} \left[ h_{i,j+1,k} - h_{i,j,k} \right] \\
+ T_{y[i,j\pm\frac{1}{2}]} \left[ h_{i,j-1,k} - h_{i,j,k} \right] \\
+ T_{y[i,j\pm\frac{1}{2}]} \left[ h_{i,j+1,k} - h_{i,j,k} \right] \\
= S \left[ h_{i,j,k} - h_{i,j,k-1} \right] \\
+ \frac{q_w(i,j) K_x}{\Delta x \Delta y \Delta t} \left[ H_{x(i,j)} - h_{i,j,k} \right] \quad (11)
\]

where

- \( i,j,k \) are indices in the \( x, y, \) and time dimensions, respectively;
- \( \Delta x, \Delta y, \Delta t \) are increments in the \( x, y, \) and time dimensions, respectively; and
- \( q_w \) is the volumetric rate of withdrawal or recharge at the \( (i,j) \) node, \( L^3/T \).

Note that \( k \) represents the new time level and \( k-1 \) represents the previous time level. To avoid confusion between tensor sub-
scripts and nodal indices, the latter are separated by commas.

The finite-difference equation (eq 11) is solved numerically for each node in the grid using an iterative alternating-direction implicit (ADI) procedure. The derivation and solution of the finite-difference equation and the use of the iterative ADI procedure have been previously discussed in detail in the literature. Some of the more relevant references include Pinder and Bredehoeft (1968), Prickett and Lonququist (1971), and Trescott, Pinder, and Larson (1976).

After the head distribution has been computed for a given time step, the velocity of ground-water flow is computed at each node using an explicit finite-difference form of equation 3. For example, the velocity in the x direction at node \((i,j)\) would be computed as

\[
V_x(i,j) = \frac{K_{zx}(i,j)}{\varepsilon} \frac{(h_{i-1,j,k} - h_{i+1,j,k})}{2\Delta x}.
\] (12)

The velocity in the x direction can also be computed on the boundary between node \((i,j)\) and node \((i+1,j)\) using the following equation:

\[
V_x(i+\frac{1}{2},j) = \frac{K_{zx}(i+\frac{1}{2},j)}{\varepsilon} \frac{(h_{i,j,k} - h_{i+1,j,k})}{\Delta x}
\] (13)

where the hydraulic conductivity on the boundary is computed as the harmonic mean of the hydraulic conductivities at the two adjacent nodes.

Expressions similar to equations 12 and 13 are used to compute the velocities in the y direction at \((i,j)\) and \((i,j+\frac{1}{2})\) respectively. Note that equation 13, which computes the head difference over a distance \(\Delta x\), is more accurate than equation 12, which computes the head difference over \(2\Delta x\).

**Transport equation**

**Method of characteristics**

The method of characteristics is used in this model to solve the solute-transport equation. This method was developed to solve hyperbolic differential equations. If solute transport is dominated by convective transport, as is common in many field problems, then equation 4 may closely approximate a hyperbolic partial differential equation and be highly compatible with the method of characteristics. Although it is difficult to present a rigorous mathematical proof for this numerical scheme, it has been successfully applied to a variety of field problems. The development of this technique for problems of flow through porous media has been presented by Garder, Peaceman, and Pozzi (1964), Pinder and Bredehoeft (1973), and Reddell and Sunada (1970), and Reddell and Sunada (1970). Garder, Peaceman, and Pozzi (1964) state that this technique does not introduce numerical dispersion (artificial dispersion resulting from the numerical calculation process). They and Reddell and Sunada (1970) also compared solutions obtained using the method of characteristics with those derived by analytical methods and found good agreement for the cases investigated. Applications of the method to field problems have been documented by Bredehoeft and Pinder (1973), Konikow and Bredehoeft (1974), Robertson (1974), Robson (1974), and Konikow (1977).

The approach taken by the method of characteristics is not to solve equation 4 directly, but rather to solve an equivalent system of ordinary differential equations. Konikow and Grove (1977, eq 61) show that by considering saturated thickness as a variable and by expanding the convective transport term, equation 4 may be rewritten as

\[
\frac{\partial C}{\partial t} = \frac{1}{b} \frac{\partial}{\partial x_i} \left( bD_{x_i} \frac{\partial C}{\partial x_i} - V_x \frac{\partial C}{\partial x} \right) + \frac{C(S \frac{\partial h}{\partial t} + W - \frac{\partial b}{\partial t}) - CW}{eb}. \] (14)

Equation 14 is the form of the solute-transport equation that is solved in the computer program presented in this report. For convenience we may also write equation 14 as

\[
\frac{\partial C}{\partial t} = \frac{1}{b} \frac{\partial}{\partial x_i} \left( bD_{x_i} \frac{\partial C}{\partial x_i} - V_x \frac{\partial C}{\partial x} - V_y \frac{\partial C}{\partial y} + F \right). \] (15)
where
\[ C(S \frac{\partial h}{\partial t} + W - \frac{\partial b}{\partial t}) - C'W \]
\[ F = \frac{\partial b}{\partial t} \]  
(16)

Next consider representative fluid particles that are convected with flowing ground water. Note that changes with time in properties of the fluid, such as concentration, may be described either for fixed points within a stationary coordinate system as successive fluid particles pass the reference points, or for reference fluid particles as they move along their respective paths past fixed points in space. Aris (1962, p. 78) states that "associated with these two descriptions are two derivatives with respect to time." Thus \( \frac{\partial C}{\partial t} \) is the rate of change of concentration as observed from a fixed point, whereas \( \frac{dC}{dt}\) is the rate of change as observed when moving with the fluid particle. Aris (1962) calls the latter the material derivative.

The material derivative of concentration may be defined as
\[ \frac{dC}{dt} = \frac{\partial C}{\partial t} + \frac{\partial C}{\partial x} \frac{dx}{dt} + \frac{\partial C}{\partial y} \frac{dy}{dt} \]  
(17)

Note the correspondence of the second and third terms on the right side of equation 15 with the second and third terms on the right side of equation 17. The latter includes the material derivatives of position, which are defined by velocity. Thus for the \( x \) and \( y \) components, respectively, of position and velocity we have
\[ \frac{dx}{dt} = V_x \]  
(18)
and
\[ \frac{dy}{dt} = V_y. \]  
(19)

If we next substitute the right sides of equations 16, 18, and 19 for the corresponding terms in equation 17, we obtain
\[ \frac{dC}{dt} = \frac{1}{b} \frac{\partial}{\partial x} (bD_v \frac{\partial C}{\partial x}) + F. \]  
(20)

The solutions of the system of equations comprising equations 18-20 may be given as
\[ x = x(t); \quad y = y(t); \quad \text{and} \quad C = C(t) \]  
(21)
and are called the characteristic curves of equation 15.

Given solutions to equations 18–20, a solution to the partial differential equation (eq 15) may be obtained by following the characteristic curves. This is accomplished numerically by introducing a set of moving points (or reference particles) that can be traced within the stationary coordinates of the finite-difference grid. Garder, Peaceman, and Pozzi (1964, p. 27) state, "Each point corresponds to one characteristic curve, and values of \( x \), \( y \), and \( C \) are obtained as functions of \( t \) for each characteristic." Each point has a concentration and position associated with it and is moved through the flow field in proportion to the flow velocity at its location. Intuitively, the method may be visualized as tracing a number of fluid particles through a flow field and observing changes in chemical concentration in the fluid particles as they move.

**Particle tracking**

The first step in the method of characteristics involves placing a number of traceable particles or points in each cell of the finite-difference grid to form a set of points that are distributed in a geometrically uniform pattern throughout the area of interest. It was found that placing from four to nine points per cell provided satisfactory results for most two-dimensional problems. The location or position of each particle is specified by its \( x \)- and \( y \)-coordinates in the finite-difference grid. The initial concentration assigned to each point is the initial concentration associated with the node of the cell containing the point.

For each time step every point is moved a distance proportional to the length of the time increment and the velocity at the location of the point. (See fig. 1.) The new position of a point is thus computed with the following finite-difference forms of equations 18 and 19:
\[ x_{p,k} = x_{p,k-1} + \delta x_p = x_{p,k-1} + \Delta t V_x x(x_{p,k-1}, y_{p,k}) \]  
(22)
Figure 1.—Part of hypothetical finite-difference grid showing relation of flow field to movement of points.

\[ y_{p,k} = y_{p,k-1} + \delta y_p = y_{p,k-1} + \Delta t V_y(x_{(p,k)}, y_{(p,k)}) \]  

where

- \( p \) is the index number for point identification;
- \( \delta x_p \) and \( \delta y_p \) are the distances moved in the \( x \) and \( y \) directions, respectively.

The \( x \) and \( y \) velocities at the position of any particular point \( p \), indicated as \( V_{x(y)(p,k)} \), for time \( k \) are calculated through bilinear interpolation over the area of half of a cell using the \( x \) and \( y \) velocities computed at adjacent nodes and cell boundaries. For example, figure 2 illustrates that the velocity in the \( x \) direction of point \( p \), located in the southeast quadrant of cell \((i,j)\), would be computed using bilinear interpolation between the \( x \) velocities computed with equations 12 and 13 at \((i,j)\), \((i,j+1)\), \((i+1/2,j)\), and \((i+1,j+1/2)\). Similarly, the velocity in the \( y \) direction of point \( p \) would be based on the \( y \) velocities computed at \((i,j)\), \((i+1,j)\), \((i,j+1/2)\) and \((i+1,j+1/2)\).

After all points have been moved, the concentration at each node is temporarily assigned the average of the concentrations of all points then located within the area of that cell; this average concentration is denoted as \( C_{i,j,*} \). The time index is distinguished with an asterisk here because this temporarily assigned average concentration represents the new time level only with respect to convective transport. The moving points simulate convective transport because the concentration at each node of the grid will change with each time step as different points having different concentrations enter and leave the area of that cell.

**Finite-difference approximations**

The total change in concentration in an aquifer may be computed by solving equations 18–20. Equations 18 and 19, which are related to changes in concentration caused
by convective transport alone, are solved by the movement of points as described previously. The changes in concentration caused by hydrodynamic dispersion, fluid sources, divergence of velocity, and changes in saturated thickness are calculated using an explicit finite-difference approximation to equation 20, which can be expressed as

$$
\Delta C_{i,j,k} = \Delta t \left[ \frac{1}{b} \frac{\partial}{\partial x_i} (bD_{ij} \frac{\partial C}{\partial x_i}) + F \right].
$$

Equation 24)

Note that a solution to equation 20 requires the computation of the change in concentration at the tracer particles. However, primarily because of the difficulty in computing the concentration gradient at a large number of moving points, the change in concentration represented by equation 20 is solved at each node of the grid rather than directly at the location of each point. The material derivative of concentration on any characteristic curve (or for any tracer particle) is then related to the change in concentration for a node during one time step, which was computed with the solution to equation 24.

The right side of equation 24 can be considered as the sum of two separate terms, as follows:

$$
\Delta C_{i,j,k} = (\Delta C_{i,j,k})_1 + (\Delta C_{i,j,k})_II
$$

(25)

where

$$
(\Delta C_{i,j,k})_1 = \frac{\Delta t}{b} \left[ \frac{\partial}{\partial x_i} (bD_{ij} \frac{\partial C}{\partial x_i}) \right]
$$

(26)

and

$$
(\Delta C_{i,j,k})_II = \Delta t F
$$

and

$$
(\Delta C_{i,j,k})_II = \Delta t F
$$

First we will examine the change in concentration due to dispersion, partly following the development of Reddell and Sunada (1970). The right side of equation 24 can be expanded according to the summation convention of tensor notation to obtain

$$
(\Delta C_{i,j,k})_1 = \frac{\Delta t}{b} \left[ \frac{\partial}{\partial x} \left( bD_{xx} \frac{\partial C}{\partial x} + bD_{xy} \frac{\partial C}{\partial y} \right) + \frac{\partial}{\partial y} \left( bD_{yx} \frac{\partial C}{\partial x} + bD_{yy} \frac{\partial C}{\partial y} \right) \right].
$$

(28)

A finite-difference approximation for the derivative in the $x$ direction at $(i,j)$ may be written as

$$
\frac{\partial}{\partial x} \left( bD_{xx} \frac{\partial C}{\partial x} + bD_{xy} \frac{\partial C}{\partial y} \right)
$$

$$
= \frac{\partial}{\partial x} \left( bD_{xx} \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial x} \left( bD_{xy} \frac{\partial C}{\partial y} \right)
$$

$$
= \left. \left( bD_{xx} \frac{\partial C}{\partial x} \right) \right|_{i+1/2,j} - \left. \left( bD_{xx} \frac{\partial C}{\partial x} \right) \right|_{i-1/2,j} + \frac{\Delta x}{\Delta x} \left( bD_{xy} \frac{\partial C}{\partial y} \right)_{i+1/2,j} - \left( bD_{xy} \frac{\partial C}{\partial y} \right)_{i-1/2,j}
$$

(29)

In the following expansion of equation 29 it is implied that concentrations ($C$) are known from the previous ($k-1$) time level; hence, equation 29 is an explicit finite-difference equation. The spatial derivatives of concentration at $(i+1/2,j)$ may be approximated by

$$
\left. \left( \frac{\partial C}{\partial x} \right) \right|_{i+1/2,j} = \frac{C_{i+1,j} - C_{i,j}}{\Delta x}
$$

(30)

and

$$
\left. \left( \frac{\partial C}{\partial y} \right) \right|_{i+1/2,j} = \frac{C_{i+1/2,j+1} - C_{i+1/2,j-1}}{2\Delta y}
$$

(31)

Because concentrations are defined only at nodes, we must express the right side of equation 31 in terms of concentrations at nodes. Assuming that the concentration at a
cell boundary is approximately equal to the average (arithmetic mean) of the concentrations at adjacent nodes, we have

\[
C_{i \pm \frac{1}{2}, j + 1} = \frac{C_{i+1, j} + C_{i+1, j+1}}{2} \quad (32)
\]

and

\[
C_{i + \frac{1}{2}, j - 1} = \frac{C_{i, j} + C_{i+1, j-1}}{2} \quad (33)
\]

Substitution of equations 32 and 33 into equation 31 results in:

\[
\frac{\partial}{\partial x} (bD_{xx} \frac{\partial C}{\partial x} + bD_{xy} \frac{\partial C}{\partial y}) = \frac{bD_{xx}(C_{i+1, j} - C_{i, j})}{(\Delta x)^2} - \frac{bD_{xx}(C_{i, j} - C_{i-1, j})}{(\Delta x)^2}
\]

\[
+ \frac{bD_{xy}(C_{i+1, j} + C_{i+1, j+1} - C_{i, j-1} - C_{i+1, j-1})}{4\Delta x \Delta y} - \frac{bD_{xy}(C_{i-1, j} + C_{i, j+1} - C_{i-1, j-1} - C_{i, j})}{4\Delta x \Delta y}
\]

A finite-difference approximation for the derivative in the \( y \) direction in equation 28 may be developed for node \((i,j)\) in an analogous manner to equation 37 to produce

\[
\frac{\partial}{\partial y} (bD_{yy} \frac{\partial C}{\partial y} + bD_{yx} \frac{\partial C}{\partial x})
\]

\[
\frac{(bD_{yy} \frac{\partial C}{\partial y})_{i+\frac{1}{2}, j} - (bD_{yy} \frac{\partial C}{\partial y})_{i-\frac{1}{2}, j}}{\Delta y} + \frac{(bD_{yx} \frac{\partial C}{\partial x})_{i+\frac{1}{2}, j} - (bD_{yx} \frac{\partial C}{\partial x})_{i-\frac{1}{2}, j}}{\Delta y}
\]

\[
\frac{bD_{xy}(C_{i+1, j} - C_{i, j})}{(\Delta y)^2} - \frac{bD_{xy}(C_{i, j} - C_{i-1, j})}{(\Delta y)^2}
\]

\[
+ \frac{bD_{xy}(C_{i+1, j} + C_{i+1, j+1} - C_{i, j-1} - C_{i+1, j-1})}{4\Delta x \Delta y} - \frac{bD_{xy}(C_{i-1, j} + C_{i, j+1} - C_{i-1, j-1} - C_{i, j})}{4\Delta x \Delta y}
\]

Equation 28 may then be solved explicitly by substituting the relationships expressed by equations 37 and 38 for the terms within brackets on the right side of equation 28.
Next we will examine the change in concentration denoted by equation 27. Substituting explicit finite-difference approximations for the terms in equation 27, we have

\[
(\Delta C_{i,j,k})_{II} = \frac{\Delta t}{c_{i,j,k}} \left[ C_{i,j,k-1} \left( S \left[ \frac{b_{i,j,k} - b_{i,j,k-1}}{\Delta t} \right] W_{i,j,k} \right) + C' \left( \frac{W_{i,j,k}}{\Delta t} \right) \right].
\]

Equations 28, 37, 38, and 39 together provide a solution to equation 24, which in turn allows us to solve equation 20 and complete the definition of the characteristic curves of equation 15.

Because the processes of convective transport, hydrodynamic dispersion, and mixing are occurring continuously and simultaneously, equations 18, 19, and 20 should be solved simultaneously. However, equations 18 and 19 are solved by particle movement based on implicitly computed heads while equation 20 is solved explicitly with respect to concentrations. Because the change in concentration at a source node due to mixing is proportional to the difference in concentration between the node and the source fluid (see eq 27), the accuracy of estimating the concentration at the node during a time increment will clearly affect the computed change. Similarly, because the change in concentration due to dispersion is proportional to the concentration gradient at a point, the accuracy of estimating the concentration gradient will clearly affect the accuracy of the numerical results. As the position of a front or breakthrough curve advances with time, say from the \(k-1\) to \(k\) time level, the concentration gradient at any fixed reference point and the concentration differences at sources are continuously changing. The consequent limitations imposed by estimating nodal concentrations in a strict explicit manner can be minimized by using a two-step explicit procedure in which equation 24 is solved at each node by giving equal weight to concentration gradients computed from the concentrations at the previous time level \((k-1)\) and to concentration gradients computed from concentrations at time level \((k^*)\), which represents the convected position of the front at the new time level \((k)\) prior to adjustments of concentration for dispersion and mixing. Figure 3 illustrates the sequence of calculations to solve equations 18–20 over a given time increment. First the concentration gradients at the previous time level \((k-1)\) are determined at each node. Then the front is convected to a new position for time level \(k^*\) based on the velocity of flow and length of the time increment. Next the concentration gradients at each node are recomputed for the new position of the front. The concentration distribution for the new frontal position is then adjusted at each node in two steps: first based on concentration gradients at \(k-1\) and second based on concentration gradients at \(k^*\).

The finite-difference approximation to equation 24 may thus be expressed as

\[
\Delta C_{i,j,k} = \frac{0.5 \Delta t}{b} \left[ \frac{\partial}{\partial x_i} (bD_{ij} \frac{\partial C_{(k-1)}}{\partial x_j}) + \frac{C_{(k-1)} (S \frac{\partial h}{\partial t} + W - \epsilon \frac{\partial b}{\partial t}) - C'W}{\epsilon} \right] + \frac{0.5 \Delta t}{b} \left[ \frac{\partial}{\partial x_i} (bD_{ij} \frac{\partial C_{(k^*)}}{\partial x_j}) + \frac{C_{(k^*)} (S \frac{\partial h}{\partial t} + W - \epsilon \frac{\partial b}{\partial t}) - C'W}{\epsilon} \right]
\]

in which the appropriate finite-difference approximations for the terms within brackets are indicated by equations 37, 38, and 39.

The new nodal concentrations at the end of time increment \(k\) are computed as

\[
C_{i,j,k} = C_{i,j,k*} + \Delta C_{i,j,k}
\]
where $C_{i,j,k^*}$ is the average of the concentrations of all points in cell $(i,j)$ after equations 22 and 23 were solved for all points for time step $k$, and $\Delta C_{i,j,k}$ is the change in concentration caused by hydrodynamic dispersion, sources, and sinks, as calculated in equation 40.

Because the concentrations of points in a cell vary about the concentration of the node, the change in concentration computed at a node using equation 40 cannot be applied directly in all cases to the concentrations of the points. If the change in concentration at the node ($\Delta C_{i,j,k}$) is positive, the increase is simply added to the point concentrations. But if the concentration change is negative, it is applied to points in that cell as a percentage decrease in concentration at each point that is equal to the percentage decrease at the node. This technique preserves a mass balance within each cell, but when a decrease in concentration is computed for a node, it will also prevent a possible but erroneous computation of negative concentrations at those points that had a concentration less than that at the node.

### Stability criteria

The explicit numerical solution of the solute-transport equation has a number of stability criteria associated with it. These may require that the time step used to solve the flow equation be subdivided into a number of smaller time increments to accurately solve the solute-transport equation.

First, Reddell and Sunada (1970, p. 62) show that for an explicit finite-difference solution of equation 26 to be stable,
Substituting equation 47 into equation 45 results in

$$\frac{\Delta t}{\epsilon b_{i,j,k}} \leq 1.0.$$  (48)

Solving equation 48 for $\Delta t$ at all nodes yields the following criterion:

$$\Delta t \leq \min \left( \frac{\epsilon b_{i,j,k}}{W_{i,j,k}} \right).$$  (49)

A third type of stability check involves the movement of points computed by equations 22 and 23 to simulate convective transport. The distance a particle moves is defined as

$$\delta x = \Delta t V_{x_{i,j,k}}$$  (50)

and

$$\delta y = \Delta t V_{y_{i,j,k}}.$$  (51)

In effect, this constitutes a linear spatial extrapolation of the position of a particle from one time step to the next. Where streamlines are curvilinear, the extrapolated position of a particle will deviate from the streamline on which it was previously located. This deviation introduces an error into the numerical solution that is proportional to $\Delta t$. Thus, it is thought that an accurate computation of concentration changes caused by convective transport requires the maintenance of a relatively uniformly spaced field of marker particles that are moving along relatively smooth and continuous pathlines. Also, if $\delta x$ is greater than $\Delta x$, or $\delta y$ is greater than $\Delta y$, it might be possible for particles to move beyond the boundaries of the grid during one time increment. Thus, for a given velocity field and grid, some restriction must be placed on the size of the time increment to assure that neither $\delta x$ nor $\delta y$ exceed some critical distances, called $\delta x^*$ and $\delta y^*$. Therefore

$$\delta x \leq \delta x^*$$  (52)

and

$$\delta y \leq \delta y^*.$$  (53)

These critical distances can be related to the dimensions of the finite-difference grid by

$$\delta x^* = \gamma \Delta x$$  (54)

and
\[ \delta y^* = \gamma \Delta y \]  
(55)

where \( \gamma \) is the fraction of the grid dimensions that particles will be allowed to move \((0 < \gamma \leq 1)\).

If we replace the terms in equations 52 and 53 with the corresponding terms from equations 50, 51, 54, and 55, we have

\[ \Delta t V_z[x_{(p,b)}, y_{(p,b)}] \leq \gamma \Delta x \]
(56)

and

\[ \Delta t V_y[x_{(p,b)}, y_{(p,b)}] \leq \gamma \Delta y. \]
(57)

Because these criteria are governed by the maximum velocities in the system, and since the computed velocity of a tracer particle will always be less than or equal to the maximum velocity computed at a node or cell boundary, we have to check only the latter. Substituting the grid velocities and solving equations 56 and 57 for \( \Delta t \) results in

\[ \Delta t \leq \frac{\gamma \Delta x}{(V_x)_{\text{max}}} \]
(58)

and

\[ \Delta t \leq \frac{\gamma \Delta y}{(V_y)_{\text{max}}}. \]
(59)

If the time step used to solve the flow equation exceeds the smallest of the time limits determined by equations 43, 49, 58, or 59, then the time step will be subdivided into the appropriate number of smaller time increments required for solving the solute-transport equation.

**Boundary and initial conditions**

Obtaining a solution to the equations that describe ground-water flow and solute transport requires the specification of boundary and initial conditions for the domain of the problem. Specifications for solving the flow equation must be compatible with the solution of the solute-transport equation. Several different types of boundary conditions can be incorporated into the solute-transport model. Two general types are incorporated in this model; these are constant-flux and constant-head conditions. These can be used to represent the real boundaries of an aquifer as well as to represent artificial boundaries for the model. The use of the latter can help to minimize data requirements and the areal extent of the modeled part of the aquifer.

A constant-flux boundary can be used to represent aquifer underflow, well withdrawals, or well injection. A finite flux is designated by specifying the flux rate as a well discharge or injection rate for the appropriate nodes. A no-flow boundary is a special case of a constant-flux boundary. The numerical procedure used in this model requires that the area of interest be surrounded by a no-flow boundary. Thus the model will automatically specify the outer rows and columns of the finite-difference grid as no-flow boundaries. No-flow boundaries can also be located elsewhere in the grid to simulate natural limits or barriers to ground-water flow. No-flow boundaries are designated by setting the transmissivity equal to zero at appropriate nodes, thereby precluding the flow of water or dissolved chemicals across the boundaries of the cell containing that node.

A constant-head boundary in the model can represent parts of the aquifer where the head will not change with time, such as recharge boundaries or areas beyond the influence of hydraulic stresses. In this model constant-head boundaries are simulated by adjusting the leakage term (the last term on the right side of equation 11) at the appropriate nodes. This is accomplished by setting the leackance coefficient \( (K_s/m) \) to a sufficiently high value (such as 1.0 \( \text{s}^{-1} \)) to allow the head in the aquifer at a node to be implicitly computed as a value that is essentially equal to the value of \( H_n \), which in this case would be specified as the desired constant-head altitude. The resulting rate of leakage into or out of the designated constant-head cell would equal the flux required to maintain the head in the aquifer at the specified constant-head altitude.

If a constant-flux or constant-head boundary represents a fluid source, then the chemical concentration in the source fluid \( (C') \) must also be specified. If the boundary represents a fluid sink, then the concentration of the produced fluid will equal the concen-
tration in the aquifer at the location of the sink.

Because solute transport directly depends upon hydraulic and concentration gradients, the head and concentration in the aquifer at the start of the simulation period must be specified. The initial conditions can be determined from field data and (or) from previous simulations. It is important to note that the simulation results may be sensitive to variations or errors in the initial conditions. In discussing computed heads, Trescott, Pinder, and Larson (1976, p. 30) state:

If initial conditions are specified so that transient flow is occurring in the system at the start of the simulation, it should be recognized that water levels will change during the simulation, not only in response to the new pumping stress, but also due to the initial conditions. This may or may not be the intent of the user.

Mass balance

Mass balance calculations are performed after specified time increments to help check the numerical accuracy and precision of the solution. The principle of conservation of mass requires that the cumulative sums of mass inflows and outflows (or net flux) must equal the accumulation of mass (or change in mass stored). The difference between the net flux and the mass accumulation is the mass residual \( R_m \) and is one measure of the numerical accuracy of the solution. Although a small residual does not prove that the numerical solution is accurate, a large error in the mass balance is undesirable and may indicate the presence of a significant error in the numerical solution.

The model uses two methods to estimate the error in the mass balance. Both are based on the magnitude of the mass residual, \( R_m \), which is computed from

\[
R_m = \Delta M_s - M_f
\]

where

\( \Delta M_s \) is the change in mass stored in the aquifer, \( M_s \); and

\( M_f \) is the net mass flux, \( M \).

The two mass terms, \( \Delta M_s \) and \( M_f \), are evaluated using the following equations:

\[
\Delta M_s = \sum_{i,j} b_{i,j} \Delta x \Delta y (C_{i,j,k} - C_{i,j,0}) \quad (61a)
\]

where \( C_{i,j,0} \) is the initial concentration at node \( (i,j) \), \( M/L^3 \); and

\[
M_f = \sum_{i,j,k} W_{i,j,k} \Delta x \Delta y \Delta t \frac{C_{i,j,k}}{c_{i,j,k}}. \quad (61b)
\]

The percent error \( (E) \) in the mass balance is computed first by comparing the residual with the average of the net flux and net accumulation, as

\[
E = \frac{100.0 (M_f - \Delta M_s)}{0.5 (M_f + \Delta M_s)}. \quad (62)
\]

This is a good measure of the accuracy of the numerical solution when the flux and the change in mass stored are relatively large. However, equation 62 does not account for the initial mass of solute in the aquifer. If total fluxes are very small compared to the initial mass of solute in the aquifer, then equation 62 might indicate a relatively large error when the numerical solution is actually quite accurate. Therefore, the error may also be computed a second way by comparing the residual with the initial mass of solute \( (M_o) \) present in the aquifer, as

\[
E = \frac{100.0 (M_f - \Delta M_s)}{M_o}. \quad (63)
\]

Equation 63 provides a good measure of the accuracy of the numerical solution when fluxes are zero or relatively small. But when \( M_o \) is zero or very small in comparison to \( \Delta M_s \), then \( E \) becomes meaningless. This problem can be overcome by correcting \( M_o \) in the denominator of equation 63 for the net mass flux, resulting in

\[
E = \frac{100.0 (M_f - \Delta M_s)}{M_o - M_f}. \quad (64)
\]

Note that as \( M_f \) becomes very small, equation 64 approaches equation 63, and as \( M_o \) becomes very small, \( E \) becomes just a comparison of the residual with the net flux. In the latter case \( E \) is a mass balance indicator similar to \( E \) in equation 62. Thus, \( E \) is considered a more reliable and versatile indicator of numerical accuracy than is \( E \). Either one or both of \( E \) and \( E \) are computed by the model, as appropriate.
Special problems

There are a number of special problems associated with the use of the method of characteristics to solve the solute-transport equation. Some of these problems are associated with the movement and tracking of particles, while other problems are related to the computational transition between the concentrations of particles within a cell and the average concentration at that node. We will next describe the more significant problems and the procedures used to minimize errors that might result from them.

One possible problem is related to no-flow boundaries. Neither water nor dissolved chemicals can be allowed to cross a no-flow boundary. However, under certain conditions it might be possible for the interpolated velocity at the location of a particle near a no-flow boundary to be such that the particle will be convected across the boundary during one time increment. Figure 4 illustrates such a possible situation, which arises from the deviation between the curvilinear flow line and the linearly projected particle path. If a particle is convected across a no-flow boundary, then it is relocated within the aquifer by reflection across the boundary, as also shown in figure 4. This correction thus will tend to relocate the particle closer to the true flow line.

Fluid sources and sinks also require special treatment. Because they tend to represent singularities in the velocity field, the use of a central difference formulation (eq 12) to compute the velocity at a node may indicate zero or very small velocities at the nodes. Therefore, the velocity components at a source or sink node cannot be used for interpolation of the velocity at a point within or adjacent to that cell. To help maintain radial flow to or from a sink or source, respectively, the velocities computed on the boundaries of source or sink cells are assigned to that node. The appropriate boundary velocities are determined on the basis of the quadrant of interest. This can be illustrated by referring again to figure 2. For a point located in the southeast quadrant of cell (i,j), the x velocity at node (i,j) would be set equal to \( V_x(i+\frac{1}{2},j) \) and the y velocity to \( V_y(i+\frac{1}{2},j) \). Corresponding adjustments are made for points in other quadrants, so that the magnitude and direction of velocity at the node are not fixed for a given time increment, but depend on the relative location of the point of interest within the cell. A similar approximation is made when a point of interest is located in a cell adjacent to a source or sink. Thus, if the same point, p, in figure 2 were located in an unstressed cell but the adjacent cell (i+1,j) represented a source or sink, then the y velocity at the node \((i+1,j)\) would be approximated by \( V_y(i+1,j+\frac{1}{2}) \) in order to estimate the y velocity at point p. A corresponding approximation for the x velocity at node \((i,j+1)\) would be made using \( V_x(i+\frac{1}{2},j+1) \) if a source or sink were located at \((i,j+1)\).

The maintenance of a reasonably uniform and continuous spacing of points requires special treatment in areas where sources and sinks dominate the flow field. Points will continually move out of a cell that represents a source, but few or none will move in to re-
place them and thereby maintain a continuous stream of points. Thus, whenever a point that originated in a source cell moves out of that source cell, a new point is introduced into the source cell to replace it. Placement of new points in a source cell is compatible with and analogous to the generation of fluid and solute mass at the source.

The procedure used to replace points in source cells that are adjacent to no-flow boundaries is illustrated in figure 5. Here a steady, uniformly spaced stream of points is maintained by generating a new point at the same relative position in the source cell as the new position in the adjacent cell of the point that left the source cell. As an example, point 7 was convected from cell \((i-1,j)\) to cell \((i,j)\). So the replacement point (22) was placed at a location within cell \((i-1,j)\) that is identical to the new location of point 7 within cell \((i,j)\).

The procedure used to replace points in source cells that lie within the aquifer and not adjacent to a no-flow boundary is illustrated in figure 6. Here a steady, uniformly spaced stream of particles is maintained by generating a new point in the source cell at the original location of the point that left the source cell. When a relatively strong source is imposed on a relatively weak regional flow field, as illustrated in figure 6a, then radial flow will be maintained in the area of the source, and all initial and replacement points will move symmetrically away from node \((i,j)\). For example, after point 7 moves from cell \((i,j)\) to \((i+1, j-1)\), the replacement point (18) is positioned at time \(k\) in cell \((i,j)\) at the same location as the initial position of point 7. Although the replacement procedure illustrated earlier by figure 5 would work just as well for the case illustrated in figure 6a, it would not be satisfactory for the situation presented in figure 6b, which illustrates the imposition of a relatively weak source in a relatively strong regional flow field. In this case the velocity distribution within the source cell does not possess radial symmetry, and the velocity within the upgradient part of the source cell is much lower than the velocity within the downgradient part of the source cell. Replacement of points at original locations in source cells, as illustrated in figure 6b, will maintain a steady stream of points leaving the source cell in proportion to the velocity field. However, the use of the procedure illustrated in figure 5 for the case presented in figure 6b would result in the accumulation of

![Figures 5 and 6](image-url)
points in the low-velocity area of the source cell \((i,j)\), with few points being replaced into the high-velocity area, where convective transport is the greatest.

Although we normally expect points to be convected out of source cells, figure 6b also demonstrates the possibility that points may sometimes enter a source cell. This can also occur when two or more source cells of different strengths are adjacent to each other. An erroneous multiplication of points might then result if points that did not originate in a particular source cell are replaced when they in turn are convected out of that source cell. Therefore, points leaving a source cell are replaced only if they had originated in that source cell.

Hydraulic sinks also require some special treatment. Points will continually move into a cell representing a strong sink, but few or none will move out. To avoid the resultant crowding and stagnation of tracer points, any point moving into a sink cell is removed from the flow field after the calculations for that time increment have been completed. The numerical removal of points which enter
sink cells is analogous to the withdrawal of fluid and solute mass through the hydraulic sink. The combination of creating new points at sources and destroying old points at sinks will tend to maintain the total number of points in the flow field at a nearly constant value.

Both the flow model and the transport model assume that sources and sinks act over the entire cell area surrounding a source or sink node. Thus, in effect, heads and concentrations computed at source or sink nodes represent average values over the area of the cell. Part of the total concentration change computed at a source node represents mixing between the source water at one concentration and the ground water at a different concentration (eq 39). It can be shown from the relationship between the source concentration \( C'_{i,j,k} \) and the aquifer concentration \( C_{i,j,k-1} \), as indicated by equation 44, that the following constraints generally must be met in a source cell:

\[
C_{i,j,k} \leq C'_{i,j,k} \quad \text{for} \quad C'_{i,j,k} > C_{i,j,k-1} \quad (65a)
\]

and

\[
C_{i,j,k} \geq C'_{i,j,k} \quad \text{for} \quad C'_{i,j,k} < C_{i,j,k-1} \quad (65b)
\]

If it is assumed that the sources act over the area of the source cell and that there is complete vertical mixing, then these same constraints should also apply to all points within the cell. Because of the possible deviation of the concentrations of individual points within a source cell from the average concentration, the change in concentration computed at a source node \( \Delta C_{i,j,k} \) should not be applied directly to each of the points in the cell. Rather, at the end of each time increment the concentration of each point in a source cell is updated by setting it equal to the final nodal concentration. Although this may introduce a small amount of numerical dispersion by eliminating possible concentration variations within the area of a source cell, it prevents the adjustment of the concentration at any point in the source cell to a value that would violate the constraints indicated by equation 65.

In areas of divergent flow there may be a problem because some cells can become void of points where pathlines become spaced widely apart. This would result in a calculation of zero change in concentration at a node due to convective transport, although the nodal concentration would still be adjusted for changes caused by hydrodynamic dispersion (eq 28). Also, some numerical dispersion is generated at nodes in and adjacent to the cells into which the convective transport of solute was underestimated because of the resulting error in the concentration gradient. This might not cause a serious problem if only a few cells in a large grid became void or if the voiding were transitory (that is, if upgradient points were convected into void cells during later or subsequent time increments). Figure 6a illustrates radial flow, which represents the most severe case of divergent flow. Here it can be seen that when four points per cell are used to simulate convective transport, then in the numerical procedure four of the eight surrounding cells would erroneously not receive any solute by convection from the adjacent source. If eight points per cell were used initially, then at a distance of two rows or columns from the source only 8 of 16 cells would be on pathlines originating in the source cell. So, while increasing the initial number of points per cell would help, it is obvious that for purely radial flow, an impractically large initial number of points per cell would be required to be certain that at least one particle pathline passes from the source through every cell in the grid.

The problem of cells becoming void of particles can be minimized by limiting the number of void cells to a small percentage of the total number of cells that represent the aquifer. If the limit is exceeded, the numerical solution to the solute-transport equation is terminated at the end of that time increment and the "final" concentrations at that time are saved. Next the problem is reinitialized at the time of termination by regenerating the initial particle distribution throughout the grid and assigning the "final" concentrations at the time of termination as new "initial" concentrations for nodes and particles. The solution to the solute-transport
equation is then simply continued in time from this new set of "initial" conditions until the total simulation period has elapsed. This procedure preserves the mass balance within each cell but also introduces a small amount of numerical dispersion by eliminating variations in concentration within individual cells.

To help minimize the amount of numerical dispersion resulting from the regeneration of points, the program also includes an optimization routine that attempts to maintain an approximation of the previous concentration gradient within a cell. The optimization routine aims to meet the following constraints:

\[
\sum_{n=1}^{N_p} \frac{C_n^*}{N_p} = C_{i,j} \quad (66a)
\]

\[C_{i,j} \leq C_n^* \leq C_{i,m} \quad \text{for} \quad C_{i,j} \leq C_{i,m} \quad (66b)
\]

and

\[C_{i,m} \leq C_n^* \leq C_{i,j} \quad \text{for} \quad C_{i,j} \geq C_{i,m} \quad (66c)
\]

where

- \(C_n^*\) is the concentration of the \(n\)th point in cell \((i,j)\), \(M/L^3\);
- \(N_p\) is the total number of points initially placed in a cell; and
- \(C_{i,m}\) is the concentration at node \((l,m)\), which represents a cell adjacent to \((i,j)\) and on a line that starts at \((i,j)\) and extends through the coordinates of the point \((n)\) of interest, as illustrated in figure 7, \(M/L^3\).

Note that equation 66a simply indicates that a mass balance must be preserved in a cell regardless of the range in variation of point concentrations within the cell. Equations 66b and c indicate that the concentration of any point must lie between \(C_{i,j}\) and the concentration at the node adjacent to particle \(n\). The coordinates of the adjacent node would take on values of \(l=\pm i\) or \(l=\pm 1\) and \(m=\pm j\) or \(m=\pm 1\). For example, figure 7 shows that for point 2, the coordinates \((l,m)\) would equal \((i,j-1)\), while for point 3, \((l,m)\) would equal \((i+1,j-1)\). The optimization routine is written so that if equations 66a–c cannot be satisfied simultaneously for node \((i,j)\) within two iterations, then to avoid further computational delay all \(C_n^*\) are simply set equal to \(C_{i,j}\).

### Computer Program

The computer program serves as a means of translating the numerical algorithm into machine executable instructions. The purpose of this chapter is to describe the overall structure of the program and to present a detailed description of its key elements, thereby providing a link between the numerical methods and the computer code. We hope that this link will make it easier for the model user to understand and, if necessary, modify the program. The FORTRAN IV source program developed for this model is listed in attachment I and includes almost 2,000 lines. For reference purposes columns 73–80 of each line contain a label that is numbered sequentially within each subroutine. The definition of selected variables used in the program is presented in attachment II; this glossary therefore also serves as a key for relating the program variables.
to their corresponding mathematical terms. The computer program is compatible with many scientific computers; it has been successfully run on Honeywell, IBM, DEC, and CDC computers.

**General program features**

The program is segmented into a main routine and eight subroutines. The name and primary purpose of each segment are listed in Table 1. Each program segment will be described in more detail in later sections of this chapter.

<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>Control execution.</td>
</tr>
<tr>
<td>PARLOD</td>
<td>Data input and initialization.</td>
</tr>
<tr>
<td>ITERAT</td>
<td>Compute head distribution.</td>
</tr>
<tr>
<td>GENPT</td>
<td>Generate or reposition particles.</td>
</tr>
<tr>
<td>VELO</td>
<td>Compute hydraulic gradients, velocities,</td>
</tr>
<tr>
<td></td>
<td>dispersion equation coefficients, and time</td>
</tr>
<tr>
<td></td>
<td>increment for stable solution to transport</td>
</tr>
<tr>
<td></td>
<td>equation.</td>
</tr>
<tr>
<td>MOVE</td>
<td>Move particles.</td>
</tr>
<tr>
<td>CNCON</td>
<td>Compute change in chemical concentrations</td>
</tr>
<tr>
<td></td>
<td>and compute mass balance for transport model.</td>
</tr>
<tr>
<td>OUTPT</td>
<td>Print head distribution and compute mass balance</td>
</tr>
<tr>
<td></td>
<td>for flow model.</td>
</tr>
<tr>
<td>CHMOT</td>
<td>Print concentrations, chemical mass balance, and</td>
</tr>
<tr>
<td></td>
<td>observation well data.</td>
</tr>
</tbody>
</table>

The major steps in the calculation procedures are summarized in figure 8, which presents a simplified flow chart of the overall structure of the computer program. The flow chart illustrates that the tracer particles may have to be moved more than once to complete a given time step. In other words, the time step used to implicitly solve the flow equation may have to be subdivided into a number of smaller time increments for the explicit solution of the solute-transport equation. The maximum time increments allowable for the explicit calculations are computed automatically by the model. Thus, the model user cannot specify an erroneously large increment or an inefficiently small increment for solving the solute-transport equation. For transient flow problems, some discretion is still required in the specification of the initial time step and of the time-step multiplier, as discussed by Trescott, Pinder, and Larson (1976, p. 38-40).

The general program presented here is written to allow a grid having up to 20 rows and 20 columns. Because the numerical procedure requires that the outer rows and columns represent no-flow boundaries, the aquifer itself is then limited to maximum dimensions of 18 rows and 18 columns. If a problem requires a larger grid, then the appropriate arrays must be redimensioned accordingly. These arrays are contained in COMMON statements PRMK, HEDA, HEDB, CHMA, CHMC, and DIFUS, and in DIMENSION statements on lines C170, G200, H140, and I160.

The program allows the specification of one pumping well per node. The wells can represent injection (recharge) or withdrawal (discharge). If more than one well exists within the area of a cell, then the flux specified for that node should represent the net rate of injection or withdrawal of all wells in that cell. The model assumes that stresses are constant with time during each pumping period (NPMP). But the total number of wells, as well as their locations, flux rates, and source concentrations, may be changed for successive pumping periods. The program also allows the specification of observation wells at as many as five nodes in the grid. For nodes that are designated as observation wells, at the end of the simulation period or after every 50 time increments the model will print a summary table of the head and concentration at the previous time increments.

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 can save much time in the preparation of input data by easily equating each code number with a desired boundary condition, flux, or source concentration.
Figure 5.—Simplified flow chart illustrating the major steps in the calculation procedure.

Program segments

MAIN

The primary purpose of the MAIN routine is to control the overall execution sequence of the program. Subroutines for input, execution, and output are called from MAIN and the elapsed time simulated is compared with the desired total simulation period. Also, lines A500–A580 serve to store (or...
record) observation well data for transient flow problems.

Subroutine PARLOD

All input data are read through subroutine PARLOD. These data define the properties, boundaries, initial conditions, and stresses for the aquifer, as well as spatial grid and time-step factors. The values of many variables are also initialized here. After the data are read, some preliminary calculations are made, such as (1) determining time increments for the flow model (lines B780–B890), (2) computing the harmonic mean transmissivities in the x and y directions (B1670–B1800), (3) adjusting transmissivity for anisotropy (B1810–B1820), (4) computing iteration parameters (B1840–B1910 and B2880–B2980), and (5) checking for possible inconsistencies among the input data (B3140–B3290). A printout is also provided of all input data so that the data may be rechecked and each run identified.

Subroutine ITERAT

This subroutine solves a finite-difference approximation of the flow equation (eq 11) using an iterative ADI procedure. The matrix generated by the finite-difference approximation is solved using the Thomas algorithm, as described by von Rosenberg (1964, p. 113). Row calculations are made in lines C270–C610, and column calculations are made in lines C630–C970. The calculations are assumed to have converged on a solution if the maximum difference at all nodes between heads computed along rows and heads computed along columns is less than the specified tolerance. Convergence is checked on lines C940–C960. Note that here (for example, lines C380, C700, C980, and C1150) and in other subroutines the thickness array (THCK) is used to check whether a node is in the aquifer.

It should also be noted here that the flow model, as written, assumes that the transmissivity of the aquifer is independent of the head (or saturated thickness) and remains constant with time. If this assumption is not appropriate to the particular aquifer system being modeled, then the solution algorithm presented in this subroutine should be modified accordingly. For example, flow models published by Prickett and Lonnquist (1971, p. 43–45) and Trescott, Pinder, and Larson (1976) include such a modification.

All parameters involved in the calculation of heads are defined as double precision variables and all calculations involving these parameters are performed in double precision. The number of double precision variables and operations can be reduced significantly if the program is to be executed on a high-precision scientific computer, thereby improving the efficiency of the model by reducing computer storage requirements and execution time.

The iterative ADI procedure used to solve the finite-difference equations is not necessarily the best possible solution technique for all problems. For example, it may be difficult to obtain a solution using the iterative ADI procedure for cases of steady-state flow when internal nodes in the grid have zero transmissivity and for cases in which the transmissivity is highly anisotropic. In such cases, a strongly implicit procedure, such as the one documented by Trescott, Pinder, and Larson (1976), should be substituted for the solution algorithm contained in subroutine ITERAT.

Subroutine GENPT

The primary purpose of subroutine GENPT is to generate a uniform initial distribution of tracer particles throughout the finite-difference grid. This is done either at the start of a simulation period or at an intermediate time when too many cells have become void of particles. In the latter case, the program attempts to preserve an approximation of the previous concentration gradient within each cell (lines D1420–D2040).

The placement of particles is accomplished in lines D510–D1410. The program allows the placement of either four, five, eight, or nine particles per cell. Of course each option will result in a slightly different geometry
and density of points, as illustrated by figure 9. The most regular or uniform patterns are produced when four or nine particles per cell are specified. If a different number of particles per cell or a different placement geometry are desired, this subroutine could be modified accordingly.

As particles are moved or convected through the grid during the calculation procedure, there is a need to remove particles at fluid sinks and create particles at fluid sources. A buffer array (called LIMBO) is created on lines D430-D480 that contains particles that can be added later to the grid at sources and that also contains space to store particles removed at sinks or discharge boundaries.

Subroutine VELO

Subroutine VELO accomplishes three objectives. First, it computes the flow velocities at nodes and on cell boundaries by solving equations having the form of equations 12 and 13. The velocities are computed on lines E420-E680. Second, the dispersion equation coefficients are calculated. These coefficients represent terms factored out of equations 37 and 38, as follows:

\[
\text{DISP}(\text{IX},\text{IY},1) = \frac{(bD_{se})_{(i+,j)}}{(\Delta x)^2} \quad (67a)
\]
\[
\text{DISP}(\text{IX},\text{IY},2) = \frac{(bD_{sv})_{(i,j+,j)}}{(\Delta y)^2} \quad (67b)
\]
\[
\text{DISP}(\text{IX},\text{IY},3) = \frac{(bD_{sv})_{(i+,j+)}}{4\Delta x\Delta y} \quad (67c)
\]
\[
\text{DISP}(\text{IX},\text{IY},4) = \frac{(bD_{sy})_{(i,j+,j+)}}{4\Delta x\Delta y} \quad (67d)
\]

Note that each dispersion coefficient \((D_{se}, D_{sv}, D_{sv}, D_{sy})\) is computed on cell boundaries using the relationships expressed in equations 8-10. Therefore, the equation coefficients computed by equation 67 are stored as forward values from the indicated node in the DISP array. Third, this subroutine computes (on lines E1050-E1240 and E1800-E1930) the minimum number of particle moves (NMOV) required to solve the transport equation for the given time step so that the maximum time increment for the transport equation solution will not exceed any of the criteria indicated by equations 43, 49, 58, and 59.

Subroutine MOVE

Although this subroutine has only one main function, which is to move the tracer particles in accordance with equations 22 and 23, it is the longest and perhaps the most complex segment of the program. The complexities are mainly introduced by the treatment of particles at the various types of boundary conditions. To help illustrate the calculation procedure followed within subroutine MOVE, a flow chart is presented in figure 10. The numbers in the flow chart indicate the corresponding lines in subroutine MOVE where the indicated operation is executed.

If a node represents a fluid source or sink, then particles must be respectively created or destroyed in these cells. If the value of pumpage (REC) at a node does not equal zero, then the node is assumed to represent either a fluid source (for \(\text{REC}<0\)) or a fluid sink (for \(\text{REC}>0\)). Recharge or discharge can also be represented by the RECH array. But it is assumed that this type of flux is sufficiently diffuse so that it does not induce areas or points of strongly divergent or convergent flow and therefore particles need not be created or destroyed at these nodes. Note that here and in other subroutines the presence of a constant-head boundary is tested by checking the value of leakance (VPRM).
Figure 10.—Generalized flow chart of subroutine MOVE. Numbers indicate line numbers where the operation is executed.
at each node. If VPRM exceeds 0.09, it is assumed that the node represents a constant-head boundary condition and is treated as a fluid source or sink accordingly. At a constant-head node, the difference in head between the aquifer and the source bed is used to determine whether the node represents a fluid source or sink (for example, lines F2500–F2520).

**Subroutine CNCON**

This subroutine computes the change in concentration at each node and at each particle for the given time increment. Equation 39, which denotes the change in concentration resulting from sources, divergence of velocity, and changes in saturated thickness, is solved on lines G350–G610. On the G520 the value of the storage coefficient is checked to determine whether the aquifer is confined or unconfined. It assumes that if $S < 0.005$, then the aquifer is confined and $\partial h / \partial t = 0$. If $S = 0.005$, the model assumes that $\partial h / \partial t = \partial h / \partial t$. If this criterion is not appropriate to a particular aquifer system, then line G520 should be modified accordingly. The change in concentration caused by hydrodynamic dispersion is computed on lines G640–G770 as indicated by equations 37 and 38.

The nodal changes in concentration caused by convective transport are computed on lines G850–G940. The number of cells that are void of particles at the new time level are also counted in this set of statements on lines G880–G910, and then compared with the critical number of void cells (NZCRIT) to determine if particles should be regenerated at initial positions before the next time level is started (lines G960–G1020).

The new (time level $k$) concentrations at nodes are computed on the basis of the previous concentration at time $k-1$ and the change during $k-1$ to $k$. The adjustment at nodes is accomplished on lines G1060–G1180, while the concentration of particles is adjusted on lines G1210–G1360.

A mass balance for the solute is next computed (lines G1400–G1730) at the end of each time increment. In computing the mass of solute withdrawn or leaking out of the aquifer at fluid sinks, the concentration at the sink node is assumed to equal the nodal concentration computed at time level $k-1$.

**Subroutine OUTPT**

This subroutine prints the results of the flow model calculations. When invoked, the subroutine prints (1) the new hydraulic head matrix (lines H190–H260), (2) a numeric map of head values (H300–H390), and (3) a drawdown map (H510–H710). This subroutine also computes a mass balance for the flow model and estimates its accuracy (II420–II820). A mass balance is performed both for cumulative volumes since the initial time and for flow rates during the present time step. The mass balance results are printed on lines II840–II930.

**Subroutine CHMOT**

This subroutine prints (1) maps of concentration (lines I250–I380), (2) change in concentration from initial conditions (I440–I580), and (3) the results of the cumulative mass balance for the solute (I670–I860). The accuracy of the chemical mass balance is estimated on lines I610–I660 using equations 62 and 64. The former is not computed if there was no change in the total mass of solute stored in the aquifer. The latter is not computed if the initial concentrations were zero everywhere. Lines I890–I1140 serve to print the head and concentration data recorded at observation wells. These data are recorded after each time step for a transient flow problem and after each particle movement for a steady-state flow problem. The data are printed after every 50 time increments and at the end of the simulation period.

**Evaluation of Model**

**Comparison with analytical solutions**

The accuracy of the numerical solution to the solute-transport equation can be evalu-
ated in part by analyzing relatively simple problems for which analytical solutions are available and then comparing the numerical calculations with the analytical solution. Figure 11 presents such a comparison for a problem of one-dimensional steady-state flow through a homogeneous isotropic porous medium. The analytical solution is obtained with the following equation presented by Bear (1972, p. 627):

\[
\frac{C(x,t) - C_o}{C_1 - C_o} = \frac{1}{2} \text{erfc} \left( \frac{x - qt/\epsilon}{\sqrt{4D_\epsilon t}} \right)
\]

where \( \text{erfc} \) is the complimentary error function, and

\[ q = \frac{q_s}{V} \] is the specific discharge, \( LT^{-1} \).

Bear (1972, p. 627) shows that equation 68 is subject to the following initial conditions:

\[
t \leq 0, \quad -\infty < x < 0, \quad C = C_0
\]

\[
0 \leq x < +\infty, \quad C = C_1
\]

and to the following boundary conditions:

\[
t > 0, \quad x = \pm \infty, \quad \frac{\partial C}{\partial x} = 0
\]

\[
x = +\infty, \quad C = C_1
\]

\[
x = -\infty, \quad C = C_0.
\]

The general computer program presented in this report was modified in three simple ways for application to a problem equivalent to the one for which the analytical solution was derived. First, the program's arrays were redimensioned to 3 by 50 rather than 20 by 20. The aquifer (or column of porous medium) was thus represented by a 1-by-48 array of nodes. A grid spacing of 10 ft (3.05 m) was used. Second, the flow velocity was specified as a constant value, rather than being computed implicitly on the basis of hydraulic gradients and hydraulic conductivity. Third, the first (upstream) node of the aquifer was specified as a constant-concentration boundary, so that the concentration at node (2,2) was always equal to \( C_o \) of

Figure 11.—Comparison between analytical and numerical solutions for dispersion in one-dimensional, steady-state flow.
equation 68. In the analysis of one-dimensional test problems, it was assumed that porosity equals 0.35, velocity equals \(3.0 \times 10^{-4}\) ft/s \((9.1 \times 10^{-5}\) m/s\), and time equals 10.0 days.

As shown in figure 11, comparisons between the analytical and numerical solutions were made for two different values of dispersivity. For the higher dispersion there was essentially an exact agreement between the two curves. In the case of low dispersion, there is a very small difference at some nodes between the concentrations computed analytically and those computed numerically. This difference is caused primarily by the error in computing the concentration at a node as the arithmetic average of the concentrations of all particles located in that cell. This is not considered to be a serious problem since this error is not cumulative. Also note in the case of low dispersion that the grid spacing \((10\) ft or \(3.05\) m) was coarse relative to the width of the breakthrough curve between concentrations of 0.05 and 0.95. Nevertheless, the numerical model still accurately computed the shape and position of the front.

In computing the numerical solutions shown in figure 11 the program was executed using nine particles per cell and with CELDIS = 0.50 \((\gamma\) in equations 54–55). The 10-day simulation required 52 time increments and used about 40 seconds of cpu on a Honeywell 60/68 computer.

An analytical solution is also available for the problem of plane radial flow in which a well continuously injects a tracer at constant rate \(q_w\) and constant concentration \(C_o\). Bear (1972, p. 638) indicates that the following equation is appropriate for this problem (although there are some limitations discussed by Bear):

\[
C = C_o \cdot \text{erfc} \left( \frac{r^2/2 - Gt}{\sqrt{4/3a_r}} \right) 
\]

(69)

where

\[
G = \frac{q_w}{2\pi r_b}; \quad \text{and} \quad \bar{r} = (2Gt)^{1/2} \text{ is the average radius of the body of injected water, } L.
\]

Again, the general computer program had to be somewhat modified to permit a suitable comparison to be made between the analytical solution and the numerical model. One change involved the direct calculation of velocity at any point based on its distance from the well using the following equation:

\[
V = \frac{q_w}{2\pi \bar{r} b}. \quad (70)
\]

The other significant change was made in subroutine GENPT to allow the initial placement of 16 particles per cell, rather than the present maximum of 9. In the analysis of test problems for radial flow, it was assumed that porosity equals 0.35, the injection rate \(q_w\) equals 1.0 ft\(^3\)/s \((0.028\) m\(^3\)/s\), saturated thickness equals 10.0 ft \((3.05\) m\), and longitudinal dispersivity equals 10.0 ft \((3.05\) m\).

The application of the method of characteristics, which was written for two-dimensional Cartesian coordinates, to a problem involving radially symmetric divergent flow represents a severe test of the model. Nevertheless, it can be seen in figure 12 that there is good agreement between the analytical and numerical solutions after both relatively short and long times. However, the presence of some numerical dispersion is evident, particularly for the longer time. The numerical dispersion is introduced in part during the regeneration of particles after the number of cells void of particles has exceeded the critical number. The geometry of initial particle placement minimized this problem in cells that lay in the same row or column of the grid as the injection well. The circles in figure 12, which indicate concentration values computed at these nodes, agree closely with the analytical solution. The greatest errors occur at nodes on radii from the injection well that are neither parallel to nor 45° from the main axes of the grid. These results indicate that this Cartesian coordinate model is not best suited for application to purely radial flow problems. However, if radially divergent flow is limited to areas of several
rows and columns within a more uniform regional flow field, the model will accurately compute concentration distributions. To apply the method of characteristics to a problem of plane radial flow, it would be best to rewrite the program in a system of radial coordinates, which should improve the accuracy for those problems to the same order shown in figure 11 for the analysis of one-dimensional flow.

**Mass balance tests**

The accuracy and precision of the numerical solution can also be partly evaluated by computing the magnitude of the error in the mass balance. The mass balance error will depend on the nature of the problem and will vary from one time step to the next. During the development of the program, the model was applied to a variety of hypothetical solute-transport problems to assure its flexibility, transferability, and accuracy under a wide range of conditions. To illustrate the range in mass balance errors that might be expected and some of the factors that affect it, several of these problems are presented here.

**Test problem 1—spreading of a tracer slug**

The first test described here was designed to evaluate the accuracy of simulating the processes of convective transport and dispersion independent of the effects of chemical sources. Thus, a slug of tracer was initially placed in four cells of a grid whose boundary conditions generated a steady-state flow field that was moderately divergent in some places and moderately convergent in other places, as illustrated in figure 13. The aquifer was assumed to be homogeneous and isotropic. Because flow was assumed to be in steady state, the storage coefficient was set equal to 0.0. The parameters used to define problem
1 are listed in Table 2. The slug of known mass was then allowed to spread down-gradient for a period of 2.0 years.

Table 2.—Model parameters for test problem 1

<table>
<thead>
<tr>
<th>Aquifer properties</th>
<th>Numerical parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K=0.005$ ft/s</td>
<td>$\Delta x=900$ ft</td>
</tr>
<tr>
<td>(1.5×10^{-2} m/s)</td>
<td>(274 m)</td>
</tr>
<tr>
<td>$b=20.0$ ft</td>
<td>$\Delta y=900$ ft</td>
</tr>
<tr>
<td>(6.1 m)</td>
<td>(274 m)</td>
</tr>
<tr>
<td>$S=0.0$</td>
<td>CELDIS=0.49</td>
</tr>
<tr>
<td>$\varepsilon=0.30$</td>
<td>NPTPND=9</td>
</tr>
<tr>
<td>$a_t/a_s=0.30$</td>
<td></td>
</tr>
</tbody>
</table>

The model first computed a steady-state head distribution, shown in Figure 13, and velocity field. The model required 12 time increments (or particle movements) to simulate a 2.0-year period. The model was run to simulate conditions of no dispersion ($a_t=0.0$ ft) as well as moderate dispersion ($a_t=100$ ft or 30.5 m). The mass balance error computed using equation 64 is shown in Figure 14 for both conditions. In these tests the error averages 1.9 percent and is always within a range of ±8 percent. Much of the error is related to the calculation of nodal concentrations based on the arithmetic mean of particle concentrations in each cell. When a particle moves across a cell boundary, its area of influence shifts entirely from the first node to the second. Thus, depending on the local density of points and local concentration gradients, the use of an arithmetic mean to compute nodal concentrations may give too much weight to some particles and too little weight to others. The use of a weighted mean, in which the weighting factor is a function of the distance between a node and a particle, reduced the error to some degree. But the improvement in precision was small compared with the increase in computational requirements, so this algorithm was not included in the general program. Because the error caused by using an arithmetic mean is not cumulative, it is not considered a serious
Figure 14.—Mass balance errors for test problem 1.

Figure 15.—Grid, boundary conditions, and flow field for test problem 2.
problem. Furthermore, figure 14 shows that the error decreases for a higher dispersivity because dispersion smooths out sharp fronts and minimizes strong concentration gradients.

**Test problem 2—effects of wells**

The second problem was designed to evaluate the application of the model to problems in which the flow field is strongly influenced by wells. The grid and boundary conditions used to define this problem are illustrated in figure 15. The problem consists of one injection well and one withdrawal well, whose effects are superimposed on a regional flow field controlled by two constant-head boundaries. The parameters for problem 2 are defined in table 3. The aquifer was also assumed to be homogeneous and isotropic. The model simulated a period of 2.4 years and assumed steady-state flow.

The model required 18 time increments (or particle movements) to simulate a 2.4-year period of solute transport. Problem 2 was also evaluated for conditions of no dispersion \((\alpha_L=0.0\text{ ft})\) as well as moderate dispersion \((\alpha_L=100\text{ ft or }30.5\text{ m})\). The mass balance error was computed using equation 62 and is shown in figure 16 for both conditions. The average of the 36 values shown in figure 16 is \(-0.06\%\); the error always falls within the range of \(\pm8\%\). It can be seen that in this case the errors are essentially coincident for almost 1 year, after which the error appears to be dependent on the magnitude of dispersion. However, the model output showed that when \(\alpha_L=100\text{ ft (30.5 m)}\), the leading edge of the breakthrough curve (or chemical front) reaches the constant-head sink just prior to 1.0 year. When \(\alpha_L=0.0\text{ ft}\), the leading edge of the breakthrough curve still had not entered the constant-head sink after 2.4 years. Because the two curves in figure 16 are essentially coincident prior to 1.0 year, it thus appears that the divergence of the two curves is not caused directly by the difference in dispersivity. Rather, it is related to the difference in arrival times at the hydraulic sinks and is a direct effect of the manner in which con-

---

**Table 3.—Model parameters for test problems 2 and 3**

<table>
<thead>
<tr>
<th>Aquifer properties and stresses</th>
<th>Numerical parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>(K=0.005\text{ ft/s (1.5x10^{-5} m/s)})</td>
<td>(\Delta z=900\text{ ft (274 m)})</td>
</tr>
<tr>
<td>(b=20.0\text{ ft (6.1 m)})</td>
<td>(\Delta y=900\text{ ft (274 m)})</td>
</tr>
<tr>
<td>(S=0.0)</td>
<td>CELDIS=0.50</td>
</tr>
<tr>
<td>(\varepsilon=0.30)</td>
<td>NPTPND=9</td>
</tr>
<tr>
<td>(\alpha_L/\alpha_R=0.30)</td>
<td></td>
</tr>
<tr>
<td>(C_L=100.0)</td>
<td></td>
</tr>
<tr>
<td>(C_0=0.0)</td>
<td></td>
</tr>
<tr>
<td>(q_w=1.0\text{ ft}^3/\text{s (0.028 m$^3$/s)})</td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure 16.—Mass balance errors for test problem 2.**
centrations are computed at sink nodes and (or) the method of estimating the mass of solute removed from the aquifer at sink nodes during each time increment.

**Test problem 3—effects of user options**

In addition to the input options that control the form or frequency of the output, there are two execution parameters that must be specified by the user and influence the accuracy, precision, and efficiency (or computational cost) of the solution to a particular problem. These execution parameters are the initial number of particles per node (NPTPND) and the maximum fraction of the grid dimensions that particles are allowed to move ($\gamma$ in equations 54–55 or CELDIS in the program). The third test problem was designed to allow an evaluation of both of these parameters. As illustrated in figure 17, this problem consists of one withdrawal well located in a regional flow field that is controlled by two constant-head boundaries. The contamination sources are three central nodes along the upgradient constant-head boundary. The model parameters for test problem 3 are the same as for test problem 2, as listed in table 3. However, for test problem 3 solutions were obtained using a range in values for CELDIS and NPTPND.

The solution to this problem was found to be sensitive to the density of tracer particles used in the simulation. Figure 18 shows how the error in the mass balance varied with time for cases of NPTPND equal to 4, 5, 8, and 9. Table 4 lists the execution time and the mean and standard deviation of the mass balance error for each case. These data clearly indicate that the accuracy and precision
Figure 18.—Effect of NPTPND on mass balance error for test problem 3; CELDIS=0.50 in all cases.

Table 4.—Effect of NPTPND on accuracy, precision, and efficiency of solution to test problem 3

<table>
<thead>
<tr>
<th>NPTPND</th>
<th>CPU-seconds</th>
<th>Mass balance error (percent)</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>12.8</td>
<td>1.49</td>
<td>5.33</td>
</tr>
<tr>
<td>5</td>
<td>14.0</td>
<td>.90</td>
<td>2.29</td>
</tr>
<tr>
<td>8</td>
<td>17.9</td>
<td>.48</td>
<td>1.53</td>
</tr>
<tr>
<td>9</td>
<td>19.2</td>
<td>.26</td>
<td>.69</td>
</tr>
</tbody>
</table>

1 The program was executed on a Honeywell 60/66 computer; CELDIS=0.50.

The solution are directly proportional to particle density, while the efficiency of the solution is inversely related to NPTPND. In other words, a better solution will cost more. It is important to note that the oscillations or scatter shown in figure 18 decrease with time and that there is essentially no difference among the solutions and among the mass balance errors for times greater than about 1.5 years.

Next the effect of CELDIS (or γ) was evaluated for test problem 3 by setting NPTPND=9 and running the model with several possible values of CELDIS. Figure 19 shows how the error in the mass balance varied with time for cases of CELDIS equal to 0.25, 0.50, 0.75, and 1.00. Table 5 lists the execution time and the mean and standard deviation of the mass balance error for each case. These data indicate that the relationship between CELDIS and the mass balance error is not as simple and straightforward as for NPTPND. It is apparent that the results for 0.50, 0.75, and 1.00 are similar, and of these, the results for CELDIS=0.50 ap-
Figure 19.—Effect of CELDIS on mass balance error for test problem 3; NPTPND=9 in all cases.

pear to be the best. However, when CELDIS was reduced to 0.25, the error oscillated strongly for about 1.5 years before apparently converging to a small error within the range of the other curves. This oscillation occurred because the maximum distance a particle could move (25 percent of the grid dimensions) was less than the spacing between particles (33 percent of the grid dimensions for NPTPND=9). Thus, convective transport across the boundaries of cells could not be adequately represented for any single time step in those parts of the grid where the concentration was changing significantly with time. But over two successive time increments the error would average out to a minimum. As the contaminated area increases in size over time, the error in computed concentrations at cells near the front (that is, in areas of steep concentration gradient) becomes an increasingly smaller percentage of the total mass of solute present in the aquifer. Hence, the mass balance error generally tends to approach a minimal range with time for these types of problems.

The effects of NPTPND and CELDIS on the mass balance error are problem-dependent. In problems for which CELDIS is not the limiting stability criterion, varying CELDIS will have no effect on the solution. Because of the possible tradeoff between accuracy and efficiency, it is recommended in general that the model user specify NPTPND as 4 or 5 and CELDIS as 0.75 to 1.0 for runs made during the early stages of model calibration when frequent runs are made and maximum efficiency is desired. For final runs when maximum accuracy is desired, set NPTPND equal to 9 and CELDIS equal to 0.50.

Possible program modifications

The program presented here represents a basic and general solute-transport model. Some program modifications may be desirable or even necessary to allow the model to be applied efficiently to a particular field problem. Some changes might require only minor adjustments, while others might involve major rewriting of the program. The purpose of this section is to discuss some of the modifications that might commonly be considered, and that might be incorporated into the present basic model, rather than using an entirely different solution technique.
Coordinate system and boundary conditions

After the finite-difference grid is designed, the first program modification that should be made is to modify the array dimensions for the specific grid used. This will permit the most efficient use of computer storage. The array sizes should be set equal to NX, NY, and NPMAX, which are specified on Input Card 2. The maximum number of particles, NPMAX, may be computed from the following equation:

\[ N_{\text{PMAX}} = (\text{NX}-2) (\text{NY}-2) (\text{NPTPND}) + (N_s) (\text{NPTPND}) + 250 \]  

(71)

where

\( N_s \) is the number of nodes that represent fluid sources, either at wells or at constant-head cells.

The values of NX and NY should be substituted for the 20-by-20 arrays contained in COMMON statements PRMK, HEDA, HEDB, CHMA, CHMC, and DIFUS, and in DIMENSION statements on lines C170, G200, H140, and I160. The value of NPMAX should replace 3200 in the PART array in all the CHMA COMMON statements.

Although this program is designed for application to two-dimensional areal flow problems, it can be applied directly to two-dimensional cross sections. In this case the \( z \)-coordinate would replace the \( y \)-coordinate. Then the user would have to assume and specify unit width (THCK array) for \( \Delta y \) and substitute hydraulic conductivity for transmissivity in data set 3 of attachment III. If the problem involves transient flow, then specific storage \( (S_s) \) should be substituted for the storage coefficient. Also, if recharge or discharge is to be specified through the RECH array (data set 5), values should be divided by the thickness of the layer \( (\Delta z) \) to reduce the dimensionality of the stress rate to \( (T^{-1}) \) rather than \( (LT^{-1}) \) as indicated in the documentation. In applying the cross-sectional model to a field problem it is important that conditions meet the inherent assumption that there exist no significant components of flow into or out of the plane of the section. Because this assumption would probably be impossible to meet in the vicinity of a pumping well, the use of the REC array (data set 2) should usually be limited to representing special or known-flux boundary conditions.

The program can also be applied directly and simply to one-dimensional problems. In this case one of the dimensions (NX or NY) should be reduced to a value of 3, of which the outer two are used to represent the no-flow boundaries around the one-dimensional row or column.

The most complex type of change would involve rewriting the program for application to other than a two-dimensional rectangular grid. One possibility includes problems of flow to or from wells in which radial symmetry can be assumed. This would allow variables to be expressed in terms of \( r-z \) coordinates. Another possibility is to simulate three-dimensional flow in \( x-y-z \) coordinates. A three-dimensional finite-difference flow model is available (Trescott, 1975) and would be compatible with the method-of-characteristics solution to the solute-transport equation.

It is sometimes convenient to separately associate certain parts of the grid or certain boundary conditions with corresponding field conditions or hydrologic units. The analysis of flow patterns and water-quality changes may then be aided by performing separate mass balances (or budgets) for each characteristic type of node. The nodal types or zones can be conveniently identified through the NODEID array. Then the mass balance routines in subroutines CNCON and (or) OUTPT would have to be modified to tally fluxes separately for each NODEID; for an example, see Konikow (1977). Similarly, if a coupled stream-aquifer system is being considered, a separate subroutine may be added to route streamflow downstream and progressively account for ground-water gains and losses and for tributary inflow or diversions. An example of such a modification is discussed by Konikow and Bredehoeft (1974).

For certain types of problems it may be desirable to be able to specify a constant-concentration boundary condition. The pro-
gram could be modified to allow this by using a predetermined value or range in values of NODEID to identify this type of boundary. Then a statement could be added between lines G1090 and G1100 to reset the concentration at the node equal to the constant concentration where this condition is specified. The value of the constant concentration can be stored in the CNRECH array. Note that the mass balance calculation as presently written will not account for the mass of solute added or removed at a constant-concentration boundary.

**Basic equations**

The basic equations that are solved by this model were derived under a number of limiting assumptions. Some of these assumptions can be overcome through modifications of the basic equations and corresponding changes in the program.

The program assumes that molecular diffusion is negligible. But if it is necessary to consider the process of molecular diffusion in a particular problem, the coefficient of hydrodynamic dispersion \( D_h \) can be redefined as the sum of the coefficient of mechanical dispersion, which is defined by the right side of equation 5, and a coefficient of molecular diffusion. The consequent program modification would have to be made only in subroutine VELO (lines E1280-E1680).

The solute-transport equation can also be modified to include the effects of first-order chemical reactions, as was done by Robertson (1974). The reaction term could be included in the right side of equation 39. The corresponding program modification would be required in subroutine CNCON.

In certain problems the range in concentrations may be so great that the dependence of fluid properties, such as density and viscosity, on the concentration may have to be considered because of the dependence of fluid flow on variations in fluid properties. In this case the flow equation (eq 1) would have to be rewritten in terms of fluid pressure, rather than hydraulic head, such as equation 15 of Bredehoeft and Pinder (1973, p. 197). Then the program can be modified to iterate between the solutions to the flow and solute-transport equations if the change in fluid properties at any node exceeds some criterion during one time increment.

The flow equation can also be modified for application to unconfined aquifers in which the saturated thickness is a direct function of water-table elevation. This would require the inclusion of steps in subroutine ITERAT to correct the transmissivity for changes in saturated thickness. Such a feature is included in the two-dimensional flow model documented by Trescott, Pinder, and Larson (1976).

**Input and output**

The input and output formats have been designed for flexibility of use and general compatibility with the analysis of a variety of types of flow problems. If any of the formats are not suitable for use with a particular problem, they should be modified accordingly. All input formats are described in attachment III and contained in subroutine PARLOD in the program.

It has been assumed that several aquifer parameters are constant and uniform in space, such as storage coefficient, effective porosity, and dispersivity. If any of these are known to vary in space, they should be redefined as two-dimensional arrays. Then statements to allow these arrays to be read into the program should be added to subroutine PARLOD. Similarly, values of leakage and source concentrations (CNRECH) are only read in data set 7, where values can be associated only with a limited number of unique node identification codes. If the variations of these parameters are known on a more detailed scale, then they too can be read as additional data sets by adding appropriate statements to subroutine PARLOD. For example, a typical sequence of statements for reading one data set is represented by lines B2650-B2750, where the initial water-table elevations (data set 8) are read. This sequence of statements can then be replicated for reading in a different data set and inserted into subroutine PARLOD.
A labeled listing of the input data deck for test problem 3 is provided in attachment IV. This example illustrates the use of the data input formats specified in attachment III and shows that only a few data cards are required by the model to simulate a relatively simple problem. This example will also allow the user to verify that his program deck and computer yield essentially the same results as obtained by the documented program. Thus, selected parts of the output for test problem 3 are included in attachment V. Not all of the printed output from test problem 3 has been duplicated in attachment III. Instead, it contains only a sufficient selection to illustrate the type and form of output provided by the model, as well as to allow the user to compare his calculated values of critical parameters, such as head, velocity, and concentration, with the values computed by the documented model.

Conclusions

The model presented in this report can simulate the two-dimensional transport and dispersion of a nonreactive solute in either steady-state or transient ground-water flow. The program is general and flexible in that it can be readily and directly applied to a wide range of types of problems, as defined by aquifer properties, boundary conditions, and stresses. However, some program modifications may be required for application to specialized problems or conditions not included in the general model.

The accuracy of the numerical results can be evaluated by comparison with analytical solutions only for relatively simple and idealized problems; in these cases there was good agreement between the numerical and analytical results. Mass balance tests also help to evaluate the accuracy and precision of the model results. The error in the mass balance is generally less than 10 percent. The range in mass balance errors is commonly the greatest during the first few time increments, but tends to decrease and stabilize with time. For some problems the accuracy and precision of the numerical results may be sensitive to the initial number of particles placed in each cell and to the size of the time increments, as determined by the stability criteria for the solute-transport equation. The results of several numerical experiments suggest that the accuracy and precision of the results are essentially independent of the magnitude of the dispersion coefficient, and comparable accuracies are attained for high, low, or zero dispersivities.

References Cited


COMPUTER PROGRAM AND RELATED DATA
MODEL OF SOLUTE TRANSPORT IN GROUND WATER

Attachment I

FORTRAN IV Program Listing

********************************************************************************************

DOUBLE PRECISIONDMIN,DEXP,DLOG,DABS
REAL *8TMRX,VPRM,HIrHR,HC,HK,WT,RECH,TIM,AOPT,TITLE
REAL *8XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR
REAL *8TINT,ALPHAl,ANITP
COMMON /PRMI/ NTIM,NPHP,NPNT,NITP,N,NX,NY,NP,NREC,INT,NNX,NNY,NUMO
1BS,NMOV,NPMAX,ITMAX,NZCRIT,IPRNT,NPTPND,NPNTM,NNPNTL,NNPNTD,NN
2NPCHN,NPDEL
COMMON /PRMI/ NODEID(20,20),NPCELL(20,20),LIMBO(500),IXOBS(5),IYOB
1S(5)
COMMON /HEDA/ THCK(20,20),PERM(20,20),TMW(5,50),TMOS(50),ANFCTR
COMMON /HEDA/ TMRX(20,20,2),VPRM(20,20),HI(20,20),HR(20,20),HC(20,20)
120,WT(20,20),RECH(20,20),REC(20,20),RECH(20,20),TIM(100),AOPT(20,20),T
2ITLE(10),XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR
COMMON /CHMA/ PART(5,320),CONC(20,20),TMN(5,50),VX(20,20),VY(20,20)
120,CONT(20,20),CNRECH(20,20),POROS,SMICH,BETA,TIMV,STORM,STORM
21,CMSIN,CMSOUT,FLMIN,FLMGT,SUMIO,CELDS,DLTRAT,CE

---LOAD DATA---

INT=0
CALL PARLOD
CALL GENPT

---START COMPUTATIONS---

---COMPUTE ONE PUMPING PERIOD---
DO 150 INT=1,NPHP
IF (INT.GT.1) CALL PARLOD
---COMPUTE ONE TIME STEP---
DO 130 N=1,NTIM
IPRNT=0
---LOAD NEW DELTA T---
TINT=SUMT-PYR*(INT-1)
TDEL=DMIN(TIM(N),PYR-TINT)
SUMT=SUMT+TDEL
TIM(N)=TDEL
REMN=MOD(N,NPHP)

---STORE OBS. WELL DATA FOR TRANSIENT FLOW PROBLEMS---
IF (REMN.EQ.0 .OR .N.EQ.NTIM) CALL OUTPT
CALL VELO
CALL MOVE

---STORE OBS. WELL DATA FOR TRANSIENT FLOW PROBLEMS---
IF (S.EQ.0 .OR .N.EQ.NTIM) CALL OUTPT
CALL VELO
CALL MOVE

150 CONTINUE
TECHNIQUES OF WATER-RESOURCES INVESTIGATIONS

FORTRAN IV program listing—Continued

---OUTPUT ROUTINES---
120 IF (REM EQ.0.0 .OR. NTIM .OR. MOD(N,50).EQ.0) CALL CHMOT
130 CONTINUE

---SUMMARY OUTPUT---
140 CONTINUE
IPRNT=1
CALL CHMOT
STOP

SUBROUTINE PARLOD
DOUBLE PRECISION DMIN,DEXP,DLOG, DABS
REAL *8 THXVPRM,HR,HK,WT,REC,RECH,TIM,AOPT,TITLE
REAL *8 XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,PRINT,TMIN,PYR
REAL *8 ALPHA,ANIT
COMMON /PRHI/ NTIH,NPMP, NPNT, NITP, NX, NY, NPBMFX, NPNTM, NPNTVL, NPNTD, N
2PNCHV, NPDELC
COMMON /PRMK/ NODEID(20,20), NPCELL(20,20), LIMBO(500), IXOB(5), IYOB(5)
COMMON /HEDA/ THCK(20,20), PERM(20,20), TMWL(5,50), TMBS(50), ANFCTR
COMMON /HEDB/ TMRX(20,20,2), VPRM(20,20), HI(20,20), HR(20,20), HC(20,20)
120, HK(20,20), WT(20,20), REC(20,20), RECH(20,20), TIM(100), AOPT(20,20)
2TITLE(10), XDEL, YDEL, S, AREA, SUMT, RHO, PARAM, TEST, TOL, PRINT, TMIN, PYR
COMMON /CHMA/ PART(3,3200), CONC(20,20), TMCN(5,50), VX(20,20), VXY(20,20)
120, CONINT(20,20), CNRECH(20,20), POROS, SUMTCH, BETAIN, TIMV, STORM, STORM
21, CMSIN, CMSOUT, FLMIN, FLMOT, SUMIO, CELDIS, DLTRAT, CSTOREM
COMMON /BALM/ TOLQ
COMMON /XINV/ DXINV, DYINV, ARINV, PORINV
COMMON /CHMC/ SUMC(20,20), VXBDY(20,20), VYBDY(20,20)

IF (INT GT 1) GO TO 10
WRITE (6,750)
READ (5,720) TITLE
WRITE (6,730) TITLE

---INITIALIZE TEST AND CONTROL VARIABLES---
STORM=0.0
TEST=0.0
TOLQ=0.0
SUMT=0.0
SUMTCH=0.0
INT=0
IPRNT=0
NCA=0
N=0
IMOV=0
NMOV=0

---LOAD CONTROL PARAMETERS---
READ (5,740) NTIM, NPMP, NX, NY, NPMAX, NPNT, NITP, NUMBS, ITMAX, NREC, NPNTM, NPNTVL, NPNTD, NPNCHV, NPDELC
READ (5,800) PINT, TOL, POROS, BETAIN, TIMV, TINIT, XDEL, YDEL, DLTRAT, CEL
1DIS, ANFCTR
PYR=PINT*86400.0*365.25
NX=NX-1
MODEL OF SOLUTE TRANSPORT IN GROUND WATER

FORTRAN IV program listing—Continued

**---PRINT CONTROL PARAMETERS---**
WRITE (6,760)  
IF (NPTPND.LT.4.OR.NPTPND.GT.9.OR.NPTPND.EQ.6.OR.NPTPND.EQ.7) WRITE (6,880)
1PDEL  
IF (NPNTMV.EQ.0) NPNTMV=999
GO TO 20  
C **---READ DATA TO REVISE TIME STEPS AND STRESSES FOR SUBSEQUENT PUMPING PERIODS---**
10 READ (5,1060) ICHK  
IF (ICHK.LE.0) RETURN  
READ (5,1070) NTIM,NPNT,ITMAX,NREC,NUMBS,NRECS,NPNTMV,NPNTVL,NPNTD,NPDEL  
1NPNCV,NPNT,NITP,ITMAX,NREC,NUMBS,NRECS,NPNTMV,NPNTVL,NPNTD,NPDEL  
1NPNCV,NPNT,NITP,ITMAX,NREC,NUMBS,NRECS,NPNTMV,NPNTVL,NPNTD,NPDEL  
C **---LIST TIME INCREMENTS---**
20 DO 30 J=1,100  
TIM(J)=0.0  
30 CONTINUE  
TIM(1)=TINIT  
IF (S.EQ.0.0) GO TO 50  
DO 40 K=2,NTIM  
40 TIM(K)=TIM(K-1)  
WRITE (6,470)  
WRITE (6,490) TIM  
GO TO 60  
50 TIM(1)=PYR  
WRITE (6,480) TIM(1)  
C **---INITIALIZE MATRICES---**
60 IF (INT.GT.1) GO TO 100  
D0 70 IY=1,NY  
D0 70 IX=1,NX  
VPRM(IX,IY)=0.0  
PERM(IX,IY)=0.0  
THCK(IX,IY)=0.0  
RECH(IX,IY)=0.0  
CNRECH(IX,IY)=0.0  
REC(IX,IY)=0.0  
NODEID(IX,IY)=0  
TMRX(IX,IY)=0.0  
TMRX(IX,IY,2)=0.0  
HI(IX,IY)=0.0  
HR(IX,IY)=0.0  
HC(IX,IY)=0.0  
HK(IX,IY)=0.0  
WT(IX,IY)=0.0  
VX(IX,IY)=0.0
FORTRAN IV program listing—Continued

VY(IX,IY)=0.0
VXBDY(IX,IY)=0.0
VBBDY(IX,IY)=0.0
CONC(IX,IY)=0.0
CONINT(IX,IY)=0.0
SUMC(IX,IY)=0.0
70 CONTINUE
C

********************* Read Observation Well Locations ***************
C
IF (NUMOBS.LE.0) GO TO 100
WRITE (6,900)
DO 80 J=1,NUMOBS
READ (5,700) IX,IY
WRITE (6,810) J,IX,IY
IXOBS(J)=IX
80
WRITE (6,960)
DO 90 I=1,NUMOBS
DO 90 J=1,SO
TMULC(I,J)=0.0
90
C

******************* Read Pumpage Data (X-Y Coordinates and Rate in CFS) ***
C
--- Signs: Withdrawal = Pos.; Injection = Neg. ---
C
100 IF (NREC.LE.0) GO TO 120
WRITE (6,910)
DO 110 I=1,NREC
READ (5,710) IX,IY,FCTR,CNREC
IF (FCTR.LT.0.0) CNRECHC(IX,IY)=CNREC
REC(IX,IY)=FCTR
110 WRITE (6,820) IX,IY,REC(IX,IY),CNRECHC(IX,IY)
C

************************* Read Transmissivity in Ft**2/sec into VPRM Array ****
C
--- FCTR = TRANSMISSIVITY MULTIPLIER ---> FT**2/SEC ---
C
WRITE (6,530)
READ (5,550) INPUT,FCTR
DO 160 IY=1,NY
IF (INPUT.EQ.1) READ (5,560) (VPRM(IX,IY),IX=1,NX)
DO 150 IX=1,NX
IF (INPUT.EQ.1) GO TO 130
VPRM(IX,IY)=VPRM(IX,IY)*FCTR
150
160
130 VPRM(IX,IY)=FCTR
140 IF (IX.EQ.1.OR.IX.EQ.NX) VPRM(IX,IY)=0.0
IF (IY.EQ.1.OR.IY.EQ.NY) VPRM(IX,IY)=0.0
150 CONTINUE
160 WRITE (6,520) (VPRM(IX,IY),IX=1,NX)
C

********************* Set Up Coefficient Matrix (Block-Centered) *************
C
--- Average Transmissivity --> Harmonic Mean ---
C
170 PIES=3.1415927*3.1415927/2.0
YNS=NY*NY
MODEL OF SOLUTE TRANSPORT IN GROUND WATER

FORTRAN IV program listing—Continued

XNS=NX*NX
HMIN=2.0
DO 180 IY=2,NNY
DO 180 IX=2,NNX
IF (VPRM(IX,IY).EQ.0.0) GO TO 180
TMRX(IX,IY,1)=2.0*VPRM(IX,IY)*VPRM(IX+1,IY)/(VPRM(IX,IY)*XDEL+VPRM)
1(IX+1,IY)*XDEL
TMRX(IX,IY,2)=2.0*VPRM(IX,IY)*VPRM(IX,IY+1)/(VPRM(IX,IY)*YDEL+VPRM)
1(IX,IY+1)*YDEL)

---ADJUST COEFFICIENT FOR ANISOTROPY---
TMRX(IX,IY,1)=TMRX(IX,IY,2)*ANFCTR

---COMPUTE MINIMUM ITERATION PARAMETER (HMIN)---
IF (TMRX(IX,IY,1).EQ.0.0) GO TO 180
IF (TMRX(IX,IY,2).EQ.0.0) GO TO 180
RAT=TMRX(IX,IY,1)*XDEL/(TMRX(IX,IY,2)*YDEL)
HMX=PIES/CXNS*1.0*RAT)
HMY=PIES/CYNS*1.0*1.0/RAT))
IF (HMX.LT.HMIN) HMIN=HMX
IF (HMY.LT.HMIN) HMIN=HMY

180 CONTINUE

---READ AQUIFER THICKNESS---
WRITE (6,510)
READ (5,550) INPUT,FCTR
DO 210 IY=1,NNY
IF (INPUT.EQ.1) READ (5,540) (THCK(IX,IY),IX=1,NX)
DO 200 IX=1,NX
IF (INPUT.NE.1) GO TO 190
THCK(IX,IY)=THCK(IX,IY)*FCTR
GO TO 200
200 CONTINUE

190 IF (VPRM(IX,IY).NE.0.0) THCK(IX,IY)=FCTR

---READ DIFFUSE RECHARGE AND DISCHARGE---
WRITE (6,630)
READ (5,550) INPUT,FCTR
DO 240 IY=1,NNY
IF (INPUT.EQ.1) READ (5,560) (RECH(IX,IY),IX=1,NX)
DO 230 IX=1,NX
IF (INPUT.NE.1) GO TO 220
RECH(IX,IY)=RECH(IX,IY)*FCTR
GO TO 230
220 IF (THCK(IX,IY).NE.0.0) RECH(IX,IY)=FCTR
230 CONTINUE

---COMPUTE PERMEABILITY FROM TRANSMITTIVITY---
---COUNT NO. OF CELLS IN AQUIFER---
DO 250 IX=1,NX
DO 250 IY=1,NY
IF (THCK(IX,IY).EQ.0.0) GO TO 250
PERM(IX,IY)=VPRM(IX,IY)/THCK(IX,IY)
NCA=NCA+1
250 PERM(IX,IY)=0.0

C
AAQ=NCA*AREA
NZCRIT=(NCA+25)/50
WRITE (6,620)
DO 260 IY=1,NNY
260 WRITE (6,650) (PERM(IX,IY),IX=1,NX)
TECHNIQUES OF WATER-RESOURCES INVESTIGATIONS

FORTRAN IV program listing—Continued

WRITE (6,630) NCA,AAQ,N2CRIT

C **********************************************
C ---READ NODE IDENTIFICATION CARDS---
C ---SET VERT. PERM., SOURCE CONC., AND DIFFUSE RECHARGE---
C ---SPECIFY CODES TO FIT YOUR NEEDS---
C
WRITE (6,570)
READ (5,550) INPUT,FCTR
DO 280 IY=1,NY
 IF (INPUT.EQ.1) READ (5,640) (NODEID(IX,IY),IX=1,NX)
 DO 270 IX=1,NX
    270 IF (INPUT.NE.1.AND.THCK(IX,IY).NE.0.0) NODEID(IX,IY)=FCTR
 280 WRITE (6,580) CNODEID(IX,IY),IX=1,NX)
 WRITE (6,920) NCODER
 IF (NCODES.LE.0) GO TO 310
 WRITE (6,930)
 DO 300 IJ=1,NCODES
 READ (5,850) ICODE,FCTR,IFCTR,OVERRD
 DO 290 IX=1,NX
    DO 290 IY=1,NY
       IF (NODEID(IX,IY).NE.ICODE) GO TO 290
      VPRM(IX,IY)=FCTR1
      CNRECH(IX,IY)=FCTR2
 IF (OVERRD.NE.0) RECH(IX,IY)=FCTR3
 290 CONTINUE
 WRITE (6,860) ICODE,FCTR1,FCTR2
 300 IF (OVERRD.NE.0) WRITE (6,1100) FCTR3
 310 WRITE (6,590)
    DO 320 IY=1,NY
     320 WRITE (6,520) (VPRM(IX,IY),IX=1,NX)
 C **********************************************
 C ---READ WATER-TABLE ELEVATION---
 C
 WRITE (6,670)
 READ (5,550) INPUT,FCTR
 DO 350 IY=1,NY
 IF (INPUT.EQ.1) READ (5,660) (WT(IX,IY),IX=1,NX)
 DO 340 IX=1,NX
 340 IF (INPUT.NE.1) GO TO 330
 WT(IX,IY)=WT(IX,IY)+FCTR
 330 IF (THCK(IX,IY).NE.0.0) WT(IX,IY)=FCTR
 340 CONTINUE
 WRITE (6,680) (WT(IX,IY),IX=1,NX)
 C **********************************************
 C ---SET INITIAL HEADS---
 C
 DO 360 IX=1,NX
 360 HK(IX,IY)=HI(IX,IY)
 CALL OUTPT
 C **********************************************
 C ---COMPUTE ITERATION PARAMETERS---
 C
 DO 370 ID=1,NITP
 AOPT(ID)=0.0
 370 CONTINUE
 ANITP=NTIP-1
 ALPHAI=DEXP(DLOG(1.0/HMIN)/ANITP)
 AOPT(1)=HMIN
 DO 380 IP=2,NTIP
 380 AOPT(IP)=AOPT(IP-1)*ALPHA1
**FORTRAN IV program listing—Continued**

```fortran
C
C
C
390
400
410
C
C
430
440
C
C
470
480
490
500
510
520
530
540
550
560
570
580
590
600
610
620
630
```

```
WRITE (6,450)
WRITE (6,460) AOPT
******tt****t****************************************************
---READ INITIAL CONCENTRATIONS AND COMPUTE INITIAL MASS STORED---
READ (5,550) INPUT,FCTR
DO 420 IY=1,NY
IF (INPUT.EQ.1) READ (5,660) (CONC(IX,IY),IX=1,NX)
DO 410 IX=1,NX
IF (THCK(IX,IY).NE.0.0) CONC(IX,IY)=FCTR
CONINT(IX,IY)=TCON(IX,IY)*THCK(IX,IY)*AREA*POROS
CONTINUE
******tt****t****************************************************
---CHECK DATA SETS FOR INTERNAL CONSISTENCY---
DO 440 IX=1,NX
DO 440 IY=1,NY
IF (THCK(IX,IY).GT.0.0) GO TO 430
IF (TMRX(IX,IY).GT.0.0) WRITE (6,940) IX,IY
IF (TMRX(IX,IY).GT.0.0) WRITE (6,950) IX,IY
IF (NODEID(IX,IY).GT.0) WRITE (6,960) IX,IY
IF (WT(IX,IY).NE.0.0) WRITE (6,970) IX,IY
IF (RECH(IX,IY).NE.0.0) WRITE (6,980) IX,IY
IF (REC(IX,IY).NE.0.0) WRITE (6,990) IX,IY
IF (THCK(IX,IY).GT.0.0) WRITE (6,1000) IX,IY
CONTINUE
********tt******************************************************
RETURN
***********tt***************************************************
```

```
450 FORMAT (1H1,20HITERATION PARAMETERS)
460 FORMAT (3H ,16,20,6)
470 FORMAT (1H1,27HTIME INTERVALS (IN SECONDS))
480 FORMAT (1H1,15X,17HSTEADY-STATE FLOW//5X,57HTIME INTERVAL (IN SEC)
1 FOR SOLUTE-TRANSPORT SIMULATION = 6G12.5)
490 FORMAT (3H ,10G12.5)
500 FORMAT (3H ,20F5.1)
510 FORMAT (1H1,22HAQUIFER THICKNESS (FT))
520 FORMAT (3H ,20F5.2)
530 FORMAT (1H1,30HTRANSMISSIVITY MAP (FT*FT/SEC))
540 FORMAT (20G3.0)
550 FORMAT (11G10.0)
560 FORMAT (20G4.1)
570 FORMAT (1H,23HNODE IDENTIFICATION MAP//)
580 FORMAT (1H ,2015)
590 FORMAT (1H1,45HVERTICAL PERMEABILITY/THICKNESS (FT/(FT*SEC)))
600 FORMAT (1H0,10X,12HX-Y SPACING:)
610 FORMAT (1H ,12X,10G12.5).
620 FORMAT (1H1,24HPERMEABILITY MAP (FT/SEC))
630 FORMAT (1H0,10X,44HNO. OF FINITE-DIFFERENCE CELLS IN AQUIFER =
1,14//10X,28HAREA OF AQUIFER IN MODEL = 6G12.5,10H SQ. FT.////1
20X,47HNZCRIT (MAX. NO. OF CELLS THAT CAN BE VOID OF/20X,56HPARTI
```
3CLES: IF EXCEEDED, PARTICLES ARE REGENERATED = 14/

640 FORMAT (2011) 

650 FORMAT (3H 20F5.3) 

660 FORMAT (20G4.0) 

670 FORMAT (1H1,1TH WATER TABLE) 

680 FORMAT (1H 20F5.3) 

690 FORMAT (1HO,10X,19H AREA OF ONE CELL = G12.4) 

700 FORMAT (212) 

710 FORMAT (212,2G8.2) 

720 FORMAT (10A8) 

730 FORMAT (1HO,10A8) 

740 FORMAT (17I4) 

750 FORMAT (1H1,7HNU.S.G.S. METHOD-OF-CHARACTERISTICS MODEL FOR SOLUTE 

1 TRANSPORT IN GROUND WATER) 

760 FORMAT (1HO,21X,21HI N P U T D A T A) 

770 FORMAT (12O8) 

780 FORMAT (120,21X,16H TIME PARAMETERS//13X,40H TIM (MAX. NO. OF TIME 

STEPS) = 16/13X,40HNPMP (NO. OF PUMPING PERIODS) = 16/13X,39H PINT 

(PUMPING PERIOD IN YEARS) = 16/13X,39HTIM (TIME INCREMENT MULTIPLIER) = 16/13X,39HINIT (INIT 

IAL TIME STEP IN SEC.) = 16/13X,39HHRATR (RATIO OF TRANSVERSE TO 

13X,39HNLONGITUDINAL SPERSIVITY) = 16/13X,39HANFCTR (RATIO OF T-YY TO T-XX) 

5 = 16/13X,39HDIFFUSE RECHARGE AND DISCHARGE (FT/SEC)) 

800 FORMAT (125,0) 

810 FORMAT (1H 15X,12,5X,12,4X,12) 

820 FORMAT (1H 17X,21,3X,35X,25,X,25,1H) 

830 FORMAT (1HO,13H RECHARGE AND DISCHARGE (FT/SEC)) 

840 FORMAT (1H 1P10E10.2) 

850 FORMAT (12,36G10.2) 

860 FORMAT (1HO,7X,12,7X,10.3,5X,9.2) 

870 FORMAT (1HO,21X,20H EXECUTION PARAMETERS//13X,39H NITP (NO. OF TIME 

SEGMENTS) = 14/13X,39HTOL (CONVERGENCE CRITERIA - ADIP) = 14/13X,28H 

RAT (RECHARGE TO CONCENTRATED FLOW) = 14/13X, 17H VEL (VELOCITY 

OF FLOW) = 14/13X, 16H NITP (NO. OF TIME STEPS) = 14/13X, 15HNPMTD (PRINT OPT.-CONC. CHANGE = 1) 

58H NPTPND (NO. PARTICLES PER NODE) = 14/13X, 14H NCODES (FOR NODE IDENT.) = 14/13X, 13H NPHCREC (NO. OF PUMPING WELLS) = 14/13X, 12H NPTPND (NO. PARTICLES PER NODE) = 14/13X, 11H NREC (NO. OF OBSERVATION WELLS) = 14/13X, 10H NCPNT (NO. OF NODE IDENT. CODES SPECIFIED) = 14/13X, 9H NCHRECHARGE)
FORTRAN IV program listing—Continued

940 FORMAT (1X,5X,61H*** WARNING *** THCK.EQ.0.0 AND TMRX(X).GT.0.0)
1 AT NODE IX =14,6H, IY =14)
950 FORMAT (1X,5X,61H*** WARNING *** THCK.EQ.0.0 AND TMRX(Y).GT.0.0)
1 AT NODE IX =14,6H, IY =14)
960 FORMAT (1X,5X,61H*** WARNING *** THCK.EQ.0.0 AND NODEID.GT.0.0)
1 AT NODE IX =14,6H, IY =14)
970 FORMAT (1X,5X,61H*** WARNING *** THCK.EQ.0.0 AND T.AT.0.0 AT N)
1ODE IX =14,6H, IY =14)
980 FORMAT (1X,5X,61H*** WARNING *** THCK.EQ.0.0 AND REC.NE.0.0 AT)
1ODE IX =14,6H, IY =14)
990 FORMAT (1X,5X,61H*** WARNING *** THCK.EQ.0.0 AND REC.NE.0.0 AT)
1ODE IX =14,6H, IY =14)
1000 FORMAT (1X,5X,61H*** WARNING *** THCK.EQ.0.0 AND NODEID.GT.0.0)
1 AT NODE IX =14,6H, IY =14)
1010 FORMAT (1X,5X,61H*** WARNING *** THCK.EQ.0.0 AND TMRXCYJ.GT.O.0)
1 AT NODE IX =14,6H, IY =14)
1020 FORMAT (1X,5X,61H*** WARNING *** PERM.EQ.0.0 AND NODEID.GT.0.0)
1 AT NODE IX =14,6H, IY =14)
1030 FORMAT (1X,5X,61H*** WARNING *** PERM.EQ.0.0 AND TMRXCYJ.GT.O.0)
1 AT NODE IX =14,6H, IY =14)
1040 FORMAT (1X,5X,61H*** WARNING *** PERM.EQ.0.0 AND NODEID.GT.0.0)
1 AT NODE IX =14,6H, IY =14)
1050 FORMAT (1X,5X,61H*** WARNING *** PERM.EQ.0.0 AND NODEID.GT.0.0)
1 AT NODE IX =14,6H, IY =14)
1060 FORMAT (11)
1070 FORMAT (11)
1080 FORMAT (11)
1090 FORMAT (11)
1100 FORMAT (11)

END

SUBROUTINE ITERAT
DOUBLE PRECISION DMIN1,DEXP,LOGID,ABS
REAL * TMRX,VPRM,HI,HR,HC,HK,WT,REC,RECH,TIM,AOPT,TITLE
REAL * XDEL,YDEL,S,AREAISUMT,RHO,PARAM,TEST,TOL,PINT,HMIN,PYR
REAL * B,GIW,AIC,E,FIDRIDCITBARITMK,COEF,LH,PRK,BR,KR,CHK,QL,QRH
COMMON /PRMI/ NTIM,NPMP,NPNT,INITPININX,NY,NPINRE,NT,NNX,NNY,NUMO,BS,NV,IHCV,NPMAX,NPNTM,NPNTD,NPNTV, NPNTVL
COMMON /PRMK/ NODEID,C20,C20,ICELL,C20,C20,ICEL,TIME,500,IXOB5
COMMON /HEDA/ THCK(20,20),PERM(20,20),TMW(5,50),TMODS(50),APRCFR
COMMON /HEDB/ TMRX(20,20),VPRM(20,20),HI(20,20),HR(20,20),HC(20,20),HK(20,20),WT(20,20),REC(20,20),RECH(20,20),TIM(100),AOPT(100),T
COMMON /BALK/ TOLT9
COMMON /XINV/ DXINV,DXINVV,ARAVRNP,ORIN
DIMENSION w(20), B(20), G(20)
C **************************** C
C ---COMPUTE ROW AND COLUMN---
C ---CALL NEW ITERATION PARAMETER---
C
10 REMN=MOD(KOUNT,NITP)
IF (REMN,EQ.0) NTH=0
NTH=NTH+1
PARAM=AOPT(NTH)
C **************************** C
C ---ROW COMPUTATIONS---
C
TEST=0.0
RHO=S/TIM(N)
BRK=-RHO
DO 50 IY=1,NY
DO 20 M=1,NX
W(M)=0.0
B(M)=0.0
G(M)=0.0
20 CONTINUE
DO 30 IX=1,NX
IF (THCK(IX,IY).EQ.0.0) GO TO 30
COEF=VPRM(IX,IY)
QL=COEF*WT(IX,IY)
A=TMRX(IX-1,IY+1)*DXINV
C=TMRX(IX,IY+1)*DXINV
E=TMRX(IX-1,IY)*DXINV
F=TMRX(IX,IY)*DXINV
TBAR=A+C+E+F
TMK=TBAR*PARAM
BLH=-A-C-RHO-COEF-TMK
BRH=E+F-TMK
DC=BRH*HC(IX,IY)+BRK*HK(IX,IY)-E*HC(IX-1,IY-1)-F+HC(IX,IY+1)+QL+RECH
1(IX,IY)+REC(IX,IY)*AINV
W(IX)=BLH-A*B(IX-1)
B(IX)=C/W(IX)
G(IX)=(DC-A*G(IX-1))/W(IX)
30 CONTINUE
C
C ---column computations---
DO 90 IX=1,NX
DO 60 M=1,NY
W(M)=0.0
B(M)=0.0
60 G(M)=0.0
DO 70 IY=1,NY
IF (THCK(IX,IY).EQ.0.0) GO TO 70
COEF=VPRM(IX,IY)
QL=COEF*WT(IX,IY)
A=TMRX(IX,IY+1)*DYINV
C=TMRX(IX,IY)*DYINV
E=TMRX(IX-1,IY)*DXINV
F=TMRX(IX,IY+1)*DXINV
TBAR=A+C+E+F
TMK=TBAR*PARAM
BLH=-A-C-RHO-COEF-TMK
BRH=E+F-TMK
DC=BRH*HC(IX,IY)+BRK*HK(IX,IY)-E*HC(IX-1,IY-1)-F+HC(IX,IY+1)+QL+RECH
1(IX,IY)+REC(IX,IY)*AINV
W(IX)=BLH-A*B(IX-1)
B(IX)=C/W(IX)
G(IX)=(DC-A*G(IX-1))/W(IX)
70 CONTINUE
C
C ---back substitution---
DO 80 J=2,NX
DO 40 IY=1,NX
IS-NX-IJ
40 HR(IS,IY)=G(IS)-8(IS)*HR(IS+1,IY)
50 CONTINUE
C
C ---column computations---
DO 90 IX=1,NX
DO 60 M=1,NY
W(M)=0.0
B(M)=0.0
60 G(M)=0.0
DO 70 IY=1,NY
IF (THCK(IX,IY).EQ.0.0) GO TO 70
COEF=VPRM(IX,IY)
QL=COEF*WT(IX,IY)
A=TMRX(IX,IY+1)*DYINV
C=TMRX(IX,IY)*DYINV
E=TMRX(IX-1,IY)*DXINV
F=TMRX(IX,IY+1)*DXINV
TBAR=A+C+E+F
TMK=TBAR*PARAM
BLH=-A-C-RHO-COEF-TMK
BRH=E+F-TMK
DC=BRH*HC(IX,IY)+BRK*HK(IX,IY)-E*HC(IX-1,IY-1)-F+HC(IX,IY+1)+QL+RECH
1(IX,IY)+REC(IX,IY)*AINV
W(IX)=BLH-A*B(IX-1)
B(IX)=C/W(IX)
G(IX)=(DC-A*G(IX-1))/W(IX)
70 CONTINUE
C
C ---back substitution---
DO 80 J=2,NX
MODEL OF SOLUTE TRANSPORT IN GROUND WATER

FORTRAN IV program listing—Continued

IJ=J-1  C 900
IB=NY-IJ  C 910
HCCI,IB)=G(IB)-B(IB)*HC(IX,IB+1)  C 920
IF (THCK(IX,IB).EQ.0.0) GO TO 80  C 930
CHK=DABS(HC(IX,IB)-HR(IX,IB))  C 940
IF (CHK.GT.TOL) TEST=1.0  C 950
80 CONTINUE  C 960
90 CONTINUE  C 970

KOUNT=KOUNT+1  C 980
IF (TEST.GT.0.0) TEST=1.0  C 990

---TERMINATE PROGRAM -- ITMAX EXCEEDED---  C 1030

100 WRITE (6,160)  C 1040
DO 110 IX=1,NX  C 1050
DO 110 IY=1,NY  C 1060
110 HK(IX,IY)=HC(IX,IY)  C 1070
CALL OUTPT  C 1080
STOP  C 1090

---SET NEW HEAD (HK)---  C 1120

120 WRITE (6,140) N  C 1130
WRITE (6,150) KOUNT  C 1140
RETURN  C 1150

---COMPUTE LEAKAGE FOR MASS BALANCE---  C 1180

130 CONTINUE  C 1190

---COMPUTE LEAKAGE FOR MASS BALANCE---  C 1200

140 CONTINUE  C 1210

150 FORMAT (1H015X#64H*+* EXECUTION TERMINATED -- MAX. NO. ITERATION
160 FORMAT (1H015X#64H*+* EXECUTION TERMINATED -- MAX. NO. ITERATION
1S EXCEEDED ***/26X21HFINAL OUTPUT FOLLOWS:)  C 1610

END

SUBROUTINE GENPT  C 1370-

REAL *VPRM,H1,HR,HC,HK,WT,REC,RECH,TIM,AOPT,TITLE  D 10
REAL *BDEL,YDEL,AREA,SMRTI,HRO,PARAM,TEST,TOL,PRINT,HMIN,PYR  D 20
COMMON /PRMI/ NIIM,NPMP,NPNT,NIP,NX,NY,NP,REC,INT,NX,NY,NUMO  D 30
COMMON /PMMK/ NODEID(20,20),NPCELL(20,20),LIAMBO(500),IXOBS(5),IYOB  D 40
1S(5)  D 50
COMMON /HEDA/ THCK(20,20),PERM(20,20),TMWL(5,50),TMBS(50),ANFCRT  D 60
COMMON /HEDB/ TMX(20,20,2),VPRM(20,20),HI(20,20),HR(20,20),HC(20,20)  D 70
COMMONEH1(5,50),VX(20,20),AOP(20,20),T  D 80
2LEVNC(20,20),VX(20,20),RECH(20,20),TIM(100),AOP(20,20),T  D 90
COMMON /CHMA/ PART(3200),CONC(20,20),TMCN(5,50),VX(20,20),VY(20,20)  D 100
COMMON /CHMB/ PART(3200),CONC(20,20),TMCN(5,50),VX(20,20),VY(20,20)  D 110
COMMON /CHMC/ PART(3200),CONC(20,20),TMCN(5,50),VX(20,20),VY(20,20)  D 120
COMMON /CHMD/ PART(3200),CONC(20,20),TMCN(5,50),VX(20,20),VY(20,20)  D 130
COMMON /CHME/ PART(3200),CONC(20,20),TMCN(5,50),VX(20,20),VY(20,20)  D 140
**FORTRAN IV program listing—Continued**

```fortran
ZIrCHSIN CMSOUT FLWINI FMLOT SUMIO CELDIS DLTRAT CSTORM
DIMENSION RP(8), RN(8), IPT(8)

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
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
CMSIN=0.0
CMSOUT=0.0
FLMIN=0.0
FLMOT=0.0
SUMIO=0.0
C *******************************************************
10 DO 20 ID=1,3
   DO 20 IN=1,NPMAX
20 PART(ID,IN)=0.0
DO 30 IA=1,8
   RP(IA)=0.0
   RN(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
50 LIMBO(IL)=IND
IND=IND+1
C *******************************************************
C -----INSERT PARTICLES-----
DO 410 IX=1,NX
   DO 610 IY=1,NY
      IF (THCK(IX,IY).EQ.0.0) GO TO 410
      KR=0
      TESTZ=0.0
      METH=1
      NPCELL(IX,IY)=0
      C1=CONC(IX,IY)
      IF (C1.LE.1.0E-05) TEST2=1.0
      IF (VPMR(IX,IY).GT.0.09) TEST2=1.0
      IF (REC(IX,IY).NE.0.0) TEST2=1.0
      IF (THCK(IX+1,IY+1).EQ.0.0,OR,THCK(IX+1,IY-1).EQ.0.0,OR,THCK(IX-1,IY+1).EQ.0.0,OR,THCK(IX-1,IY-1).EQ.0.0) TEST2=1.0
      IF (THCK(IX,IY+1).EQ.0.0,OR,THCK(IX+1,IY-1).EQ.0.0,OR,THCK(IX+1,IY).EQ.0.0,OR,THCK(IX+1,IY+1).EQ.0.0) TEST2=1.0
      IF (THCK(IX,IY).EQ.0.0,OR,THCK(IX-IY+1).EQ.0.0,OR,THCK(IX-IY-1).EQ.0.0,OR,THCK(IX-IY).EQ.0.0) TEST2=1.0
      CNDPE=C1*(1.0+F1)
      IF (TEST.LT.98.0,OR,TEST2.GT.0.0) GO TO 70
      SUMC=SUMC+CONC(IX+1,IY)+CONC(IX-1,IY)+CONC(IX,IY+1)+CONC(IX,IY-1)
      IF (NCHK.EQ.4) GO TO 60
      SUMC=SUMC+CONC(IX+1,IY+1)+CONC(IX+1,IY-1)+CONC(IX-1,IY+1)+CONC(IX-1,IY-1)
50 AVC=SUMC/NCHK
      IF (AVC.GT.C1) METH=2
60 DO 140 IT=1,2
```

**End of program listing**
MODEL OF SOLUTE TRANSPORT IN GROUND WATER

FORTRAN IV program listing—Continued

EVET = (-1.0)**IT
DO 140 IS = 1, 2
EVES = (-1.0)**IS
PART(1, IND) = IX + F1 * EVET
PART(2, IND) = IY + F1 * EVES
PART(2, IND) = -PART(2, IND)
PART(3, IND) = C1
IF (TEST.LT.98.0 OR TEST2.GT.0.0) GO TO 130
IXD = IX + FVET
IYD = IY + EVES
KR = KR + 1
IPT(KR) = IND
IF (METH.EQ.2) GO TO 80
PART(3, IND) = CNODE + CONC(IXD, IYD) * F1
GO TO 90
80 PART(3, IND) = 2.0 * C1 * CONC(IXD, IYD) / (C1 + CONC(IXD, IYD))
GO TO 130
90 IF (C1 - CONC(IXD, IYD)) 100.110.120
100 RP(KR) = CONC(IXD, IYD) - PART(3, IND)
RN(KR) = C1 - PART(3, IND)
GO TO 130
110 RP(KR) = 0.0
RN(KR) = 0.0
GO TO 130
120 RP(KR) = C1 - PART(3, IND)
RN(KR) = CONC(IXD, IYD) - PART(3, IND)
130 IND = IND + 1
140 CONTINUE
C
IF (NPTPND.EQ.5 OR NPTPND.EQ.9) GO TO 150
GO TO 160
C --- PUT ONE PARTICLE AT CENTER OF CELL ---
150 PART(1, IND) = IX
PART(2, IND) = IY
PART(3, IND) = C1
IND = IND + 1
C
--- PLACE NORTH, SOUTH, EAST, AND WEST PARTICLES ---
160 IF (NPTPND.LT.8) GO TO 290
CNODE = C1 * (1.0 - F2)
DO 280 IT = 1, 2
EVET = (-1.0)**IT
PART(1, IND) = IX + F2 * EVET
PART(2, IND) = IY
PART(3, IND) = C1
IF (TEST.LT.98.0 OR TEST2.GT.0.0) GO TO 220
IXD = IX + EVET
KR = KR + 1
IPT(KR) = IND
IF (METH.EQ.2) GO TO 170
PART(3, IND) = CNODE + CONC(IXD, IYD) * F2
GO TO 180
170 PART(3, IND) = 2.0 * C1 * CONC(IXD, IYD) / (C1 + CONC(IXD, IYD))
180 IF (C1 - CONC(IXD, IYD)) 190.200.210
190 RP(KR) = CONC(IXD, IYD) - PART(3, IND)
RN(KR) = C1 - PART(3, IND)
GO TO 220
200 RP(KR) = 0.0
RN(KR) = 0.0
GO TO 220
210 RP(KR) = C1 - PART(3, IND)
RN(KR) = CONC(IXD, IYD) - PART(3, IND)
220 IND = IND + 1
PART(1, IND) = IX
```
PART(2,IND) = IY + F2 * EVET
PART(2,IND) = -PART(2,IND)
PART(3,IND) = C1
IF (TEST .LT. 98.0 OR. TEST2 .GT. 0.0) GO TO 280
IYD = IY + EVET
KR = KR + 1
IPT(KR) = IND
IF (METH .EQ. 2) GO TO 230
PART(3,IND) = CONC(INC) + CONC(IYD) * F2
GO TO 240

230 PART(3,IND) = 2.0 * C1 * CONC(IYD) / (C1 + CONC(IYD))
240 IF (C1 - CONC(IYD)) .LT. 0.0 OR. C1 .GT. 260 GO TO 270
250 RPAR(KR) = CONC(IYD) - PART(3,IND)
RN(KR) = C1 - PART(3,IND)
GO TO 280
260 RP(KR) = 0.0
RN(KR) = 0.0
GO TO 280
270 RP(KR) = C1 - PART(3,IND)
RN(KR) = CONC(IYD) - PART(3,IND)
280 IND = IND + 1
C
290 IF (TEST .LT. 98.0 OR. TEST2 .GT. 0.0) GO TO 410
SUMPT = 0.0
C
--- COMPUTE CONC. GRADIENT WITHIN CELL ---
DO 300 KPT = 1, NCHK
IK = IPT(KPT)
300 SUMPT = PART(3,IK) + SUMPT
CBAR = SUMPT / NCHK
C
--- CHECK MASS BALANCE WITHIN CELL AND ADJUST PT. CONCS. ---
SUMPT = 0.0
IF (CBAR - C1) .LT. 0.0 GO TO 330
310 CRCT = 1.0 - (CBAR / C1)
IF (METH .EQ. 1) CRCT = CBAR / C1
DO 320 KPT = 1, NCHK
IK = IPT(KPT)
PART(3,IK) = PART(3,IK) + RP(KPT) * CRCT
CBARN = SUMPT / NCHK
GO TO 350
330 CRCT = 1.0 - (C1 / CBAR)
IF (METH .EQ. 1) CRCT = C1 / CBAR
DO 340 KPT = 1, NCHK
IK = IPT(KPT)
PART(3,IK) = PART(3,IK) + RN(KPT) * CRCT
340 SUMPT = SUMPT + PART(3,IK)
CBARN = SUMPT / NCHK
350 IF (CBARN .EQ. C1) GO TO 410
C
--- CORRECT FOR OVERCOMPENSATION ---
CRCT = C1 / CBARN
DO 380 KPT = 1, NCHK
IK = IPT(KPT)
PART(3,IK) = PART(3,IK) + CRCT
C
--- CHECK CONSTRAINTS ---
IF (PART(3,IK) - C1) .LT. 0.0 OR. PART(3,IK) .GT. CLIM GO TO 360
360 CLIM = C1 - RP(KPT) + RN(KPT)
IF (PART(3,IK) .LT. CLIM) GO TO 390
GO TO 380
370 CLIM = C1 + RP(KPT) - RN(KPT)
IF (PART(3,IK) .GT. CLIM) GO TO 590
380 CONTINUE
GO TO 410
```
390 TEST2=1.0
   DO 400 KPT=1,NCHK
   IK=IPT(KPT)
400 PART(3,IK)=C1
410 CONTINUE
   NP=NCHK
   IF (INT(EQ.0)) CALL CHMOT
   RETURN
C
END
SUBROUTINE VELO
DOUBLE PRECISION DMINT,DEXP,DLOG,DABS
REAL +8TMRX~VPRM~HI~HR~HC,HK,WT,REC,RECH,TIM,AOPT~TITLE
REAL *~XDELIYDEL~S~AREA,SUMT,RHO,PARAM,TEST,TOL~PINT,HMIN,PYR
REAL +~RATEISLEAKIDIV
COMMON /PRMI/ NTIMINPMPINPNTINITPIN,NXINYINPINRECIINTINNX~NNY~NUMO
1BS8NM0v~IMCv~NpMAX~ITMAX0N2C~IT,IpR~TI~pTp~~,~p~T~~,~~~~~~,Np~T~,~
COMMON /PRMK/ NODEID~2O,2O~rNPCELL~2O,2O~rLIMBO~5OO~,IXOBS~5~,IYOB
IS(S)
COMMON /HEDA/ THCK~20~20~rPERM~20,20~~TMWL~5~50~~TM0BS~50~~ANFCTR
COMMON /HEDB/ TMAX(20,20)~VPRM(20,20)~HI(20,20)~HR(20,20)~HC(20,20)
120)~HK(20,20)~WT(20,20)~REC(20,20)~RECH(20,20)~TIM(100)~AOPT(20)~T
2TLE(10)~XDEL~YDEL~S~AREA~SUMT~RHO~PARAM~TEST~TOL~PINT~HMIN~PYR
E 130
COMMON /XINV/ DXINO~DYINO~AVIN~ARINV~PORINV
E 140
COMMON /CHMA/ PART(3~3200)~CONC(20,20)~TMCN(5,50)~VX(20,20)~VY(20,20)
120)~CONINT(20,20)~CNRECH(20,20)~POROS~SUMTCH,BETA,TIMV,STORM,STORM
E 150
21~CMSIN~CMSOUT~FLMIN~FLMOT~S~ARGE~SUM0~CELDIS~DLTRAT~TSTORM
E 160
COMMON /CHMC/ SUMC(20,20)~VXBDRY(20,20)~VYBDY(20,20)
E 170
COMMON /DIIFUS/ DISP(20,20,4)
E 180
C
---COMPUTE VELOCITIES AND STORE---
C
   VMAX=1.0E-10
   VMAX=1.0E-10
   VMXBD=1.0E-10
   VMYBD=1.0E-10
   LIM=O
C
DO 20 IX=1,NX
DO 20 IY=1,NY
DO 10 IZ=1,NZ
10 DISP(IX,IY,IZ)=0.0
C
IF (THCK(IY,IY).EQ.0.0) GO TO 20
   RATE=REC(IY,IY)/AREA
   SLEAK=(HK(IY,IY)-WT(IY,IY))*VPRM(IY,IY)
   DIV=RATE+SLEAK+RECH(IY,IY)
   VMXBD=VMXBD+DIV
   VMYBD=VMYBD+DIV
C
---VELOCITIES AT NODES---
C
   GRDX=(HK(IY-1,IY)-HK(IY+1,IY))*DXINO*0.5
   IF (THCK(IY-1,IY).EQ.0.0) GRDX=(HK(IY-1,IY)-HK(IY+1,IY))*DXINO
   IF (THCK(IY+1,IY).EQ.0.0) GRDX=(HK(IY+1,IY)-HK(IY-1,IY))*DXINO
   IF (THCK(IY,IY).EQ.0.0) GRDX=HK(IY,IY)*DXINO
   VX(IY,IY)=PERM(IY,IY)*GRDX*VORINV
   ABVX=ABS(VX(IY,IY))
   IF (ABVX.GT.VMAX) VMAX=ABVX
C
---Y-DIRECTION---
C
   GRDY=(HK(IY,IY+1)-HK(IY,IY-1))*DYINO*0.5
   IF (THCK(IY+1,IY).EQ.0.0) GRDY=(HK(IY+1,IY)-HK(IY,IY-1))*DYINO
   IF (THCK(IY-1,IY).EQ.0.0) GRDY=(HK(IY-1,IY)-HK(IY,IY+1))*DYINO
   IF (THCK(IY,IY).EQ.0.0) GRDY=HK(IY,IY)*DYINO
   VY(IY,IY)=PERM(IY,IY)*GRDY*VORINV
   ABVY=ABS(VY(IY,IY))
   IF (ABVY.GT.VMAX) VMAX=ABVY
C
IF (THCK(IX, IY+1).EQ.0.0)  GRDY=(HK(IX, IY-1)-HK(IX, IY))*DYINV  E 520
IF (THCK(IX, IY-1).EQ.0.0.AND. THCK(IX, IY+1).EQ.0.0)  GRDY=0.0  E 530
VY(IX, IY)=PERM(IX, IY)*GRDY*PORINV*ANFCTR  E 540
ABY=ABS(VY(IX, IY))  E 550
IF (ABY.GT.VMAY) VMAY=ABY  E 560
C
C ----V ELOCITIES AT CELL BOUNDARIES----
GRDX=(HK(IX, IY)-HK(IX+1, IY))*DXINV  E 580
PERMX=2.0*PERM(IX, IY)*PERM(IX+1, IY)/(PERM(IX, IY)+PERM(IX+1, IY))  E 590
VXBDY(IX, IY)=PERMX*GRDX*PORINV  E 600
GRDY=(HK(IX, IY)-HK(IX+1, IY))*DYINV  E 610
PERMY=2.0*PERM(IX, IY)*PERM(IX+1, IY)/(PERM(IX, IY)+PERM(IX+1, IY))  E 620
VYBDY(IX, IY)=PERMY*GRDY*PORINV*ANFCTR  E 630
ABVX=ABS(VXBDY(IX, IY))  E 640
ABVY=ABS(VYBDY(IX, IY))  E 650
IF (ABVX.GT.VMXBD) VMXBD=ABVX  E 660
IF (ABVY.GT.VMYBD) VMYBD=ABVY  E 670
C
IF (DIV.GE.0.0) GO TO 20  E 700
TDIV=(POROS*THCK(IX, IY))/DABS(DIV)  E 710
IF (TDIV.LT.TMV) TMV=TDIV  E 720
20 CONTINUE  E 730
C
C **********PR IN VELOCITIES**********  E 740
IF (NPNTVL.EQ.0) GO TO 80  E 750
IF (NPNTVL.EQ.2) GO TO 90  E 760
IF (NPNTVL.EQ.1.AND.N.EQ.1) GO TO 90  E 770
GO TO 80  E 780
30 WRITE (6,320)  E 790
WRITE (6,330)  E 800
DO 40 IY=1,NY  E 810
40 WRITE (6,350) (VX(IX, IY), IX=1,NX)  E 820
WRITE (6,340)  E 830
DO 50 IY=1,NY  E 840
50 WRITE (6,350) (VXBDY(IX, IY), IX=1,NX)  E 850
WRITE (6,360)  E 860
WRITE (6,330)  E 870
DO 60 IY=1,NY  E 880
60 WRITE (6,350) (VY(IX, IY), IX=1,NX)  E 890
WRITE (6,340)  E 900
DO 70 IY=1,NY  E 910
70 WRITE (6,350) (VYBDY(IX, IY), IX=1,NX)  E 920
C
C PUNCH VELOCITIES  E 930
80 IF (NPNCHV.EQ.0) GO TO 110  E 940
IF (NPNCHV.EQ.2) GO TO 90  E 950
IF (NPNCHV.EQ.1.AND.N.EQ.1) GO TO 90  E 960
GO TO 110  E 970
90 WRITE (7,510) NX,NY,XDEL,YDEL,VMAX,VMAY  E 980
DO 100 IY=1,NY  E 990
100 WRITE (7,520) (VX(IX, IY), IX=1,NX)  E 1000
100 WRITE (7,520) (VY(IX, IY), IX=1,NX)  E 1010
C
C ***************************************************  E 1020
C
C ----COMPUTE NEXT TIME STEP----  E 1030
110 WRITE (6,390)  E 1040
WRITE (6,400) VMAX,VMAY  E 1050
WRITE (6,410) VMXBD,VMYBD  E 1060
TDEX=CELDIS*XDEL/VMAX  E 1070
TDELY=CELDIS*YDEL/VMAY  E 1080
TDELX=CELDIS*XDEL/VMXBD  E 1090
TDELY=CELDIS*YDEL/VMYBD  E 1100
TIMV=MIN1(TDELX,TDELY,TDELX,TDELY)  E 1110
WRITE (6,310) TMV,TIMV  E 1120
C
C **TECHNIQUES OF WATER-RESOURCES INVESTIGATIONS**

FORTRAN IV program listing—Continued
IF (TMV.LT.TIMV) GO TO 120
LIM=-1
GO TO 130
120 TIMV=TMV
LIM=1
130 NTIMV=TIM(N)/TIMV
NMMOV=NTIMV+1
WRITE (6,420) TIMV,NTIMV,NMOV
TIMV=TIM(N)/NMMOV
WRITE (6,370) TIM(N)
WRITE (6,380) TIMV
C
IF (BETA.EQ.0.0) GO TO 200
C****************************************************************
C---COMPUTE DISPERSION COEFFICIENTS---
ALPHA=BETA
ALNG=ALPHA
TRAN=DLTRAT*ALPHA
XXZ=XDEL*XDEL
YY2=YDEL*YDEL
XY2=4.0*XDEL*YDEL
DO 150 IX=ZTNX
DO 150 IYM=ZNY
IF (THCK(IX+IY).EQ.0.0) GO TO 150
VXE=VXBDY(IX+IY)
VYS=VYBDY(IX+IY)
VXE=VXBDY(IX+IY)
VYS=VYBDY(IX+IY)
IF (THCK(IX+IY).EQ.0.0) GO TO 140
---FORWARD COEFFICIENTS: X-DIRECTION---
VYE=(VYBDY(IX+IY-I)+VYBDY(IX+IY-I)+VYS+VYBDY(IX+IY))/4.0
VXE2=VX*VXE
VYE2=VY*VYE
VMGE=SQRT(VXE2+VYE2)
IF (VMGE.LT.1.0E-20) GO TO 140
DALN=ALNG*VMGE
DTRN=TRAN*VMGE
VMGE2=VMGE*VMGE
140 IF (THCK(IX+IY+1).EQ.0.0) GO TO 150
VXS=(VXBDY(IY-IX)+VX+VXBDY(IY+IX+1)+VXBAY(IY+1,IY))/4.0
VYS2=VY*VYS
VXS2=VX*VXS
VMGS=SQRT(VXS2+VYS2)
IF (VMGS.LT.1.0E-20) GO TO 150
DALN=ALNG*VMGS
DTRN=TRAN*VMGS
VMGS2=VMGS*VMGS
150 CONTINUE
C****************************************************************
C---ADJUST CROSS-PRODUCT TERMS FOR ZERO THICKNESS---
DO 160 IX=ZTNX
DO 160 IYM=ZNY
IF (THCK(IX+IY+1).EQ.0.0.OR.THCK(IX+IY+1).EQ.0.0.OR.THCK(IX+IY-1).EQ.0.0.OR.THCK(IX+IY+1).EQ.0.0) DI
S(IX+IY+3)=0.0
IF (THCK(IX+IY+1).EQ.0.0.OR.THCK(IX+IY+1).EQ.0.0.OR.THCK(IX+1,IY+1).EQ.0.0) DI
S(IX+IY+1)=0.0
160 CONTINUE
C****************************************************************
C---FORWARD COEFFICIENTS: Y-DIRECTION---
DTRN=TRAN*VMGE
VMGS2=VMGS*VMGS
TECHNIQUES OF WATER-RESOURCES INVESTIGATIONS

FORTRAN IV program listing—Continued

1),EQ.0.0.OR.THCK(IY-1,IX+1),EQ.0.0) DISP(IX,IY,4)=0.0  
160 CONTINUE  
C  **************************************************  
C  ---CHECK FOR STABILITY OF EXPLICIT METHOD----  
C  **************************************************  
TIMDIS=0.0  
DO 170 IX=2,NNX  
DO 170 IY=2,NNY  
TDCO=DISP(IX,IY,1)+DISP(IX,IY,2)  
170 IF (TDCO.GT.TIMDIS) TIMDIS=TDCO  
TIMDC=0.5/TIMDIS  
WRITE (6,440) TIMDC  
NTIMD=TIM(N)/TIMDC  
NDISP=NTIMD+1  
IF (NDISP.LE.NMOV) GO TO 180  
NMOV=NDISP  
TIMV=TIM(N)/NMOV  
LIM=0  
180 WRITE (6,430) TIMV,NTIMD,NMOV  
C  **************************************************  
C  ---ADJUST DISPERSION EQUATION COEFFICIENTS FOR SATURATED THICKNESS---  
C  **************************************************  
DO 190 IX=2,NNX  
DO 190 IY=2,NNY  
BAVX=0.5*(THCK(IX,IY)+THCK(IX+1,IY))  
BAVY=0.5*(THCK(IX,IY)+THCK(IX,IY+1))  
DI SP(IX+1,IY)=DISP(IX+1,IY)*BAVX  
DI SP(IX, IY+1)=DISP(IX, IY+1)*BAVY  
DI SP(IX+1, IY+1)=DISP(IX+1, IY+1)*BAVX  
DI SP(IX+2, IY)=DISP(IX+2, IY)*BAVY  
DI SP(IX+2, IY+1)=DISP(IX+2, IY+1)*BAVX  
DI SP(IX+3, IY)=DISP(IX+3, IY)*BAVY  
DI SP(IX+3, IY+1)=DISP(IX+3, IY+1)*BAVX  
190 DISP(IX,IY,4)=DISP(IX,IY,4)*BAVY  
C  **************************************************  
200 IF (LIM) 210,220,230  
210 WRITE (6,530)  
GO TO 240  
220 WRITE (6,540)  
GO TO 240  
230 WRITE (6,550)  
240 IF (NPNTD.EQ.0) GO TO 300  
IF (NPNTD.EQ.2) GO TO 250  
IF (NPNTD.EQ.1) AND (N.EQ.1) GO TO 250  
GO TO 300  
250 WRITE (6,450)  
WRITE (6,460)  
DO 260 IY=1,NNY  
260 WRITE (6,500) (DISP(IX,IY,1),IX=1,NNX)  
WRITE (6,470)  
DO 270 IY=1,NNY  
270 WRITE (6,520) (DISP(IX,IY,2),IX=1,NNX)  
WRITE (6,480)  
DO 280 IY=1,NNY  
280 WRITE (6,500) (DISP(IX,IY,3),IX=1,NNX)  
WRITE (6,490)  
DO 290 IY=1,NNY  
290 WRITE (6,500) (DISP(IX,IY,4),IX=1,NNX)  
300 RETURN  
C  **************************************************  
310 FORMAT (1H,TMV (MAX. INJ.) =,G12.5/20H TIMV (CEL DIS) =,6E23.5)
END

SUBROUTINE MOVE

REAL *TMRXIVPRMIHIIHRIHCIHKIWTIRECIREC, RECH~TIM, AOPT, TITLE
REAL *XDELIYDELISIAREAISUMTIRHOIPARAMITESTITOL~PHMIN~PYPYR

COMMON /PRMI/ NTIM~NPMPINPNTINITPINNXNYNP~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N
COMMON /PRMK/ NODEID~2O,2O~rNPCELL~2O,2O~,LIMBO~5OO~.IXOES~S~,IYOB
COMMON /HEDA/ THCK(~~I~~),PERM(~O,~O~~TMWL~S~~~~~TMOBS~~~~,ANFCTR
COMMON /HEDB/ TMRX(~~I~~I~),VPRM(~~,~~)~HI(~~),HR(Z~,~~).HC(~~,
COMMON /XINV/ DXINV~DYINVIARINVIPORINV
COMMON /CHMA/ PART(3,3200rCONC(20,20),VX(20,20),VY(20,20)
COMMON /CHMC/ SUMCC~~I~~).VXBDYC~~.~~~),VYBDYC~~.~~~)
DIMENSION XNEW(4)r YNEWC4)r DIST(4)

C ***+~t****t*t*****+t***+t**t**+tt*+**+*t~~.~~~~~~~~~~...~~~.~~~
WRITE (6,680) NMOV
SUMTCH=SUMT-TIM(N)

C SUBROUTINE MOVE 
REAL *TMRXIVPRMIHIIHRIHCIHKIWTIRECIREC, RECH~TIM, AOPT, TITLE
REAL *XDELIYDELISIAREAISUMTIRHOIPARAMITESTITOL~PHMIN~PYPYR

C COMMON /PRMI/ NTIM~NPMPINPNTINITPINNXNYNP~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N~N
COMMON /PRMK/ NODEID~2O,2O~rNPCELL~2O,2O~,LIMBO~5OO~.IXOES~S~,IYOB
COMM
JFLAG = 1
   IF (PART(1, IN) .GT. 0.0) GO TO 20
   JFLAG = -1
   PART(1, IN) = -PART(1, IN)
20  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  YOLD = PART(2, IN)
   IY = YOLD + 0.5
   IF (THCK(IX, IY) .EQ. 0.0) GO TO 560

C  ****************************************************
C  --- COMPUTE NEW LOCATION AND LOCATE CLOSEST NODE ---
C  --- LOCATE NORTHWEST CORNER ---
C
   IVX = XOLD
   IVY = YOLD
   IXE = IVX + 1
   IYS = IVY + 1

C  ****************************************************
C  --- LOCATE QUADRANT, VEL. AT 4 CORNERS, CHECK FOR BOUNDARIES ---
C
   CELDX = XOLD - IX
   CELDY = YOLD - IY
   IF (CELDX .EQ. 0.0 .AND. CELDY .EQ. 0.0) GO TO 280
   IF (CELDX .GE. 0.0 .OR. CELDY .GE. 0.0) GO TO 70

C  --- PT. IN NW QUADRANT ---
C
   VXNW = VXBDY(IVX, IVY)
   VXNE = VX(IXE, IVY)
   VXSW = VXBDY(IVX, IYS)
   VXSE = VX(IXE, IYS)
   VYNW = VYBDY(IVX, IVY)
   VYNE = VY(IXE, IVY)
   VYSW = VYBDY(IXE, IYS)
   VYSE = VY(IXE, IYS)
   IF (THCK(IVX, IVY) .EQ. 0.0) GO TO 50
   IF (REC(IXE, IVY) .EQ. 0.0 .AND. VPRM(IXE, IVY) .LT. 0.09) GO TO 40
   VXNE = VXNW
   40  IF (REC(IVX, IYS) .EQ. 0.0 .AND. VPRM(IVX, IYS) .LT. 0.09) GO TO 50
       VYSW = VYNW
   50  IF (REC(IXE, IYS) .EQ. 0.0 .AND. VPRM(IXE, IYS) .LT. 0.09) GO TO 270
       IF (THCK(IXE, IYS) .EQ. 0.0) GO TO 60
           VXSE = VXSW
   60  IF (THCK(IXE, IYS) .EQ. 0.0) GO TO 270
       VYSE = VYNE
       GO TO 270

C  --- PT. IN NE QUADRANT ---
C
   VXNW = VX(IVX, IVY)
   VXNE = VXBDY(IVX, IVY)
   VXSW = VXBDY(IVX, IYS)
   VXSE = VX(IXE, IYS)
   VYNW = VYBDY(IVX, IVY)
   VYNE = VYBDY(IXE, IVY)
   VYSW = VY(IVX, IYS)
   VYSE = VY(IXE, IYS)
   IF (CELDX .EQ. 0.0) GO TO 120
   IF (THCK(IXE, IVY) .EQ. 0.0) GO TO 100
   IF (REC(IXE, IVY) .EQ. 0.0 .AND. VPRM(IXE, IVY) .LT. 0.09) GO TO 90
   VXNW = VXNE
   90  IF (REC(IXE, IYS) .EQ. 0.0 .AND. VPRM(IXE, IYS) .LT. 0.09) GO TO 100
FORTRAN IV program listing—Continued

VYSE=VYNE
100 IF (REC(Ivx,Iys).EQ.0.0.AND.VPRM(Ivx,Iys).LT.0.09) GO TO 270
IF (THCK(Ixe,Iys).EQ.0.0) GO TO 110
VXSW=VXSE
110 IF (THCK(Ivx,Ivy).EQ.0.0) GO TO 270
VYSW=VYNW
GO TO 270
120 IF (REC(Ivx,Iys).EQ.0.0.AND.VPRM(Ivx,Iys).LE.0.09) GO TO 270
IF (THCK(Ivx,Ivy).EQ.0.0) GO TO 270
VYSW=VYNW
GO TO 270
130 IF (CELDy.LE.0.0.OR.CELDX.GE.0.0) GO TO 190
C ---PT, IN SW QUADRANT---
140 VXNW=VB(DY(Ivx,Ivy)
VXNE=VX(Ixe,Ivy)
VXSW=VB(DY(Ivx,Iys)
VXSE=VX(Ixe,Iys)
VYNW=VY(Ivx,Ivy)
VYNE=VY(Ixe,Ivy)
VYSW=VYBDY(Ivx,Ivy)
VYSE=VYBDY(Ixe,Ivy)
IF (CELDy.EQ.0.0) GO TO 180
IF (THCK(Ivx,Ivy).EQ.0.0) GO TO 160
IF (REC(Ivx,Ivy).EQ.0.0.AND.VPRM(Ivx,Ivy).LT.0.09) GO TO 150
VYNW=VYSW
150 IF (REC(Ixe,Iys).EQ.0.0.AND.VPRM(Ixe,Iys).LT.0.09) GO TO 180
IF (THCK(Ivx,Ivy).EQ.0.0) GO TO 160
160 IF (REC(Ixe,Ivy).EQ.0.0.AND.VPRM(Ixe,Ivy).LT.0.09) GO TO 270
IF (THCK(Ivx,Ivy).EQ.0.0) GO TO 170
VXNW=VXNW
170 IF (THCK(Ixe,Iys).EQ.0.0) GO TO 270
VYNE=VYSE
GO TO 270
GO TO 270
180 IF (REC(Ixe,Ivy).EQ.0.0.AND.VPRM(Ixe,Ivy).LE.0.09) GO TO 270
IF (THCK(Ivx,Ivy).EQ.0.0) GO TO 270
VXNW=VXNW
GO TO 270
C ---PT, IN SE QUADRANT---
190 IF (CELDy.LE.0.0.OR.CELDX.LE.0.0) GO TO 260
C 200 VXNW=VX(Ivx,Ivy)
VXNE=VX(Ixe,Ivy)
VXSW=VX(Ivx,Iys)
VXSE=VX(Ixe,Iys)
VYNW=VY(Ivx,Ivy)
VYNE=VY(Ixe,Ivy)
VYSW=VYBDY(Ivx,Ivy)
VYSE=VYBDY(Ixe,Ivy)
IF (CELDy.EQ.0.0) GO TO 240
IF (CELDx.EQ.0.0) GO TO 250
IF (THCK(Ixe,Iys).EQ.0.0) GO TO 220
IF (REC(Ixe,Ivy).EQ.0.0.AND.VPRM(Ixe,Ivy).LT.0.09) GO TO 210
VYNE=VYSE
210 IF (REC(Ivx,Iys).EQ.0.0.AND.VPRM(Ivx,Iys).LE.0.09) GO TO 220
VXSW=VXSE
220 IF (REC(Ivx,Ivy).EQ.0.0.AND.VPRM(Ivx,Ivy).LE.0.09) GO TO 270
IF (THCK(Ixe,Ivy).EQ.0.0) GO TO 230
VXNW=VXNW
230 IF (THCK(Ivx,Iys).EQ.0.0) GO TO 270
VYNW=VYSW
GO TO 270
TECHNIQUES OF WATER-RESOURCES INVESTIGATIONS

FORTRAN IV program listing—Continued

240 IF (REC(IX, IVY).EQ.0.0.AND.VPRM(IX, IVY).LE.0.09) GO TO 270
IF (THCK(IXE, IVY).EQ.0.0) GO TO 270
VXNW=VXNE
GO TO 270
250 IF (REC(IX, IVY).EQ.0.0.AND.VPRM(IX, IVY).LE.0.09) GO TO 270
IF (THCK(IX, IYS).EQ.0.0) GO TO 270
VYNW=VYSW
GO TO 270
C
260 IF (C EldX.EQ.0.0.AND.CeldY.LT.0.0) GO TO 80
IF (C EldX.LT.0.0.AND.CeldY.EQ.0.0) GO TO 140
IF (C EldX.GT.0.0.AND.CeldY.EQ.0.0) GO TO 200
IF (C EldX.EQ.0.0.AND.CeldY.GT.0.0) GO TO 200
WRITE (6,690) INIX, IY
270 CONTINUE
C
**********+t*********t********************************************
---BILINEAR INTERPOLATION---
CLEDX=XOLD-IX
CLEDXH=AMOD(CeldX,0.5)
CLEDX=CLEDXH+2.0
CLEDY=YOLD-IV
C
**********+t*********t********************************************
---X VELOCITY---
VXNW=VXNW*(1.0-Celdx)+VXNE*Celdx
IF (THCK(IX, IVY).EQ.0.0.OR.THCK(IXE, IVY).EQ.0.0) VXN=VXNW+VXNE
VXSW=VXSW*(1.0-Celdx)+VXSE*Celdx
IF (THCK(IX, IYS).EQ.0.0.OR.THCK(IXE, IYS).EQ.0.0) VXS=VXSW+VXSE
XVEL=XVN*(1.0-CeldY)+VXS*CeldY
IF (THCK(IX, IVY).EQ.0.0.AND.THCK(IXE, IVY).EQ.0.0) XVEL=VXS
IF (THCK(IX, IYS).EQ.0.0.AND.THCK(IXE, IYS).EQ.0.0) XVEL=VXS
C
---Y VELOCITY---
VYNW=VYNW*(1.0-Celdy)+VYSW*CeldY
IF (THCK(IX, IVY).EQ.0.0.OR.THCK(IXE, IVY).EQ.0.0) VYW=VYNW+VYSW
VYNE=VYNE*(1.0-CeldY)+VYSE*CeldY
IF (THCK(IX, IYS).EQ.0.0.OR.THCK(IXE, IYS).EQ.0.0) VYE=VYNE+VYSE
YVEL=VYN*(1.0-Celdx)+VYE*Celdx
IF (THCK(IX, IVY).EQ.0.0.AND.THCK(IXE, IVY).EQ.0.0) YVEL=VYE
IF (THCK(IX, IYS).EQ.0.0.AND.THCK(IXE, IYS).EQ.0.0) YVEL=VYE
C
GO TO 290
280 XVEL=VX(IX, IY)
YVEL=VY(IX, IY)
290 DISTX=XVEL*CONST1
DISTY=YVEL*CONST2
C
**********+t*********t********************************************
---BOUNDARY CONDITIONS---
TEMPX=XOLD+DISTX
TEMPY=YOLD+DISTY
INX=TEMPX+0.5
INTMPY=TEMPY+0.5
IF (THCK(INX, IY).GT.0.0) GO TO 330
C
**********+t*********t********************************************
---X BOUNDARY---
IF (THCK(INX, IY).EQ.0.0) GO TO 300
PART(1,IN)=TEMPX
GO TO 310
300 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
C
MODEL OF SOLUTE TRANSPORT IN GROUND WATER

FORTRAN IV program listing—Continued

```
INX=PART(1,IN)+0.5
TEMPX=PART(1,IN)
C ******************************************************
C ---Y BOUNDARY---
310 IF (THCK(INX,INY).EQ.0.0) GO TO 320
    PART(2,IN)=TEMPY
    GO TO 340
C ******************************************************
320 BEYON=TEMPY-IY
    IF (BEYON_LT.0.0) BEYON=BEYON+0.5
    IF (BEYON_GT.0.0) BEYON=BEYON-0.5
    INY=PART(2,IN)+0.5
    TEMPY=PART(2,IN)
    GO TO 340
330 PART(1,IN)=TEMPX
    PART(2,IN)=TEMPY
340 CONTINUE
C ******************************************************
C ---SUM CONCENTRATIONS AND COUNT PARTICLES---
SUMC(INX,INY)=SUMC(INX,INY)+PART(3,IN)
NPCELL(INX,INY)=NPCELL(INX,INY)+1
C ******************************************************
C ---CHECK FOR CHANGE IN CELL LOCATION---
IF (IX_EQ.INX.AND.IY_EQ.INY) GO TO 580
C ---CHECK FOR CONST. - HEAD BDY. OR SOURCE AT OLD LOCATION---
IF (REC(IX,IX).LT.0.0) GO TO 350
IF (REC(IX,IX).GT.0.0) GO TO 360
IF (VPBM(IX,IX).LT.0.09) GO TO 540
IF (WT(IX,IX).GT.HK(IX,IX)) GO TO 350
IF (WT(IX,IX).LT.HK(IX,IX)) GO TO 360
GO TO 540
C ******************************************************
C ---CREATE NEW PARTICLES AT BOUNDARIES---
350 IF (IFLAG.GT.0) GO TO 550
    KFLAG=1
360 DO 370 IL=1,500
    IF (LIMBO(IL).EQ.0) GO TO 370
    IP=LIMBO(IL)
    IF (IP_LT.IN) GO TO 380
370 CONTINUE
C ******************************************************
C ---GENERATE NEW PARTICLE---
IF (NPTM_EQ.NP.MAX) GO TO 600
    NPTM=NPTM+1
    IP=NPTM
    GO TO 390
380 LIMBO(IL)=0
390 IF (KFLAG.EQ.0) GO TO 520
    IF (THCK(IX+1,IX+1).EQ.0.0.OR.THCK(IX+1,IX-1).EQ.0.0.OR.THCK(IX+1,IX-1).EQ.0.0) GO TO 520
    IF (THCK(IX+1,IX+1).EQ.0.0.OR.THCK(IX+1,IX-1).EQ.0.0.OR.THCK(IX+1,IX-1).EQ.0.0) GO TO 520
    IF (CENTER) GO TO 270
C ******************************************************
370 IF (JFLAG.EQ.0) GO TO 500
    JJ=4
    AN=TEMPY-YOLD
    AD=TEMPX-XOLD
    DIST=SQRT((AD+AD)+(AN*AN))
    IF (AD_EQ.0.0) GO TO 410
    SLOPE=AN/AD
```

BI=YOLD-SLOPE*XOLD
XC1=IX-F1
XC2=IX+F1
YC1=IY-F1
YC2=IY+F1

---COMPUTE NEW COORDINATES AND VERIFY---

DO 400 IK=1,4
YNEW(IX)=0.0
XNEW(IX)=0.0

400 DIST(IX)=0.0
YNEW(1)=(SLOPE*XC1)+BI
XNEW(1)=XC1
YNEW(2)=(SLOPE*XC2)+BI
XNEW(2)=XC2
IF (SLOPE.EQ.0.0) GO TO 420
YNEW(3)=YC1
XNEW(3)=(YC1-BI)/SLOPE
YNEW(4)=YC2
XNEW(4)=(YC2-BI)/SLOPE
GO TO 430

410 YNEW(1)=IY-F1
XNEW(1)=XOLD
YNEW(2)=IY+F1
XNEW(2)=XOLD

420 JJ=2
430 DO 440 II=1,JJ
440 DIST(II)=SQRT((XNEW(II)-TEMPX)**2+(YNEW(II)-TEMPY)**2)*1.00001
IACC=0
DISTCK=2.0
DO 460 IG=1,JJ
IF (DIST(IG).GE.DISTMV.AND.DIST(IG).LT.DISTCK) GO TO 450
GO TO 460

450 IXC=XNEW(IG)+0.5O
IYC=YNEW(IG)+0.5O
IF ((IXC.NE.IX.OR.IYC.NE.IY) GO TO 460
IACC=IG
DISTCK=DIST(IG)
GO TO 470

460 CONTINUE
IF (IACC.LT.1.OR.IACC.GT.4) GO TO 510
IF (XNEW(IACC).EQ.XC1.OR.XNEW(IACC).EQ.XC2) GO TO 470
IF (YNEW(IACC).EQ.YC1.OR.YNEW(IACC).EQ.YC2) GO TO 480
GO TO 510

470 IF (YNEW(IACC).LT.YC1) YNEW(IACC)=YC1
IF (YNEW(IACC).GT.YC2) YNEW(IACC)=YC2
GO TO 490

480 IF (XNEW(IACC).LT.XC1) XNEW(IACC)=XC1
IF (XNEW(IACC).GT.XC2) XNEW(IACC)=XC2

490 PART(1,IP)=XNEW(IACC)
PART(2,IP)=YNEW(IACC)
GO TO 530

500 PART(1,IP)=IX
PART(2,IP)=IY
GO TO 530

510 PART(1,IP)=XOLD
PART(2,IP)=YOLD
GO TO 530

C ---IF EDGE SOURCE OR SINK---

C ---X POSITION---

520 DLX=INX-IX
PART(1,IP)=TEMPX-DLX

C ---Y POSITION---

DLY=INY-IY
FORTRAN IV program listing—Continued

```fortran
PART(2,IP)=TEMPY-DLY
IF (KFLAG.GT.0) GO TO 530
C   ---IF SINK---
SUMC(IX,IY)=SUMC(IX,IY)+CONC(IX,IY)
NPCELL(IX,IY)=NPCELL(IX,IY)+1
C 530 PART(2,IP)=-PART(2,IP)
PART(3,IP)=CONC(IX,IY)
IF (REC(IX,IY).EQ.0.0) GO TO 540
C ***-----------------------------------------------***
C ---CHECK FOR DISCHARGE BOUNDARY AT NEW LOCATION---
C 540 IFLAG=1.0
550 IF (VPRM(INX,INY).GT.0.9.AND.WT(INX,INY).LT.HK(INX,INY)) GO TO 56
IF (REC(INX,INY).EQ.0.0) GO TO 560
GO TO 590
C ***-----------------------------------------------***
C ---PUT PT. IN LIMBO---
C 560 PART(1,IN)=0.0
PART(2,IN)=0.0
PART(3,IN)=0.0
DO 570 ID=1,N0
      IF (LIMBO(ID).GT.0.0) GO TO 570
LIMBO(ID)=IN
GO TO 590
C 570 CONTINUE
C 580 IF (IFLAG.LT.0) PART(2,IN)=TEMPY
      IF (JFLAG.LT.0) PART(1,IN)=-TEMPX
C 590 CONTINUE
C ---END OF LOOP---
C 620 SUMTCH=SUMTCH+TIMV
C 620 CONTINUE
C ---RESTART MOVE IF PT. LIMIT EXCEEDED---
C 660 WRITE (6,700) IMOV,IN
      TEST=100.0
      CALL GENPT
      DO 610 IX=1,NX
          DO 610 IY=1,NY
              SUMC(IX,IY)=0.0
      SUMTCH=SUMTCH+TIMV
      NP=NPTM
      WRITE (6,670) NPIIMOV
      CALL CNCON
      IF (C+GT.0.0) GO TO 640
      IF (NUMOBS.LE.0) GO TO 640
      J=MOD(IMOV,50)
      IF (J.EQ.0) J=50
      TMWLS(J)=SUMTCH
      DO 630 I=1,NUMOBS
          TMWLS(I,J)=HKXOBS(I),IYOBS(I)
      630 TMWLS(I,J)=CONC(XOBS(I),IYOBS(I)
      ---PRINT CHEMICAL OUTPUT---
      640 IF (IMOV.GE.NMOV) GO TO 660
```
IF (MOD(IMOV,NPNTMW).EQ.0.OR.MOD(IMOV,50).EQ.0) CALL CHMOT

RETURN

C  ***************************************************************
C **********+***************************+***************************
C  ***************************************************************
C
C
C
C
C
C

FORMAT (1HC2X,2HNP,7X2H=8X,1I4,10X,11HIMOV = 8X,1I4)   CALL CHMOT
C *****+************+****t***t******************.*..**.**.*********.*
C

660 IF CMODCIMOV,NPNTMV~.EQ.O.OR.MODCIMOV,SO).EQ.O~ CALL CHMOT
C *****************************************************
660 RETURN
C

C
C

10 FORMAT (HCIZXIZHNP~~X,ZH* ,8X,I4rlOX~11HIMOV = r8x1I4)
680 FORMAT (1H0110X161HN0. OF PARTICLE NOVES REQUIRED TO COMPLETE THIS
1 TIME STEP = ,I4//)
690 FORMAT (1H0,SX,53H*** WARNING

1 NO. ,15llH ** QUADRANT NOT LOCATED FOR PT.

1 NO. ,15llH , IN CELL ~214)
700 FORMAT (1H015X017H

1 ** NOTE

1 **rlOX,23HNPTM.EQ.NPNAX --- IMOV=

1I4rSX18HPT. NO .=,14rSX,lOHCALL GENPT/) 

END 
SUBROUTINE CNCON
REAL *~TMRXIVPRM,HIIHRIHC~HK,WT,REC,RECH,TIM,AOPT~TITLE
REAL *8XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,P~NT,HMIN,PYR
REAL 8FLW
COMMON /PRMI/ NTIMINPMPINPNTINITP,N~NX,NY,NP~NREC~INTINNX~NNY~NUMO
COMMON /PRMK/ NODEID~2O,2O~rNPCELL~20.2O~,LIM~O~~00~,IXO~S~~~~IYO~
COMMON /HEDA/ THCK(20,20),PERM(20,20),TMWL(5,50),TMOBS(50),ANFCTR
COMMON /HEDB/ TMXR(20,20,2),VPRM(20,20),HR(20,20),HR(20,20),HC(20,20),
10 WX(20,20),WT(20,20),RECH(20,20),RECH(20,20),RECH(20,20),RECH(20,20),
20 TIM(10)*AOPT(20),T1M(10)*AOPT(20),T1M(10)*AOPT(20),T1M(10)*AOPT(20),
ZTITLE(10)*AOPT(20),ZTITLE(10)*AOPT(20),ZTITLE(10)*AOPT(20),ZTITLE(10)*AOPT(20),
COMMON /XIHV/ DXINV,DYINV,ARINV,PRINV
COMMON /HCMA/ PART(3,3200),CONC(20,20),TMCN(5,50),VX(20,20),VY(20,20),
120 WX(20,20),WT(20,20),RECH(20,20),RECH(20,20),RECH(20,20),RECH(20,20),
130 CONT(20,20),CNRECH(20,20),CNRECH(20,20),CNRECH(20,20),CNRECH(20,20),
140 SUMTCH,BETA,TIMV,STORM,STORM
150 CMSIN,CMSOUT,FLMIN,FLMOT,SUMIO,CELDIS,DLTRAT,STORM
160 COMMON /DIFUS/ DISP(20,20,4)
170 COMMON /CHWC/ SUMCC20,20),VXBDYC20,20),VYBDYC20,20)
180 COMMON /CHMA/ PART0r3200~,C0NC~20,20~,VX~20,20~,VY~20,20~,CONINT~2O~2O~,CNRECH~2O~2O~,POROS,SMU
190 TCH~~ETA,TIMV~STORM~STORM
200 COMMON /DIFUS/ DISP(20,20,4)
210 COMMON /CHWC/ SUMCC20,20),VXBDYC20,20),VYBDYC20,20)
220 DIMENSION CNCNCC20,20), CN0lDC20120)

C
C
C

C  ***************************************************************
C **********+***************************+***************************
C  ***************************************************************
C
C
C
C
C
C

SUBROUTINE CNCON
REAL *~TMRXIVPRM,HIIHRIHC~HK,WT,REC,RECH,TIM,AOPT~TITLE
REAL *8XDEL,YDEL,S,AREA,SUMT,RHO,PARAM,TEST,TOL,P~NT,HMIN,PYR
REAL 8FLW
COMMON /PRMI/ NTIMINPMPINPNTINITP,N~NX,NY,NP~NREC~INTINNX~NNY~NUMO
COMMON /PRMK/ NODEID~2O,2O~rNPCELL~20.2O~,LIM~O~~00~,IXO~S~~~~IYO~
COMMON /HEDA/ THCK(20,20),PERM(20,20),TMWL(5,50),TMOBS(50),ANFCTR
COMMON /HEDB/ TMXR(20,20,2),VPRM(20,20),HR(20,20),HR(20,20),HC(20,20),
10 WX(20,20),WT(20,20),RECH(20,20),RECH(20,20),RECH(20,20),RECH(20,20),
20 TIM(10)*AOPT(20),T1M(10)*AOPT(20),T1M(10)*AOPT(20),T1M(10)*AOPT(20),
ZTITLE(10)*AOPT(20),ZTITLE(10)*AOPT(20),ZTITLE(10)*AOPT(20),ZTITLE(10)*AOPT(20),
COMMON /XIHV/ DXINV,DYINV,ARINV,PRINV
COMMON /HCMA/ PART(3,3200),CONC(20,20),TMCN(5,50),VX(20,20),VY(20,20),
120 WX(20,20),WT(20,20),RECH(20,20),RECH(20,20),RECH(20,20),RECH(20,20),
130 CONT(20,20),CNRECH(20,20),CNRECH(20,20),CNRECH(20,20),CNRECH(20,20),
140 SUMTCH,BETA,TIMV,STORM,STORM
150 CMSIN,CMSOUT,FLMIN,FLMOT,SUMIO,CELDIS,DLTRAT,STORM
160 COMMON /DIFUS/ DISP(20,20,4)
170 COMMON /CHWC/ SUMCC20,20),VXBDYC20,20),VYBDYC20,20)
180 COMMON /CHMA/ PART0r3200~,C0NC~20,20~,VX~20,20~,VY~20,20~,CONINT~2O~2O~,CNRECH~2O~2O~,POROS,SMU
190 TCH~~ETA,TIMV~STORM~STORM
200 COMMON /DIFUS/ DISP(20,20,4)
210 COMMON /CHWC/ SUMCC20,20),VXBDYC20,20),VYBDYC20,20)
220 DIMENSION CNCNCC20,20), CN0lDC20120)
DIV=RATE+SLEAK+RECH(IY,IY)
IF (S.EQ.0.0) GO TO 30
DERH=(HK(IY,IY)-HR(IY,IY))/TIM(N)
DIV=DIV+DERH
IF (S.LT.0.05) GO TO 30

..NOTE: ABOVE STATEMENT ASSUMES THAT S=0.005 SEPARATES CONFINED
C FROM UNCONFINED CONDITIONS; THIS CRITERION SHOULD BE
C CHANGED IF FIELD CONDITIONS ARE DIFFERENT.

DELC=EQFCT2*(C1*(DIV+POROS*DERH)-RATE*CNREC+SLEAK*CLKCN+RECH(IY,IY))
GO TO 40

30 DELC=EQFCT2*(C1-DIV-RATE*CNREC-SLEAK*CLKCN-RECH(IY,IY))
GO TO 40

40 CNCCN(IY,IY)=CNCCN(IY,IY)+DELC

---DISPERSION WITH TENSOR COEFFICIENTS---

IF (BETA.EQ.O.0) GO TO 50
X1=DISP(IY,IY)+(CONC(IY,IY)-C1)
X2=DISP(IY-1,IY)+(CONC(IY-IY)-C1)
Y1=DISP(IY,IY-2)+(CONC(IY-IY+1)-C1)
Y2=DISP(IY,IY-1,2)+(CONC(IY-IY-1)-C1)
XX1=DISP(IY,IY-3)+(CONC(IY-IY+1)+CNREC(IY+1,IY+1)-CONC(IY,IY-1)-C1)
XX2=DISP(IY,IY-1,3)*(CONC(IY-IY+1)+CNREC(IY+1,IY+1)-CONC(IY-IY-1)-C1)
YY1=DISP(IY,IY-4)*(CONC(IY+1,IY)+CNREC(IY+1,IY+1)-CONC(IY-IY-1)-C1)
YY2=DISP(IY,IY-1,4)*(CONC(IY+1,IY)+CNREC(IY+1,IY+1)-CONC(IY-IY-1)-C1)

50 CNCCN(IY,IY)=CNCCN(IY,IY)+EQFCT1*(X1+X2+Y1+Y2+XX1+XX2+YY1+YY2)

60 CONTINUE

---CHANGE CONCENTRATIONS AT NODES DUE TO CONVECTION---

70 DO 90 IX=1,NX
DO 90 IY=1,NY
IF (THCK(IX,IY),EQ.0.0) GO TO 120
CONC(IX,IY)=CONC(IX,IY)+CNCCN(IX,IY)

80 CONTINUE

---CHECK NUMBER OF CELLS VOID OF PTS.---

90 CONTINUE
TECHNIQUES OF WATER-RESOURCES INVESTIGATIONS

FORTRAN IV program listing—Continued

NPCELL(IX,IY)=0
SUMC(IX,IY)=0.0
IF (CONC(IX,IY).LE.0.0) GO TO 130
CNCPC=CNCPC(IX,IY)/CONC(IX,IY)
SUMC(IX,IY)=CNCPC
GO TO 130
120 IF (CONC(IX,IY).GT.0.0) WRITE (6,310) IX,IY,CONC(IX,IY)
CONC(IX,IY)=0.0
130 CONTINUE

C ******************************************************
C **CHANGE CONCENTRATION OF PARTICLES**
C DO 180 IN=1,NP
IF (PART(1,IN).EQ.0.0) GO TO 180
INX=ABS(PART(1,IN))+0.5
INY=ABS(PART(2,IN))+0.5
180 CONTINUE

C ******************************************************
C **UPDATE CONC. OF PTS. IN SINK/SOURCE CELLS**
IF (REC(IX,IY).NE.0.0) GO TO 140
IF (VPRM(IX,IY).LE.0.0) GO TO 150
140 PART(3,IN)=CONC(IX,IY)
GO TO 180
150 IF (CNCPC(IX,IY).LT.0.0) GO TO 170
160 PART(3,IN)=PART(3,IN)+CNCPC(IX,IY)
GO TO 180
170 IF (CONC(IX,IY).LE.0.0) GO TO 160
IF (SUMC(IX,IY).LT.-1.0) GO TO 160
PART(3,IN)=PART(3,IN)+PART(3,IN)+SUMC(IX,IY)
180 CONTINUE

WRITE (6,280) TINCN),TIMVISUNTCH

C ******************************************************
C **COMPUTE MASS BALANCE FOR SOLUTE**
CSTORM=0.0
STORM=0.0
DO 270 IX=1,NX
DO 270 IY=1,NY
IF (THCK(IX,IY).EQ.0.0) GO TO 270
SUNCC(IX,IY)=0.0
C
---COMPUTE MASS OF SOLUTE IN STORAGE---
STORM=STORM+CONC(IX,IY)*THCK(IX,IY)*ARPOR
C
---ACCOUNT FOR MASS PUMPED IN, OUT, RECHARGED, & DISCHARGED---
IF (REC(IX,IY)) 190,210.190
190 CMSOUT=CMSOUT+REC(IX,IY)*CNOLD(IX,IY)*TIMV
GO TO 210
200 CMSIN=CMSIN+REC(IX,IY)*CNRECH(IX,IY)*TIMV
210 IF (REC(IX,IY)) 230,240.220
220 CMSOUT=CMSOUT+REC(IX,IY)*CNOLD(IX,IY)*TVA
GO TO 240
230 CMSIN=CMSIN+REC(IX,IY)*CNRECH(IX,IY)*TVA
C ******************************************************
C **ACCOUNT FOR BOUNDARY FLOW---
240 IF (VPRM(IX,IY).EQ.0.0) GO TO 270
FLW=VPRM(IX,IY)*WT(IX,IY)-HK(IX,IY)
IF (FLW.GT.0.0) GO TO 250
IF (FLW.LT.0.0) GO TO 260
GO TO 270
C
---MASS IN BOUNDARY DURING TIME STEP---
250 FLMIN=FLMIN+FLW*CNRECH(IX,IY)*TVA
GO TO 270
C
---MASS OUT DURING TIME STEP---
260 FLMOT=FLMOT+FLW*CNOLD(IX,IY)*TVA
270 CONTINUE
C ******************************************************
C **COMPUTE CHANGE IN MASS OF SOLUTE STORED---


FORTRAN IV program listing—Continued

C Model of solute transport in ground water
C
C ************************************************************
C --- REGENERATE PARTICLES IF 'NZCRIT' EXCEEDED ---
C
C *** REGENERATE PARTICLES IF 'NZCRIT' EXCEEDED --- CALL GENPT ***
C
C ******************************************************
C
C 280 FORMAT (3H12HTIM(N) = R1G12.5,10X,11HTIMV = R1G12.5,10X
C 19HSTIM(N) = R1G12.5)
C 290 FORMAT (1H0,5X,4OHNUMBER OF CELLS WITH ZERO PARTICLES = R1G14,5X,9
C 1HMV = R1G14/)
C 300 FORMAT (1H0,5X,44H*** NZCRIT EXCEEDED --- CALL GENPT ***)
C 310 FORMAT (1H0,5X,6HNPCELL/) 
C 320 FORMAT (1H0,2X,6HNPCCELL/) 
C 330 FORMAT (1H0,4X,2013) 
C END
C
C SUBROUTINE OUTPT
REAL *8 TMRX, VPRM, HI, HR, HC, HK, WT, REC, RECH, TIM, AOPT, TITLE
REAL *8 XDEL, YDEL, S, AREA, SUMT, RHO, PARAM, TEST, TOL, PINT, HMIN, PYR
COMMON /PRMI/ NTIM, NPMP, NPNT, NITP, N, NX, NY, NP, NREC, INT, NNX, NNY, NUMO
COMMON /PRMK/ NODEID(20,20), NPCELL(20,20), LIMBO(500), IxOBS(5), IYOB(5)
COMMON /HEDA/ THCK(20,20), PERM(20,20), TMWL(50), TMQBS(50), ANFCTR
COMMON /HEDB/ TMRX(20,20,2), VPRM(20,20), HI(20,20), HR(20,20), HC(20,20)
COMMON /BALM/ TOTLP
DIMENSION IHC(50)
C
C ******************************************************
C TIMD=SUMT/86400.
C TIMY=SUMT/(86400,0+365,25)
C
C --- PRINT HEAD VALUES ---
C WRITE (6,120)
C WRITE (6,130) N
C WRITE (6,140) SUMT
C WRITE (6,150) TIMD
C WRITE (6,160) TIMY
C WRITE (6,170)
C DO 10 IY=1, NY
C 10 WRITE (6,180) (HK(IY,IY),IY=1,NX)
C IF (N.EQ.0) GO TO 110
C
C --- PRINT HEAD MAP ---
C WRITE (6,120)
C WRITE (6,130) N
C WRITE (6,140) SUMT
C WRITE (6,150) TIMD
C WRITE (6,160) TIMY
C WRITE (6,170)
C DO 30 IY=1, NY
C 30 WRITE (6,190) (IN(IY),IY=1,NX)
C
--- COMPUTE WATER BALANCE AND DRAWDOWN ---
QSTR = 0.0
PUMP = 0.0
TPUM = 0.0
QIN = 0.0
QOUT = 0.0
QNET = 0.0
DEQL = 0.0
JCK = 0
PCTERR = 0.0
WRITE (6, 290)

DO 80 IY = 1, NY
DO 70 IX = 1, NX
IH(IX) = 0.0
IF (THCK(IX, IY).EQ.0.0) GO TO 70
TPUM = RECC(IX, IY) + REC(IX, IY) * AREA + TPUM
IF (VPRM(IX, IY).EQ.0.0) GO TO 60
DELQ = VPRM(IX, IY) * AREA * (WT(IX, IY) - HK(IX, IY))
IF (DELQ.GT.0.0) GO TO 40
QOUT = QOUT + DELQ
GO TO 50
40 QIN = QIN + DELQ
50 QNET = QNET + DELQ
60 DDRW = HI(IX, IY) - HK(IX, IY)
IH(IX) = DDRW + 0.5
QSTR = QSTR + IH(IX) * AREA
70 CONTINUE

--- PRINT DRAWDOWN MAP ---
WRITE (6, 300) (IH(IX), IX = 1, NX)

CONTINUE

PUMP = TPUM * SUMT
DELS = - QSTR / SUMT
ERRMB = PUMP - TOTLQ - QSTR
DEN = PUMP + TOTLQ
IF (ABS(DEN).EQ.ABS(ERRMB)) JCK = 1
IF (DEN.EQ.0.0) GO TO 100
IF (JCK.EQ.1) GO TO 90
PCTERR = ERRMB * 200.0 / DEN
GO TO 100
90 IF (QIN.EQ.0.0) GO TO 100
PCTERR = 100.0 * QNET / QIN

--- PRINT MASS BALANCE DATA FOR FLOW MODEL ---
WRITE (6, 240)
WRITE (6, 250) PUMP
WRITE (6, 260) QIN
WRITE (6, 270) TPUM
WRITE (6, 280) PCTERR
WRITE (6, 290) QOUT
WRITE (6, 300) TOTLQ
WRITE (6, 310) DELS
IF (JCK.EQ.1) WRITE (6, 280) PCTERR

RETURN

120 FORMAT (1H1, 13H HEAD DISTRIBUTION - ROW)
130 FORMAT (1X, 13H NUMBER OF TIME STEPS = , I11)
140 FORMAT (8X, 16H TIME(SECONDS) = , 1G12.5)
MODEL OF SOLUTE TRANSPORT IN GROUND WATER

FORTRAN IV program listing—Continued

150 FORMAT (8X,16HTIME(DAYS) = $1E12.5) H1030
160 FORMAT (8X,16HTIME(YEARS) = $1E12.5) H1040
170 FORMAT (1H ) H1050
180 FORMAT (1H0,10F12.7/10F12.7) H1060
190 FORMAT (1H0,120I4) H1070
200 FORMAT (1H0,2X,33HRATE MASS BALANCE -- (IN C.F.S.) //1OX,8HQIN = $G12.5)
201 FORMAT (1H0,17X,8HQOUT = $G12.5) H1080
202 FORMAT (1H0,17X,8HQNET = $G12.5) H1090
210 FORMAT (1H0,17X,8HPUM = $G12.5) H1100
220 FORMAT (1H0,17X,8HTPUM = $G12.5) H1110
230 FORMAT (1H0,12X,H12.5) H1120
240 FORMAT (1H0,2X,23HCUMULATIVE MASS BALANCE/) H1130
250 FORMAT (1H0,2X,23HCUMULATIVE NET PUMPAGE = $G12.5) H1140
260 FORMAT (1H0,2X,23HCUMULATIVE NET LEAKAGE = $E12.5) H1150
270 FORMAT (1H0,7X,23HMASS BALANCE RESIDUAL = $G12.5) H1160
280 FORMAT (1H0,7X,23HERROR (AS PERCENT) = $G12.5/) H1170
290 FORMAT (1H0,8HDRAWDOWN) H1180
300 FORMAT (3H $2015) H1190
END H1200-
SUBROUTINE CHHOT

REAL TMRX, VRPRM, HIR, HCR, HHR, WTR, REC, RECH, TIM, AOPT, TITLE
REAL XDELY, YDEL, SAREA, SUMT, RH0, PARAM, TEST, TOL, PINT, HMIN, PYR
COMMON /PRMI/ NTIM, NPMPINPNT, INITPININXINYINXINXINXINXINXINX
COMMON /PRMK/ NODEID(20,20), NPCELL(20,20), LIMBO(500), IXOBS(5), IYOB(5)
COMMON /HEDA/ THCK(20,20), PERM(20,20), TMWL(5,50), TMOLB(50), ANFCTR
COMMON /HEAD/ TMRX, VRPRM, HIR, HCR, HHR, WTR, REC, RECH, TIM, AOPT, TITLE
COMMON /CHMA/ PART(3,3,200), CONC(20,20), TMCC(5,50), VX(20,20), VY(20,20),
CONC(20,20), CNRCCH(20,20), POROS, SUMCII, DETA, TIM, STORM, STORM
COMMON /CHSA/ PART(3,3,200), CONC(20,20), TMCC(5,50), VX(20,20), VY(20,20),
CONC(20,20), CNRCCH(20,20), POROS, SUMCII, DETA, TIM, STORM, STORM
DIMENSION IC(20)
C ***********************************************************************
TMFY=86400.0/365.25 I 170
TMYR=SUMT/TMFY I 180
TCHD=SUMCH/86400.0 I 190
TCHYR=SUMCH/TMFY I 200
IF (IPRNT.GT.0) GO TO 100 I 210
C ***************---PRINT CONCENTRATIONS---*************** I 220
WRITE (6,160) I 230
WRITE (6,170) N I 240
IF (N.EQ.0) WRITE (6,180) TIM(N) I 250
WRITE (6,190) SUMT I 260
WRITE (6,200) SUMCH I 270
WRITE (6,210) TCHD I 280
WRITE (6,220) TCHYR I 290
WRITE (6,230) IMOV I 300
WRITE (6,240) DO 20 IX=1,NX I 310
DO 20 IY=1,NY I 320
DO 20 10 ICX=1,NX I 330
10 ICX=ICX+1
WRITE (6,240) (ICX(IY), IY=1,NY) I 340
C ***************---PRINT CHANGES IN CONCENTRATION---*************** I 350
WRITE (6,250)
WRITE (6,170) N
WRITE (6,180) TIM(N)
WRITE (6,190) SUMT
WRITE (6,210) TMYR
WRITE (6,220) TCHD
WRITE (6,240) CONC(IX,IY)+CONINT(IX,IY)
WRITE (6,220) DO 40 IY=1,NY
WRITE (6,450) SUMTCH
WRITE (6,460) TCHYR
WRITE (6,380) IMOV
WRITE (6,210) DO 30 IX=1,NX
CNG=CONC(IX,IY)-CONINT(IX,IY)
30 IC(IX)=CNG
WRITE (6,240) IC(IX),IX=1,NX
C ---PRINT MASS BALANCE DATA FOR SOLUTE---
50 RESID=SUMIO-CSTORM
IF (SUMIO.EQ.0.0) GO TO 60
RESID=SUMIO-CSTORM
ERR1=RESID*200.0/(SUMIO+CSTORM)
60 IF (STORMI.EQ.0.0) GO TO 80
ERR3=-100.O*RESID/(STORMI-SUMIO)
WRITE (6,220)
WRITE (6,260) FLMIN
WRITE (6,270) FLMAX
RECDIN=-CMSIN
RECDOUT=-CMSOUT
WRITE (6,290) RECDIN
WRITE (6,300) RECDOUT
WRITE (6,310) STORMI
WRITE (6,320) STORM
WRITE (6,330) CSTORM
IF (SUMIO.EQ.0.0) GO TO 80
WRITE (6,340)
WRITE (6,350) RESID
WRITE (6,360) ERR1
WRITE (6,370) ERR3
80 IF (STORMI.EQ.0.0) GO TO 100
WRITE (6,380)
WRITE (6,390) TITLE
IF (NUMOBS.LE.0) GO TO 150
WRITE (6,400) INT
IF (S.GT.0.0) WRITE (6,410)
IF (S.EQ.0.0) WRITE (6,420)
C ---PRINT HYDROGRAPHS AFTER 50 STEPS OR END OF SIMULATION---
90 IF (MOD(IMOV,N).EQ.0.0) GO TO 100
IF (MOD(N,50).EQ.0.0) GO TO 100
GO TO 150
100 WRITE (6,390) TITLE
WRITE (6,400) INT
C ---TABULATE HYDROGRAPH DATA---
MODZ=0
IF (S.GT.0.0) GO TO 110
NTO=NUMOV
IF (NUMOBS.GT.50) NTO=MOD(IMOV,50)
GO TO 120
110 NTO=NTIM
IF (NTIM.GT.50) NTO=MOD(N,50)
120 IF (NTO.EQ.0) NTO=50
DO 140 J=1,NUMOBS
140
TMYR=0.0
WRITE (6,430) IXOBS(J),IYOBS(J)
WRITE (6,440) M=1,NTO
TMYR=TMOBS(M)/TMYF
140 CONTINUE
150 RETURN
C ****************************************************************
C
C
160 FORMAT (1H1,13HCONCENTRATION/)  
170 FORMAT (1X,23HNUMBER OF TIME STEPS = ,115)  
180 FORMAT (8X,16HDELTA T = ,1G12.5)  
190 FORMAT (8X,16HTIME (SECONDS) = ,1G12.5)  
200 FORMAT (8X,21HCHEM. TIME (DAYS) = ,1E12.5)  
210 FORMAT (8X,16HTIME (YEARS) = ,1E12.5)  
220 FORMAT (1H )  
230 FORMAT (1H,23HCHANGE IN CONCENTRATION/)  
240 FORMAT (1H0,2015)  
250 FORMAT (1H0,21HCHEMICAL MASS BALANCE)  
260 FORMAT (8X,25HMASS IN BOUNDARIES = ,1E12.5)  
270 FORMAT (8X,25HMASS OUT BOUNDARIES = ,1E12.5)  
280 FORMAT (8X,25HMASS PUMPED OUT = ,1E12.5)  
290 FORMAT (8X,25HMASS PUMPED IN = ,1E12.5)  
300 FORMAT (8X,25HINFLOW MINUS OUTFLOW = ,1E12.5)  
310 FORMAT (8X,25HINITIAL MASS STORED = ,1E12.5)  
320 FORMAT (8X,25HPRESENT MASS STORED = ,1E12.5)  
330 FORMAT (8X,25HCHANGE MASS STORED = ,1E12.5)  
340 FORMAT (1H5X,53HCOMPARE RESIDUAL WITH NET FLUX AND MASS ACCUMULATION/)  
350 FORMAT (1H0,25HMASS BALANCE RESIDUAL = ,1E12.5)  
360 FORMAT (8X,25HERROR (AS PERCENT) = ,1E12.5)  
370 FORMAT (1H0,25HCOMPARE INITIAL MASS STORED WITH CHANGE IN MASS STORED/)  
380 FORMAT (1X,23H NO. MOVES COMPLETED = ,115)  
390 FORMAT (1H1,10AB/)  
400 FORMAT (1H0,5X,65HTIME VERSUS HEAD AND CONCENTRATION AT SELECTED OBSERVATION POINTS/)  
410 FORMAT (1H0,16X,19HPUMPING PERIOD NO., ,14///)  
420 FORMAT (1H0,15X,21HSTEADY-STATE SOLUTION///)  
430 FORMAT (1H0,20X,22HOBJS.WELL NO., X, Y, 17X, 1HN, 6X,40HHEAD (FT))  
440 FORMAT (1H58X,12X,6xF7.1,8X,F7.1,8X,F7.2)  
450 FORMAT (1H0,2X,21HCHEM. TIME (SECONDS) = ,1E12.5)  
460 FORMAT (1H0,2X,21HCHEM. TIME (YEARS) = ,1E12.5)  
END
Attachment II
Definition of Selected Program Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAQ</td>
<td>area of aquifer in model</td>
</tr>
<tr>
<td>ALNG</td>
<td>BETA</td>
</tr>
<tr>
<td>ANFCTR</td>
<td>anisotropy factor (ratio of $T_x$ to $T_y$)</td>
</tr>
<tr>
<td>AOPT</td>
<td>iteration parameters</td>
</tr>
<tr>
<td>AREA</td>
<td>area of one cell in finite-difference grid</td>
</tr>
<tr>
<td>BETA</td>
<td>longitudinal dispersivity of porous medium</td>
</tr>
<tr>
<td>CELDIS</td>
<td>maximum distance across one cell that a particle is permitted to move in one step (as fraction of width of cell)</td>
</tr>
<tr>
<td>CLKCN</td>
<td>concentration of leakage through confining layer or streambed</td>
</tr>
<tr>
<td>CMSIN</td>
<td>mass of solute recharged into aquifer</td>
</tr>
<tr>
<td>CMSOUT</td>
<td>mass of solute discharged from aquifer</td>
</tr>
<tr>
<td>CNCCN</td>
<td>change in concentration due to dispersion and sources</td>
</tr>
<tr>
<td>CNCPCT</td>
<td>change in concentration as percentage of concentration at node</td>
</tr>
<tr>
<td>CNOLD</td>
<td>concentration at node at end of previous time increment</td>
</tr>
<tr>
<td>CNREC</td>
<td>concentration of well withdrawal or injection</td>
</tr>
<tr>
<td>CNRECH</td>
<td>concentration in fluid source</td>
</tr>
<tr>
<td>CONC</td>
<td>concentration in aquifer at node</td>
</tr>
<tr>
<td>CONINT</td>
<td>concentration in aquifer at start of simulation</td>
</tr>
<tr>
<td>C1</td>
<td>CONC at node (IX, IY)</td>
</tr>
<tr>
<td>DALN</td>
<td>longitudinal dispersion coefficient</td>
</tr>
<tr>
<td>DDRW</td>
<td>drawdown</td>
</tr>
<tr>
<td>DELQ</td>
<td>volumetric rate of leakage across a confining layer or streambed</td>
</tr>
<tr>
<td>DELS</td>
<td>rate of change in ground-water storage</td>
</tr>
<tr>
<td>DERH</td>
<td>change in head with respect to time</td>
</tr>
<tr>
<td>DISP</td>
<td>dispersion equation coefficients</td>
</tr>
<tr>
<td>DISTX</td>
<td>distance particle moves in $x$-direction during time increment</td>
</tr>
<tr>
<td>DISTY</td>
<td>distance particle moves in $y$-direction during time increment</td>
</tr>
<tr>
<td>DLTRAT</td>
<td>ratio of transverse to longitudinal dispersivity</td>
</tr>
<tr>
<td>DTRN</td>
<td>transverse dispersion coefficient</td>
</tr>
<tr>
<td>FCTR</td>
<td>multiplication or conversion factor</td>
</tr>
<tr>
<td>FLMIN</td>
<td>solute mass entering modeled area during time step</td>
</tr>
<tr>
<td>FLMOT</td>
<td>solute mass leaving modeled area during time step</td>
</tr>
<tr>
<td>GRDX</td>
<td>hydraulic gradient in $x$-direction</td>
</tr>
<tr>
<td>GRDY</td>
<td>hydraulic gradient in $y$-direction</td>
</tr>
<tr>
<td>HC</td>
<td>head from column computation</td>
</tr>
<tr>
<td>HI</td>
<td>initial head in aquifer</td>
</tr>
<tr>
<td>HK</td>
<td>computed head at end of time step</td>
</tr>
<tr>
<td>HMIN</td>
<td>minimum iteration parameter</td>
</tr>
<tr>
<td>HR</td>
<td>head from row computation in subroutine ITERAT; elsewhere HR represents head from previous time step</td>
</tr>
<tr>
<td>IMOV</td>
<td>particle movement step number</td>
</tr>
<tr>
<td>INT</td>
<td>pumping period number</td>
</tr>
<tr>
<td>IPRNT</td>
<td>print control index for hydrographs</td>
</tr>
<tr>
<td>ITMAX</td>
<td>maximum permitted number of iterations</td>
</tr>
<tr>
<td>IROBS</td>
<td>$x$-coordinate of observation point</td>
</tr>
<tr>
<td>IOBYS</td>
<td>$y$-coordinate of observation point</td>
</tr>
<tr>
<td>KOUNT</td>
<td>iteration number for ADIP</td>
</tr>
<tr>
<td>LIMRO</td>
<td>array for temporary storage of particles</td>
</tr>
<tr>
<td>N</td>
<td>time step number</td>
</tr>
<tr>
<td>NCA</td>
<td>number of aquifer nodes in model</td>
</tr>
<tr>
<td>NCODES</td>
<td>number of node identification codes</td>
</tr>
<tr>
<td>NITP</td>
<td>number of iteration parameters</td>
</tr>
<tr>
<td>NMOV</td>
<td>number of particle movements (or time increments) required to complete time step</td>
</tr>
<tr>
<td>NODEID</td>
<td>node identification code</td>
</tr>
<tr>
<td>NP</td>
<td>total number of active particles in grid</td>
</tr>
<tr>
<td>NPCELL</td>
<td>number of particles in a cell during time increment</td>
</tr>
<tr>
<td>NPMAX</td>
<td>maximum number of available particles</td>
</tr>
<tr>
<td>NPM</td>
<td>number of pumping periods or simulation periods</td>
</tr>
<tr>
<td>NPNT</td>
<td>number of time steps between printouts</td>
</tr>
<tr>
<td>NPTPN</td>
<td>initial number of particles per node</td>
</tr>
<tr>
<td>NREC</td>
<td>number of pumping wells</td>
</tr>
<tr>
<td>NTIM</td>
<td>number of time steps</td>
</tr>
<tr>
<td>NUMOBS</td>
<td>number of observation wells</td>
</tr>
<tr>
<td>NX</td>
<td>number of nodes in $x$-direction</td>
</tr>
<tr>
<td>NY</td>
<td>number of nodes in $y$-direction</td>
</tr>
<tr>
<td>NZCRIT</td>
<td>maximum number of cells that can be void of particles</td>
</tr>
<tr>
<td>NZERO</td>
<td>number of cells that are void of particles at the end of a time increment</td>
</tr>
<tr>
<td>PARAM</td>
<td>iteration parameter for current iteration</td>
</tr>
<tr>
<td>PART</td>
<td>1. $x$-coordinate of particle; 2. $y$-coordinate of particle; 3. concentration of particle. Also note that the signs of coordinates are used as flags to store information on original location of particle.</td>
</tr>
<tr>
<td>PERT</td>
<td>hydraulic conductivity (in $LT^{-1}$)</td>
</tr>
<tr>
<td>PINT</td>
<td>pumping period in years</td>
</tr>
<tr>
<td>POROS</td>
<td>effective porosity</td>
</tr>
<tr>
<td>PUMP</td>
<td>cumulative net pumpage</td>
</tr>
<tr>
<td>PYR</td>
<td>total duration of pumping period (in seconds)</td>
</tr>
<tr>
<td>QNET</td>
<td>net water flux (in $LT^{-1}$)</td>
</tr>
</tbody>
</table>
Definition of selected program variables—Continued

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QSTR</td>
<td>cumulative change in volume of water in storage</td>
</tr>
<tr>
<td>REC</td>
<td>point source or sink; negative for injection, positive for withdrawal (in (LT^{-1}))</td>
</tr>
<tr>
<td>RECH</td>
<td>diffuse recharge or discharge; negative for recharge, positive for discharge (in (LT^{-2}))</td>
</tr>
<tr>
<td>RN</td>
<td>range in concentration between regenerated particle and adjacent node having lower concentration</td>
</tr>
<tr>
<td>RP</td>
<td>range in concentration between regenerated particle and adjacent node having higher concentration</td>
</tr>
<tr>
<td>S</td>
<td>storage coefficient (or specific yield)</td>
</tr>
<tr>
<td>SLEAK</td>
<td>rate of leakage through confining layer or streambed</td>
</tr>
<tr>
<td>STORM</td>
<td>change in total solute mass in storage (by summation)</td>
</tr>
<tr>
<td>STORM1</td>
<td>initial mass of solute in storage</td>
</tr>
<tr>
<td>SUMC</td>
<td>summation of concentrations of all particles in a cell</td>
</tr>
<tr>
<td>SUMIO</td>
<td>change in total solute mass in storage (from inflows—outflows)</td>
</tr>
<tr>
<td>SUMT</td>
<td>total elapsed time (in seconds)</td>
</tr>
<tr>
<td>SUMTCH</td>
<td>cumulative elapsed time during particle moves (in seconds)</td>
</tr>
<tr>
<td>THCK</td>
<td>saturated thickness of aquifer</td>
</tr>
<tr>
<td>TIM</td>
<td>length of specific time step (in seconds)</td>
</tr>
<tr>
<td>TIMD</td>
<td>elapsed time in days</td>
</tr>
<tr>
<td>TIMY</td>
<td>elapsed time in years</td>
</tr>
<tr>
<td>TIMV</td>
<td>length of time increment for particle movement (in seconds)</td>
</tr>
<tr>
<td>TIMX</td>
<td>time step multiplier for transient flow problems</td>
</tr>
<tr>
<td>TINIT</td>
<td>size of initial time step for transient flow problems (in seconds)</td>
</tr>
<tr>
<td>TITLE</td>
<td>problem description</td>
</tr>
<tr>
<td>TMCN</td>
<td>computed concentrations at observation points</td>
</tr>
<tr>
<td>TMOBS</td>
<td>elapsed times for observation point records</td>
</tr>
<tr>
<td>TMRX</td>
<td>transmissivity coefficients (harmonic means on cell boundaries; forward values are stored)</td>
</tr>
<tr>
<td>TMWL</td>
<td>computed heads at observation points</td>
</tr>
<tr>
<td>TOL</td>
<td>convergence criteria (ADIP)</td>
</tr>
<tr>
<td>TOTLQ</td>
<td>cumulative net leakage through confining layer or streambed</td>
</tr>
<tr>
<td>TRAN</td>
<td>transverse dispersivity of porous medium</td>
</tr>
<tr>
<td>VMAX</td>
<td>maximum value of VX</td>
</tr>
<tr>
<td>VMAXY</td>
<td>maximum value of VY</td>
</tr>
<tr>
<td>VMGE</td>
<td>magnitude of velocity vector</td>
</tr>
<tr>
<td>VMXBD</td>
<td>maximum value of VXBDY</td>
</tr>
<tr>
<td>VMYBD</td>
<td>maximum value of VYBDY</td>
</tr>
<tr>
<td>VPRM</td>
<td>initially used to read transmissivity values at nodes; then after line B22', VPRM equals leakance factor for confining layer or streambed (vertical hydraulic conductivity/ thickness). If VPRM≥0.09, then the program assumes that the node is a constant-head boundary and is flagged for subsequent special treatment in calculating convective transport.</td>
</tr>
<tr>
<td>VX</td>
<td>velocity in z-direction at a node</td>
</tr>
<tr>
<td>VXBDY</td>
<td>velocity in z-direction on a boundary between nodes</td>
</tr>
<tr>
<td>VY</td>
<td>velocity in y-direction at a node</td>
</tr>
<tr>
<td>VYBDY</td>
<td>velocity in y-direction on a boundary between nodes</td>
</tr>
<tr>
<td>WT</td>
<td>initial water-table or potentiometric elevation, or constant head in stream or source bed</td>
</tr>
<tr>
<td>XDEL</td>
<td>grid spacing in x-direction</td>
</tr>
<tr>
<td>XOLD</td>
<td>x-coordinate of particle at end of previous time increment</td>
</tr>
<tr>
<td>XVEL</td>
<td>velocity of particle in x-direction</td>
</tr>
<tr>
<td>YDEL</td>
<td>grid spacing in y-direction</td>
</tr>
<tr>
<td>YOLD</td>
<td>y-coordinate of particle at end of previous time increment</td>
</tr>
<tr>
<td>YVEL</td>
<td>velocity of particle in y-direction</td>
</tr>
</tbody>
</table>
## Attachment III
### Data Input Formats

<table>
<thead>
<tr>
<th>Card</th>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description of problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-80</td>
<td>10A8</td>
<td>TITLE</td>
<td>Maximum number of time steps in a pumping period (limit=100)*.</td>
</tr>
<tr>
<td>2</td>
<td>1-4</td>
<td>I4</td>
<td>NTIM</td>
<td>Number of pumping periods. Note that if NPMP&gt;1, then data set 10 must be completed.</td>
</tr>
<tr>
<td>5-8</td>
<td>I4</td>
<td>NPMP</td>
<td></td>
<td>Number of pumping periods. Note that if NPMP&gt;1, then data set 10 must be completed.</td>
</tr>
<tr>
<td>9-12</td>
<td>I4</td>
<td>NX</td>
<td></td>
<td>Number of nodes in z direction (limit=20)*.</td>
</tr>
<tr>
<td>13-16</td>
<td>I4</td>
<td>NY</td>
<td></td>
<td>Number of nodes in y direction (limit=20)*.</td>
</tr>
<tr>
<td>17-20</td>
<td>I4</td>
<td>NPMax</td>
<td></td>
<td>Maximum number of particles (limit=3200)*. (See eq 71.)</td>
</tr>
<tr>
<td>21-24</td>
<td>I4</td>
<td>NPNT</td>
<td></td>
<td>Time-step interval for printing hydraulic and chemical output data.</td>
</tr>
<tr>
<td>25-28</td>
<td>I4</td>
<td>NITP</td>
<td></td>
<td>Number of iteration parameters (usually 4≤NITP=7).</td>
</tr>
<tr>
<td>29-32</td>
<td>I4</td>
<td>NUMOBS</td>
<td></td>
<td>Number of observation points to be specified in a following data set (limit=5)*.</td>
</tr>
<tr>
<td>33-36</td>
<td>I4</td>
<td>ITMAX</td>
<td></td>
<td>Maximum allowable number of iterations in ADIP (usually 100=ITMAX=200).</td>
</tr>
<tr>
<td>37-40</td>
<td>I4</td>
<td>NREC</td>
<td></td>
<td>Number of pumping or injection wells to be specified in a following data set.</td>
</tr>
<tr>
<td>41-44</td>
<td>I4</td>
<td>NPTPND</td>
<td></td>
<td>Initial number of particles per node (options=4, 5, 8, 9).</td>
</tr>
<tr>
<td>45-48</td>
<td>I4</td>
<td>NCODES</td>
<td></td>
<td>Number of node identification codes to be specified in a following data set (limit=10)*.</td>
</tr>
<tr>
<td>49-52</td>
<td>I4</td>
<td>NPNTMV</td>
<td></td>
<td>Particle movement interval (IMOV) for printing chemical output data. (Specify 0 to print only at end of time steps.)</td>
</tr>
<tr>
<td>53-56</td>
<td>I4</td>
<td>NPNTVL</td>
<td></td>
<td>Option for printing computed velocities (0=do not print; 1=print for first time step; 2=print for all time steps).</td>
</tr>
<tr>
<td>57-60</td>
<td>I4</td>
<td>NPNTD</td>
<td></td>
<td>Option for printing computed dispersion equation coefficients (option definition same as for NPNTVL).</td>
</tr>
<tr>
<td>61-64</td>
<td>I4</td>
<td>NPDELIC</td>
<td></td>
<td>Option for printing computed changes in concentration (0=do not print; 1=print).</td>
</tr>
<tr>
<td>65-68</td>
<td>I4</td>
<td>NPNCHV</td>
<td></td>
<td>Option to punch velocity data (option definition same as for NPNTVL). When specified, program will punch on unit 7 the velocities at nodes.</td>
</tr>
</tbody>
</table>

*See footnotes at end of table.*
### Data input formats—Continued

<table>
<thead>
<tr>
<th>Card</th>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1–5</td>
<td>G5.0</td>
<td>PINT</td>
<td>Pumping period in years.</td>
</tr>
<tr>
<td></td>
<td>6–10</td>
<td>G5.0</td>
<td>TOL</td>
<td>Convergence criteria in ADIP (usually TOL≤0.01).</td>
</tr>
<tr>
<td>11–15</td>
<td></td>
<td>G5.0</td>
<td>POROS</td>
<td>Effective porosity.</td>
</tr>
<tr>
<td>16–20</td>
<td></td>
<td>G5.0</td>
<td>BETA</td>
<td>Characteristic length, in feet (=longitudinal dispersivity).</td>
</tr>
<tr>
<td>21–25</td>
<td></td>
<td>G5.0</td>
<td>S</td>
<td>Storage coefficient (set $S=0$ for steady flow problems).</td>
</tr>
<tr>
<td>26–30</td>
<td></td>
<td>G5.0</td>
<td>TIMX</td>
<td>Time increment multiplier for transient flow problems. TIMX is disregarded if $S=0$.</td>
</tr>
<tr>
<td>31–35</td>
<td></td>
<td>G5.0</td>
<td>TINIT</td>
<td>Size of initial time step in seconds. TINIT is disregarded if $S=0$.</td>
</tr>
<tr>
<td>36–40</td>
<td></td>
<td>G5.0</td>
<td>XDEL</td>
<td>Width of finite-difference cell in $x$ direction, in feet.</td>
</tr>
<tr>
<td>41–45</td>
<td></td>
<td>G5.0</td>
<td>YDEL</td>
<td>Width in finite-difference cell in $y$ direction, in feet.</td>
</tr>
<tr>
<td>46–50</td>
<td></td>
<td>G5.0</td>
<td>DLTRAT</td>
<td>Ratio of transverse to longitudinal dispersivity.</td>
</tr>
<tr>
<td>51–55</td>
<td></td>
<td>G5.0</td>
<td>CELDIS</td>
<td>Maximum cell distance per particle move (value between 0 and 1.0).</td>
</tr>
<tr>
<td>56–60</td>
<td></td>
<td>G5.0</td>
<td>ANFCTR</td>
<td>Ratio of $T_v$ to $T_s$.</td>
</tr>
</tbody>
</table>

### Data set

<table>
<thead>
<tr>
<th>Data set</th>
<th>Number of cards</th>
<th>Format</th>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Value of NUMOBS (limit=5) *</td>
<td>2I2</td>
<td>IXOBS, IYOBS</td>
<td>$x$ and $y$ coordinates of observation points. This data set is eliminated if NUMOBS is specified as =0.</td>
</tr>
<tr>
<td>2</td>
<td>Value of NREC</td>
<td>2I2, 2G6.2</td>
<td>IX, IY, REC, CNRECH</td>
<td>$x$ and $y$ coordinates of pumping (+) or injection (−) wells, rate in $ft^3/s$, and if an injection well, the concentration of injected water. This data set is eliminated if NREC=0.</td>
</tr>
<tr>
<td>3</td>
<td>a. 1</td>
<td>I1, G10.0</td>
<td>INPUT, FCTR</td>
<td>Parameter card for transmissivity.</td>
</tr>
<tr>
<td></td>
<td>b. Value of NY</td>
<td>2G04.1</td>
<td>VPRM</td>
<td>Array for temporary storage of transmissivity data, in $ft^3/s$. For an anisotropic aquifer, read in values of $T_s$ and the program will adjust for anisotropy by multiplying $T_v$ by ANFCTR.</td>
</tr>
<tr>
<td>4</td>
<td>a. 1</td>
<td>I1, G10.0</td>
<td>INPUT, FCTR</td>
<td>Parameter card for THICK.</td>
</tr>
<tr>
<td></td>
<td>b. Value of NY</td>
<td>2G03.0</td>
<td>THICK</td>
<td>Saturated thickness of aquifer, in feet.</td>
</tr>
<tr>
<td>5</td>
<td>a. 1</td>
<td>I1, G10.0</td>
<td>INPUT, FCTR</td>
<td>Parameter card for RECH.</td>
</tr>
<tr>
<td></td>
<td>b. Value of NY</td>
<td>2G04.1</td>
<td>RECH</td>
<td>Diffuse recharge (−) or discharge (+), in $ft/s$.</td>
</tr>
<tr>
<td>6</td>
<td>a. 1</td>
<td>I1, G10.0</td>
<td>INPUT, FCTR</td>
<td>Parameter card for NODEID.</td>
</tr>
<tr>
<td></td>
<td>b. Value of NY</td>
<td>201</td>
<td>NODEID</td>
<td>Node identification matrix (used to define constant-head nodes or other boundary conditions and stresses).</td>
</tr>
</tbody>
</table>

---

See footnotes at end of table.
### TECHNIQUES OF WATER-RESOURCES INVESTIGATIONS

#### Data input formats—Continued

<table>
<thead>
<tr>
<th>Data set</th>
<th>Number of cards</th>
<th>Format</th>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
<td>I2, 3G10.2, ICODE, FCTR1, FCTR2, FCTR3, OVERRD</td>
<td>Value of NCODES (limit=10)*</td>
<td>Instructions for using NODEID array. When NODEID=ICODE, program sets leakance=FCTR1, CNRECH=FCTR2, and if OVERRD is nonzero, RECH=FCTR3. Set OVERRD=0 to preserve values of RECH specified in data set 5.</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>I1, G10.0 INPUT, FCTR</td>
<td>b. Value of NY (limit=20)*</td>
<td>Parameter card† for WT. Initial water-table or potentiometric elevation, or constant head in stream or source bed, in feet.</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>I1, G10.0 INPUT, FCTR</td>
<td>a. 1</td>
<td>Parameter card† for CONC. Initial concentration in aquifer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20G4.0 WT</td>
<td>b. Value of NY (limit=20)*</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>I1</td>
<td>a. 1</td>
<td>This data set allows time step parameters, print options, and pumping data to be revised for each pumping period of the simulation. Data set 10 is only used if NPMP &gt;1. The sequence of cards in data set 10 must be repeated (NPMP -1) times (that is, data set 10 is required for each pumping period after the first).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1014,3G5.0 NTIM, NPNT, NITP, ITMAX, NREC, NPNTMV, NPNTVL, NPNTD, NPDELC, NPNCHV, PINT, TIMX, TINIT</td>
<td>b. 1</td>
<td>Parameter to check whether any revisions are desired. Set ICHK=1 if data are to be revised, and then complete data set 10b and c. Set ICHK=0 if data are not to be revised for the next pumping period, and skip rest of data set 10.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2I2, 2G8.2 IX, IY, REC, CNRECH</td>
<td>c. Value of NREC</td>
<td>Thirteen parameters to be revised for next pumping period; the parameters were previously defined in the description of data cards 2 and 3. Only include this card if ICHK=1 in previous part a.</td>
</tr>
</tbody>
</table>

---

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

† The parameter card must be the first card 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 from cards as described by part b. Then FCTR is a multiplication factor for the values read in the data set.
Attachment IV

Input Data for Test Problem 3

Card 1  TEST PROBLEM NO. 3 (STEADY FLOW, 1 WELL, CONSTANT-HEAD BOUNDARIES)
Card 2  1 1 9 10320 1 7 2 100 1 9 2 10 1 0 0 0
Card 3  2.5 0.001 0.30 100 0.0 0.0 0.0 900 900 0.3 0.50 1.0

Data Set 1  
  5 4
  5 7

Data Set 2  4 7 1.0

Data Set 3  0 0.1

Data Set 4  0 20.0

Data Set 5  0 0.0

  1 1.0
  000000000
  022111220
  000000000
  000000000

Data Set 6  
  000000000
  000000000
  000000000
  022222222
  000000000

Data Set 7  
  2 1.0 0.0 0.0 0
  1 1.0 100.0 0.0 0
  1 1.0

  0.0100 100.100 100 100.100 100.0 0.0

Data Set 8

  0.0 75. 75. 75. 75. 75. 75. 0.0

Data Set 9  0 0.0
Attachment V
Selected Output for Test Problem 3

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

TEST PROBLEM NO. 3 (STEADY FLOW, 1 WELL, CONSTANT-HEAD BOUNDARIES)

INPUT DATA

GRID DESCRIPTORS

NX (NUMBER OF COLUMNS) = 9
NY (NUMBER OF ROWS) = 10
XDEL (X-DISTANCE IN FEET) = 900.0
YDEL (Y-DISTANCE IN FEET) = 900.0

TIME PARAMETERS

NTIM (MAX. NO. OF TIME STEPS) = 1
NPMP (NO. OF PUMPING PERIODS) = 1
PINT (PUMPING PERIOD IN YEARS) = 2.50
TIMX (TIME INCREMENT MULTIPLIER) = 0.00
TINIT (INITIAL TIME STEP IN SEC.) = 0.

HYDROLOGIC AND CHEMICAL PARAMETERS

S (STORAGE COEFFICIENT) = 0.600006
POROS (EFFECTIVE POROSITY) = 0.30
BETA (CHARACTERISTIC LENGTH) = 100.0
DLTRAT (RATIO OF TRANSVERSE TO LONGITUDINAL DISPERSIVITY) = 0.30
ANFCLN (RATIO OF T-YY TO T-XX) = 1.600000

EXECUTION PARAMETERS

NITP (NO. OF ITERATION PARAMETERS) = 7
TOL (CONVERGENCE CRITERIA - ADIP) = 0.0001
ITMAX (MAX. NO. OF ITERATIONS - ADIP) = 100
CELDIS (MAX. CELL DISTANCE PER MOVE OF PARTICLES - M.O.C.) = 0.500
NPMAX (MAX. NO. OF PARTICLES) = 3200
NPTPAD (NO. PARTICLES PER NODE) = 9

PROGRAM OPTIONS

NPNT (TIME STEP INTERVAL FOR COMPLETE PRINTOUT) = 1
NPNTKV (MOVE INTERVAL FOR CHEM. CONCENTRATION PRINTOUT) = 10
NPNTVL (PRINT OPTION=VELOCITY
O=NO; 1=FIRST TIME STEP;
2=ALL TIME STEPS) = 1
NPNTD (PRINT OPTION=DISP.COEFF.
O=NO; 1=FIRST TIME STEP;
2=ALL TIME STEPS) = 0
NUMOBS (NO. OF OBSERVATION WELLS FOR HYDROGRAPH PRINTOUT) = 2
NPAC (NO. OF PUMPING WELLS) = 1
NCODES (FOR NODE IDENT.) = 2
NPNCWH (PUNCH VELOCITIES) = 0
NPDELG (PRINT OPT.-CONC. CHANGE) = 0
MODEL OF SOLUTE TRANSPORT IN GROUND WATER

Selected output for test problem 3—Continued

STEADY-STATE FLOW

TIME INTERVAL (IN SEC) FOR SOLUTE-TRANSER SIMULATION = 0.78894d+08

LOCATION OF OBSERVATION WELLS

<table>
<thead>
<tr>
<th>NO.</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

LOCATION OF PUMPING WELLS

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>RATE (IN CFS)</th>
<th>CONC.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>7</td>
<td>1.00</td>
<td>0.0</td>
</tr>
</tbody>
</table>

AREA OF ONE CELL = 0.8100d+06

X-Y SPACING:
900.00
900.00

TRANSMISSIVITY MAP (FT*FT/SEC)

<table>
<thead>
<tr>
<th></th>
<th>0.00</th>
<th>0.00</th>
<th>0.00</th>
<th>0.00</th>
<th>0.00</th>
<th>0.00</th>
<th>0.00</th>
<th>0.00</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
</tr>
</tbody>
</table>

AQUIFER THICKNESS (FT)

<table>
<thead>
<tr>
<th></th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>0.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>0.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>0.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>0.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>0.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>0.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>0.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
</tbody>
</table>
### Diffuse Recharge and Discharge (ft/sec)

<table>
<thead>
<tr>
<th></th>
<th>0.00d+00</th>
<th>0.00d+00</th>
<th>0.00d+00</th>
<th>0.00d+00</th>
<th>0.00d+00</th>
<th>0.00d+00</th>
<th>0.00d+00</th>
<th>0.00d+00</th>
<th>3.33d+00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>3.33d+00</td>
</tr>
<tr>
<td>0.00d+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>3.33d+00</td>
</tr>
<tr>
<td>0.00d+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>0.00a+00</td>
<td>3.33d+00</td>
</tr>
<tr>
<td></td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>0.00d+00</td>
<td>3.33d+00</td>
</tr>
</tbody>
</table>

### Permeability Map (ft/sec)

<table>
<thead>
<tr>
<th></th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

### Summary

- **No. of Finite-Difference Cells in Aquifer**: 56
- **Area of Aquifer in Model**: 0.45360e+08 sq. ft.
- **NZCRIT (Max. No. of Cells That Can Be Void of Particles; If Exceeded, Particles Are Regenerated)**: 1
MODEL OF SOLUTE TRANSPORT IN GROUND WATER

Selected output for test problem 3—Continued

NODE IDENTIFICATION MAP

0 0 0 0 0 0 0 0 0
0 2 2 1 1 1 2 2 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 2 2 2 2 2 2 2 0
0 0 0 0 0 0 0 0 0

NO. OF NODE IDENT. CODES SPECIFIED = 2

THE FOLLOWING ASSIGNMENTS HAVE BEEN MADE:

<table>
<thead>
<tr>
<th>CODE NO.</th>
<th>LEAKANCE</th>
<th>SOURCE CONC.</th>
<th>RECHARGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.00e+01</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.100e+01</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

VERTICAL PERMEABILITY/THICKNESS (FT/(FT*SEC))

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 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 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 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

WATER TABLE

0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 100. 100. 100. 100. 100. 100. 100. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 75. 75. 75. 75. 75. 75. 75. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0.
Selected output for test problem 3—Continued

ITERATION PARAMETERS
0.246749e-01
0.457299e-01
0.847539e-01
.157080
.291125
.539560
1.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000

CONCENTRATION

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

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

N = 1
NUMBER OF ITERATIONS = 20
### Head Distribution - Row

**Number of Time Steps** = 1

**Time (Seconds)** = 0.78894d+08

**Time (Days)** = 0.91313e+03

**Time (Years)** = 0.25000e+01

<table>
<thead>
<tr>
<th>Z</th>
<th>0.0000000000</th>
<th>0.0000000000</th>
<th>0.0000000000</th>
<th>0.0000000000</th>
<th>0.0000000000</th>
<th>0.0000000000</th>
<th>0.0000000000</th>
<th>0.0000000000</th>
<th>0.0000000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---

### Selected output for test problem 3—Continued
### Selected output for test problem 3—Continued

#### Drawdown

| Time  | Drawdown
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>-95 -95 -95 -95 -95 -95</td>
</tr>
<tr>
<td>0</td>
<td>-91 -91 -91 -91 -91 -91</td>
</tr>
<tr>
<td>0</td>
<td>-87 -87 -87 -87 -87 -87</td>
</tr>
<tr>
<td>0</td>
<td>-83 -83 -82 -83 -83 -84</td>
</tr>
<tr>
<td>0</td>
<td>-79 -79 -76 -79 -80 -80</td>
</tr>
<tr>
<td>0</td>
<td>-77 -76 -76 -77 -77 -77</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

#### Cumulative Mass Balance

- **Cumulative Net Pumpage**: $0.78894e+08$
- **Water Release from Storage**: $0.00000e+00$
- **Cumulative Net Leakage**: $0.78895e+08$
- **Mass Balance Residual**: $-767.00$
- **Error (as percent)**: $-0.97219e-03$

#### Rate Mass Balance — (in C.F.S.)

- **In**: 2.7857
- **Out**: -1.7857
- **Net**: 1.0000

#### X Velocities

<table>
<thead>
<tr>
<th>Time</th>
<th>Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.000</td>
<td>0.935e-14 -0.924e-14</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.699e-13 -0.131e-12 -0.139e-12 -0.996e-13 -0.712e-13</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.757e-07 -0.566e-06 -0.106e-05 -0.112e-05 -0.807e-05 -0.577e-05</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.528e-06 -0.229e-06 -0.113e-05 -0.254e-05 -0.263e-05 -0.182e-05 -0.127e-05</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.217e-05 -0.186e-05 -0.165e-05 -0.536e-05 -0.502e-05 -0.320e-05 -0.214e-05</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.628e-05 -0.718e-05 -0.198e-05 -0.122e-04 -0.891e-05 -0.488e-05 -0.305e-05</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.137e-04 -0.282e-04 -0.186e-05 -0.326e-04 -0.135e-04 -0.588e-05 -0.337e-05</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.573e-05 -0.749e-05 -0.112e-05 -0.101e-04 -0.677e-05 -0.342e-05 -0.204e-05</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.708e-12 -0.925e-12 -0.139e-12 -0.125e-11 -0.835e-12 -0.422e-12 -0.252e-12</td>
</tr>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

#### X Velocities — At Nodes

<table>
<thead>
<tr>
<th>Time</th>
<th>Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.000</td>
<td>0.935e-14 -0.278e-13 -0.112e-12 -0.149e-12 -0.128e-12 -0.712e-13</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.757e-07 -0.908e-06 -0.121e-05 -0.104e-05 -0.577e-05</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.528e-06 -0.697e-07 -0.220e-05 -0.289e-05 -0.236e-05 -0.127e-05</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.217e-05 -0.161e-05 -0.492e-05 -0.579e-05 -0.425e-05 -0.214e-05</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.628e-05 -0.934e-05 -0.133e-04 -0.114e-04 -0.670e-05 -0.305e-05</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.137e-04 -0.427e-04 -0.463e-04 -0.187e-04 -0.839e-04 -0.337e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.573e-05 -0.925e-05 -0.115e-04 -0.874e-05 -0.479e-05 -0.204e-05</td>
</tr>
<tr>
<td>0.000</td>
<td>-0.708e-12 -0.114e-11 -0.142e-11 -0.108e-11 -0.592e-12 -0.252e-12</td>
</tr>
</tbody>
</table>
### Selected output for test problem 3—Continued

#### Y Velocities at Nodes

<table>
<thead>
<tr>
<th>Y</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.752e-04</td>
<td>0.753e-04</td>
<td>0.751e-04</td>
<td>0.742e-04</td>
<td>0.729e-04</td>
<td>0.719e-04</td>
<td>0.711e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.752e-04</td>
<td>0.754e-04</td>
<td>0.756e-04</td>
<td>0.743e-04</td>
<td>0.729e-04</td>
<td>0.717e-04</td>
<td>0.710e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.748e-04</td>
<td>0.759e-04</td>
<td>0.768e-04</td>
<td>0.748e-04</td>
<td>0.725e-04</td>
<td>0.709e-04</td>
<td>0.701e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.736e-04</td>
<td>0.764e-04</td>
<td>0.811e-04</td>
<td>0.756e-04</td>
<td>0.715e-04</td>
<td>0.693e-04</td>
<td>0.684e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.694e-04</td>
<td>0.751e-04</td>
<td>0.957e-04</td>
<td>0.749e-04</td>
<td>0.685e-04</td>
<td>0.664e-04</td>
<td>0.658e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.594e-04</td>
<td>0.591e-04</td>
<td>0.590e-04</td>
<td>0.599e-04</td>
<td>0.611e-04</td>
<td>0.621e-04</td>
<td>0.626e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.497e-04</td>
<td>0.428e-04</td>
<td>0.214e-04</td>
<td>0.447e-04</td>
<td>0.540e-04</td>
<td>0.582e-04</td>
<td>0.599e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.468e-04</td>
<td>0.411e-04</td>
<td>0.318e-04</td>
<td>0.433e-04</td>
<td>0.520e-04</td>
<td>0.568e-04</td>
<td>0.589e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

#### On Boundaries

<table>
<thead>
<tr>
<th>Y</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.752e-04</td>
<td>0.753e-04</td>
<td>0.751e-04</td>
<td>0.742e-04</td>
<td>0.729e-04</td>
<td>0.719e-04</td>
<td>0.711e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.751e-04</td>
<td>0.756e-04</td>
<td>0.757e-04</td>
<td>0.745e-04</td>
<td>0.728e-04</td>
<td>0.714e-04</td>
<td>0.708e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.746e-04</td>
<td>0.762e-04</td>
<td>0.779e-04</td>
<td>0.751e-04</td>
<td>0.722e-04</td>
<td>0.704e-04</td>
<td>0.695e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.725e-04</td>
<td>0.767e-04</td>
<td>0.844e-04</td>
<td>0.760e-04</td>
<td>0.707e-04</td>
<td>0.682e-04</td>
<td>0.673e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.662e-04</td>
<td>0.736e-04</td>
<td>0.107e-03</td>
<td>0.738e-04</td>
<td>0.665e-04</td>
<td>0.646e-04</td>
<td>0.643e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.525e-04</td>
<td>0.644e-04</td>
<td>0.111e-04</td>
<td>0.461e-04</td>
<td>0.560e-04</td>
<td>0.596e-04</td>
<td>0.609e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.468e-04</td>
<td>0.411e-04</td>
<td>0.318e-04</td>
<td>0.433e-04</td>
<td>0.520e-04</td>
<td>0.568e-04</td>
<td>0.589e-04</td>
</tr>
<tr>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

#### Stability Criteria --- M.O.C.

- \( VMAX = 3.26e-05 \)
- \( VMAX = 9.57e-05 \)
- \( VMXB = 4.65e-05 \)
- \( VMXB = 1.07e-04 \)
- \( TMV (MAX, INJ.) = 0.11955e+08 \)
- \( TMV (CELDR) = 0.142045e+07 \)
- \( TIMV = 4.20e+06 \)
- \( NTIMV = 18 \)
- \( NMOV = 19 \)
- \( TIM (N) = C.78894d+08 \)
- \( TIMEUL = C.41523e+07 \)
- \( TIMSP = C.30413e+08 \)
- \( TIMV = 4.15e+06 \)
- \( NTIMD = 2 \)
- \( NMOV = 19 \)

*The limiting stability criterion is CELDIS*

*No. of particle moves required to complete this time step = 19*
**Selected output for test problem 3—Continued**

**CONCENTRATION**

**NUMBER OF TIME STEPS** = 1

**DELTA T** = 0.788944e+08

**TIME (SECONDS)** = 0.788944e+08

**CHEM. TIME (SECONDS)** = 0.788944e+08

**CHEM. TIME (DAYS)** = 0.91313e+03

**TIME (YEARS)** = 0.25000e+01

**CHEM. TIME (YEARS)** = 0.25000e+01

**NO. MOVES COMPLETED** = 19

0 0 0 0 0 0 0 0 0 0
0 0 2 98 100 98 2 0 0 0
0 0 4 96 100 96 4 0 0 0
0 0 7 92 99 93 7 0 0 0
0 0 9 89 96 88 9 0 0 0
0 1 10 81 89 80 10 1 0 0
0 1 8 56 73 46 8 1 0 0
0 0 2 20 35 19 3 0 0 0
0 0 0 1 5 3 0 0 0 0
0 0 0 0 0 0 0 0 0 0

**CHEMICAL MASS BALANCE**

**MASS IN BOUNDARIES** = 0.94642e+10

**MASS OUT BOUNDARIES** = -0.13340e+08

**MASS PUMPED IN** = 0.00000e+00

**MASS PUMPED OUT** = -0.96281e+09

**INFLOW MINUS OUTFLOW** = 0.84881e+10

**INITIAL MASS STORED** = 0.00000e+00

**PRESENT MASS STORED** = 0.84631e+10

**CHANGE MASS STORED** = 0.84631e+10

**COMPARE RESIDUAL WITH NET FLUX AND MASS ACCUMULATION**

**MASS BALANCE RESIDUAL** = 0.24910e+08

**ERROR (AS PERCENT)** = 0.29390e+00
**Selected output for test problem 3—Continued**

**TEST PROBLEM NO. 3 (STEADY FLOW, 1 WELL, CONSTANT-HEAD BOUNDARIES)**

**TIME VERSUS HEAD AND CONCENTRATION AT SELECTED OBSERVATION POINTS**

**PUMPING PERIOD NO. 1**

**STEADY-STATE SOLUTION**

<table>
<thead>
<tr>
<th>OBS. WELL NO.</th>
<th>X</th>
<th>Y</th>
<th>N</th>
<th>HEAD (FT)</th>
<th>CONC. (MG/L)</th>
<th>TIME (YEARS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>4</td>
<td></td>
<td>0.0</td>
<td>3.0</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>92.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.13</td>
</tr>
<tr>
<td>3</td>
<td>92.0</td>
<td>0.2</td>
<td>1.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>2.9</td>
<td>6</td>
<td>2.9</td>
<td>0.39</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>15.5</td>
<td>0.66</td>
<td>15.5</td>
<td>0.66</td>
<td>0.79</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>33.0</td>
<td>1.31</td>
<td>33.0</td>
<td>1.31</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>53.1</td>
<td>1.05</td>
<td>53.1</td>
<td>1.05</td>
<td>1.18</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>64.6</td>
<td>1.18</td>
<td>64.6</td>
<td>1.18</td>
<td>1.32</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>72.9</td>
<td>1.32</td>
<td>72.9</td>
<td>1.32</td>
<td>1.45</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>79.8</td>
<td>1.45</td>
<td>79.8</td>
<td>1.45</td>
<td>1.58</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>85.4</td>
<td>1.58</td>
<td>85.4</td>
<td>1.58</td>
<td>1.71</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>89.4</td>
<td>1.71</td>
<td>89.4</td>
<td>1.71</td>
<td>1.84</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>92.2</td>
<td>1.84</td>
<td>92.2</td>
<td>1.84</td>
<td>1.97</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>94.3</td>
<td>1.97</td>
<td>94.3</td>
<td>1.97</td>
<td>2.11</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>95.8</td>
<td>2.11</td>
<td>95.8</td>
<td>2.11</td>
<td>2.24</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>97.8</td>
<td>2.24</td>
<td>97.8</td>
<td>2.24</td>
<td>2.37</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>98.4</td>
<td>2.37</td>
<td>98.4</td>
<td>2.37</td>
<td>2.50</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>98.7</td>
<td>2.50</td>
<td>98.7</td>
<td>2.50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Selected output for test problem 3—Continued

<table>
<thead>
<tr>
<th>TIME (YEARS)</th>
<th>CONC. (MG/L)</th>
<th>N</th>
<th>HEAD (FT)</th>
<th>X</th>
<th>Y</th>
<th>OBS. WELL NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>0.3</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>0.4</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>0.5</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>0.6</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>0.7</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>0.8</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
</tbody>
</table>

*U.S. GOVERNMENT PRINTING OFFICE: 1978—201—359.159*