A Three-Dimensional Method-of-Characteristics Solute-Transport Model *(MOC3D)*

By L.F. Konikow, D.J. Goode, and G.Z. Hornberger

U. S. GEOLOGICAL SURVEY

Water-Resources Investigations Report 96-4267

Reston, Virginia
1996

U.S. DEPARTMENT OF THE INTERIOR
BRUCE BABBITT, Secretary

U.S. GEOLOGICAL SURVEY
Gordon P. Eaton, Director

The use of firm, trade, and brand names in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey.

For additional information write to:

Leonard F. Konikow
U.S. Geological Survey
Water Resources Division
431 National Center
Reston, VA 20192

Copies of this report can be purchased from:

U.S. Geological Survey
Information Services
Box 25286
Federal Center
Denver, CO 80225

PREFACE

The computer model described in this report is designed to simulate the transport and dispersion of a single solute in ground water flowing through porous media. The model is developed as a module for the U.S. Geological Survey's (USGS) *MODFLOW* ground-water model, and it is intended to be the first of a family of alternative solution methods for the solute-transport equation that will be compatible with *MODFLOW.*

This model, named *MOC3D,* was developed through modifications of an existing two-dimensional code (*MOC*), which was documented originally by Konikow and Bredehoeft (1978). Although extensive testing of *MOC3D* indicates that this model will yield reliable calculations for a wide variety of field problems, the user is cautioned that the accuracy and efficiency of the model can be affected significantly for certain combinations of values for parameters and boundary conditions. Development of alternative codes that will optimize the accuracy and efficiency of solving the solute-transport equation for a broader range of conditions is planned.

The code for this model will be available for downloading over the Internet from a USGS software repository. The repository is accessible on the World Wide Web (WWW) from the USGS Water Resources Information web page at URL **http://h2o.usgs.gov/**. The URL for the public repository is: **http://h2o.usgs.gov/software/**. The public anonymous FTP site is on the Water Resources Information server (h2o.usgs.gov or 130.11.50.175) in the /pub/software directory. When this code is revised or updated in the future, new versions or releases will be made available for downloading from these same sites.

Acknowledgments. The authors appreciate the helpful model evaluation and review comments provided by USGS colleagues H. I. Essaid, W. B. Fleck, and S. P. Garabedian.

CONTENTS

Page

ABSTRACT 1

INTRODUCTION 1

THEORETICAL BACKGROUND AND GOVERNING EQUATIONS 3

- Ground-Water Flow Equation 3
- Average Interstitial Velocity 4
- Governing Equation for Solute Transport 4
- Dispersion Coefficient 6
- Review of Assumptions 8

NUMERICAL METHODS 8

- Ground-Water Flow Equation 9
- Average Interstitial Velocity 10
- Solute-Transport Equation 11
  - Method of Characteristics 11
  - Particle Tracking 14
    - Linear interpolation and semi-analytic integration 15
    - Bilinear interpolation and explicit integration 18
    - Discussion—Choosing an appropriate interpolation scheme 20
  - Decay 20
  - Node Concentrations 21
  - Finite-Difference Approximations 21
  - Stability and Accuracy Criteria 23
  - Mass Balance 25
  - Special Problems 26
  - Review of *MOC3D* Assumptions and Integration with *MODFLOW* 30

COMPUTER PROGRAM 33

- General Program Features 34
- Program Segments 35

MODEL TESTING AND EVALUATION 41

- One-Dimensional Steady Flow 41
- Three-Dimensional Steady Flow 45
- Two-Dimensional Radial Flow and Dispersion 49
- Point Initial Condition in Uniform Flow 50
- Constant Source in Nonuniform Flow 55
- Relative Computational Efficiency 60

CONCLUSIONS 62

REFERENCES 62

APPENDIX A

- FINITE-DIFFERENCE APPROXIMATIONS 64

APPENDIX B

- DATA INPUT INSTRUCTIONS FOR *MOC3D* 70

Page

MODFLOW Name File 70

MODFLOW Source and Sink Packages 70

MOC3D Input Data Files 70

MOC3D Transport Name File (CONC) 71

Main *MOC3D* Package Input (MOC) 72

Source Concentration in Recharge File (CRCH) 77

Observation Well File (OBS) 77

APPENDIX C

ANNOTATED EXAMPLE INPUT DATA SET FOR SAMPLE PROBLEM 78

APPENDIX D

SELECTED OUTPUT FOR SAMPLE PROBLEM 82

FIGURES

1–3. Diagrams showing:

1. Notation used to label rows, columns, and nodes within one layer (k) of a three-dimensional, block-centered, finite-difference grid for *MOC3D* 9

2. Representative three-dimensional grid for *MOC3D* illustrating notation for layers ... 9

3. Part of a hypothetical finite-difference grid showing relation of flow field to movement of points (or particles) in method-of-characteristics model for simulating solute transport (modified from Konikow and Bredehoeft, 1978) 12

4. Graph showing representative change in breakthrough curve from time level t to t+1. Note that concentration changes are exaggerated to help illustrate the sequence of calculations. Curve for C^{t+adv} represents the concentration distribution at time t+1 due to advection only. (Modified from Konikow and Bredehoeft, 1978.) 13

5–11. Diagrams showing:

5. Spatial weights used in linear interpolation method to estimate V_x at location of a particle in cell j,i,k 16

6. Interpolation factors used in bilinear interpolation method to estimate horizontal components of velocity, V_x and V_y, at the position of a particle located in the southeast quadrant of cell j,i,k 19

7. Possible movement of a particle near an impermeable (no-flow) boundary (modified from Konikow and Bredehoeft, 1978) 27

8. Replacement of particles in fluid-source cells (a) for case of negligible regional flow and (b) for case of relatively strong regional flow 28

9. One layer of finite-difference grid illustrating use of uniformly-spaced transport subgrid for *MOC3D* within variably-spaced primary grid for *MODFLOW* 31

Page

10. Alternative *MODFLOW* approaches to vertical discretization of an aquifer system (a) consisting of two high-permeability units separated by a confining layer consisting of a low-permeability clay. In the fully three-dimensional representation (b), the clay unit would be represented by one or more model layers. In the quasi-three-dimensional approach (c), heads are not calculated in the clay unit, which is represented more simply by vertical conductance terms between the layers above and below it; in this case, the bottom of layer 1 coincides with the top of layer 2. (Modified from figs. 10-12 of McDonald and Harbaugh, 1988.) 32

11. Double time-line illustrating the sequence of progression in the *MOC3D* model for solving the flow and transport equations. This example is for transient flow and three stress periods (*NPER* = 3) of durations $PERLEN_1$, $PERLEN_2$, and $PERLEN_3$. Each time step for solving the flow equation (of duration *DELT*) is divided into one or more time increments (of duration *TIMV*) for solving the transport equation; all particles are moved once during each transport time increment. For illustration purposes, the sequence of solving the two equations is labeled for the first five time steps of the first stress period, and the indices for counting time steps for flow and time increments for transport are labeled for the fourth time step 34

12. Generalized flow chart for *MOC3D* 36

13. Simplified flow chart for the transport loop, which is shown as a single element in fig. 12 37

14–16. Diagrams showing:

14. Default initial particle configurations for a one-dimensional simulation using (a) one, (b) two, (c) three, and (d) four particles per cell 38

15. Default initial particle configurations for a two-dimensional simulation using (a) one, (b) four, (c) nine, and (d) sixteen particles per cell 38

16. Default initial particle configurations for a three-dimensional simulation using (a) one, (b) eight, and (c) 27 particles per cell. For clarity, only the volume of a single cell is illustrated 38

17. Flow chart for MOVE and MOVEBI subroutines of *MOC3D*, which is shown as a single element in fig. 13 39

18–24. Graphs showing:

18. Numerical (*MOC3D*) and analytical solutions at three different locations for solute transport in a one-dimensional, steady flow field. Parameter values for this base case are listed in Table 11 43

19. Numerical (*MOC3D*) and analytical solutions at three different locations for solute transport in a one-dimensional, steady flow field for case of increased dispersivity (α_L = 1.0 cm, D_{xx} = 0.1 cm^2/s, and other parameters as defined in Table 11) 44

20. Detailed view of numerical and analytical solutions for early times (t < 10 s) at the first node downgradient from the inflow source boundary for same problem as shown in fig. 19 44

21. Detailed view of numerical and analytical solutions for early times (t < 10 s) at the first node downgradient from the inflow source boundary for same conditions as shown in fig. 20, except that the initial number of particles per node, *NPTPND*, equals 50 44

Page

22. Numerical (*MOC3D*) and analytical solutions at three different times for same one-dimensional, steady flow, solute-transport problem shown in fig. 18 45

23. Numerical (*MOC3D*) and analytical solutions after 240 seconds for three different retardation factors for same problem represented in fig. 22 45

24. Numerical (*MOC3D*) and analytical solutions for four different times for solute transport in a one-dimensional, steady flow field for case with decay at rate of $\lambda = 0.01\ s^{-1}$. All other parameters as defined in Table 11 45

25–32. Plots showing:

25. Concentration contours for (a) analytical and (b) numerical solutions in the horizontal plane containing the solute source (layer 1) for three-dimensional solute transport in a uniform steady flow field. Parameters are defined in Table 12 47

26. Concentration contours for (a) analytical and (b) numerical solutions in the vertical plane parallel to the flow direction and aligned with column 2 for three-dimensional solute transport in a uniform steady flow field. Parameters are defined in Table 12 48

27. Concentration contours for (a) analytical and (b) numerical solutions in the vertical plane transverse to the flow direction and aligned with row 12 for three-dimensional solute transport in a uniform steady flow field. Parameters are defined in Table 12. Flow is towards the reader 48

28. Initial particle positions within the source cell for radial flow case (based on custom particle placement and *NPTPND* = 46). The relative coordinates on the *x*- and *y*-axes shown for the cell (1,1) are the same for any cell of the grid; this relative coordinate system is used for the custom definition of particle locations in the input file 50

29. Contours of relative concentrations calculated using (a) analytical and (b) numerical models for solute transport in a steady radial flow field. Source concentration is 1.0 and source is located in cell (1,1). Grid spacing is 10.0 m 50

30. Concentration contours for (a, c) analytical and (b, d) numerical solutions for transport of a point initial condition in uniform flow in the *x*-direction. The *z*-component of flow is zero, but there is dispersion in all three directions. Contour values are the log of the concentrations 53

31. Concentration contours for (a) analytical and (b, c, d) numerical solutions for transport of a point initial condition in uniform flow at 45 degrees to *x* and *y*. Contour values are the log of the concentrations 54

32. Concentration contours showing effects on areas of negative concentrations related to decreasing CELDIS factor in *MOC3D* in simulation of flow at 45 degrees to grid having 24 rows and 24 columns of nodes: (a) CELDIS = 0.50; (b) CELDIS = 0.25; and (c) CELDIS = 0.10 55

33. Schematic representation of transport domain and boundary conditions for nonuniform-flow test problem of Burnett and Frind (1987); front surface represents a plane of symmetry (modified from Burnett and Frind, 1987, fig. 4) 56

Page

34–38. Plots showing:

34. Initial particle positions within a cell for the Burnett-Frind (1987) test case (based on custom particle placement and *NPTPND* = 3). The relative coordinates on the *x*- and *y*-axes shown for the cell (1,1) are the same for any cell of the grid; this relative coordinate system is used for the custom definition of particle locations in the input file 57

35. Two-dimensional simulation results for nonuniform-flow test case showing plume positions as contours of relative concentration: (a) finite-element model (modified from Burnett and Frind, 1987, fig. 8a), and (b) *MOC3D*. Contour interval is 0.2 relative concentration 58

36. Three-dimensional simulation results for nonuniform-flow test case in which α_{T_H} = 0.1 m and α_{T_V} = 0.01 m: (a) finite-element model (modified from Burnett and Frind, 1987, fig. 8c), and (b) *MOC3D*. Plume positions are represented by contours of relative concentration; contour interval is 0.2 relative concentration 58

37. Perspective view of *MOC3D* results for three-dimensional problem of constant source in nonuniform flow and unequal transverse dispersivity coefficients. This visualization of the plume was generated from particle concentrations using a three-dimensional visualization software package and is derived from the same simulation that is the basis of fig. 36b. Note that a piece of the plume near the source is cut away (a "chair" cut) to expose a clearer view of the degree of transverse spreading in the selected vertical and horizontal planes. Shading increments are in relative-concentration intervals of 0.20, and interval bounds range between 0.9 and 0.1. Concentrations less than 0.10 are transparent 59

38. Three-dimensional simulation results for nonuniform-flow test case in which $\alpha_{T_H} = \alpha_{T_V}$ = 0.01 m: (a) finite-element model (modified from Burnett and Frind, 1987, fig. 9b), and (b) *MOC3D*. Plume positions are represented by contours of relative concentration; contour interval is 0.2 relative concentration 60

TABLES

1. *MOC3D* subroutines controlling simulation preparation 37

2. *MOC3D* subroutines controlling transport time factors 37

3. *MOC3D* subroutines controlling velocity calculations and output 38

4. *MOC3D* subroutines controlling particle tracking, concentration calculations, and output for particle and concentration data 38

5. *MOC3D* subroutines controlling dispersion calculations and output 40

6. *MOC3D* subroutines controlling *MODFLOW* source/sink package calculations 40

7. *MOC3D* subroutines controlling cumulative calculations relating to sources and sinks ... 40

8. *MOC3D* subroutines controlling observation wells 40

9. *MOC3D* subroutines controlling mass balance calculations and output 41

10. Miscellaneous *MOC3D* subroutines 41

Page

11. Parameters used in *MOC3D* simulation of transport in a one-dimensional, steady-state flow system 42

12. Parameters used in *MOC3D* simulation of transport from a continuous point source in a three-dimensional steady-state flow system 47

13. Parameters used in *MOC3D* simulation of two-dimensional, steady-state, radial flow case 49

14. Parameters used in *MOC3D* simulation of three-dimensional transport from a point source with flow in the *x*-direction and flow at 45 degrees to *x* and *y* 52

15. Parameters used in *MOC3D* simulation of transport in a vertical plane from a continuous point source in a nonuniform, steady-state, two-dimensional flow system described by Burnett and Frind (1987) 57

16. Comparison of *MOC3D* simulation times for selected test cases on various computer platforms 61

17. Formats associated with *MOC3D* print flags 74

ABSTRACT

This report presents a model, *MOC3D*, that simulates three-dimensional solute transport in flowing ground water. The model computes changes in concentration of a single dissolved chemical constituent over time that are caused by advective transport, hydrodynamic dispersion (including both mechanical dispersion and diffusion), mixing (or dilution) from fluid sources, and mathematically simple chemical reactions (including linear sorption, which is represented by a retardation factor, and decay). The transport model is integrated with *MODFLOW*, a three-dimensional ground-water flow model that uses implicit finite-difference methods to solve the transient flow equation. *MOC3D* uses the method of characteristics to solve the transport equation on the basis of the hydraulic gradients computed with *MODFLOW* for a given time step. This implementation of the method of characteristics uses particle tracking to represent advective transport and explicit finite-difference methods to calculate the effects of other processes. However, the explicit procedure has several stability criteria that may limit the size of time increments for solving the transport equation; these are automatically determined by the program. For improved efficiency, the user can apply *MOC3D* to a subgrid of the primary *MODFLOW* grid that is used to solve the flow equation. However, the transport subgrid must have uniform grid spacing along rows and columns. The report includes a description of the theoretical basis of the model, a detailed description of input requirements and output options, and the results of model testing and evaluation. The model was evaluated for several problems for which exact analytical solutions are available and by benchmarking against other numerical codes for selected complex problems for which no exact solutions are available. These test results indicate that the model is very accurate for a wide range of conditions and yields minimal numerical dispersion for advection-dominated problems. Mass-balance errors are generally less than 10 percent, and tend to decrease and stabilize with time.

INTRODUCTION

This report describes and documents a computer model *(MOC3D)* for calculating transient changes in the concentration of a single solute in a three-dimensional ground-water flow field. The calculations require the numerical solution of 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 solute concentration within the flow system. By coupling the flow equation with the solute-transport equation, the model can be applied to both steady-state and transient ground-water 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 because of four distinct processes: (1) advective transport, in which dissolved chemicals are moving with (are being carried by) the flowing ground water; (2) hydrodynamic dispersion, in which molecular and ionic diffusion and mechanical dispersion (related mostly to variations in fluid velocity through the porous media) cause the paths of dissolved molecules and ions to diverge and spread from the average direction of ground-water flow; (3) fluid sources, where water of one composition is introduced into and mixes with water of a different composition; and (4) reactions, in which some amount of the solute is added to or removed from the ground water because of chemical, biological, and (or) physical reactions in the water or between the water and the solid aquifer materials.

The *MOC3D* model is integrated with *MODFLOW*, the U.S. Geological Survey's (USGS) modular, three-dimensional, finite-difference, ground-water flow model (McDonald and Harbaugh, 1988; Harbaugh and McDonald, 1996a and 1996b). *MODFLOW* solves the ground-water flow equation, and the reader is referred to the documentation for that model and its subsequent modules for complete details. In this report it is assumed that the reader is familiar with the *MODFLOW* family of codes. The numerical solution to the solute-transport equation is based on the method-of-characteristics, which is advantageous (relative to other standard numerical schemes) for transport problems in which advection is a dominant process. The *MOC3D* code is based largely on *MOC*, the USGS's two-dimensional, method-of-characteristics, solute-transport model (Konikow and Bredehoeft, 1978; Goode and Konikow, 1989).

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. *MOC3D* is offered as a basic tool that is applicable to a wide range of field problems involving solute transport. However, there are some situations in which the model results could be inaccurate or model operation inefficient. The report includes guidelines for recognizing and avoiding these types of problems.

The computer program is written in FORTRAN, and has been developed in a modular style, similar to the *MODFLOW* model. At present, the model is not compatible with all the modules for *MODFLOW* that describe secondary flow processes or features, such as streamflow routing, subsidence, and rewetting of dry cells. As assumed by *MODFLOW*, it is also assumed by *MOC3D* that fluid properties are homogeneous and that concentration changes do not significantly affect the fluid density or viscosity, and hence the fluid velocity. Within the finite-difference grid used to solve the flow equation in *MODFLOW*, the user is able to specify a window or subgrid over which *MOC3D* will solve the solute-transport equation. This feature can significantly enhance the overall efficiency of the model by avoiding calculation effort where it is not needed. However, *MOC3D* also requires that the horizontal (row and column) grid spacing be uniform within the subgrid, although an expanding or nonuniform spacing may be applied outside of the subgrid boundaries.

The types of reactions incorporated into *MOC3D* are restricted to those that can be represented by a first-order rate reaction, such as radioactive decay, or by a retardation factor, such as instantaneous, reversible, sorption-desorption reactions governed by a linear isotherm and constant distribution coefficient (K_d). This is somewhat more restrictive than the two-dimensional model, *MOC*, which allowed the representation of nonlinear isotherms.

The report includes a detailed description of the numerical methods used to solve the solute-transport equation. The data requirements, input format specifications, program options, and output formats are all structured in a general manner that should be compatible with the types of data available for many field problems. We have attempted to maximize the use of existing *MODFLOW* output modules and styles in developing the *MOC3D* output options and features.

THEORETICAL BACKGROUND AND GOVERNING EQUATIONS

Mathematical equations that describe ground-water flow and transport processes may be developed from the fundamental principle of conservation of mass of fluid or of solute. A statement of conservation of mass (or continuity equation) may be combined with a mathematical description of the relevant process to obtain a differential equation describing flow or transport (see, for example, Bear, 1979; Freeze and Cherry, 1979; or Domenico and Schwartz, 1990).

Ground-Water Flow Equation

A quantitative description of ground-water flow is a prerequisite to accurately simulating solute transport in aquifers. A general form of the equation describing the transient flow of a compressible fluid in a heterogeneous anisotropic aquifer may be derived by combining Darcy's Law with the continuity equation. A general ground-water flow equation may be written in Cartesian tensor notation as:

$$\frac{\partial}{\partial x_i}\left(K_{ij}\frac{\partial h}{\partial x_i}\right)=S_s\frac{\partial h}{\partial t}+W \qquad (1)$$

where K_{ij} is the hydraulic conductivity of the porous media (a second-order tensor), LT^{-1}; h is the hydraulic (or potentiometric) head, L; S_s is the specific storage, L^{-1}; t is time, T; W is the volumetric flux per unit volume (positive for inflow and negative for outflow), T^{-1}; and x_i are the Cartesian coordinates, L. The summation convention of Cartesian tensor analysis is implied in eq. 1. Equation 1 can generally be applied if isothermal conditions prevail, the porous medium deforms only vertically, the volume of individual grains remains constant during deformation, Darcy's Law applies (and gradients of hydraulic head are the only driving force), and fluid properties (density and viscosity) are homogeneous and constant.

In general, the properties of porous media vary in space. Although fluid sources and sinks may vary in space and time, they have been lumped into one term *(W)* in the previous development for convenience in notation. Also, at any given location, more than one process may be adding or removing fluid simultaneously from the system, such as well withdrawals, diffuse recharge from precipitation, and evapotranspiration. However, the solution to the governing equation depends only on the net flux from sources and sinks as a function of time at each location.

If the principal axes of the hydraulic conductivity tensor are aligned with the *x-y-z* coordinate axes, the cross-product terms of the hydraulic conductivity tensor are eliminated; that is, $K_{ij} = 0$ when $i \neq j$. The ground-water flow equation may then be written to include explicitly all hydraulic conductivity terms as:

$$\frac{\partial}{\partial x}\left(K_{xx}\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_{yy}\frac{\partial h}{\partial y}\right)+\frac{\partial}{\partial z}\left(K_{zz}\frac{\partial h}{\partial z}\right)-W=S_s\frac{\partial h}{\partial t} \qquad (2)$$

where K_{xx}, K_{yy}, and K_{zz} are values of hydraulic conductivity along the x, y, and z coordinate axes, LT^{-1}. Equation 2 is identical to eq. 1 of McDonald and Harbaugh (1988, p. 2-1).

Under these same assumptions, Darcy's law may be written as:

$$q_x=-K_{xx}\,{}^{\partial h}\!/_{\partial x} \qquad (3a)$$

$$q_y=-K_{yy}\,{}^{\partial h}\!/_{\partial y} \qquad (3b)$$

$$q_z=-K_{zz}\,{}^{\partial h}\!/_{\partial z} \qquad (3c)$$

where q is the specific discharge, LT^{-1}.

Average Interstitial Velocity

The migration and mixing of chemicals dissolved in ground water will obviously be affected by the velocity of the flowing ground water. The specific discharge, q_i, calculated from eq. 3 represents the volumetric flux per unit cross-sectional area. Thus, to calculate the actual average interstitial velocity of ground water, one must account for the actual cross-sectional area through which flow is occurring. This is usually calculated as follows:

$$V_i = \frac{q_i}{\varepsilon} \tag{4}$$

where V_i is the average interstitial velocity (or seepage velocity), LT^{-1}; and ε is the effective porosity (dimensionless) of the porous medium. Assuming the same grid alignment as stated for eq. 2, it can also be written in terms of Darcy's law as:

$$V_i = -\frac{K_{ii}}{\varepsilon}\frac{\partial h}{\partial x_i}. \tag{5}$$

Governing Equation for Solute Transport

The principle of conservation of mass requires that the net mass of solute entering and leaving a specified volume of aquifer during a given time interval must equal the accumulation or loss of mass stored in that volume during the interval. This relation may be expressed mathematically in a general governing equation for solute transport in three dimensions in an incompressible fluid flowing through a porous medium as (see Bear, 1979, p. 239-243; and Goode and Konikow, 1989):

$$\frac{\partial(\varepsilon C)}{\partial t} + \frac{\partial(\rho_b \overline{C})}{\partial t} + \frac{\partial}{\partial x_i}(\varepsilon C V_i) - \frac{\partial}{\partial x_i}\left(\varepsilon D_{ij}\frac{\partial C}{\partial x_j}\right) - \Sigma C'W + \lambda\left(\varepsilon C + \rho_b \overline{C}\right) = 0 \tag{6}$$

where ε is porosity, C is volumetric concentration (mass of solute per unit volume of fluid, ML^{-3}), ρ_b is the bulk density of the aquifer material (mass of solids per unit volume of aquifer, ML^{-3}), $\overline{C}$ is the mass concentration of solute sorbed on or contained within the solid aquifer material (mass of solute per unit mass of aquifer material, MM^{-1}), V is a vector of interstitial fluid velocity components (LT^{-1}), D is a second-rank tensor of dispersion coefficients (L^2T^{-1}), W is a volumetric fluid sink ($W<0$) or fluid source ($W>0$) rate per unit volume of aquifer (T^{-1}), C' is the volumetric concentration in the sink/source fluid (ML^{-3}), and λ is the decay rate (T^{-1}).

The decay term in eq. 6 typically represents radioactive decay of both the free and sorbed solute. A radioactive decay rate is usually expressed as a half-life ($t_{1/2}$). The half-life is the time required for the concentration to decrease to one-half of the original value, and is related to the decay rate as:

$$t_{1/2} = \frac{(\ln 2)}{\lambda}. \tag{7}$$

In limited cases, the decay term can also adequately represent chemical decomposition or biodegradation. However, if in these latter cases there is also a sorbed phase present, it must be assured that the decay process occurs at the same rate for both the dissolved and sorbed phases, an assumption that is true for radioactive decay.

The concentration in the fluid leaving the aquifer at fluid sinks is commonly assumed to have the same concentration as the fluid in the aquifer (that is, $C' = C$ for $W<0$). The summation for the sink/source term allows for multiple fluid sinks and sources having different associated source concentrations. The assumption of fluid incompressibility means that all changes in fluid storage are represented by changes in porosity in the three-dimensional transport equation.

Equation 6 contains velocity divergence terms that can be eliminated (Konikow and Grove, 1977). This removes several derivatives from the transport equation, which may reduce errors during the numerical solution, as well as removing the direct effect of fluid sinks from the governing equation. The accumulation and divergence terms in eq. 6 can be expanded:

$$\frac{\partial(\varepsilon C)}{\partial t} = \varepsilon\frac{\partial C}{\partial t} + C\frac{\partial \varepsilon}{\partial t} \qquad (8)$$

$$\frac{\partial}{\partial x_i}(\varepsilon C V_i) = \varepsilon V_i\frac{\partial C}{\partial x_i} + C\frac{\partial}{\partial x_i}(\varepsilon V_i). \qquad (9)$$

Substituting these expressions into eq. 6, adding $(WC\text{-}WC) = 0$, and rearranging terms yields

$$\varepsilon\frac{\partial C}{\partial t} + \frac{\partial(\rho_b\overline{C})}{\partial t} + \varepsilon V_i\frac{\partial C}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\varepsilon D_{ij}\frac{\partial C}{\partial x_j}\right)$$
$$-\Sigma\left[W(C'-C)\right] + \lambda\left(\varepsilon C + \rho_b\overline{C}\right)$$
$$+C\left[\frac{\partial \varepsilon}{\partial t} + \frac{\partial}{\partial x_i}(\varepsilon V_i) - \Sigma W\right] = 0. \qquad (10)$$

The last term on the left side of eq. 10 contains a bracketed term, which is an expression of fluid continuity. If fluid continuity is satisfied, then the bracketed term is zero, leaving:

$$\frac{\partial C}{\partial t} + \frac{\partial(\rho_b\overline{C})}{\varepsilon\partial t} + V_i\frac{\partial C}{\partial x_i} - \frac{\frac{\partial}{\partial x_i}\left(\varepsilon D_{ij}\frac{\partial C}{\partial x_j}\right)}{\varepsilon}$$
$$-\frac{\Sigma\left[W(C'-C)\right]}{\varepsilon} + \lambda\left(C + \frac{\rho_b\overline{C}}{\varepsilon}\right) = 0. \qquad (11)$$

This form of the governing equation can be designated as "flow-equation-removed" and is advantageous, though not required, for the method-of-characteristics numerical solution because the divergence of velocity does not appear. Under these assumptions, including incompressible fluid, the porosity is not constant unless the flow system is in a steady-state equilibrium (Goode, 1990b). Therefore, we update the porosity to account for elastic changes in aquifer volume caused by transient changes in hydraulic head using the method proposed by Goode (1990b). Testing and evaluation of this approach for incorporation into *MOC3D* is documented by Goode and Konikow (1991).

The governing equation can be further simplified for the case of reversible, instantaneous, equilibrium sorption of the solute governed by a linear isotherm. For this case, the sorbed concentration, $\overline{C}$, is given by:

$$\overline{C} = K_d C \qquad (12)$$

where K_d is the sorption coefficient, or distribution coefficient, which is assumed to be constant in time. The accumulation in the sorbed phase can be expressed as:

$$\frac{\partial(\rho_b\overline{C})}{\partial t} = K_d\frac{\partial(\rho_b C)}{\partial t} = \rho_b K_d\frac{\partial C}{\partial t} \qquad (13)$$

if it is assumed that the aquifer bulk density is constant in time. Substituting eqs. 12 and 13 into eq. 11 gives:

$$\left(1 + \frac{\rho_b K_d}{\varepsilon}\right)\frac{\partial C}{\partial t} + V_i\frac{\partial C}{\partial x_i} - \frac{\frac{\partial}{\partial x_i}\left(\varepsilon D_{ij}\frac{\partial C}{\partial x_j}\right)}{\varepsilon}$$
$$-\frac{\Sigma\left[W(C'-C)\right]}{\varepsilon} + \lambda C\left(1 + \frac{\rho_b K_d}{\varepsilon}\right) = 0. \qquad (14)$$

The terms controlling sorption can be combined into a single parameter—the retardation factor (R_f), which is defined as:

$$R_f = 1 + \frac{\rho_b K_d}{\varepsilon}. \qquad (15)$$

R_f may be slightly variable in time if the porosity changes due to transient flow effects. We ignore this possible minor variability and assume that the retardation factor is constant in time. Substituting eq. 15 into eq. 14 yields:

$$\frac{\partial C}{\partial t}+\frac{V_i}{R_f}\frac{\partial C}{\partial x_i}-\frac{1}{\varepsilon R_f}\frac{\partial}{\partial x_i}\left(\varepsilon D_{ij}\frac{\partial C}{\partial x_j}\right)$$

$$-\frac{\Sigma\left[W(C'-C)\right]}{\varepsilon R_f}+\lambda C=0. \qquad (16)$$

This is the form of the governing equation solved in *MOC3D*.

We convert eq. 16 from an Eulerian framework to a Lagrangian one through the material derivative, yielding a simpler form of the governing equation (for example, see Konikow and Bredehoeft, 1978, p. 6) for the concentration of a reference point moving with the retarded velocity (V/R_f):

$$\frac{\mathrm{d}C}{\mathrm{d}t}-\frac{1}{\varepsilon R_f}\frac{\partial}{\partial x_i}\left(\varepsilon D_{ij}\frac{\partial C}{\partial x_j}\right)$$

$$-\frac{\Sigma\left[W(C'-C)\right]}{\varepsilon R_f}+\lambda C=0. \qquad (17)$$

Although this concentration is now that of a moving point in space, we retain the same symbol, C, as a matter of convenience.

The mathematical solution of the governing equations requires the specification of certain initial and boundary conditions. Because the transport equation is always solved for transient conditions, the initial concentration must be specified throughout the domain within which solute transport occurs (which may be equal to or smaller than the domain in which the flow equation is applied and solved).

The specification of a constant-concentration boundary condition at one or more nodes for the transport equation would be analogous to the use of a constant-head boundary condition for the flow equation. Although this is mathematically and numerically feasible, it is rare that a field environment would be consistent with such a constant-concentration condition. Therefore, we have not implemented the use of this type of boundary condition in this model. Instead, input concentrations must always be associated with a fluid flux.

For the transport equation, two specified mass-flux boundary conditions are used in this model. At no-flow boundaries for the flow equation, the solute mass flux is also required to be zero. The second type of specified mass boundary condition is applied when the transport subdomain is within a flow domain. That is, the boundaries of the transport subdomain do not coincide with the flow domain boundary. In this case, solute mass movement into and out of the transport subdomain is assumed to be by advection only; no dispersive solute flux can occur across a subdomain boundary, which is mathematically equivalent to a zero gradient in concentration across the boundary.

The effects of all other external fluid sinks and sources on transport are incorporated through the fluid source/sink terms *(W)* in eq. 17. For a fluid source (W>0), denoted W_+, C' in eq. 17 is the specified source concentration of the incoming fluid. For a fluid sink (W<0), denoted W_-, C' in eq. 17 is assumed to equal the concentration in the aquifer, C, at the location of the sink. In this case $W(C'-C)=(W_-)(C-C)=0$. However, if the fluid sink is associated with evaporation or transpiration, it is assumed that the fluid discharge mechanism will exclude dissolved chemicals; for this special case $C'=0$ and $W(C'-C)=(W_-)(0-C)$, which results in an increase in concentration at the location of the fluid sink.

Dispersion Coefficient

The third term in eq. 16 represents the change in concentration due to hydrodynamic dispersion. This expression is analogous to

Fick's Law describing diffusive flux. This Fickian model assumes that the driving force is the concentration gradient and that the dispersive flux occurs in a direction from higher concentrations towards lower concentrations. 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:

$$D_{ij} = \alpha_{ijmn}\frac{V_m V_n}{|V|} \qquad i,j,m,n{=}1,2,3 \quad (18)$$

where α_{ijmn} is a component of the dispersivity tensor (L), a fourth rank tensor, V_m and V_n are components of the velocity vector in the m and n directions, respectively, and $|V|$ is the magnitude of velocity:

$$|V| = \sqrt{V_x^2 + V_y^2 + V_z^2}\,. \qquad (19)$$

Scheidegger (1961) further shows that for an isotropic aquifer the dispersivity tensor can be defined in terms of the longitudinal and transverse dispersivities, α_L and α_T. This yields two dispersion coefficients oriented with the direction of flow, the longitudinal dispersion coefficient, D_L, and the transverse dispersion coefficient, D_T:

$$D_L = \alpha_L|V|\,; \qquad D_T = \alpha_T|V|. \qquad (20)$$

The directional dispersion coefficients in the *x, y,* and *z* coordinates are derived by a transformation from the *L-T* coordinates. Additionally, an isotropic diffusion coefficient, D_m, can be included to account for molecular diffusion. The diffusion coefficient includes the effects of tortuosity. These manipulations yield the components of the dispersion tensor, *D*, which includes diffusion:

$$D_{ij} = \left(D_m + \alpha_T|V|\right)\delta_{ij} + \left(\alpha_L - \alpha_T\right)\frac{V_i V_j}{|V|} \qquad (21)$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. This short-hand notation can be written explicitly as:

$$D_{xx} = \alpha_L\frac{V_x^2}{|V|} + \alpha_T\frac{V_y^2}{|V|} + \alpha_T\frac{V_z^2}{|V|} + D_m \qquad (22a)$$

$$D_{yy} = \alpha_L\frac{V_y^2}{|V|} + \alpha_T\frac{V_x^2}{|V|} + \alpha_T\frac{V_z^2}{|V|} + D_m \qquad (22b)$$

$$D_{zz} = \alpha_L\frac{V_z^2}{|V|} + \alpha_T\frac{V_y^2}{|V|} + \alpha_T\frac{V_x^2}{|V|} + D_m \qquad (22c)$$

$$D_{xy} = D_{yx} = \left(\alpha_L - \alpha_T\right)\frac{V_x V_y}{|V|} \qquad (22d)$$

$$D_{xz} = D_{zx} = \left(\alpha_L - \alpha_T\right)\frac{V_x V_z}{|V|} \qquad (22e)$$

$$D_{yz} = D_{zy} = \left(\alpha_L - \alpha_T\right)\frac{V_y V_z}{|V|}. \qquad (22f)$$

A number of field studies have indicated that transverse spreading in the vertical direction is much smaller than transverse spreading in the horizontal direction (for example, see Robson, 1974; Robson, 1978; Garabedian and others, 1991; and Gelhar and others, 1992). To allow modeling of this observed spreading pattern, Burnett and Frind (1987) made an *ad hoc* modification to the transverse terms in the preceding expressions, which we incorporate into *MOC3D*:

$$D_{xx} = \alpha_L\frac{V_x^2}{|V|} + \alpha_{T_H}\frac{V_y^2}{|V|} + \alpha_{T_V}\frac{V_z^2}{|V|} + D_m \qquad (23a)$$

$$D_{yy} = \alpha_L\frac{V_y^2}{|V|} + \alpha_{T_H}\frac{V_x^2}{|V|} + \alpha_{T_V}\frac{V_z^2}{|V|} + D_m \qquad (23b)$$

$$D_{zz} = \alpha_L\frac{V_z^2}{|V|} + \alpha_{T_V}\frac{V_y^2}{|V|} + \alpha_{T_V}\frac{V_x^2}{|V|} + D_m \qquad (23c)$$

$$D_{xy} = D_{yx} = \left(\alpha_L - \alpha_{T_H}\right)\frac{V_x V_y}{|V|} \qquad (23d)$$

$$D_{xz} = D_{zx} = \left(\alpha_L - \alpha_{T_V}\right)\frac{V_x V_z}{|V|} \qquad (23e)$$

$$D_{yz} = D_{zy} = \left(\alpha_L - \alpha_{T_V}\right)\frac{V_y V_z}{|V|} \qquad (23f)$$

where α_{T_H} is the horizontal transverse dispersivity and α_{T_V} is the vertical transverse dispersivity. If $\alpha_{T_H} = \alpha_{T_V}$, then eq. 23 reduces to eq. 22.

Review of Assumptions

A number of assumptions have been made in the development of the previous forms of the governing 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 hydraulic conductivity of the aquifer system is constant with time. Also, if the system is anisotropic, it is assumed that the principal axes of the hydraulic conductivity tensor are aligned with orthogonal frame of reference, so that the cross-product terms of the hydraulic conductivity tensor are eliminated.
3. Gradients of fluid density, viscosity, and temperature do not affect the velocity distribution.
4. No chemical reactions occur that affect the fluid or aquifer properties.
5. The dispersivity coefficients are constant with time, and the aquifer is isotropic with respect to longitudinal dispersivity.

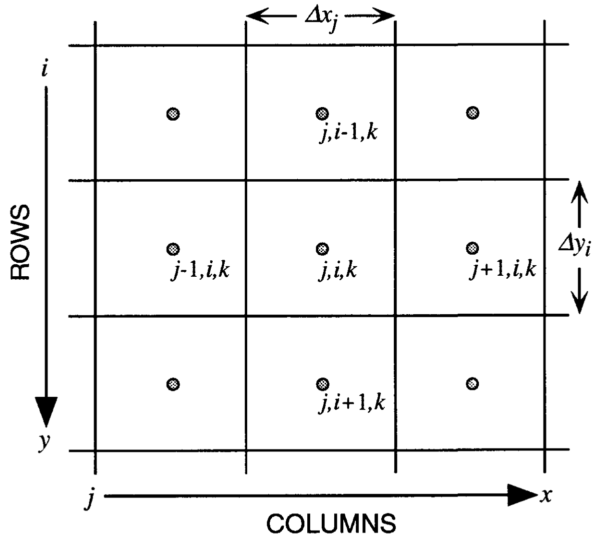
As noted by Konikow and Bredehoeft (1978), the nature of a specific field problem may be such that not all of these underlying assumptions are valid. The degree to which field conditions deviate from these assumptions will affect the applicability and reliability of the model for that problem. If the deviation from a particular assumption is significant, the governing equations and the numerical code will have to be modified to account for the appropriate processes or factors.

NUMERICAL METHODS

Because aquifers are heterogeneous and have complex boundary conditions, exact analytical solutions to the governing equations can not be obtained directly. Instead, numerical methods are used, in which the continuous variables of the governing equations are replaced with discrete variables that are defined at grid blocks (or cells or nodes). Thus, the continuous differential equation, which defines hydraulic head or solute concentration everywhere in the system, is replaced by a finite number of algebraic equations that defines the hydraulic head or concentration at specific points. This system of algebraic equations generally is solved using matrix techniques.

However, numerical methods yield only approximate (rather than exact) solutions to the governing equation (or equations); they require discretization of space and time. The variable internal properties, boundaries, and stresses of the system are approximated within the discretized format. In general, the finer the discretization, the closer the numerical solution will be to the true solution.

The notation and conventions used in this report and in the *MOC3D* code to describe the grid and to reference (or to number) nodes are described in figs. 1-2. The indexing notation used here is consistent with that used in the FORTRAN code for *MODFLOW* by McDonald and Harbaugh (1988), although not necessarily the notation used in the text of their report. Our indexing notation maintains conformity between the text of this report and the FORTRAN code in *MOC3D*, and the index order corresponds to an x,y,z sequence, which is standard in numerical models. However, our notation differs from that used in some other ground-water models in that the x-direction is indexed by "j" and increases from left to right along a row to indicate the column number. Our use of Δx and Δy is synonymous with the use of Δr and Δc, respectively, by McDonald and Harbaugh (1988). The y-direction is indexed by "i" and increases from the top of the grid to the bottom within a column to indicate the row number. Thus, in a map view of any one horizontal layer, as illustrated in fig. 1, the node representing a cell in the first row and first column of the grid would lie in the upper left corner of the grid. The z-direction represents layers and is indexed by "k." As indicated in fig. 2, the first layer ($k = 1$) in a multilayer grid would be the top (or highest elevation) layer. The saturated thickness of a cell ($b_{j,i,k}$) is equivalent to Δz.



EXPLANATION

◎ Node of finite-difference cell

$\Delta x_j = \Delta r_j$ = cell dimension in row direction

$\Delta y_i = \Delta c_i$ = cell dimension in column direction

Figure 1. Notation used to label rows, columns, and nodes within one layer (k) of a three-dimensional, block-centered, finite-difference grid for *MOC3D*.

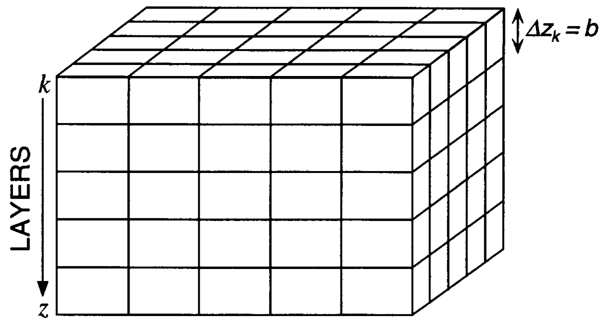


Figure 2. Representative three-dimensional grid for *MOC3D* illustrating notation for layers.

Ground-Water Flow Equation

A numerical solution of three-dimensional ground-water flow equation is obtained by the *MODFLOW* code using implicit (backward-in-time) finite-difference methods. The model was coded in FORTRAN in a modular style to allow and encourage the development of additional packages or modules that can be added on or linked to the original code. Many such packages or modules, which typically allow additional ground-water processes, hydrogeological features, solution algorithms, or input/output options to be represented or used, have been documented since *MODFLOW* was first released. Most of these are summarized by Appel and Reilly (1994).

MODFLOW is based on use of a block-centered finite-difference grid that allows variable spacing of the grid in three dimensions. Flow can be steady or transient. Layers can be simulated as confined, unconfined, or a combination of both. Aquifer properties can vary spatially and hydraulic conductivity (or transmissivity) can be anisotropic. Flow associated with external stresses, such as wells, areally distributed recharge, evapotranspiration, drains, and streams, can also be simulated through the use of specified head, specified flux, or head-dependent flux boundary conditions.

MODFLOW offers several options to solve the implicit finite-difference equations, including the Strongly Implicit Procedure (SIP), Slice-Successive Overrelaxation (SSOR) methods, or Preconditioned Conjugate-Gradient matrix solvers (for example, Hill, 1990). Successful use of *MOC3D*, which is programmed as a module to *MODFLOW*, requires a thorough familiarity with the use of *MODFLOW*. Comprehensive documentation of *MODFLOW* is presented by McDonald and Harbaugh (1988), Harbaugh and McDonald (1996a and 1996b), and the various reports for additional implemented modules.

Average Interstitial Velocity

Because advective transport and hydrodynamic dispersion both depend on the velocity of ground-water flow, the solution of the transport equation requires knowledge of the velocity (or specific discharge) field. Therefore, after the head distribution has been calculated for a given time step or steady-state flow condition, the specific discharge across every face of each finite-difference cell within the transport subgrid is calculated next.

The specific discharge components are calculated for each face using a finite-difference form of eq. 3. For example, the specific discharge in the horizontal plane in the x-direction at the block interface between cells j,i,k and $j+1,i,k$ is (after McDonald and Harbaugh, 1988):

$$q_{x(j+1/2,i,k)} = -\overline{K}_{xx(j+1/2,i,k)}(\Delta h / \Delta x)_{(j+1/2,i,k)}$$
$$= -\overline{K}_{xx(j+1/2,i,k)}\frac{\left(h_{j+1,i,k} - h_{j,i,k}\right)}{\Delta x} \qquad (24)$$

where $\overline{K}_{xx(j+1/2,i,k)}$ is the interblock hydraulic conductivity in the x-direction on the forward face of the cell and the hydraulic gradient is based on implicitly calculated heads at the adjacent nodes. (Note that the interblock hydraulic conductivity is commonly defined by the harmonic mean, but in *MODFLOW* the user can specify alternative methods for calculating the interblock hydraulic conductivity in the Block-Centered Flow [BCF] package.)

The seepage velocity at any point within a cell must be defined to represent advective transport. It is calculated at a point of interest within a finite-difference cell based on the interpolated estimate of specific discharge at that point divided by the effective porosity of the cell in which the point is located (see eqs. 4-5). The interpolation methods are discussed later in the section on "Particle Tracking."

Because the velocity is one of the most important factors controlling solute transport, it is necessary to examine closely the calculated velocity field to understand the patterns and rates of solute spreading. Therefore, the *MOC3D* model offers the user several options to output the velocity data. These options are described in Appendix B. The x-, y-, and z-components of the velocity vector at the nodes of the finite-difference cells can be computed with equivalent orders of accuracy for all three components at a single location. For example, the velocity in the x-direction at node (j,i,k) would be computed as

$$V_{x(j,i,k)} = \frac{\left(q_{x(j+1/2,i,k)} + q_{x(j-1/2,i,k)}\right)}{2\varepsilon_{(j,i,k)}}. \qquad (25)$$

Analogous expressions are used to compute the velocity components in the y- and z-directions.

Note that the specific discharge components themselves are computed on cell faces with a higher order of accuracy than are the velocity components at the nodes. For example, $q_{x(j+1/2,i,k)}$ is based on the head difference over Δx; whereas V_x at the node is based on the head difference over $2\Delta x$. However, an estimate of V_y or V_z at the same location $(j+1/2,i,k)$ would have to be at a lower order of accuracy (for example, $2\Delta x \Delta y$ for V_y). Therefore, the more accurate estimates on cell faces are used in the model as the basis for interpolation of particle velocities, but the velocities at nodes are written in the output files either to enable direct inspection by the user or to facilitate postprocessing with visualization software.

Solute-Transport Equation

The solute-transport equation is, in general, more difficult to solve accurately using numerical methods than is the ground-water flow equation, largely because the mathematical properties of the transport equation vary depending upon which terms in the equation are dominant in a particular system. Where solute transport is dominated by advective transport, as is common in many field problems, then the transport equation approximates a hyperbolic type of equation (similar to equations describing the propagation of a wave or of a shock front). In contrast, where a system is dominated by dispersive fluxes, such as might occur where fluid velocities are relatively low and aquifer dispersivities are relatively high, then the transport equation becomes more parabolic in nature (similar to the transient ground-water flow equation). To further complicate matters, because system properties and fluid velocity may vary significantly, the dominant process (and the mathematical properties of the governing equation) may vary from point to point and over time within the same domain.

Dispersion-dominated solute-transport problems (and parabolic equations in general) are quite amenable to accurate and efficient solution using standard finite-difference and finite-element numerical methods. However, in solving advection-dominated transport problems, in which a relatively sharp front (or steep concentration gradient) is moving through a system, it is numerically difficult to preserve the sharpness of the front. In such cases, numerical solutions often will include either erroneous oscillations (overshoot and undershoot) or calculate a greater dispersive flux than would occur by physical dispersion alone or than would be indicated by an exact solution of the governing equation (for example, see Pinder and Gray, 1977). That part of the calculated dispersion introduced solely by the numerical solution algorithm is called *numerical dispersion.*

Method of Characteristics

The method of characteristics was developed to solve hyperbolic differential equations. A major advantage is that the method minimizes numerical dispersion (or even eliminates it in limited cases) (Garder and others, 1964; Pinder and Cooper, 1970; Reddell and Sunada, 1970; Bredehoeft and Pinder, 1973; Konikow and Bredehoeft, 1978; Zheng, 1990). The approach taken by the method of characteristics is not to solve eq. 16 directly, but rather to solve an equivalent system of ordinary differential equations. Equation 16 can be rearranged to obtain:

$$\frac{\partial C}{\partial t} = \frac{1}{\varepsilon R_f}\frac{\partial}{\partial x_i}\left(\varepsilon D_{ij}\frac{\partial C}{\partial x_j}\right) - \frac{V_i}{R_f}\frac{\partial C}{\partial x_i} + \frac{\Sigma[W(C' - C)]}{\varepsilon R_f} - \lambda C. \qquad (26)$$

Equation 26 describes the change in concentration over time at fixed reference points within a stationary coordinate system, which is referred to as an Eulerian framework. An alternative perspective is to consider changes in concentration over time in representative fluid parcels as they move with the flow of the fluid past fixed points in space. This, in effect, is a moving coordinate system, which is referred to as a Lagrangian framework. We convert eq. 16 from an Eulerian framework to a Lagrangian one (essentially, a framework of a moving grid) through the material derivative. The material derivative of concentration with respect to time, d*C*/d*t*, describes the change in concentration in a parcel of water moving at the average interstitial velocity of water; it may be defined for a three-dimensional system as:

$$\frac{\mathrm{d}C}{\mathrm{d}t}=\frac{\partial C}{\partial t}+\frac{\partial C}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t}+\frac{\partial C}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t}+\frac{\partial C}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}t}. \qquad (27)$$

The last three terms on the right side include the material derivatives of position, which are defined by the velocity in the *x*-, *y*-, and *z*-directions. We then have:

$$\frac{\mathrm{d}x}{\mathrm{d}t}=\frac{V_x}{R_f} \qquad (28)$$

$$\frac{\mathrm{d}y}{\mathrm{d}t}=\frac{V_y}{R_f} \qquad (29)$$

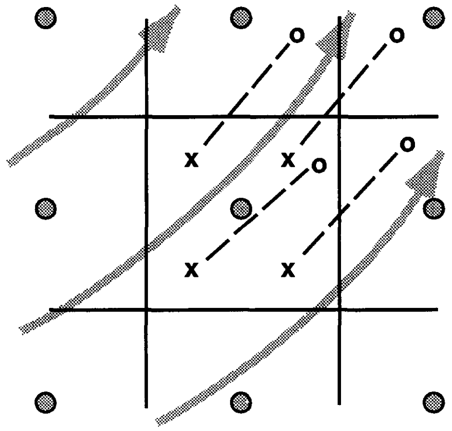
$$\frac{\mathrm{d}z}{\mathrm{d}t}=\frac{V_z}{R_f}, \qquad (30)$$

and a simpler form of the governing equation (for example, see Konikow and Bredehoeft, 1978, p. 6) for the concentration of a reference point moving with the retarded velocity (*V/R$_f$*) is obtained by substituting the right sides of eqs. 26 and 28-30 for the corresponding terms in eq. 27:

$$\frac{\mathrm{d}C}{\mathrm{d}t}=\frac{1}{\varepsilon R_f}\frac{\partial}{\partial x_i}\left(\varepsilon D_{ij}\frac{\partial C}{\partial x_j}\right)+\frac{\Sigma[W(C'-C)]}{\varepsilon R_f}-\lambda C. \qquad (31)$$

Although this concentration is now that of a moving point in space, we retain the same symbol, *C*, as a matter of convenience.

The solutions of the system of equations comprising eqs. 28-31 may be given as *x* = *x(t)*, *y* = *y(t)*, *z* = *z(t)*, and *C* = *C(t)*, and are called the characteristic curves of eq. 26. Given solutions to eqs. 28-31, a solution to the original partial differential equation may be obtained by following the characteristic curves. This may be accomplished by introducing a set of moving points (or reference particles) that can be traced within the stationary coordinates of a finite-difference grid. Each particle corresponds to one characteristic curve, and values of *x*, *y*, *z*, and *C* are obtained as functions of *t* for each characteristic (Garder and others, 1964). 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 (see fig. 3). Equation 31 can be solved using any one of several approaches, including random-walk methods (for example, see Prickett and others, 1981; and Engineering Technologies Associates, Inc., 1989). We



EXPLANATION

⇒ Flow line and direction of flow

— — — Computed path of particle

● Node of finite-difference cell

x New location of particle

o Initial location of particle

Figure 3. Part of a hypothetical finite-difference grid showing relation of flow field to movement of points (or particles) in method-of-characteristics model for simulating solute transport (modified from Konikow and Bredehoeft, 1978).

choose to solve eq. 31 using explicit finite-difference approaches because of its efficiency and relative simplicity. However, as noted by Reddell and Sunada (1970) and Konikow and Bredehoeft (1978), this approach also requires the consideration of stability criteria, which will be discussed in the section "Stability and Accuracy Criteria." In some cases, the efficiency may be adversely affected because of restrictions imposed by the stability criteria.

As noted by Konikow and Bredehoeft (1978), the processes of advection, dispersion, mixing, and reactions are occurring continuously and simultaneously. Therefore, eqs. 28-31 should be solved simultaneously, but for practical reasons, they are solved sequentially. However, the results will be sensitive to the order in which they are solved because the change in concentration due to dispersion depends on the concentration gradient, and the concentration gradient at any location may change significantly because of advection during a time increment. This is illustrated in fig. 4. As the position of a concentration front or breakthrough curve advances with time, say from time *t* at the beginning of a time increment, to time *t*+1 at the end of a time increment, the concentration gradient at any particular location or reference point is continuously changing. For example, in fig. 4 the concentration gradient at the location where the relative distance equals 0.4 is relatively steep at time *t*, as indicated by the slope of the curve labeled C^t. However, if the solute migrated in the direction of flow to a new location due to advection only without being subjected to any dispersive flux, as indicated by the curve labeled C^{t+adv}, then the concentration gradient at the same point in space would be much smaller. If the change in concentration caused by dispersion were computed by solving eq. 31 first in the

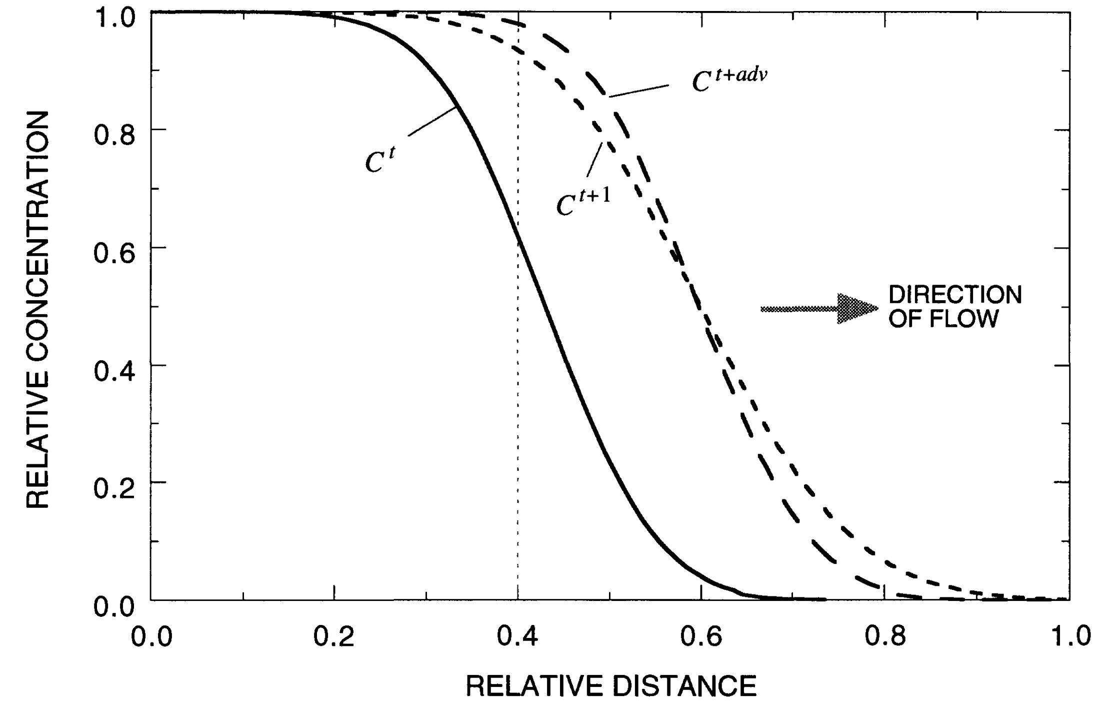


Figure 4. Representative change in breakthrough curve from time level *t* to *t*+1. Note that concentration changes are exaggerated to help illustrate the sequence of calculations. Curve for C^{t+adv} represents the concentration distribution at time *t*+1 due to advection only. (Modified from Konikow and Bredehoeft, 1978.)

sequence and using the spatial concentration gradients at the beginning of a time step, the net results would be different from those that would be computed from solving eq. 31 last in sequence and using concentration gradients after advection. Sensitivity to the sequence of solving the characteristic equations is eliminated by solving eq. 31 using concentration gradients based on the average of the concentrations at each node before and after advection. This effectively gives equal weight to the concentration gradients before and after advection while computing the solute flux due to dispersion. The averaged concentrations, designated as $C^*_{j,i,k}$, are calculated as:

$$C^*_{j,i,k} = \frac{C^t_{j,i,k} + C^{t+adv}_{j,i,k}}{2} \qquad (32)$$

where $C^{t+adv}_{j,i,k}$ is the concentration at the new time level after advection alone.

Particle Tracking

Advection in flowing ground water is simulated by particle tracking. The other terms in the governing equation—dispersion, sources, and decay—are accounted for by adjusting the concentrations associated with each particle. The concentration changes caused by dispersion and fluid sources are computed in the fixed-in-space finite-difference grid, whereas concentration changes caused by decay are calculated directly on the particles, as described below. Initial particle locations are defined through model input, and subsequent particle positions are integrated in time using the spatially and temporally variable velocity field.

The fluid velocity is derived from a block-centered finite-difference solution of the three-dimensional flow equation (McDonald and Harbaugh, 1988). The components of the velocity vector are approximated by difference expressions at interfaces between adjacent blocks occupying three-dimensional space (see eq. 24). In the quasi-three-dimensional approach taken by McDonald and Harbaugh (1988), horizontal fluxes are computed using the transmissivity of each layer and the vertical fluxes are computed using the vertical conductance, which is the vertical hydraulic conductivity divided by the vertical distance between layer centers. Thus, the components of the specific discharge vector can be represented by:

$$bq_{x(j+1/2,i,k)} = -T_{xx(j+1/2,i,k)} \frac{h_{j+1,i,k} - h_{j,i,k}}{\Delta x} \qquad (33a)$$

$$bq_{y(j,i+1/2,k)} = -T_{yy(j,i+1/2,k)} \frac{h_{j,i+1,k} - h_{j,i,k}}{\Delta y} \qquad (33b)$$

$$q_{z(j,i,k+1/2)} = -\left(\frac{K_z}{\Delta z}\right)_{(j,i,k+1/2)} \left(h_{j,i,k+1} - h_{j,i,k}\right). \qquad (33c)$$

Note that in eqs. 33a-b, a value of *b* is not explicitly calculated on the face of a cell. Instead, we calculate a value for the product *bq* on the face using a mean transmissivity. Having computed specific discharge components at block interfaces, spatial interpolation is used to estimate the specific discharge and velocity at the locations of all particles (using values of *b* and ε for the particular cell in which the particle of interest is located).

MOC3D uses either linear or bilinear spatial interpolation, as specified by the user. These two alternatives are described below in the sections on interpolation. Depending on the velocity interpolation method chosen, changes in particle position are computed either explicitly for bilinear interpolation, or semi-analytically for linear interpolation. The integration of particle position in time is also described in the sections on interpolation.

A moving point in a ground-water flow system will change velocity as it moves due to both spatial variation in velocity and temporal variations during transient flow. During a flow time step, advection is determined from velocities computed at the end of the flow time step. Temporal changes in velocity are accounted for by a step change in velocity at the start of each new flow time step. After the flow equation is solved for a new time step, specific discharges are recomputed on the basis of the new head distribution, and the movement of particles during this flow time step is based only on these specific discharges.

Linear interpolation and semi-analytic integration

Linear interpolation of specific discharge is formally consistent with the block-centered finite-difference flow solution in that the governing flow equation is satisfied locally within each cell (Goode, 1990a). In this case, each velocity component is linearly interpolated in the direction of the component of interest. The solute velocity at any particular point or location within a cell, which may be retarded with respect to the average interstitial velocity of water, is:

$$V_x = \left[\left(R_f\right)_k (\varepsilon b)_{j,i,k}\right]^{-1} \left\{bq_{x(j-1/2,i,k)} + \delta x\left[bq_{x(j+1/2,i,k)} - bq_{x(j-1/2,i,k)}\right]\right\} \qquad (34a)$$

$$V_y = \left[\left(R_f\right)_k (\varepsilon b)_{j,i,k}\right]^{-1} \left\{bq_{y(j,i-1/2,k)} + \delta y\left[bq_{y(j,i+1/2,k)} - bq_{y(j,i-1/2,k)}\right]\right\} \qquad (34b)$$

$$V_z = \left[\left(R_f\right)_k (\varepsilon)_{j,i,k}\right]^{-1} \left\{q_{z(j,i,k-1/2)} + \delta z\left[q_{z(j,i,k+1/2)} - q_{z(j,i,k-1/2)}\right]\right\} \qquad (34c)$$

where δx, δy, and δz are the spatial weights for interpolation. The spatial weights are given by:

$$\delta x = \frac{x_p - x_{j-1/2,i,k}}{\Delta x} \qquad (35a)$$

$$\delta y = \frac{y_p - y_{j,i-1/2,k}}{\Delta y} \qquad (35b)$$

$$\delta z = \frac{z_p - z_{j,i,k-1/2}}{\Delta z} \qquad (35c)$$

where the subscript p indicates the particle number and δx, δy, and δz can range from 0.0 to 1.0.

V_x at p is calculated from eqs. 34a and 35a. If a layer has a uniform porosity and thickness, and the solute were nonreactive, then V_x at p can be calculated directly from the following simplification of eq. 34a as:

$$V_{x(p)} = \delta x V_{x(j+1/2,i,k)} + (1-\delta x) V_{x(j-1/2,i,k)}. \qquad (36)$$

The linear interpolation scheme is illustrated for V_x in fig. 5, which shows a particle p located in cell j,i,k. V_x at p depends on the relative position within the cell in the x-direction only, and does not vary with changes in particle position in the y- or z-directions within the cell.

For simplicity, a dimensionless local coordinate is used to facilitate particle tracking. Internally to the program, the particle coordinates are represented, for example, in the x-direction by:

$$x_p = \left(j + \hat{x}_p\right)\Delta x \qquad (37)$$

where j is the column index of the block where a particle is located and $-0.5<\hat{x}_p<0.5$ is the particle position in the x-direction relative to the

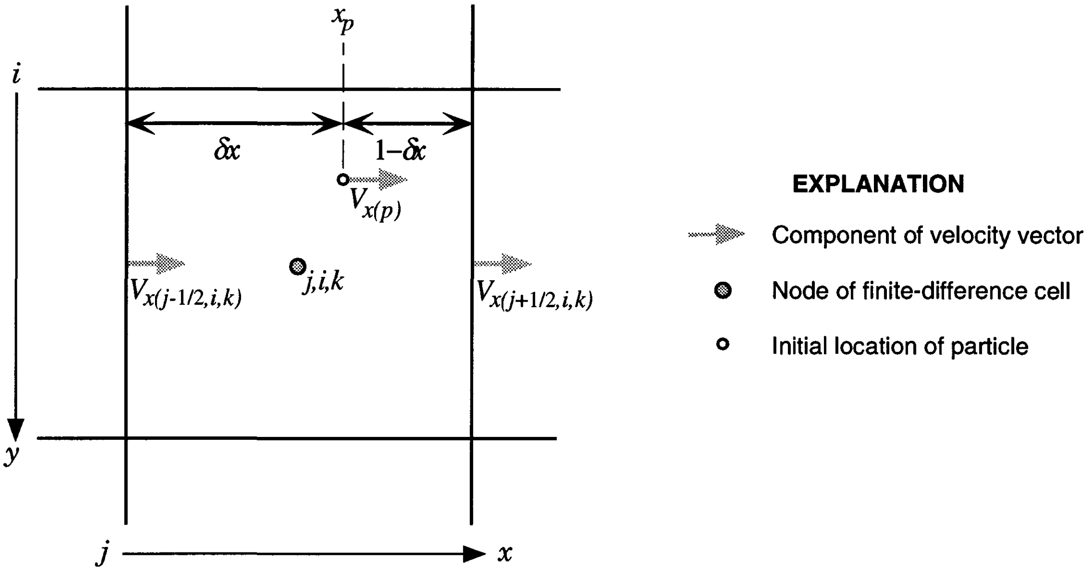


Figure 5. Spatial weights used in linear interpolation method to estimate V_x at location of a particle in cell j,i,k.

node (that is, $\hat{x}_p = (x_p/\Delta x) - j$). The value $j + \hat{x}_p$ is stored in the model and, for example, ranges from 4.5 to 5.5 for particles within column 5. Thus, the spatial weights can be represented in terms of the particle positions relative to the node, and the velocity at the location of a particle is estimated using linear interpolation as:

$$V_{x(p)} = \left[(R_f)_k (\varepsilon b)_{j,i,k}\right]^{-1} \left\{ bq_{x(j-1/2,i,k)} + \left(\hat{x}_p + 0.5\right)\left[bq_{x(j+1/2,i,k)} - bq_{x(j-1/2,i,k)}\right]\right\} \tag{38a}$$

$$V_{y(p)} = \left[(R_f)_k (\varepsilon b)_{j,i,k}\right]^{-1} \left\{ bq_{y(j,i-1/2,k)} + \left(\hat{y}_p + 0.5\right)\left[bq_{y(j,i+1/2,k)} - bq_{y(j,i-1/2,k)}\right]\right\} \tag{38b}$$

$$V_{z(p)} = \left[(R_f)_k (\varepsilon)_{j,i,k}\right]^{-1} \left\{ q_{z(j,i,k-1/2)} + \left(\hat{z}_p + 0.5\right)\left[q_{z(j,i,k+1/2)} - q_{z(j,i,k-1/2)}\right]\right\}. \tag{38c}$$

Assuming that the velocity is constant in time during a time increment for solving the transport equation, the particle position can be integrated analytically within each cell (Goode, 1990a). The integration of particle position is accomplished in a multi-step procedure. First, the velocity components in each direction at the starting position of the particle are computed using eqs. 38a-c. Then the gradient of velocity within the cell for each direction is computed from the last term of eqs. 38a-c and denominator terms appropriate for the particular direction. For example, the velocity gradient in the x-direction, g_x, is calculated by dividing the last term of eq. 38a by the product of the retardation factor, porosity, saturated thickness, and grid spacing in the x-direction, as:

$$g_x = \frac{\partial V_x}{\partial x} = \frac{bq_{x(j+1/2,i,k)} - bq_{x(j-1/2,i,k)}}{(R_f)_k (\varepsilon b)_{j,i,k} \Delta x}. \tag{39}$$

It is implicit in the linear interpolation approach that

$$\frac{\partial V_x}{\partial y} = \frac{\partial V_x}{\partial z} = 0. \qquad (40)$$

Given the solutions to eqs. 38a-c and 39, the time of travel to each cell boundary, in the x-, y-, and z-directions, can be computed analytically. For example, the time for a particle to reach or cross the x-boundary, Δt_x, is given by:

$$\Delta t_x = \frac{1}{g_x}\left[\ln\left|\frac{V_{x,0} + g_x(x_e - x_0)}{V_{x,0}}\right|\right] \qquad (41)$$

where $V_{x,0}$ is the x-velocity at the starting location of the particle, x_0 is the initial position of the particle, and x_e is the x-coordinate of the exit face of the cell. By definition, if the starting velocity is positive, then the only possible x-boundary for the particle to exit is that in the positive x-direction. If g_x is zero, or practically zero (that is, less than 10^{-20}), then V_x is assumed to be uniform in space and constant during the particle motion, and the travel time to the exit boundary can be computed directly from:

$$\Delta t_x = \frac{x_e - x_0}{V_{x,0}}. \qquad (42)$$

The travel time for a particle to exit the cell, Δt_e, is the minimum of Δt_x, Δt_y, and Δt_z. If the cell exit travel time (Δt_e) is greater than the remaining time for this time increment, then the ending particle position is within the same cell that it started during that time increment, and the particle coordinates at the end of the time increment are determined from the velocity at the initial location, the rate of change in velocity in the direction of movement, and the length of that time increment. For example, the new x-coordinate at the end of the move would be:

$$x_1 = x_0 + \frac{V_{x,0}}{g_x}\left(e^{g_x \Delta t} - 1\right) \qquad (43)$$

where the subscript "1" denotes the end of the particle movement for the time increment. If g_x, the gradient in x-velocity, is zero or practically zero (again, less than 10^{-20}), then the ending coordinate is simply:

$$x_1 = x_0 + V_{x,0}\Delta t. \qquad (44)$$

On the other hand, if the time to reach any cell boundary is less than the remaining time for this time increment *(Δt)*, then the particle exits the cell during the time increment. In this case, the particle exits the cell on the boundary for the direction having the minimum travel time (Δt_x, Δt_y, or Δt_z). For example, if Δt_x is less than Δt_y, Δt_z, and Δt, then the particle leaves the cell by crossing the x-cell boundary after the "cell exit travel time" ($\Delta t_e = \Delta t_x$) has elapsed.

The particle position must be evaluated when it leaves a cell during a time increment. By definition, the position in the direction of the exit face is known *a priori*. The other two coordinates are computed by inserting the cell exit travel time, Δt_e, into eq. 43 instead of the entire time increment, Δt. For example, for a particle that leaves a cell on the $x_{j+1/2}$ face, the y- and z-coordinates of that particle where it crosses the cell face are given by:

$$y_1 = y_0 + \frac{V_{y,0}}{g_y}\left(e^{g_y \Delta t_e} - 1\right) \qquad (45a)$$

$$z_1 = z_0 + \frac{V_{z,0}}{g_z}\left(e^{g_z \Delta t_e} - 1\right). \qquad (45b)$$

If the gradient terms in eq. 45a or eq. 45b are zero, or essentially zero, then the y and z equivalents of eq. 44 are used with $\Delta t = \Delta t_e$.

Also, if $\Delta t_e < \Delta t$, the time remaining in the time increment, $\Delta t'$, is the original time increment minus the cell exit travel time:

$$\Delta t' = \Delta t - \Delta t_e. \qquad (46)$$

Subsequent particle movement is computed as outlined above, but with a reduced time interval ($\Delta t'$) for its path in the new cell, until the entire original time increment (Δt) is exhausted. Note that in some cases, such as when a particle is located near a corner of a cell at the start of a move, it is possible that during that particular time increment this sequence of calculations may have to be repeated a third time before the move is completed for that particle.

Note that linear interpolation yields a discontinuous velocity field (Goode, 1990a). Equations 39 and 40 show that velocity components vary as a function of distance only in the direction of that particular component of the velocity vector. Thus, when linear interpolation is used, specific discharge and velocity components change abruptly at block or cell interfaces parallel to the direction indicated by that component, even for homogeneous aquifers. For example, the component of flux in the x-direction, q_x, is continuous at cell interfaces in the x-direction (at j-1/2,i,k and j+1/2,i,k), but it can change abruptly at the other four cell faces that are perpendicular to the y and z axes, even when K is uniform.

Bilinear interpolation and explicit integration

In a homogeneous aquifer the velocity field would be continuous and smoothly varying, except at the locations of strong sources and sinks, unlike the discontinuous velocity field calculated using linear interpolation. Bilinear interpolation generates such a continuous specific discharge field, as shown by Goode (1990a) for a case of two-dimensional flow. *MOC3D* includes an option to use bilinear interpolation in the x-y (horizontal) plane to determine the x- and y-velocity components, which then will be continuous and smoothly varying in the x-y plane. Because of stratigraphic layering, many aquifer systems are more heterogeneous in the vertical direction than within a horizontal plane. Hence, specific discharge and velocity might be more often expected to change abruptly (rather than smoothly) in the vertical direction. Therefore, *MOC3D* always uses linear interpolation for calculating the z-component of particle velocity.

Following Konikow and Bredehoeft (1978) and Goode (1990a), the x- and y-velocity components are bilinearly interpolated in the horizontal plane by:

$$V_{x(p)} = \left[\left(R_f\right)_k (\varepsilon b)_{j,i,k}\right]^{-1} \Big\{\left(1-F_y\right)\left[bq_{x(j-1/2,i,k)} + \left(\hat{x}_p + 0.5\right)\left(bq_{x(j+1/2,i,k)} - bq_{x(j-1/2,i,k)}\right)\right] + F_y\left[bq_{x(j-1/2,i^*,k)} + \left(\hat{x}_p + 0.5\right)\left(bq_{x(j+1/2,i^*,k)} - bq_{x(j-1/2,i^*,k)}\right)\right]\Big\} \qquad (47)$$

and

$$V_{y(p)} = \left[\left(R_f\right)_k (\varepsilon b)_{j,i,k}\right]^{-1} \Big\{\left(1-F_x\right)\left[bq_{y(j,i-1/2,k)} + \left(\hat{y}_p + 0.5\right)\left(bq_{y(j,i+1/2,k)} - bq_{y(j,i-1/2,k)}\right)\right] + F_x\left[bq_{y(j^*,i-1/2,k)} + \left(\hat{y}_p + 0.5\right)\left(bq_{y(j^*,i+1/2,k)} - bq_{y(j^*,i-1/2,k)}\right)\right]\Big\} \qquad (48)$$

where the local particle coordinates ($\hat{x}_p$ and $\hat{y}_p$) are as defined previously in eq. 37, and the interpolation factors (F_i) and adjacent node locators (j^* and i^*) are defined as:

$$F_y = -\hat{y}_p \qquad i^* = i\text{-}1 \qquad \text{for } \hat{y}_p < 0 \quad \text{(49a)}$$

$$F_y = 0.0 \qquad \text{for } \hat{y}_p = 0 \quad \text{(49b)}$$

$$F_y = \hat{y}_p \qquad i^* = i\text{+}1 \qquad \text{for } \hat{y}_p > 0 \quad \text{(49c)}$$

and

$F_x = -\hat{x}_p \qquad j^* = j\text{-}1 \qquad \text{for } \hat{x}_p < 0$ (49d)

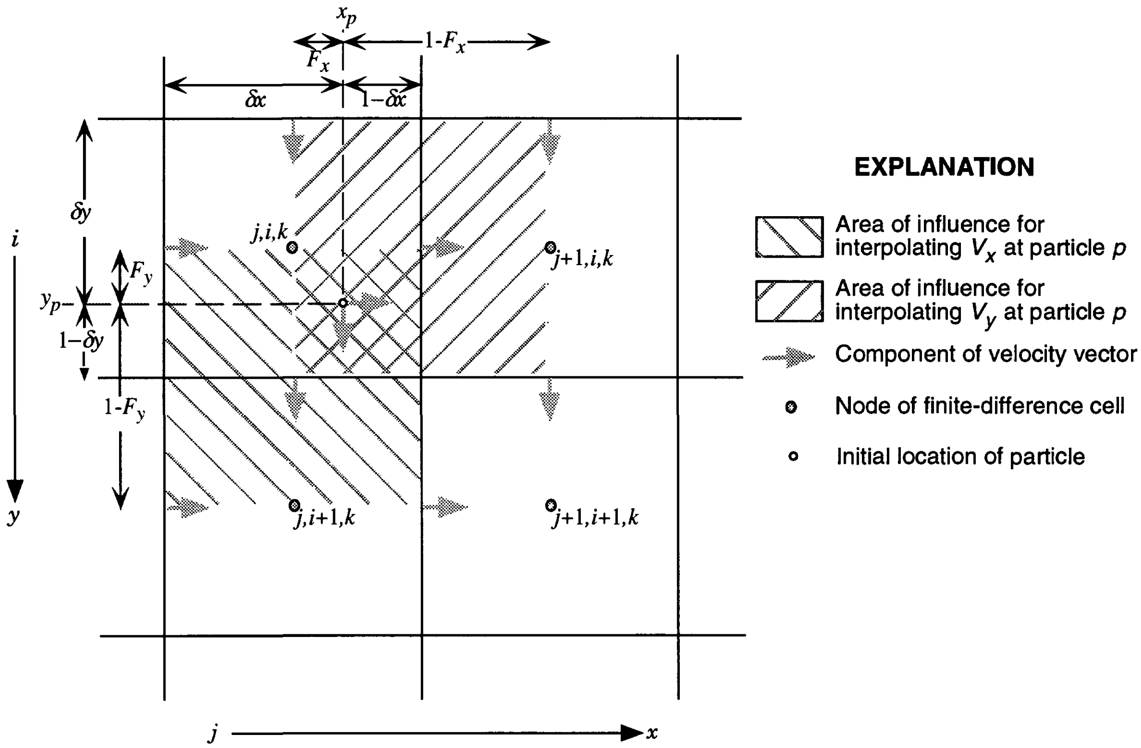
$F_x = 0.0 \qquad \text{for } \hat{x}_p = 0$ (49e)

$F_x = \hat{x}_p \qquad j^* = j\text{+}1 \qquad \text{for } \hat{x}_p > 0.$(49f)

The bilinear interpolation scheme to estimate V_x and V_y is illustrated in fig. 6, which shows a particle p located in cell j,i,k. V_x and V_y at p depend on the relative position within the cell in both the x- and y-directions. V_x and V_y at p are calculated from eqs. 47 and 48 using the weighting factors defined in eqs. 49a-f. Note that $(\hat{x}_p + 0.5)$ of eq. 47 equals δx in fig. 6, and $(\hat{y}_p + 0.5)$ of eq. 48 equals δy in fig. 6. Note that the interpolated values of V_x and V_y at p are determined using specific discharge values at differing adjacent cell faces.

Also, if the cell face adjacent to the quadrant containing a particle of interest represents a no-flow boundary, then the component of the flow parallel to that boundary should not change in the area of the half-cell between the node and that face. Therefore, in such a case, linear interpolation is used within that half-cell area for that component only. For example, in fig. 6, if cell $(j,i+1,k)$ were a no-flow cell, then V_x at p would be estimated by linear interpolation between $(j\text{-}1/2,i,k)$ and $(j+1/2,i,k)$. V_y at p would still be estimated using bilinear interpolation, as shown in fig. 6, properly recognizing that $V_y = 0.0$ at $(j,i+1/2,k)$.

Bilinear interpolation yields velocity components that depend on both the x- and y-positions, hence the semi-analytic integration of position used above is not applicable. For bilinear interpolation, particle movement is computed explicitly using the velocity of the



x_p
F_x
1-F_x
δx
1-δx
δy
i
y_p
F_y
1-δy
1-F_y
y
j,i,k
j+1,i,k
j,i+1,k
j+1,i+1,k
j
x
EXPLANATION
Area of influence for interpolating V_x at particle p
Area of influence for interpolating V_y at particle p
Component of velocity vector
Node of finite-difference cell
Initial location of particle

Figure 6. Interpolation factors used in bilinear interpolation method to estimate horizontal components of velocity, V_x and V_y, at the position of a particle located in the southeast quadrant of cell j,i,k.

particle at its starting position and the length of the time increment:

$$x_1 = x_0 + V_{x,0}\Delta t \tag{50a}$$

$$y_1 = y_0 + V_{y,0}\Delta t. \tag{50b}$$

The times required for a given particle to reach the x- and y-faces of the cell in which it is located are computed using:

$$\Delta t_x = \frac{x_e - x_0}{V_{x,0}} \tag{51}$$

and

$$\Delta t_y = \frac{y_e - y_0}{V_{y,0}}. \tag{52}$$

For the z-direction, all factors are computed as described previously in eqs. 34-45.

Also, following the same procedure as implemented in the previously described linear interpolation scheme, a particle that would cross a cell boundary during a move interval (or transport time increment) is temporarily stopped at whichever cell boundary (or face) it reaches first. At that location the particle velocity is updated using the porosity and thickness of the cell that the particle is entering, and the particle is then moved an appropriate distance in the new cell for the remainder of the time increment ($\Delta t'$, as calculated from eq. 46), or until it reaches another cell face. Computation of particle movement continues until the entire movement time interval is exhausted.

Discussion—Choosing an appropriate interpolation scheme

As noted by Goode (1990a), selecting the best interpolation scheme to determine particle velocity in a ground-water flow model depends in part on the conceptualization of aquifer heterogeneity. Linear interpolation is directly consistent with the block-centered finite-difference solution of the flow equation. However, linear interpolation produces a discontinuous velocity field, even for homogeneous media (Goode, 1990a). In the presence of strong heterogeneities between adjacent cells within a layer, it would usually be preferable to select the linear interpolation scheme.

If transmissivity within a layer is homogeneous or smoothly varying, bilinear interpolation of velocity yields more realistic pathlines for a given discretization than does linear interpolation. In such cases, the bilinear interpolation scheme may be preferable because it will yield pathlines more consistent with the conceptualization. For example, Goode (1990a) shows that bilinear interpolation yields more accurate travel times and pathlines for a case in which the interblock hydraulic conductivity is computed using the logarithmic mean (Goode and Appel, 1992) corresponding to a linear variation of hydraulic conductivity between nodes. Because vertical heterogeneity is significant in typical aquifer systems, the *MOC3D* code always uses linear interpolation for velocity in the vertical direction.

In general, the choice of interpolation scheme will only make a small difference in the final solution. Using a finer grid will invariably have a larger effect and yield a more accurate definition of the flow field, particularly in areas where the hydraulic properties are changing in space.

Decay

Decay is simulated by reducing the particle concentrations after advection. At this point the particle concentration has not yet been adjusted for dispersion and sources. However, the change in particle position accounts for advection up to time increment t+1. The loss of solute mass during a given transport time increment (Δt) because of decay processes is accounted for by computing the decayed particle concentration, C_p^d:

$$C_p^d = C_p^t \; e^{-\lambda \Delta t} \qquad (53)$$

where C_p^t is the particle concentration at the start of the move interval (and before advection).

A major advantage of calculating the effect of decay directly on the particles, rather than on the nodal concentrations, is that this procedure eliminates any possible reduction in precision (or numerical dispersion) caused by the framework transfer between the moving grid and the fixed grid (that is, from particle concentrations to average nodal concentrations and back again to particle concentrations), as is done in computing changes in concentration due to dispersion. The dispersion calculation is then based on average concentration gradients during a time increment, an average that is based on the advected and decayed concentrations at the end of the time increment and the particle concentrations at the start of the time increment.

As noted by Goode and Konikow (1989), this exponential formulation has no numerical stability restrictions associated with it. However, if the half-life is on the order of the transport time increment or smaller, then some accuracy will be lost because of the explicit decoupling of decay and other transport processes.

When a solute subject to decay enters the aquifer through a fluid source, it is assumed that the fluid source contains the solute in the concentration specified by C'. The governing equation and the *MOC3D* model assumes that decay only occurs within the ground-water system, and not within the source reservoir. In other words, for a given stress period, the model assumes that C' remains constant in time and does not decay. If the problem being simulated requires that the source fluid itself undergo decay, then the code will have to be modified to allow this.

Node Concentrations

After all particles have been moved, the concentration at each node is temporarily assigned the average concentration of all particles then located within the volume of that cell; this average concentration is denoted as $C_{j,i,k}^{adv}$.

$$C_{j,i,k}^{adv} = \frac{\sum_{p=1}^{N} C_p^d \delta\left(j_p^{t+1} = j, i_p^{t+1} = i, k_p^{t+1} = k\right)}{\sum_{p=1}^{N} \delta\left(j_p^{t+1} = j, i_p^{t+1} = i, k_p^{t+1} = k\right)} \qquad (54)$$

where the δ function is 1 if the particle is within the cell $j, i, k,$ and is zero otherwise. The time index is labeled "*adv*" because this temporarily assigned average concentration represents the new time level only with respect to advective transport and decay. With respect to the finite-difference grid, the effect of advective transport is to move particles with differing concentrations into and out of each cell.

Finite-Difference Approximations

The divergence of dispersive flux is normalized by the retardation factor and porosity to yield the rate of change in concentration. In addition, in a quasi-3D approach, changes in saturated thickness are incorporated for horizontal flux terms. The rate of change in concentration due to dispersion and mixing in cells having a fluid source can be written:

$$\left(\frac{\mathrm{d}C}{\mathrm{d}t}\right)_{j,i,k}=\frac{1}{\left(R_f\right)_k(\varepsilon b)_{j,i,k}^{t+1}}\left\{\Delta x^{-1}\left[\left(\varepsilon bD_{1m}\frac{\partial C}{\partial x_m}\right)_{j+1/2,i,k}^{*}-\left(\varepsilon bD_{1m}\frac{\partial C}{\partial x_m}\right)_{j-1/2,i,k}^{*}\right]\right.$$

$$+\Delta y^{-1}\left[\left(\varepsilon bD_{2m}\frac{\partial C}{\partial x_m}\right)_{j,i+1/2,k}^{*}-\left(\varepsilon bD_{2m}\frac{\partial C}{\partial x_m}\right)_{j,i-1/2,k}^{*}\right]$$

$$\left.+\left[\left(\varepsilon D_{3m}\frac{\partial C}{\partial x_m}\right)_{j,i,k+1/2}^{*}-\left(\varepsilon D_{3m}\frac{\partial C}{\partial x_m}\right)_{j,i,k-1/2}^{*}\right]+\sum_{W>0}\left[W_{j,i,k}\left(C'_{j,i,k}-C_{j,i,k}^{*}\right)\right]\right\} \qquad (55)$$

where subscript m is a summation index for the dispersion term. The j,i,k subscripts in eq. 55 denote the spatial finite-difference grid indexing, as discussed previously in the section "Numerical Methods." The superscript "*" indicates that the terms depend on the average of the concentration at the old time level, and the concentration at the new time level after advection (see eq. 32). These averaged concentrations are used to calculate the solute flux terms indicated by the superscript "*".

The components of the dispersive flux in each direction across cell faces are calculated using finite-difference approximations that are centered-in-space and explicit (forward-in-time). A detailed description of these finite-difference approximations is given in Appendix A.

The explicit finite-difference approach is conceptually straightforward, but only applicable for certain conditions. Reddell and Sunada (1970, p. 62) show that for an explicit finite-difference solution of eq. 31 to be stable, the following constraint must be met:

$$\frac{D_{xx}\Delta t}{(\Delta x)^2}+\frac{D_{yy}\Delta t}{(\Delta y)^2}+\frac{D_{zz}\Delta t}{(\Delta z)^2}\leq\frac{1}{2}. \qquad (56)$$

Inspection of eq. 56 shows that the constraint will be most readily met for relatively small values of the dispersion coefficient (that is, for advection-dominated problems). Note that stability does not necessarily assure accuracy. If the constraint is not met for a given set of physical parameters, then either the grid spacing must be increased (with a consequent loss of accuracy) or the time increments must be decreased until the condition is satisfied.

Note that the dispersive fluxes calculated by solving the eqs. A1-A3 include contributions related to the cross-product terms of the dispersion tensor. However, the constraint expressed in eq. 56 only includes the principal components of the dispersion tensor. In circumstances where the cross-product terms are relatively large, this can lead to a calculation of a negative concentration at a node if the calculated solute flux out of a cell during a time increment is greater than the solute mass in that cell at the end of the previous time increment. Considering eqs. 22a and 22d, for example, we can see that the cross-product terms (such as D_{xy} and D_{yx}) would be large relative to a principal diagonal coefficient (such as D_{xx}) when both V_x and V_y are significant and α_L is significantly greater than α_T. For a given velocity and fixed dispersivities, the cross-product terms are maximized when $|V_x|=|V_y|$ and flow is at 45 degrees to the grid. Negative concentrations are more likely to occur using this formulation in the presence of steep concentration gradients.

Consider the solute mass flux due to dispersion across one face of a cell, say in the x-direction across the cell face at $(j+1/2,i,k)$. The change in solute mass, $M_{f(j+1/2,i,k)}$, is equal to the rate of dispersive flux, expressed in eq. A1, multiplied by the length of the time increment and by the width of the cell:

$$
\begin{aligned}
M_{f(j+1/2,i,k)} = &-\Delta t \Delta y \varepsilon b \Bigg[\left(D_{xx}\right)^{t+1}_{j+1/2,i,k} \frac{\left(C^{*}_{j+1,i,k} - C^{*}_{j,i,k}\right)}{\Delta x} \\
&+ \left(D_{xy}\right)^{t+1}_{j+1/2,i,k} \frac{\left(C^{*}_{j,i+1,k} + C^{*}_{j+1,i+1,k} - C^{*}_{j,i-1,k} - C^{*}_{j+1,i-1,k}\right)}{4\Delta y} \\
&+ \left(D_{xz}\right)^{t+1}_{j+1/2,i,k} \frac{1}{2}\left(\frac{C^{*}_{j+1,i,k+1} - C^{*}_{j+1,i,k-1}}{2B^{t+1}_{j+1,i,k}} + \frac{C^{*}_{j,i,k+1} - C^{*}_{j,i,k-1}}{2B^{t+1}_{j,i,k}} \right) \Bigg]. \qquad (57)
\end{aligned}
$$

This mass can be compared with the solute mass in the cell at the start of the time increment, $M^{t}_{j,i,k}$, which is given by:

$$M^{t}_{j,i,k} = \Delta x \Delta y \varepsilon b C^{t}_{j,i,k}. \qquad (58)$$

The criterion in eq. 56 is equivalent to requiring that the sum of the solute mass fluxes across all faces of a cell must be less than or equal to $M^{t}_{j,i,k}$. However, because the criterion does not include the dispersive flux related to the cross-product terms, it can lead to oscillations in the solution that yield negative concentrations, although it would rarely lead to a strict stability problem. Experience shows that when oscillations do occur because of the cross-product terms, they are usually small and tend to damp out over time.

An *ad hoc* procedure was formulated and implemented in *MOC3D* to help minimize and limit the occurrence of negative concentrations due to the cross-product dispersive flux. The approach is to limit the mass flux during one time increment across each cell face, as calculated by eq. 57, to the mass available in the cell, as calculated by eq. 58. The flux across each cell face is checked independently, so this constraint will not completely eliminate negative concentrations, but our experience indicates that it will often reduce their occurrence significantly. Because the same check will be applied from both sides of a given face, if the constraint is applied, it will be applied equally from both adjacent nodes, so the procedure will not affect the global mass balance. That is, the solute mass flux into one cell always corresponds to the mass flux out of the adjacent cell, whether or not this limiting procedure is implemented for that particular cell face.

Stability and Accuracy Criteria

As noted by Konikow and Bredehoeft (1978), 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, consider the explicit finite-difference solution to calculate changes in concentration due to dispersion. Solving eq. 56 for Δt, and accounting for the effects of retardation, we see that

$$\Delta t \leq \underset{\substack{\text{(over}\\ \text{grid)}}}{\text{Min}} \left[\frac{0.5}{\dfrac{D_{xx}}{R_f(\Delta x)^2} + \dfrac{D_{yy}}{R_f(\Delta y)^2} + \dfrac{D_{zz}}{R_f(\Delta z)^2}} \right]. \qquad (59)$$

Because the solution to eq. 31 is actually written as a set of *N* equations for *N* nodes, the maximum permissible time increment is the smallest Δt computed for any individual node in the entire transport grid. The smallest Δt will then occur at the node having the largest value of

$$\frac{D_{xx}}{R_f(\Delta x)^2}+\frac{D_{yy}}{R_f(\Delta y)^2}+\frac{D_{zz}}{R_f(\Delta z)^2}.$$

Next consider the effects of mixing ground water of one concentration with injected or recharged water of a different concentration, as represented by the source terms in eq. 31. The change in concentration in a source node cannot exceed the difference between the source concentration ($C'_{j,i,k}$) and the concentration in the aquifer ($C_{j,i,k}$), and the maximum possible change occurs when a source completely flushes out the volume of water in an aquifer cell at the start of a time increment. Konikow and Bredehoeft (1978) show that this condition translates to

$$\frac{\Delta t\, W_{j,i,k}}{\varepsilon R_f} \le 1.0. \qquad (60)$$

Solving eq. 60 for Δt at all nodes yields the following criterion:

$$\Delta t \le \underset{\text{(over grid)}}{\text{Min}}\left[\frac{\varepsilon R_f}{W_{j,i,k}}\right]. \qquad (61)$$

A third type of criterion involves the movement of points to simulate advective transport. The distance a particle moves during a time increment is equal to (or approximately so in cases where particles cross a cell face and the adjacent cells have different properties) the velocity at the location of the particle times the length of the time increment. In effect, this constitutes a linear spatial extrapolation of the position of a particle from one time increment to the next. Konikow and Bredehoeft (1978) note that 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 Δt. An accurate computation of concentration changes caused by advective transport requires the maintenance of a relatively uniformly spaced field of marker particles that are moving along relatively smooth and continuous pathlines. The degree of curvilinearity of streamlines in the calculated head field is constrained by the grid spacing, as the finite-difference solution to the flow equation inherently assumes linear variations in head between adjacent nodes. Also, if the distance a particle moves in any direction during one time increment is greater than the grid spacing in that direction, it might be possible for a particle to cross a no-flow boundary (or even leave the model domain) 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 the distance a particle moves in the *x*-, *y*-, or *z*-directions during one time increment does not exceed some critical distances, which can be related to the grid spacing in each direction.

These critical distances can be related to the grid dimensions by

$$\Delta t\ V_{x(p)} \le \gamma \Delta x \qquad (62a)$$

$$\Delta t\ V_{y(p)} \le \gamma \Delta y \qquad (62b)$$

and

$$\Delta t\ V_{z(p)} \le \gamma \Delta z \qquad (62c)$$

where γ is the fraction of the grid dimensions that particles will be allowed to move (normally, $0 < \gamma \le 1$). Note that these accuracy criteria are equivalent to requiring that the Courant number be less than or equal to 1. However, the model is designed to allow the user to specify the value of γ (named *CELDIS* in the code and input instructions).

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 components computed at cell boundaries, we have to check only the latter. Substituting grid velocity components and solving eq. 62 for Δt results in

$$\Delta t \leq \frac{\gamma \Delta x}{(V_x)_{\max}} \qquad (63a)$$

$$\Delta t \leq \frac{\gamma \Delta y}{(V_y)_{\max}} \qquad (63b)$$

and

$$\Delta t \leq \frac{\gamma \Delta z}{(V_z)_{\max}}. \qquad (63c)$$

If the time step used to solve the flow equation exceeds the smallest of the time limits determined by eqs. 59, 61, or 63, then the time step will be subdivided into an appropriate number of equal-sized smaller time increments to solve the solute-transport equation so that none of these limits are exceeded. To help the user analyze the results and the grid design, the model output will include a statement clarifying which of the several criteria were limiting and at which node the limiting condition occurred.

Mass Balance

Mass balance calculations are performed to help check the numerical accuracy and precision of the solution. The principle of conservation of mass requires that the net mass flux (cumulative sum of mass inflows and outflows plus any mass lost or removed by reactions) must equal the mass accumulation (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 (Konikow and Bredehoeft, 1978).

As part of the mass balance calculations, the solute fluxes contributed by each distinct hydrologic component of the flow and transport model are accumulated and itemized separately to produce a solute budget for the system being modeled. The budget is a valuable assessment tool because it provides a measure of the relative importance of each component to the total solute budget. The budgets should always be reviewed for consistency with the conceptual model and as a "reality check" on the model calculations.

In the method of characteristics, the accuracy of the solution is associated with the concentrations being tracked on the particles. However, it is computationally difficult to compute a mass stored in the system directly from the particle concentrations because their relative positions are constantly changing and they do not explicitly track a solute or fluid mass. Therefore, the mass in storage at any time is calculated from the concentrations at the nodes of the transport subgrid of the finite-difference mesh. In that sense, the calculated and printed mass balance values are themselves only an approximation.

The mass residual is computed as

$$R_m = \Delta M_s - M_f \qquad (64)$$

where ΔM_s is the change in mass stored in the aquifer, and M_f is the net mass flux.

The above two mass balance terms are evaluated using the following equations:

$$\Delta M_s = \sum_k \sum_i \sum_j \Delta x \Delta y \left((\varepsilon b C)^n_{j,i,k} - (\varepsilon b C)^0_{j,i,k} \right) \qquad (65)$$

where $C^0_{j,i,k}$ is the initial concentration at node (j,i,k), M/L^3, and $C^n_{j,i,k}$ is the concentration at that node at the end of the time increment; and

$$M_f = \sum_k \sum_i \sum_j \sum_n W^n_{j,i,k} b \Delta x \Delta y \Delta t_n C'_{j,i,k,n}. \qquad (66)$$

For cases where W in eq. 66 represents a fluid source, C' is the specified source concentration for that node. Where W represents a fluid sink, C' is assumed to equal the average concentration in cell (j,i,k) at the beginning of that time increment (or move interval) for solving the transport equation.

The percent error *(E)* in the mass balance is calculated by relating the residual to an appropriate measure of the solute flux or mass accumulation in the system. However, the appropriate basis (or denominator) is problem dependent. Thus, the model compares the residual with the cumulative mass inflow to the system (M_i) or mass outflow from the system (M_o), whichever is dominant in a particular problem. However, if the solute mass entering or discharging the system with fluid sources and sinks is zero or very small, as it would be, for example, in a problem simulating the movement of an initial slug within the system, then comparing the residual to the mass flux in or out of the system could indicate a very large error when the numerical solution is actually quite accurate. Therefore, in these cases, the error will be computed by comparing the residual with the mass of solute stored in the aquifer, as described by Konikow and Bredehoeft (1978). The model will calculate, print, and label whichever of the following three measures of error are appropriate for the problem being simulated.

$$E_1 = \frac{100.0 R_m}{M_i} \tag{67a}$$

$$E_2 = \frac{100.0 R_m}{M_o} \tag{67b}$$

$$E_3 = \frac{100.0 R_m}{M_s}. \tag{67c}$$

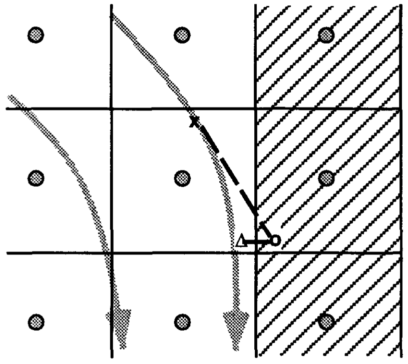
E_3 is calculated only if the mass flux in or out of the system is less than 50 percent of the initial mass stored.

Errors in the mass balance for the flow model should generally be less than 0.1 percent. However, because the solute-transport equation is more difficult to solve numerically than is the flow equation, the mass-balance error for a solute is often greater than for the fluid. Also, because the particles that represent advection in the method of characteristics are discrete in nature and because the concentrations tracked on particles are translated to the finite-difference nodes for the purpose of computing the mass balance, the mass balance error will typically exhibit an oscillatory behavior over time. However, this is not a cumulative type of error; it is usually largest for the first few time increments and then tends to balance out over time. As long as the oscillations remain within a steady range, not exceeding about ±10 percent as a guide, then the error probably does not represent a bias and is not a serious problem. Rather, the oscillations only reflect the fact the mass balance calculation is itself just an approximation. Thus, it is recommended that users examine the solute budget and residuals for their particular problem. The significance of the residual and rates of change in the residual should be assessed qualitatively relative to the nature of a particular problem, and not merely on the basis of the magnitude of the error at any one time increment.

Special Problems

As noted by Konikow and Bredehoeft (1978), a number of special problems are 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, whereas other problems are related to the computational transition between the concentrations of particles within a cell and the average concentration at that node. This section describes 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 solutes 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 advected across the boundary during one time increment. Figure 7 illustrates such a situation, which arises from the deviation between the curvilinear flow line and the linearly projected particle path. Figure 7 also shows the correction scheme built into the *MOC3D* model. If a particle is advected across a no-flow boundary, then it is relocated within the aquifer by reflection across the boundary. This correction thus will tend to relocate the particle closer to the true flow line. However, extensive testing indicates that it is unlikely that a particle will ever cross a no-flow boundary unless CELDIS > 1, which is not recommended.



EXPLANATION

▨	Zero-transmissivity cell (or no-flow boundary)
→	Flow line and direction of flow
— — —	Computed path of particle
●	Node of finite-difference cell
x	Previous location of particle
o	Computed new location of particle
Δ	Corrected new location of particle

Figure 7. Possible movement of a particle near an impermeable (no-flow) boundary (modified from Konikow and Bredehoeft, 1978).

The maintenance of a reasonably uniform and continuous spacing of particles requires special treatment in areas where strong fluid sources and sinks dominate the flow field. Strong fluid sources and sinks cause significant convergence and divergence in the flow field, which will degrade the desired uniform spacing of particles. Without special provisions, particles will continually move out of a cell that represents a strong fluid source, but few or none will move in to replace them and thereby maintain a continuous stream of particles. Thus, whenever a particle that originated in a strong fluid-source cell moves out of that source cell, a new particle is introduced into the source cell to replace it. Placement of new particles in a source cell is compatible with and analogous to the generation of fluid and solute mass at the source. On the other hand, if a fluid source or sink is very weak, it will not induce significant divergence or convergence in the flow field and have any noticeable effect on particle spacing. For cells representing weak fluid sources or sinks, particles need not be added or removed. The model user must specify explicitly whether fluid sources and sink are to be flagged as either weak or strong, so that particle tracking is implemented appropriately. Source/sink cells are identified as either strong or weak for purposes of particle control by the user-specified value of the IGENPT array (see Appendix B).

The procedure used to replace particles in source cells is illustrated in figure 8. A steady, uniformly spaced stream of particles is maintained by generating a new particle in the source cell at the original location of the particle that left the source cell. When a relatively strong fluid source is imposed on a relatively weak regional flow field, as illustrated in figure 8a, then radially divergent flow will be maintained in the area of the source, and all initial and replacement points will move symmetrically away from node *j,i,k*. For example, after particle 7 moves from cell *(j,i,k)* at the start of a time increment to cell *(j+*1*,i-*1*,k)* at the end of that time increment, the replacement particle (particle 18 in fig. 8a) is positioned at time *n* in cell *(j,i,k)* at the same location as the initial position of particle 7.

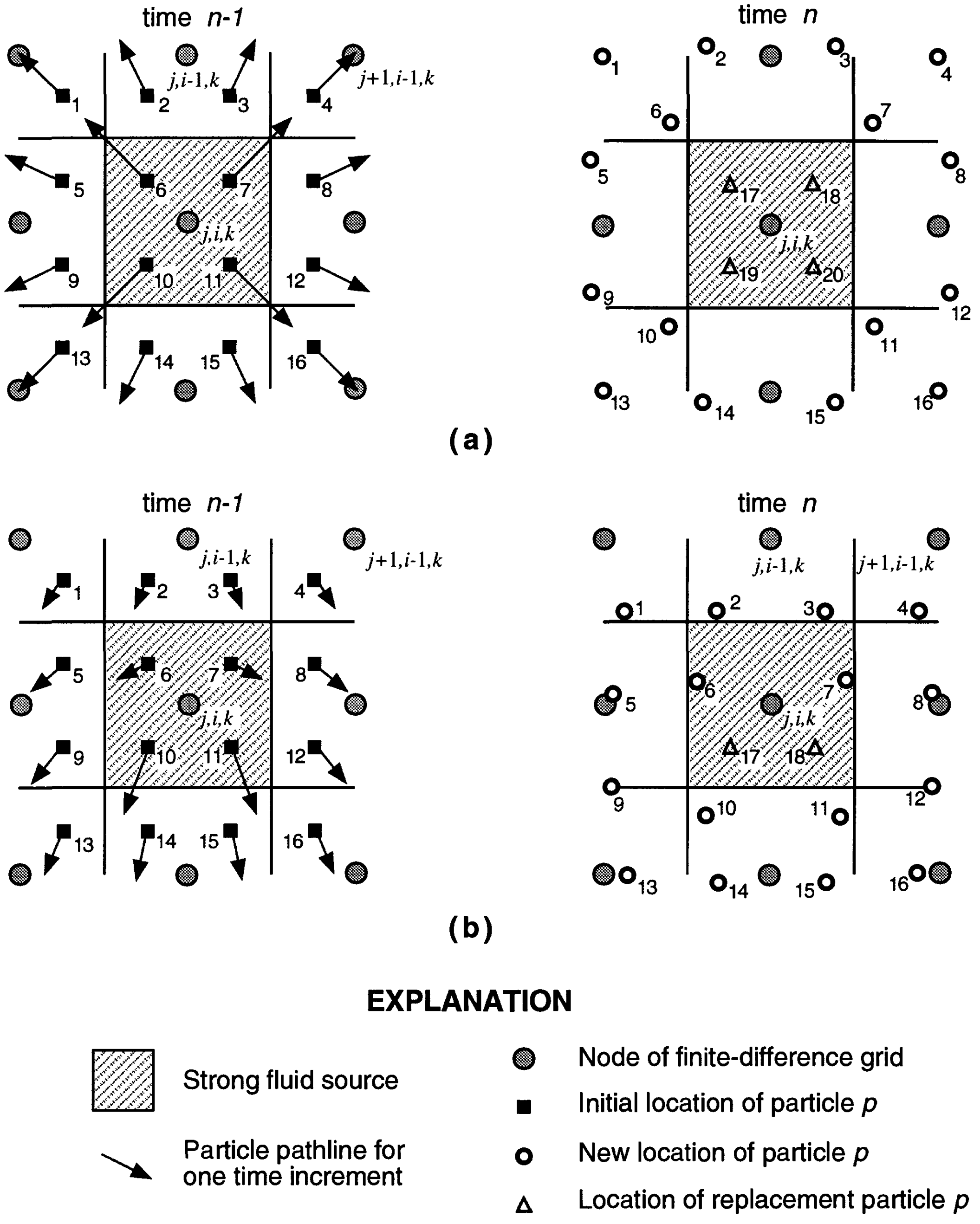

time n-1
time n
j,i-1,k
j+1,i-1,k
j,i,k
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
(a)
(b)
EXPLANATION
Strong fluid source
Particle pathline for one time increment
Node of finite-difference grid
Initial location of particle p
New location of particle p
Location of replacement particle p

Figure 8. Replacement of particles in fluid-source cells (a) for case of negligible regional flow and (b) for case of relatively strong regional flow.

Although we normally expect particles to be advected out of fluid-source cells, figure 8b demonstrates the possibility that particles may sometimes enter a source cell. When a relatively weak fluid source is imposed on a relatively strong regional flow field, the velocity distribution within the source cell does not possess radial symmetry, and the velocity within the upgradient part of the source cell is lower than the velocity within the downgradient part of the source cell. It is possible then that particles originating in upgradient cells (such as particle 2) will migrate through the designated source cell. This can also occur when two or more source cells of different strengths are adjacent to each other. Particles that leave a source cell, but did not originate in it, are not replaced because that would ultimately lead to mass balance errors in downgradient areas as proportionally too many particles will exist in downgradient areas relative to the volume of water being added at the source cell. Some particles that originated in a strong source cell may take more than one time increment to leave that cell (for example, particles 6 and 7 in fig. 8b). Although not illustrated in fig. 8, when these lower-velocity particles do leave the source cell at time $n+1$, the replacement particles will be placed at the original positions of the particles (which was at time n-1) rather than at their positions immediately prior to leaving (which was at time n).

Hydraulic sinks also require some special treatment. Particles will continually move into a cell representing a strong sink, but few or none will move out. To avoid the resultant accumulation and stagnation of tracer particles, any particle moving into a strong-sink cell is removed from the flow field after the calculations for that time increment have been completed. The numerical removal of particles that enter sink cells is analogous to the withdrawal of fluid and solute mass through the hydraulic sink. If the relative magnitude of discharge from a sink cell is not strong enough to maintain radially convergent flow, and a particle exits from the sink cell, then a replacement particle will be placed at the center of the cell. The combination of creating new particles at sources and destroying old particles 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 uniformly over the entire area or volume of the cell surrounding a source or sink node. Thus, in effect, heads and concentrations computed at source or sink nodes represent average values over the area or volume 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. 17). It can be shown from the relation between the source concentration ($C'_{j,i,k}$) and the aquifer concentration at the start of a time increment ($C^{n-1}_{j,i,k}$), that the following constraints generally must be met in a source cell:

$$C^n_{j,i,k} \le C'_{j,i,k} \quad \text{for} \quad C'_{j,i,k} > C^{n-1}_{j,i,k} \qquad (68a)$$

and

$$C^n_{j,i,k} \ge C'_{j,i,k} \quad \text{for} \quad C'_{j,i,k} < C^{n-1}_{j,i,k}. \qquad (68b)$$

If it is assumed that there is complete mixing between the source fluid and the resident fluid within the volume of a strong source cell, then these same constraints should also apply to all points within the cell. Because of the possible deviation of the concentrations of individual particles within a source cell from the average concentration, the change in concentration computed at a source node should not be applied directly to each of the particles in the cell. Rather, at the end of each time increment for solving the transport equation the concentration of each particle in a strong 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 among particles within the volume of the cell, it prevents the adjustment of the concentration for any individual particle in the source cell to a value that would violate the constraints indicated by equation 66.

In areas of divergent flow, a problem may arise because some cells can become void of particles where pathlines become spaced widely apart. This can occur, even in the absence of a strong fluid source, because of heterogeneities or boundary conditions. This would result in a calculation of no change in concentration at a node due to advective transport, although the nodal concentration would still be adjusted for changes caused by hydrodynamic dispersion. Also, some numerical dispersion is generated at nodes in and adjacent to the cells in which the advective 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 advected into void cells during later or subsequent time increments). Figure 8a 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 advective transport, then in the numerical procedure four of the eight surrounding cells would erroneously not receive any solute by advection from the adjacent source. If eight uniformly spaced particles 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. Because the *MOC3D* model is based on a rectangular Cartesian coordinate system, it is not recommended for applications to a purely radial flow problem. However, if radial flow is localized within a predominantly nonradial regional flow field, then satisfactory results should be achievable.

The problem of cells becoming void of particles can be minimized by limiting the number of void cells to a small fraction of the total number of active cells that represent the aquifer. The user specifies this fraction (FZERO) in the *MOC3D* input data file (see Appendix B). If the limit is exceeded, the numerical solution to the solute-transport equation is halted temporarily 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" concentration 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.

Review of *MOC3D* Assumptions and Integration with *MODFLOW*

Following is a brief summary of model application assumptions that have been incorporated into the *MOC3D* model. These are relevant to both grid design and model implementation. Efficient and accurate use of *MOC3D* requires the user to be aware of all of these assumptions and options.

- *MOC3D* is integrated with *MODFLOW-96* (Harbaugh and McDonald, 1996a) and will not work with earlier versions. The main *MODFLOW* subroutine is replaced with the *MOC3D* main subroutine. In addition, several *MOC3D*-specific source code files must be compiled and linked to the *MODFLOW* code.
- Particle velocities are interpolated spatially, but not over time. That is, we assume that the head distribution calculated for the end of a given time step applies during that entire time interval.
- The transport model is applied to a "window" of the grid used to solve the flow equation. This subgrid can be equal in size or smaller than the primary *MODFLOW* grid.
- Within the area of the transport subgrid, row and column discretization must be

uniformly spaced (that is, Δx and Δy must be constant, although they need not be equal to each other). The spatial discretization or rows and columns beyond the boundaries of the subgrid can be nonuniform, as allowed by *MODFLOW*, to permit calculations of head over a much larger area than the area of interest for transport simulation (see fig. 9). Vertical discretization, defined by the cell thickness, can be variable in all three dimensions. However, large variability may adversely affect numerical accuracy (as discussed in second item below).

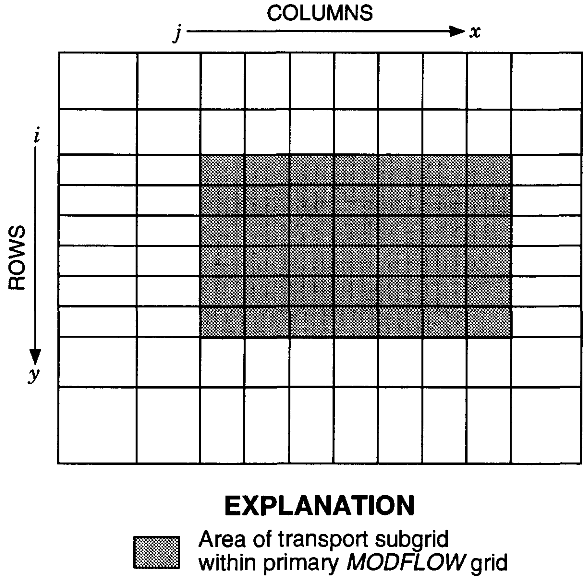


Figure 9. One layer of finite-difference grid illustrating use of uniformly-spaced transport subgrid for *MOC3D* within variably-spaced primary grid for *MODFLOW*.

- Retardation factor values and all dispersivity values are constant in each layer. Values for porosity may vary within a layer and are defined for each node (also see discussion in next item).

- The particle-tracking algorithm inherently assumes that all particles represent the concentration in an equal volume of water in a cell, where the volume equals $(\varepsilon b\Delta x\Delta y)_{j,i,k}$. Thus, although Δx and Δy are uniform, it is also very important that the variations in the product of porosity and thickness within the transport subgrid remain relatively small. Otherwise, when a particle moves into a cell having a very different volume from the cell in which it originated, the estimate of the average concentration in the new cell may become biased, which will also have an adverse effect on the overall mass balance for the solute.

- *MODFLOW* offers flexibility to the user in the conceptualization of vertical discretization (see McDonald and Harbaugh, 1988, Ch. 2). As illustrated in fig. 10a, it is common in applications of *MODFLOW* to represent the resistance to flow in a low hydraulic conductivity unit by lumping the vertical hydraulic conductivity and thickness of the confining unit into the vertical conductance term between the adjacent layers (fig. 10c). However, because transport simulation requires that travel distances be known explicitly in all directions, three-dimensional transport simulation requires fully three-dimensional flow simulation (rather than a quasi-three-dimensional analysis) within the area of the transport subgrid. That is, even if the solution to the flow equation is insensitive to heads and storage releases in the clay layer, it must still be represented by one or more model layers for the solution to the transport equation (fig. 10b). If not, it can be seen in the quasi-three-dimensional analysis (fig. 10c) that any solute crossing the lower boundary of layer 1 would immediately be located in and influenced by the properties of layer 2, and would never have been subject to the relatively long travel time through the clay.

- In *MODFLOW*, layers may be defined as representing confined or unconfined conditions, or allowed to switch between these two types, depending on the specification of the value of the parameter LAYCON. If LAYCON>0, then the user also must specify information about the elevations of the top and bottom of the layer. For layers having LAYCON=0 or LAYCON=2, *MODFLOW* assumes that transmissivity and saturated thickness remains constant; hence, the

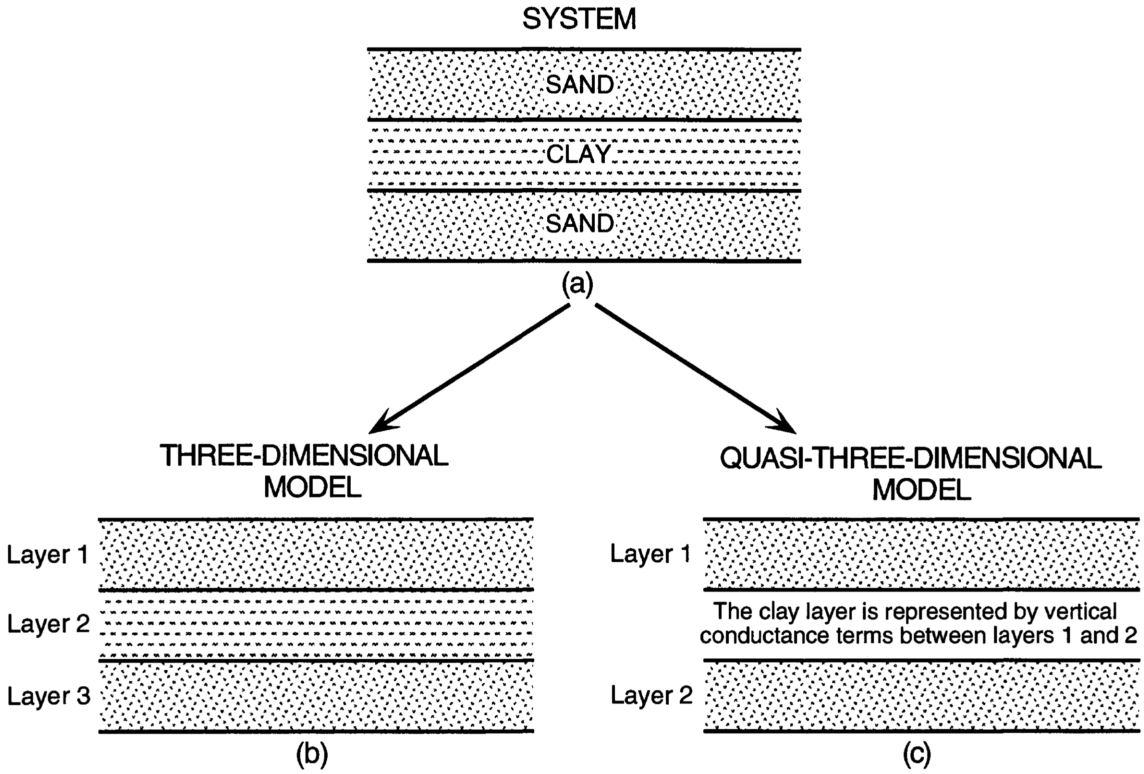

SYSTEM
SAND
CLAY
SAND
(a)
THREE-DIMENSIONAL MODEL
Layer 1
Layer 2
Layer 3
(b)
QUASI-THREE-DIMENSIONAL MODEL
Layer 1
The clay layer is represented by vertical conductance terms between layers 1 and 2
Layer 2
(c)

Figure 10. Alternative *MODFLOW* approaches to vertical discretization of an aquifer system (a) consisting of two high-permeability units separated by a confining layer consisting of a low-permeability clay. In the fully three-dimensional representation (b), the clay unit would be represented by one or more model layers. In the quasi-three-dimensional approach (c), heads are not calculated in the clay unit, which is represented more simply by vertical conductance terms between the layers above and below it; in this case, the bottom of layer 1 coincides with the top of layer 2. (Modified from figs. 10-12 of McDonald and Harbaugh, 1988.)

thickness values read from the *MOC3D* input files are used. However, for layers having LAYCON=1 or LAYCON=3, *MODFLOW* allows the transmissivity to change as a function of changes in saturated thickness in each cell on the basis of elevation data that are input for the *MODFLOW* BCF package. In these cases, thickness for a cell is defined as “TOP” minus “BOT” if the layer is confined and “HEAD” minus “BOT” if it is unconfined; the thickness values specified in the *MOC3D* input file are ignored in the calculations. However, note that thickness values must still be specified in the *MOC3D* input file in all cases.

• Concentrations associated with fluid sources are read directly from *MODFLOW* source/sink package files. In each of the package files used, the options “CBCALLOCATE” and “AUXILIARY CONC” must be included on the first line of input data. See *MODFLOW* and *MOC3D* input instructions.

• If the evapotranspiration package (EVT) is implemented, *MODFLOW* will calculate a fluid discharge (or sink) rate that is typically associated with an evapotranspirative process that removes water but excludes dissolved solids, which are retained in the remaining fluid at a consequently higher concentration. *MOC3D* assumes that for any such calculated flux, the associated source concentration (C' in eq. 17) will equal 0 rather than equaling the concentration at the node, as is assumed normally for a fluid sink. This will induce an appropriate increase in concentration at a cell representing a fluid sink due to evapotranspiration. Note that this *MOC3D* assumption should be viewed as a first-order approximation because in actuality (1) the evapotranspirative process may not be 100 percent effective in excluding solutes, depending on the particular chemical species, and (2) calculated solute concentrations may exceed the upper limits of solubility for a particular chemical constituent (and *MOC3D* does not simulate mineral precipitation).

• When the solute of interest is subject to decay, it is assumed that the solute in liquid and solid phases will decay at the same rate. If a fluid source contains the decaying solute, it is subject to decay after it enters the ground-water system, but is not decayed within its "source reservoir."

• All unit numbers specified in the name files for a particular simulation must be unique. Unit numbers 99, 98, and 97 are reserved for the *MODFLOW* name file, the batch mode input file, and the batch mode output file, so cannot be specified for any other use. However, unit numbers may be reused in separate simulation runs in batch mode.

• The model includes output options to create separate binary data files (Ftypes CNCB, VELB, and PRTB); when implemented, the model will write calculated values from the simulation for the selected variables as unformatted data. The concentration and velocity files (CNCB and VELB) use the *MODFLOW* module ULASAV to write the data (see *MODFLOW* documentation). When the velocity option VELB is implemented, the code will first write the velocities in the column direction at all nodes, then all velocities in the row direction, and finally all of the velocities in the layer direction. The velocity and concentration arrays are dimensioned to the size of the transport subgrid only. When particle data are written to a separate binary file (PRTB), the file begins with a header line that includes the move number, number of particles, and length of the transport time increment. A record for each particle in sequence follows the header line and contains the location and concentration of each particle in the following order: column coordinate, row coordinate, layer coordinate, and particle concentration.

COMPUTER PROGRAM

MOC3D is implemented as a package for *MODFLOW*. *MOC3D* uses the flow components calculated by *MODFLOW* to compute velocities across each cell face in the transport domain. The computed velocities are used in an interpolation scheme to move each particle an appropriate distance and direction to represent advection. The effects of fluid sources, dispersion, and decay on concentration are then applied to the particles.

A separate executable version of *MODFLOW*, which is adapted to link with and use the *MOC3D* module, must first be generated and then used to run *MOC3D* simulations. The *MOC3D* code is written in standard FORTRAN, and it has been successfully compiled and executed on multiple platforms, including 486- and Pentium-based personal computers, Macintosh personal computers, and Data General and Silicon Graphics Unix workstations. FORTRAN compilers for each of these platforms vary in their characteristics and may require the use of certain options to successfully compile *MOC3D*. For instance, the compiler should initialize all variables to zero. Depending on the size of the X-array (defined by LENX in the *MODFLOW* source code), options to enable the compiler to handle a large array may be needed.

Implementing *MOC3D* requires the use of a separate "name" file similar to the one used in *MODFLOW*. The principal *MOC3D* input data (such as subgrid dimensions, hydraulic properties, and particle information) are read from the main *MOC3D* data file. Other files are used for observation wells, concentrations in recharge, and several input and output options. Detailed input data requirements and instructions are presented in Appendix B. Also, a sample input data set for a test problem is included in Appendix C.

MOC3D output is routed to a main listing file, separate from the *MODFLOW* listing file. There are also several options for writing specific data to separate output files, which will facilitate graphical postprocessing. Appendix D contains output from the sample data set described in Appendix C.

General Program Features

Because the model assumes that changes in concentration do not affect the fluid properties (such as density and viscosity), the head distribution and flow field are independent of the solution to the solute-transport equation. Therefore, the flow and transport equations can be solved sequentially, rather than simultaneously. But because transport depends on fluid velocity, which is calculated from the solution to the flow equation, the sequence order must be to solve the flow equation first. This sequence is illustrated in figure 11 for a hypothetical problem involving transient flow and three stress periods. The numbered sequence from 1 through 16, which starts at the left edge of the double time line, illustrates the order of solving equations as the simulation progresses through the first five time steps of the first stress period in this hypothetical example. This figure also helps to illustrate the nomenclature used for time parameters in *MODFLOW* and *MOC3D*, as well as the relation between them.

The implicit solution to the flow equation in *MODFLOW* generally allows the use of time steps of increasing length during a given stress

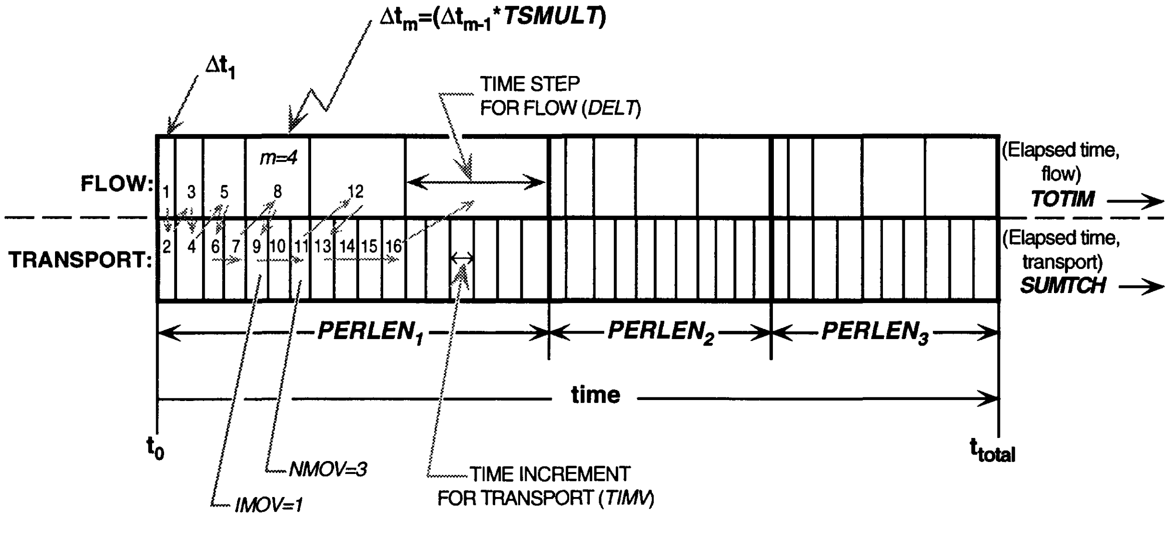


Figure 11. Double time-line illustrating the sequence of progression in the *MOC3D* model for solving the flow and transport equations. This example is for transient flow and three stress periods (*NPER* = 3) of durations *PERLEN*$_1$, *PERLEN*$_2$, and *PERLEN*$_3$. Each time step for solving the flow equation (of duration *DELT*) is divided into one or more time increments (of duration *TIMV*) for solving the transport equation; all particles are moved once during each transport time increment. For illustration purposes, the sequence of solving the two equations is labeled for the first five time steps of the first stress period, and the indices for counting time steps for flow and time increments for transport are labeled for the fourth time step.

period. The length of the first time step for solving the flow equation is calculated by *MODFLOW* on the basis of user-defined values for the number of time steps (NSTP), a time-step multiplier (TSMULT), and the length of the stress period (PERLEN). After the flow equation is solved for the first time step (Δt_1), the model compares the length of the time step for the flow equation with the limitations imposed by the stability and accuracy criteria for solving the transport equation. If any criteria are exceeded, *MOC3D* will subdivide the time step into the fewest number of equal-sized time increments that meet all of the criteria. In the example shown in figure 11, the first two time steps are small enough so that the transport equation can be solved for a single time increment of the same duration as the flow time step (that is, TIMV = DELT). As this equation-solving sequence progresses and is repeated for increasingly long time steps, the stability criteria are eventually exceeded. Figure 11 shows that for the third and subsequent time steps, the transport equation had to be solved over shorter time increments. Note that because time increments for transport are the same length (TIMV) during any given time step for flow, the length of the transport time increments will generally be slightly different between any two different flow time steps. For example, the length of the three transport time increments during the fourth flow time step (m = 4) are slightly different than the lengths of the four time increments during flow time step 5. At any point during the progress of the simulation, the elapsed time for transport is always less than or equal to the elapsed time for flow.

Transport may be simulated within a subgrid, which is a "window" within the primary *MODFLOW* grid used to simulate flow (see fig. 9). The grid dimensions are limited only by the size of the "X" array (see "Space Allocation" in the *MODFLOW* documentation). Within the subgrid, the row and column spacing must be uniform, but thickness can vary within a small range from cell to cell and layer to layer.

Many *MOC3D* subroutines are linked closely with *MODFLOW* counterparts. When possible, *MOC3D* follows *MODFLOW* subroutine structure. In general, data are defined, space is allocated in the "X" array, and simulation parameters are read just as in *MODFLOW*. The overall structure of the *MOC3D* code and its integration with *MODFLOW* are illustrated in figure 12, which shows a flow chart for the main program (excluding details of the transport calculations).

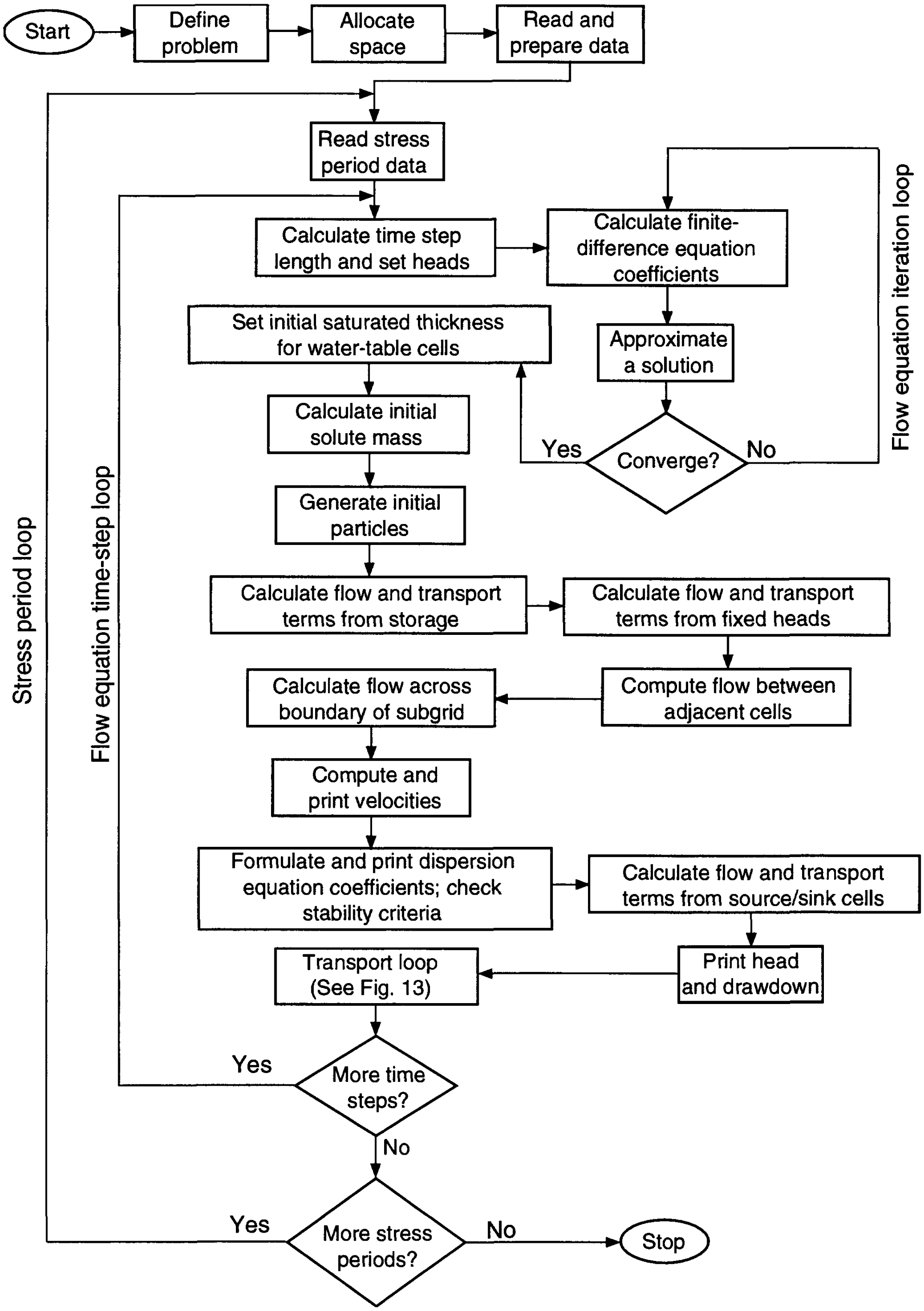
A more detailed flow chart of the program segments controlling transport calculations is shown in figure 13. The fluxes that *MODFLOW* calculates within the transport subgrid are processed by *MOC3D* subroutines to generate a transient solution to the solute-transport equation.

Program Segments

MOC3D input and output utilizes the standard *MODFLOW* array reading and writing utilities as much as possible. *MOC3D* also takes advantage of new features in *MODFLOW*, such as the option for auxiliary parameters in the source and sink packages and storing budget flows from each of those packages, as documented by Harbaugh and McDonald (1996a and 1996b). However, many subroutines in *MOC3D* do not fit into a *MODFLOW* module class. For those model users who are interested in more details about the internal structure and organization of the code, Tables 1-10 list and describe briefly each of the subroutines in *MOC3D* that are used for ten different categories of functions.

Tables 1 and 2 include subroutines that initialize and set up the transport simulation. *MOC3D* data are read and checked for consistency with each other and with several *MODFLOW* parameters. Seepage velocities are calculated on the basis of hydraulic gradients determined by the *MODFLOW* solution to the flow equation. The VELO subroutine (Table 3) is called up to three times after each solution to the flow equation is obtained—once for each dimension of the simulation.

Table 4 includes subroutines controlling particle tracking, concentration calculations, and related output. The SMOC5GP subroutine



Start
Define problem
Allocate space
Read and prepare data
Read stress period data
Calculate time step length and set heads
Calculate finite-difference equation coefficients
Approximate a solution
Converge?
Yes
No
Flow equation iteration loop
Set initial saturated thickness for water-table cells
Calculate initial solute mass
Generate initial particles
Calculate flow and transport terms from storage
Calculate flow and transport terms from fixed heads
Compute flow between adjacent cells
Calculate flow across boundary of subgrid
Compute and print velocities
Formulate and print dispersion equation coefficients; check stability criteria
Calculate flow and transport terms from source/sink cells
Print head and drawdown
Transport loop (See Fig. 13)
More time steps?
Yes
No
More stress periods?
Yes
No
Stop
Stress period loop
Flow equation time-step loop

Figure 12. Generalized flow chart for *MOC3D*.

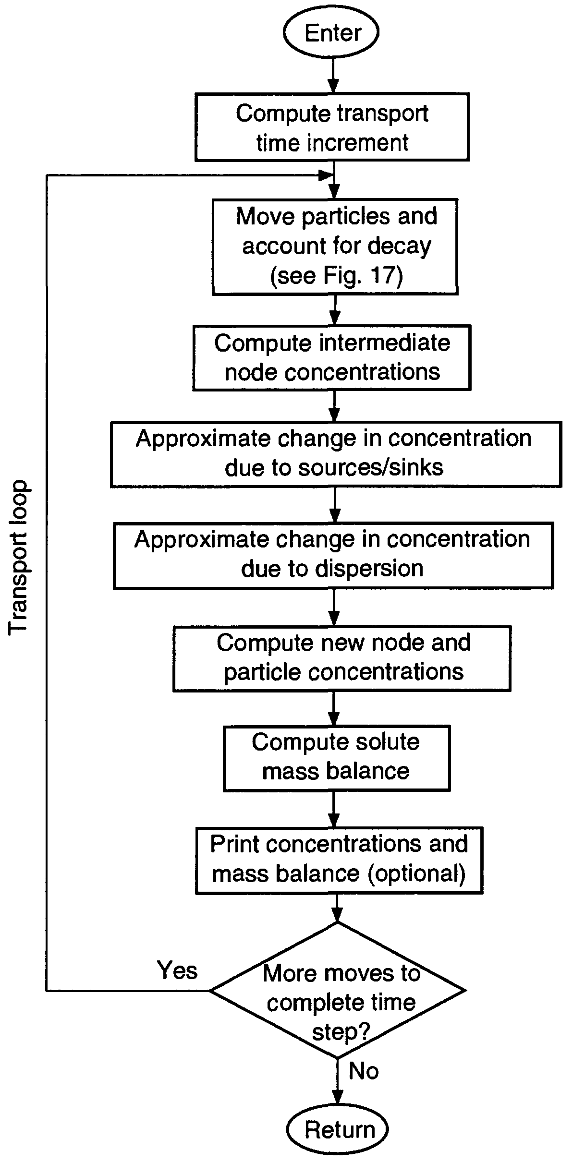

Enter
Compute transport time increment
Move particles and account for decay (see Fig. 17)
Compute intermediate node concentrations
Approximate change in concentration due to sources/sinks
Approximate change in concentration due to dispersion
Compute new node and particle concentrations
Compute solute mass balance
Print concentrations and mass balance (optional)
More moves to complete time step?
Yes
No
Return
Transport loop

Figure 13. Simplified flow chart for the transport loop, which is shown as a single element in fig. 12.

generates the initial coordinates for each tracer particle in the transport subgrid. Default particle placement patterns include several configurations for simulations in one, two, or three dimensions. Each of the configurations distributes the particles uniformly in space with respect to dimension. Figures 14 through 16 illustrate the distribution of particles with each of the default number of particles per node. Note that in some cases, it may be necessary to customize the initial positions of the particles (see the Two Dimensional Radial Flow and Dispersion test case in the Model Testing and Evaluation chapter). For a given transport time increment, particles are moved a distance and direction on the basis of the estimated velocity at the location of each particle and the length of the time increment. The particle velocity is estimated by interpolation from the velocities on adjacent cell faces to the location of a particular particle. Either linear or bilinear interpolation is used (in subroutines MOVE or MOVEBI, respectively) based on the user-selected value for the INTRPL flag (see *MOC3D* Input Instructions). However, when the bilinear interpolation option is used, particle velocity in the vertical (or "layer") direction will still be interpolated linearly. A flow chart describing details of the MOVE and MOVEBI subroutines is presented in figure 17.

Table 1. *MOC3D* subroutines controlling simulation preparation

Subroutine	Description
SMOC5O	Open *MOC* data files
MOC5DF	Define subgrid and other key parameters
MOC5AL	Allocate space in "X" array to store *MOC* data
MOC5RP	Read *MOC* parameters
MOC5CK	Check *MOC* data for consistency
SMOC5Z	Set an array to a specified constant value

Table 2. *MOC3D* subroutines controlling transport time factors

Subroutine	Description
MOC5ST	Check limiting stability criteria; compute time increment and number of moves for solute transport
MOC5AD	Update elapsed transport time

Table 3. *MOC3D* subroutines controlling velocity calculations and output

Subroutine	Description
VELO	Calculate velocities from flows across cell faces
SMOC5V	Output velocity data

Table 4. *MOC3D* subroutines controlling particle tracking, concentration calculations, and output for particle and concentration data

Subroutine	Description
SMOC5GP	Generate initial particle distribution
SMOC5P	Output particle locations and concentrations
MOVE and MOVEBI	Advect particles; compute concentrations at the end of each move; decay particle concentrations (MOVE uses linear interpolation of velocity and MOVEBI uses bilinear interpolation)
MOVTIM	Compute time for particle to reach boundary of cell
MOC5AP	Compute new node and particle concentrations at end of move
SMOC5C	Output node concentrations

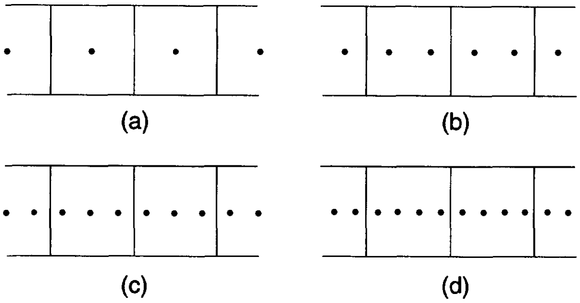


Figure 14. Default initial particle configurations for a one-dimensional simulation using (a) one, (b) two, (c) three, and (d) four particles per cell.

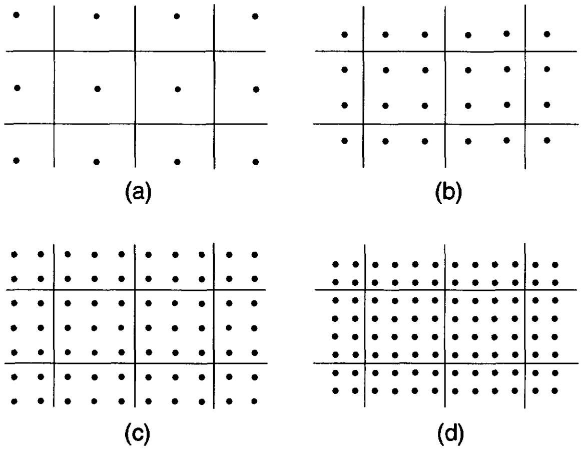


Figure 15. Default initial particle configurations for a two-dimensional simulation using (a) one, (b) four, (c) nine, and (d) sixteen particles per cell.

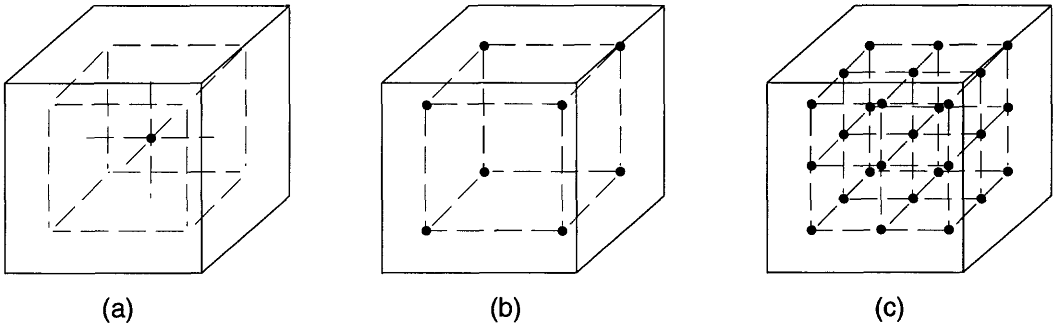


Figure 16. Default initial particle configurations for a three-dimensional simulation using (a) one, (b) eight, and (c) 27 particles per cell. For clarity, only the volume of a single cell is illustrated.

Subroutines related to dispersion calculations are included in Table 5. Dispersion coefficients are determined on cell faces. However, to facilitate improved efficiency in the code the dispersion coefficients are lumped with the porosity, thickness, and an appropriate grid dimension factor of the cell into combined parameters called "dispersion equation coefficients." For example, the dispersion equation coefficient for

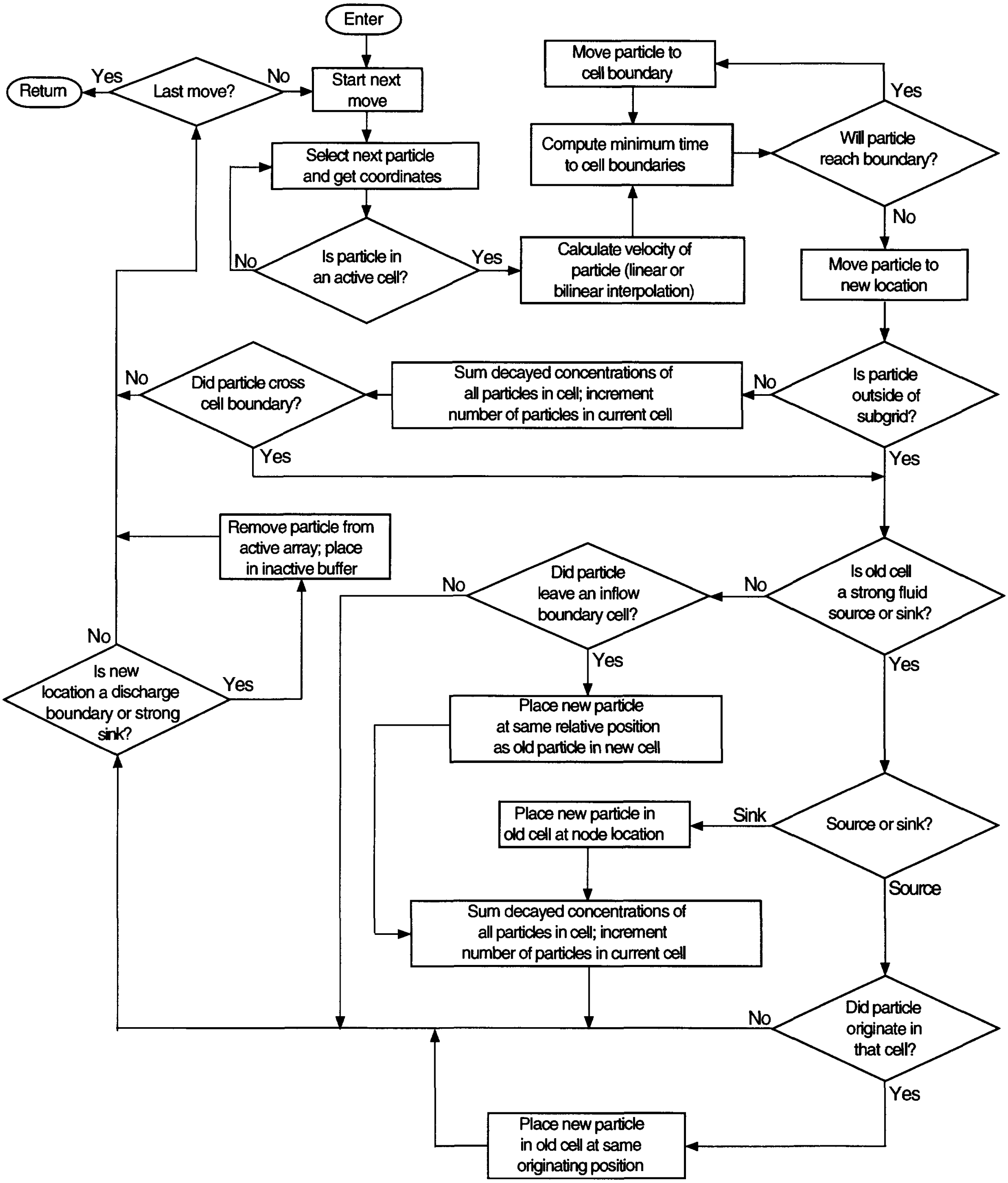

Enter
Start next move
Last move?
Yes
No
Return
Select next particle and get coordinates
Is particle in an active cell?
No
Yes
Calculate velocity of particle (linear or bilinear interpolation)
Compute minimum time to cell boundaries
Move particle to cell boundary
Will particle reach boundary?
Yes
No
Move particle to new location
Is particle outside of subgrid?
No
Yes
Sum decayed concentrations of all particles in cell; increment number of particles in current cell
Did particle cross cell boundary?
No
Yes
Remove particle from active array; place in inactive buffer
Is old cell a strong fluid source or sink?
No
Yes
Did particle leave an inflow boundary cell?
No
Yes
Place new particle at same relative position as old particle in new cell
Is new location a discharge boundary or strong sink?
No
Yes
Source or sink?
Sink
Source
Place new particle in old cell at node location
Sum decayed concentrations of all particles in cell; increment number of particles in current cell
Did particle originate in that cell?
No
Yes
Place new particle in old cell at same originating position

Figure 17. Flow chart for MOVE and MOVEBI subroutines of *MOC3D*, which is shown as a single element in fig. 13.

the $j+1/2,i,k$ face in the column direction is

$$\frac{\left(\varepsilon b D_{xx}\right)_{j+1/2,i,k}}{\Delta x}.$$

These combined coefficients are the ones that are written to the output files by SMOC5D. Appendix A includes a more detailed description of the dispersion equation coefficients.

Table 5. *MOC3D* subroutines controlling dispersion calculations and output

Subroutine	Description
DSP5FM	Calculate dispersion coefficients
SMOC5D	Output dispersion equation coefficients
DSP5AP	Use explicit finite-difference formulation to compute changes in concentration due to dispersion

Subroutines that link the *MODFLOW* source/sink package calculations of fluid flux to the *MOC3D* calculations of solute concentration and solute flux are listed in Table 6. *MODFLOW* source and sink packages contain an option called CBCALLOCATE. When used, the package will save the cell-by-cell flow terms across all faces of every source or sink cell. *MOC3D* uses these fluid fluxes to calculate solute flux to or from the source/sink nodes. Because these individual solute fluxes are required to compute the solute mass balance, the CBCALLOCATE option must always be selected when using *MOC3D*. Calculations of concentration changes at nodes caused by mixing with fluid sources are controlled by the "SRC" subroutines listed in Table 7.

Subroutines controlling observation well features are listed in Table 8. Table 9 lists subroutines related to the solute mass balance calculations. Table 10 lists subroutines related to calculating fluid storage terms in the solute-transport equation.

Table 6. *MOC3D* subroutines controlling *MODFLOW* source/sink package calculations

Subroutine	Description
CDRN5FM	Calculate solute flux to drains
CEVT5FM	Calculate solute flux to evapotranspiration
CGHB5FM	Calculate solute flux to/from general head boundary cells
CRCH5AL	Allocate space in "X" array for concentrations associated with recharge
CRCH5FM	Calculate solute flux from recharge
CRIV5FM	Calculate solute flux to/from river cells
CWEL5FM	Calculate solute flux to/from well cells

Table 7. *MOC3D* subroutines controlling cumulative calculations relating to sources and sinks

Subroutine	Description
SRC5FM	Initialize source/sink array; compute terms at fixed heads
SRC5AP	Calculate changes in concentration due to flux at sources and sinks

Table 8. *MOC3D* subroutines controlling observation wells

Subroutine	Description
OBS5DF	Read number of observation wells
OBS5AL	Allocate space in "X" array for observation well data
OBS5RP	Read observation well locations
SOBS5O	Output observation well data

Table 9. *MOC3D* subroutines controlling mass balance calculations and output

Subroutine	Description
SMOC5IM	Calculate initial solute mass stored
SMOC5BY	Calculate flows across boundaries of subgrid (for mass balance)
SMOC5BD	Compute cumulative solute mass balance
SMOC5M	Output mass balance information

Table 10. Miscellaneous *MOC3D* subroutines

Subroutine	Description
SMOC5TK	Set initial saturated thickness for water-table cells
SMOC5UP	Update fluid storage terms for transport equation

MODEL TESTING AND EVALUATION

In developing and documenting a new numerical model, it must be demonstrated that the generic model can accurately solve the governing equations for various boundary value problems. This is accomplished by demonstrating that the numerical code gives good results for problems having known solutions, such as those for which an analytical solution is available.

The accuracy of numerical solutions is sometimes sensitive to spatial and temporal discretization. Therefore, even a perfect agreement for selected test cases proves only that the numerical code *can* accurately solve the governing equations, not that it *will* under any and all circumstances.

Analytical solutions generally require that an aquifer can be assumed to have simple geometry, uniform properties, and idealized boundary and initial conditions. A major advantage of numerical methods is that they relax the simplifications required by analytical methods and allow the representation of more realistic field conditions, such as heterogeneous and anisotropic properties, irregular geometry, mixed boundary conditions, and multiple stresses that vary in time and space. However, analytical solutions approximating these complexities are unavailable for comparison. Therefore, it is difficult to prove that the numerical models can accurately solve the governing equations for the very situations for which they are most needed. For such cases, we are limited to relatively simple tests, such as benchmarking and evaluating the global mass-balance error. In the benchmarking approach, we compare the results of the *MOC3D* model for selected complex problems to results of other well accepted models. Although benchmarking is useful to improve confidence in the model, it is largely a measure of consistency and does not guarantee or measure accuracy. Overall, we have attempted to test and evaluate the *MOC3D* model for a range of conditions and problem types so that the user will gain an appreciation for both the strengths and weaknesses of this particular code. Additional testing and benchmarking of *MOC3D* is documented in Goode and Konikow (1991).

One-Dimensional Steady Flow

Wexler (1992) presents an analytical solution for one-dimensional solute transport in a finite-length aquifer system having a third-type source boundary condition. The governing equation is subject to the following boundary conditions:

$$VC' = VC - D\frac{\partial C}{\partial x}, \qquad x = 0 \qquad (69)$$

$$\frac{\partial C}{\partial x} = 0, \qquad x = L \quad (70)$$

and the following initial condition:

$$C = 0, \qquad 0<x<L. \quad (71)$$

For this test problem we assumed that the length of the system, L, is equal to 12 cm, $C' = 1.0$, and V = 0.10 cm/s. The analytical solution is given by equations 52 and 53 of Wexler (1992, p. 17). In generating an equivalent solution using *MOC3D*, we set up a one-dimensional grid having 122 cells (nodes) in the x-direction within which the flow equation was solved. The solute-transport equation was solved in a 120-cell subgrid to assure a constant velocity within the transport domain and to allow an accurate match to the boundary conditions of the analytical solution. The grid spacing was Δx = 0.1 cm. The numerical solution was implemented using three initial particles per cell (NPTPND = 3) and a CELDIS factor of 0.5. The input parameters for the model simulation are summarized in Table 11.

Two different values of dispersion coefficients were evaluated in the first set of tests. The values were D_{xx} = 0.1 and 0.01 cm^2/s, which are equivalent to α_L = 1.0 and 0.1 cm, respectively. Breakthrough curves showing concentration changes over time at three different locations as calculated with both the analytical and numerical solutions for the lower dispersion case are compared in figure 18. To improve clarity, this plot only shows every fourth data point for the numerical model results, except for the curve for x = 0.05 cm, where every data point is shown for times less than 10 seconds. Note that this distance (x = 0.05) is the first node downgradient from the source location. With the possible exception of very early time at locations very close to the source, there is essentially an exact fit between the numerical and analytical solutions. At early times and short distances the numerical solution exhibits some nonsmoothness and oscillation about the mean, which is related to the discrete nature of the particles used to represent the advection process. However, this small loss of precision is not a cumulative error, as it vanishes after moderate travel times or distances.

Table 11. Parameters used in *MOC3D* simulation of transport in a one-dimensional, steady-state flow system

Parameter	Value
$T_{xx} = T_{yy}$	0.01 cm^2/s
ε	0.1
α_L	0.1 cm
$\alpha_{T_H} = \alpha_{T_V}$	0.1 cm
PERLEN (length of stress period)	120 s
V_x	0.1 cm/s
$V_y = V_z$	0.0 cm/s
Initial concentration (C_0)	0.0
Source concentration (C')	1.0
Number of rows	122
Number of columns	1
Number of layers	1
DELR (Δx)	0.1 cm
DELC (Δy)	0.1 cm
Thickness (b)	1.0 cm
NPTPND (Initial number of particles per cell)	3
CELDIS	0.5
INTRPL (Interpolation scheme)	2

The results for the higher dispersion case are presented in figure 19. Because dispersion is a limiting stability criterion and the dispersion coefficient is ten times higher in the same grid, the transport simulation takes many more time increments (or particle moves). Thus, in figure 19 only every 100th point is shown (as small circles) for the numerical solution at the two larger distances. For these two curves, the match between the analytical

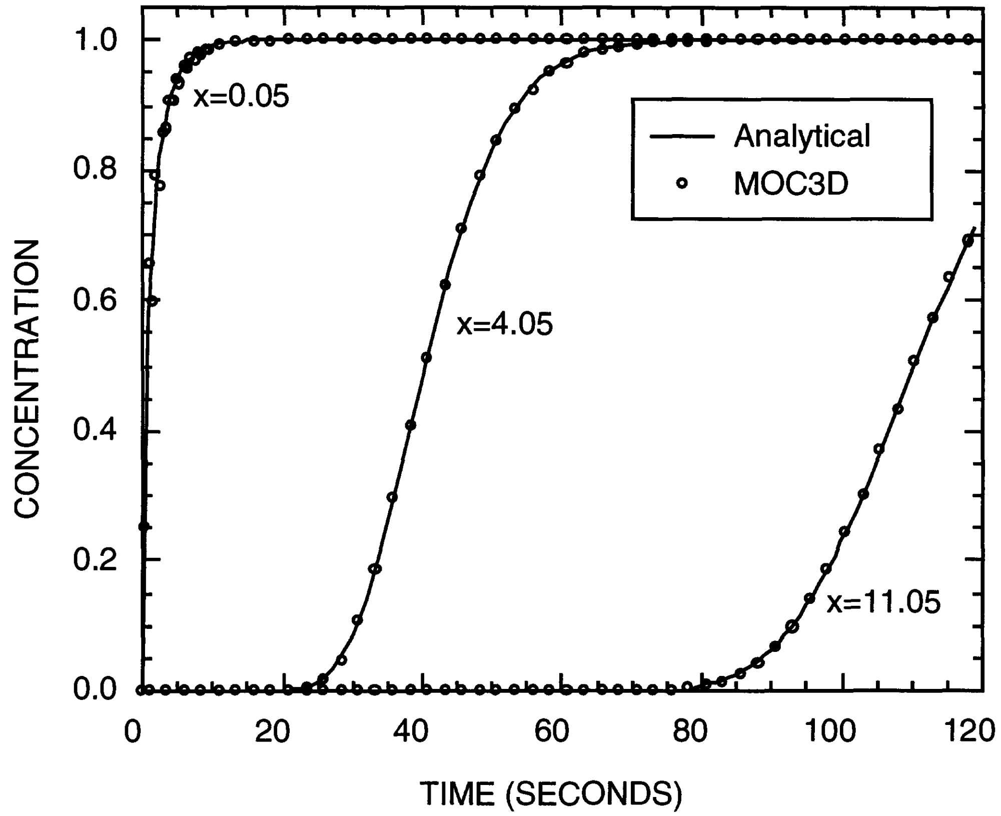


Figure 18. Numerical (*MOC3D*) and analytical solutions at three different locations for solute transport in a one-dimensional, steady flow field. Parameter values for this base case are listed in Table 11.

solution and the numerical solution is almost perfect. For the location that is close to the source, every point is plotted as a small dot to illustrate again the small loss of precision for short travel distances and times.

To help assess the significance of the oscillations and loss of precision at nodes very close to the source, the early (times less than 10 seconds) part of that breakthrough curve is replotted at a larger scale in figure 20. The oscillation is caused by the fact that the stability requirements related to the explicit solution for the dispersive flux causes the time increment for solving the transport equation to be so small that particles used to track the advective flux can only move a small fraction of the width of a cell during a given time increment. Because the distance that the particles move during one time increment (about 0.005 cm in this case) is smaller than the spacing between particles (one-third of Δx for this case in which three particles per cell are used, or 0.033 cm), particles only cross cell boundaries after every seven moves in this case. Therefore, the change in concentration caused by advection is underestimated during six moves when no particles cross a cell boundary and overestimated during the seventh move when one particle does cross the cell boundary. However, there is essentially no cumulative error and the numerical solution oscillates regularly in a small and decreasing range about the true solution. Also, the magnitude of the oscillations diminishes over time as dispersion reduces the local concentration gradients. To check this explanation and to demonstrate that

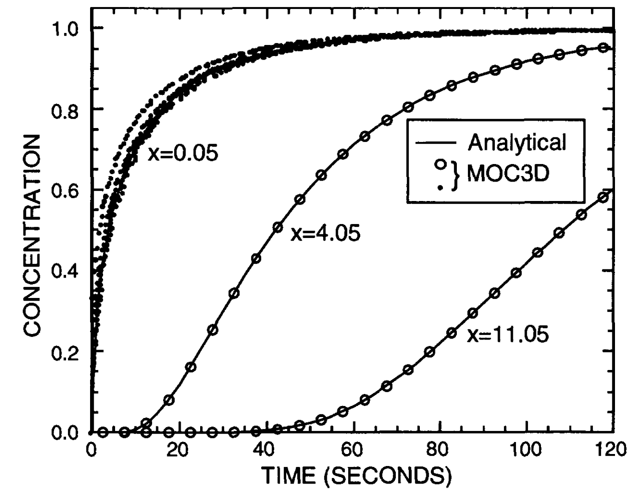


Figure 19. Numerical (*MOC3D*) and analytical solutions at three different locations for solute transport in a one-dimensional, steady flow field for case of increased dispersivity (α_L = 1.0 cm, D_{xx} = 0.1 cm^2/s, and other parameters as defined in Table 11).

the error shown in figure 20 is primarily an artifact of having too few initial particles per cell for this particular combination of parameters, and that it is not a generic deficiency in the algorithm, *MOC3D* was run for the same problem using an initial particle density of 50 particles per cell. The results for the same location for the first 10 seconds are shown in figure 21 for comparison. When 50 particles are used, the distance that each particle moves during one time increment (again about 0.005 cm) is greater than the spacing between adjacent particles (0.002 cm). For this case, the agreement between the analytical solution and the *MOC3D* results are much closer than in figure 20 and the oscillations are almost entirely eliminated.

The results of these tests can also be presented in the form of breakthrough curves that plot concentration against distance for various times. Figure 22 shows the results for the same set of parameters as shown in figure 18 (that is, the low dispersion case). For clarity in figure 22, only every fourth data point is plotted for the numerical results, except every data point is shown for distances less than 1.5 cm on the curve for t = 6 seconds. The results show an almost perfect agreement between the analytical and numerical solutions.

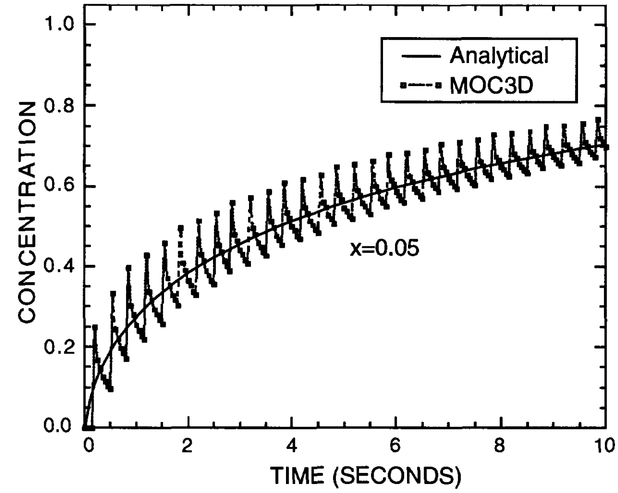


Figure 20. Detailed view of numerical and analytical solutions for early times (t < 10 s) at the first node downgradient from the inflow source boundary for same problem as shown in fig. 19.

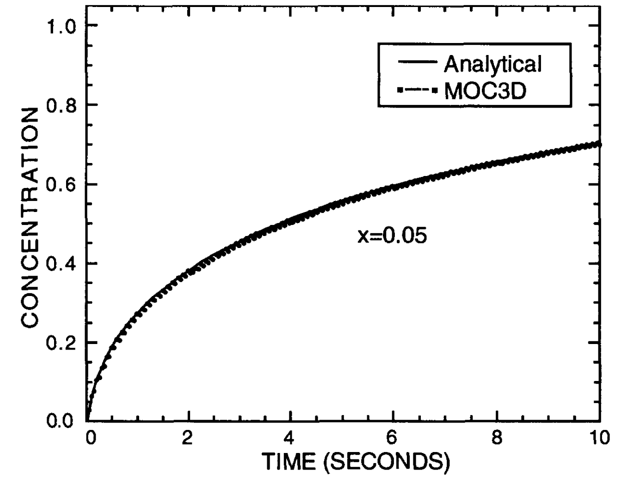


Figure 21. Detailed view of numerical and analytical solutions for early times (t < 10 s) at the first node downgradient from the inflow source boundary for same conditions as shown in fig. 20, except that the initial number of particles per node, NPTPND, equals 50.

The effect of incorporating a retardation factor to represent a linear, equilibrium, reversible, sorption process is illustrated in figure 23. This shows a comparison between the analytical solution and the *MOC3D* results for the same low-dispersion problem represented in figures 18 and 22, except that the elapsed time is 240 seconds and the three different curves are for cases in which $R_f = 2$, 4, and 40. Only every fourth data point is plotted in figure 23 for the numerical results, except for the case for $R_f = 40$, where every data point is shown for distances less than 1.5 cm. The agreement is excellent. Note that because the net effect of the retardation factor is to transform the time scale, the three sets of curves and data points shown in figure 23 for $t = 240$ s are identical to the three sets of curves and data points shown in figure 22 for shorter times and $R_f = 1.0$.

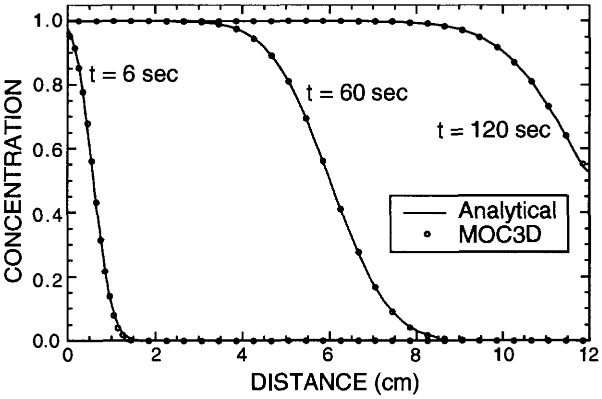


Figure 22. Numerical (*MOC3D*) and analytical solutions at three different times for same one-dimensional, steady flow, solute-transport problem shown in fig. 18.

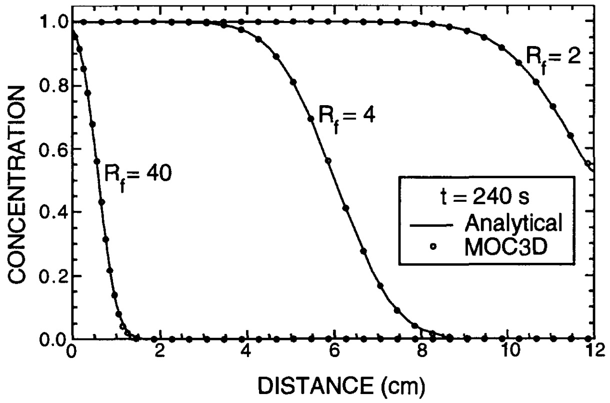


Figure 23. Numerical (*MOC3D*) and analytical solutions after 240 seconds for three different retardation factors for same problem represented in fig. 22.

The effect of decay is evaluated by specifying the decay rate as $\lambda = 0.01$ s^{-1} for the same low-dispersion, no sorption, problem as defined for figures 18 and 22. These results are presented in figure 24, which shows excellent agreement between the analytical and numerical solutions. Only every fourth data point is plotted in figure 24 for the numerical results.

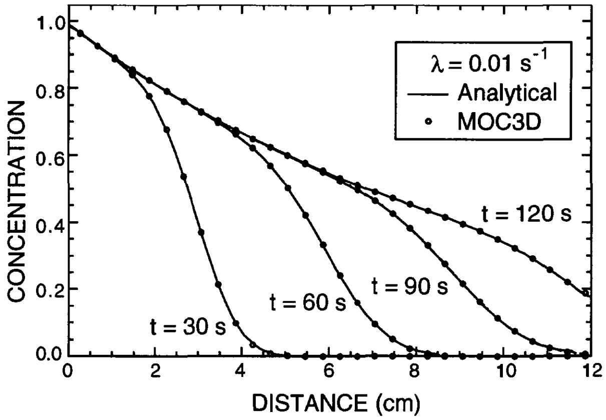


Figure 24. Numerical (*MOC3D*) and analytical solutions for four different times for solute transport in a one-dimensional, steady flow field for case with decay at rate of $\lambda = 0.01$ s^{-1}. All other parameters as defined in Table 11.

Three-Dimensional Steady Flow

To further evaluate and test *MOC3D* for three dimensional cases, we compare the numerical results with those of the analytical solution developed by Wexler (1992) for the case of three-dimensional solute transport from a continuous point source in a steady, uniform flow field in a homogeneous aquifer of infinite extent. Relative to the previous tests, and because the flow field is aligned with the grid in this test case also, this evaluation primarily is a test of the accuracy of the calculated dispersive flux in three directions. The analytical solution of Wexler (1992, p. 42-43) is subject to the following boundary conditions:

$$C = 0, \qquad x = \pm\infty \qquad (72a)$$

$$C = 0, \qquad y = \pm\infty \qquad (72b)$$

$$C = 0, \qquad z = \pm\infty \qquad (72c)$$

and the following initial condition (at $t = 0$):

$$C = 0,$$
$$-\infty < x < \infty, \ -\infty < y < \infty, \ -\infty < z < \infty. \quad (73)$$

For the analytical solution to this test problem, we assumed that $V_x = 0.1$ m/d, $V_y = V_z = 0.0$ m/d, $D_x = 0.06$ m^2/d, $D_y = 0.003$ m^2/d, $D_z = 0.0006$ m^2/d, $\varepsilon = 0.25$, and that the source strength (or solute mass flux) is $QC' = 10.0$ g/d. The analytical solution is given by equation 105 of Wexler (1992, p. 47), and assumes that the fluid source does not affect the flow field.

Whereas the analytical solution assumes an infinite aquifer, the numerical solution can only be applied to a finite system. In generating an equivalent solution using *MOC3D*, we aimed to use a grid that was sufficiently large so as to minimize any effects of the boundaries on the solution. Because of the symmetry of the problem, we only simulated one quadrant of the cross-sectional area of the aquifer downgradient from the point source. The three-dimensional transport subgrid had 30 rows of cells (nodes) at a grid spacing of 3 m that are parallel to the x-direction of the analytical solution, 12 columns at a grid spacing of 0.5 m in the y-direction, and 40 layers at a grid spacing of 0.05 m in the z-direction, within which the transport equation was solved. Boundary conditions, values of heads on boundaries, hydraulic conductivity, and porosity were specified to assure that the velocity would equal 0.1 m/d in the x-direction. Identical values for the dispersion coefficients were generated by specifying $\alpha_L = 0.6$ m, $\alpha_{T_H} = 0.03$ m, and $\alpha_{T_V} = 0.006$ m. The point source was represented in the numerical model by a combination of $Q = 1.0 \times 10^{-6}$ m^3/d and $C' = 2.5 \times 10^6$ g/m^3 (note that g/m^3 is equivalent to mg/L), which together yield one-fourth of the source flux assumed in the analytical solution. The very small fluid injection rate at the point source assures that the fluid has only a negligible effect on the flow field, as required for consistency with the analytical solution. However, another small but unavoidable difference between the analytical and numerical solutions is that the former has the solute source at a true point but the numerical model inherently assumes the solute source is within the volume of one cell. Both models were run for a total elapsed time of 400 days. The numerical solution was implemented using three initial particles per cell (NPTPND = 3) and a CELDIS factor of 0.1. The input parameters for the model simulation are summarized in Table 12.

The results of the analytical solution are compared graphically with those of *MOC3D* for three different planes in figs. 25-27. Figure 25a shows the concentrations in the x-y plane of the point source as calculated using the analytical solution and figure 25b shows the same for the *MOC3D* results. In this view, the left edge of each map is a line of symmetry, so that the map represents only half of the true problem domain. Both sets of contours were generated for an identical number of points and locations to eliminate any differences attributable solely to the contouring procedure. The results agree very closely, although a slightly greater distance of migration or spreading is evident in the *MOC3D* results, both upstream as well as downstream of the source. However, a large part of this small difference can be explained simply by the fact that the source is applied over a larger area in the horizontal plane of the *MOC3D* model, in which the length of the source cell is 3 m in the direction parallel to flow.

Figure 26 shows a comparison of the results in a vertical plane parallel to the flow direction and aligned with column 2 of the numerical grid (and $y = 0.75$ m in the analytical solution). In this view, the top edge is a line of symmetry, so that the cross section represents only the lower half of the true problem domain. Overall the agreement is very close, although the numerical results show slightly more upstream dispersion, particularly for the lower concentrations (less than 10). Figure 27 shows a comparison of the results in a vertical plane transverse to the flow direction and aligned with row 12 of the numerical grid (and $x = 34.5$ m in the analytical solution). In this view, both the left and top edges are lines of symmetry, so that this cross section represents only one quarter of the true problem domain. Once again the overall agreement is excellent. The only noticeable difference is a minor one. In the numerical results there is slightly more lateral spreading, as indicated by a displacement of the contours for the lower values of concentration by less than half a cell distance away from the axis of symmetry, but only in the upper half of the grid.

Table 12. Parameters used in *MOC3D* simulation of transport from a continuous point source in a three-dimensional steady-state flow system

Parameter	Value
$T_{xx} = T_{yy}$	0.0125 m^2/day
ε	0.25
α_L	0.6 m
α_{TH}	0.03 m
α_{TV}	0.006 m
PERLEN (length of stress period)	400 days
V_x	0.1 m/day
V_y	0.0 m/day
V_z	0.0 m/day
Source concentration (C')	2.5×10^6 g/m^3
Q (at well)	1.0×10^{-6} m^3/d
Source location	row 8, column 1, layer 1
Number of rows	30
Number of columns	12
Number of layers	40
DELR (Δx)	3 m
DELC (Δy)	0.5 m
Thickness (b)	0.05 m
NPTPND (Initial number of particles per cell)	3
CELDIS	0.1
INTRPL (Interpolation scheme)	1

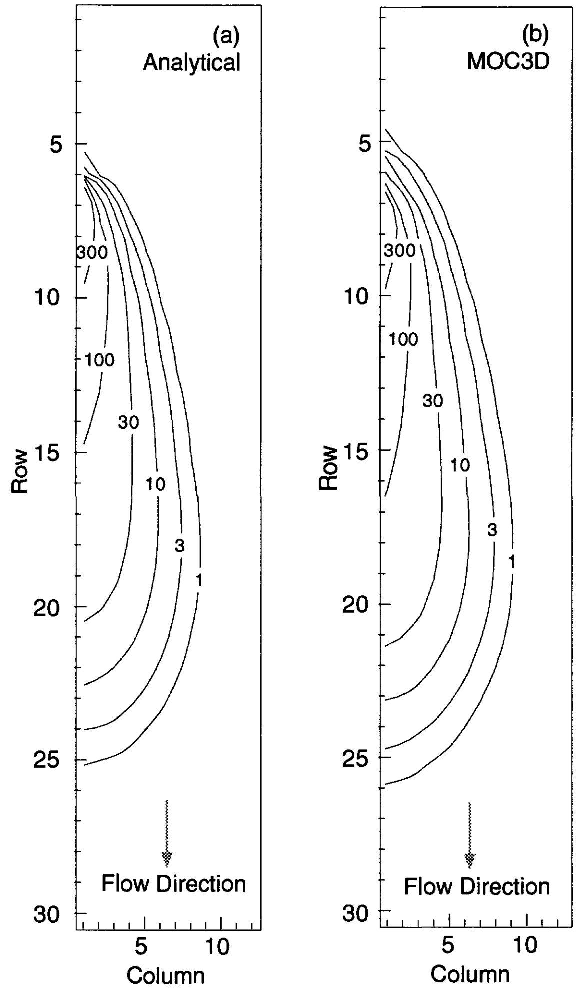


Figure 25. Concentration contours for (a) analytical and (b) numerical solutions in the horizontal plane containing the solute source (layer 1) for three-dimensional solute transport in a uniform steady flow field. Parameters are defined in Table 12.

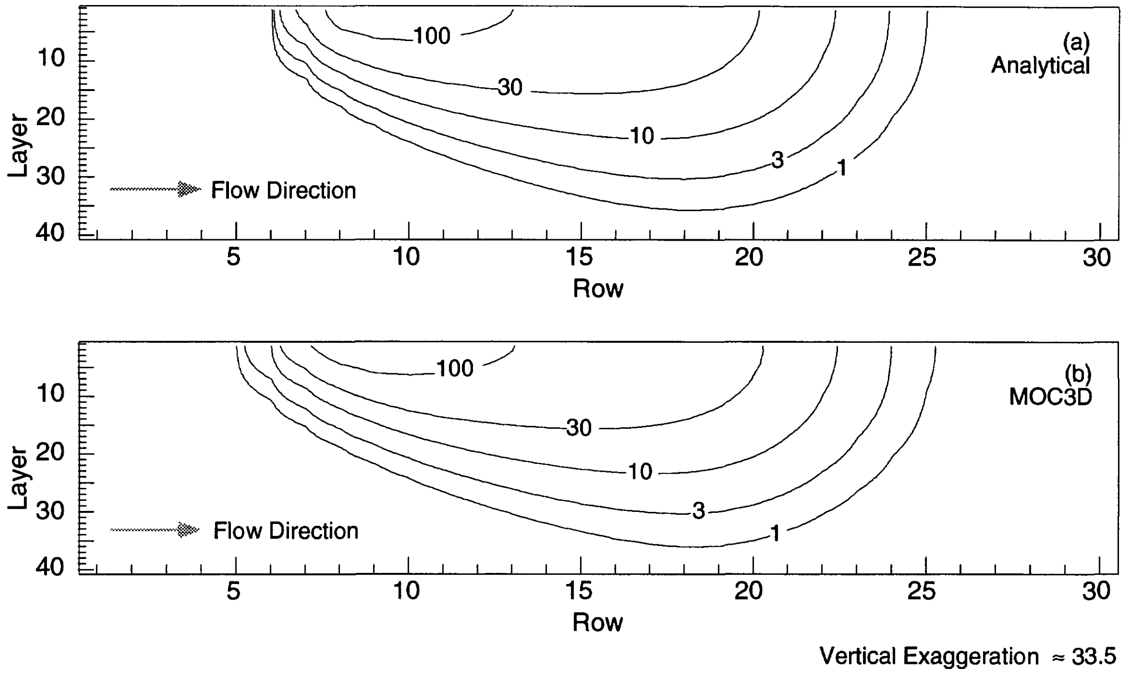


Figure 26. Concentration contours for (a) analytical and (b) numerical solutions in the vertical plane parallel to the flow direction and aligned with column 2 for three-dimensional solute transport in a uniform steady flow field. Parameters are defined in Table 12.

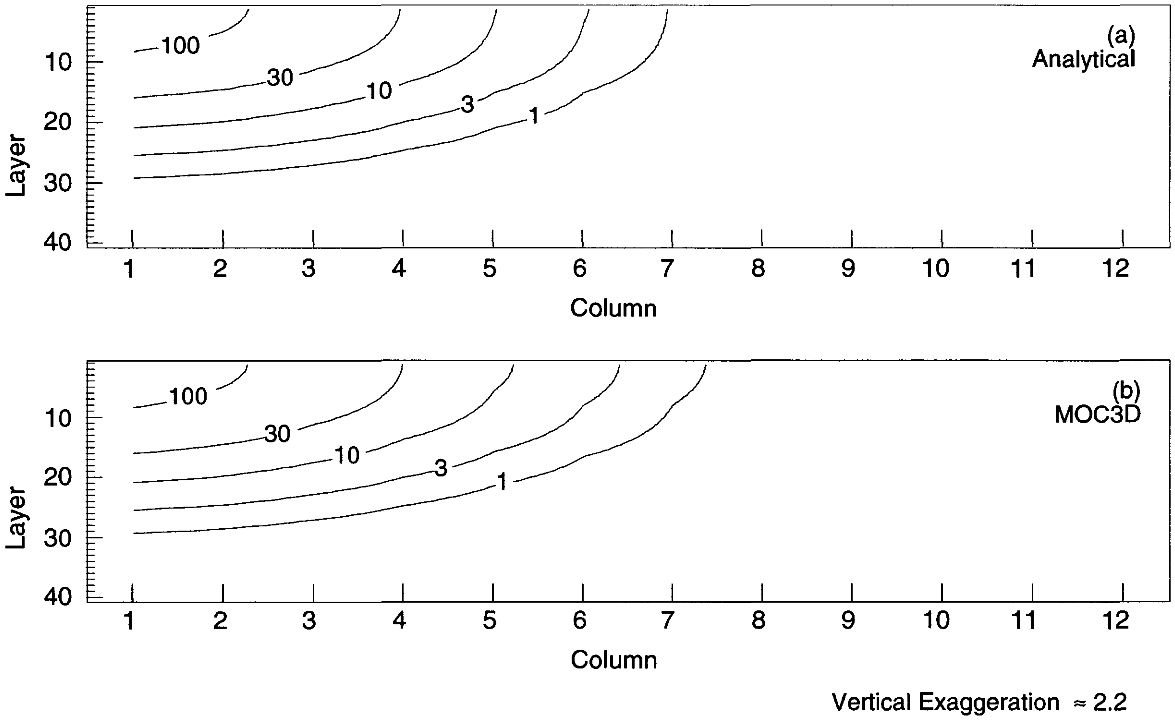


Figure 27. Concentration contours for (a) analytical and (b) numerical solutions in the vertical plane transverse to the flow direction and aligned with row 12 for three-dimensional solute transport in a uniform steady flow field. Parameters are defined in Table 12. Flow is towards the reader.

Two-Dimensional Radial Flow and Dispersion

A radial dispersion problem was used to compare the *MOC3D* solution to the analytical solution given by Hsieh (1986) for a finite-radius injection well in an infinite aquifer of two dimensions. The problem is equivalent to flow from a single injection well; the velocities vary in space and are inversely related to the distance from the injection well. The governing equation for the analytical solution is

$$\frac{\partial C}{\partial t}+\frac{A}{r}\frac{\partial C}{\partial r}=\alpha\frac{A}{r}\frac{\partial^2 C}{\partial r^2} \qquad r>r_w \quad t>0 \qquad (74)$$

where $A=Q/2\pi b\varepsilon$, r is the radial distance from the center of the well, r_w is the radius of the injection well, α is the longitudinal dispersivity in radial flow, Q is the volumetric rate of the well injection, b is the thickness of the aquifer, and ε is porosity. The initial and boundary conditions are

$$C(r,0)=0 \qquad r>r_w \qquad (75a)$$

$$C(r_w,t)=C' \qquad t>0 \qquad (75b)$$

$$C(r\to\infty,t)\to 0 \qquad t>0. \qquad (75c)$$

The radius of the well (r_w) was set to 1.0 (dimensionless) and the concentration of the injected tracer (C') was 1.0 (dimensionless) at the well.

The problem was modeled using a grid having 30 cells in the x-direction and 30 cells in the y-direction, representing one quadrant of the radial flow field (90 of 360 degrees). The initial concentration was set to 1.0 at the well node (1,1), defined by a specified flux of 56.25 m^3/h. The input parameters for the model simulation are summarized in Table 13. Initial particle positions were defined using the custom particle placement option in the input data set and were aligned in a quarter circle in two-degree increments, equidistant from the center point of radial symmetry in the upper left corner of the grid (see fig. 28). Using a large number of particles per cell in this configuration ensures that all cells in the grid will include at least one particle pathline emanating from the source cell. If one of the *MOC3D* default options was selected to place fewer particles in a regular geometric pattern where strong flow divergence exists, then some cells located just a few rows or columns away from the source cell would never receive particles originating in the source cell. This would make it impossible to calculate accurately concentration changes caused by advection. This custom configuration also parallels the expected solute distribution pattern in a radially divergent flow field; standard particle positioning (linear, quadratic, or cubic configurations) results in extreme spreading of

Table 13. Parameters used in *MOC3D* simulation of two-dimensional, steady-state, radial flow case

Parameter	Value
$T_{xx}=T_{yy}$	3.6 m^2/hour
ε	0.2
α_L	10.0 m
α_{TH}	10.0 m
α_{TV}	10.0 m
PERLEN (length of stress period)	1000 hours
Q (at well)	56.25 m^3/hour
Source concentration (C')	1.0
Number of rows	30
Number of columns	30
Number of layers	1
DELR (Δx)	10.0 m
DELC (Δy)	10.0 m
Thickness (b)	10.0 m
NPTPND (Initial number of particles per cell)	46
CELDIS	0.5
INTRPL (Interpolation scheme)	2

particles at small distances from the source, distorting contours of concentration in that area. The same pattern of particles shown for the source cell in figure 28 is repeated in all cells of the grid at the start of the simulation.

Figure 29 shows a comparison of the concentrations calculated in one quadrant after 1000 hours using both the analytical (a) and numerical (b) solutions. The two solutions are almost identical.

Point Initial Condition in Uniform Flow

A test problem for three-dimensional solute transport from an instantaneous point source, or Dirac initial condition, in a uniform flow field was used to test the *MOC3D* model. An analytical solution for an instantaneous point source in a homogeneous infinite aquifer is given by Wexler (1992), and he presents a code (named *POINT3*) for a related case for a continuous point source. The *POINT3* code was modified to solve for the desired case of an instantaneous point source. Test problems

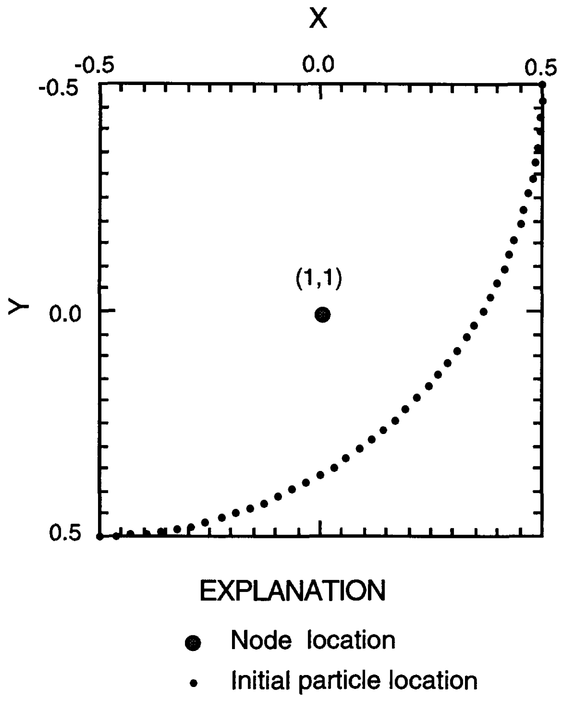


Figure 28. Initial particle positions within the source cell for radial flow case (based on custom particle placement and *NPTPND* = 46). The relative coordinates on the *x*- and *y*-axes shown for the cell (1,1) are the same for any cell of the grid; this relative coordinate system is used for the custom definition of particle locations in the input file.

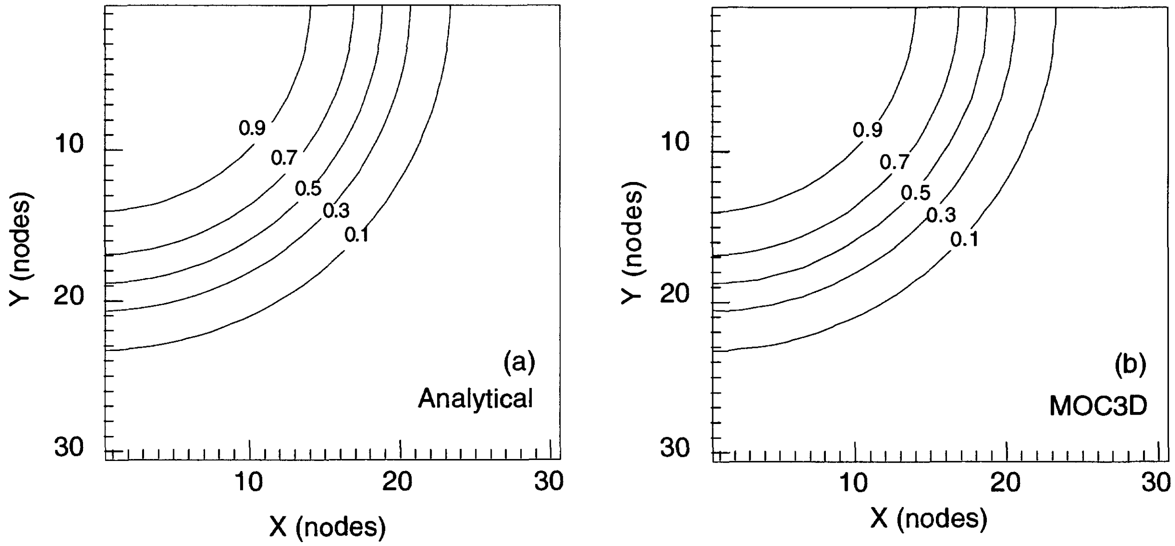


Figure 29. Contours of relative concentrations calculated using (a) analytical and (b) numerical models for solute transport in a steady radial flow field. Source concentration is 1.0 and source is located in cell (1,1). Grid spacing is 10.0 m.

were designed to evaluate the numerical solution for two cases—one in which flow is parallel to the grid (in the x-direction) and one in which flow occurs at 45 degrees to the x- and y-axes. This allows us to evaluate the accuracy of the numerical model for this basic type of problem, and also to evaluate the sensitivity of the numerical solution to the orientation of the flow relative to that of the grid.

The governing equation and boundary conditions for an instantaneous point source are (see Wexler, 1992, p. 42):

$$\frac{\partial C}{\partial t} = D_x \frac{\partial^2 C}{\partial x^2} + D_y \frac{\partial^2 C}{\partial y^2} + D_z \frac{\partial^2 C}{\partial z^2} - V \frac{\partial C}{\partial x} - \lambda C$$

$$+ \frac{Qdt}{\varepsilon} C' \bullet \delta(x - X_c)\delta(y - Y_c)\delta(z - Z_c)\delta(t - t') \qquad (76)$$

$$C = 0, \qquad x = \pm\infty \qquad (77a)$$

$$C = 0, \qquad y = \pm\infty \qquad (77b)$$

$$C = 0, \qquad z = \pm\infty \qquad (77c)$$

where V is the velocity in the direction of flow (assumed to be the x-direction in eq. 76), λ is the decay rate ($\lambda = 0$ for this problem), Q is the injection rate for the well, δ is a dirac delta function, X_C, Y_C, and Z_C are coordinates of the point source, and t' is the time at which the instantaneous point source activates.

The initial condition (at $t' = 0$) is:

$$C = 0,$$

$$-\infty < x < \infty, \; -\infty < y < +\infty, \; -\infty < z < +\infty. \qquad (78)$$

For the test case of flow in the x-direction, we assumed V_X = 1.0 m/d, and $V_Y = V_Z$ = 0.0 m/d. For flow at 45 degrees to x and y, we assumed $V_X = V_Y$ = 1.0 m/d, and V_Z = 0.0 m/d. For both cases, the distance the plume travels in the x-direction is the same for equal simulation times. Note, however, that the magnitude of velocity is higher in the latter case; therefore, there will be more dispersion in that problem during an equivalent time interval.

Several different grid spacings were used for the model simulations to help show the relation between discretization and the accuracy of the numerical results. The coarsest transport subgrid used 24 rows, 24 columns, and 24 layers with a grid spacing of 10.0 m in each direction. Subsequent runs doubled and tripled the number of rows and columns; the grid spacing was reduced accordingly so that the model domain (the aquifer volume being simulated) remained the same for all grids, and changes in accuracy would be attributable only to changes in the spatial discretization. Because advective transport occurs only in the x-y plane, the number of layers was held constant at 24. The input parameters for the simulations are presented in Table 14, which includes the several different values used for grid dimensions and spacing.

The results for both the analytical and numerical solutions for the case in which flow occurs only in the x-direction are shown in figure 30. The *MOC3D* results for the coarsest grid (fig. 30b) clearly show too much spreading transverse to flow (that is, in the y-direction) relative to the analytical solution (fig. 30a). Note that the analytical solution was contoured for values only at the same exact locations as the nodes in the grid used for the numerical solution to which it is being compared (fig. 30b); this eliminates differences for a visual comparison due only to artifacts of the contouring procedure, and is the reason that the analytical solution appears less smooth than it should. To improve numerical and contouring accuracy, a finer grid was used for both the analytical and *MOC3D* solutions (figs. 30c and 30d). Using three times as many nodes in each direction in the horizontal plane, the numerical dispersion is reduced significantly and the numerical solution very closely matches the analytical solution. (Note that the analytical solution mapped in fig. 30c is based on values calculated at nine times as many points as that in fig. 30a.)

The results of the test problem for flow at 45 degrees to the grid are shown in figure 31. The analytical solution (fig. 31a), which provides the basis for the evaluation, was solved on a 72×72 grid, and the *MOC3D* solutions are shown for a 24×24×24 grid (fig. 31b), a 48×48×24 grid (fig. 31c), and a 72×72×24 grid (fig. 31d). Unlike the previous case (where flow is aligned with the grid), the numerical results in figure 31 show a noticeable difference in the shape of the plume relative to the analytical solution. The numerically calculated "hourglass" shape is characteristic of a grid-orientation effect and is related to the cross-product terms of the dispersion tensor. When flow is oriented parallel to the grid, or when longitudinal and transverse dispersivities are equal, the cross-product terms of the dispersion equation are zero. Because flow is at 45 degrees to the grid, the cross-product terms of the dispersion equation are nonzero. The model estimates the concentration gradients associated with the cross-product terms less accurately than those associated with the diagonal terms, and therefore the overall solution is less accurate.

The magnitude of this effect is minimized by using a finer grid. Overall, the coarsest grid exhibits too much spreading, but the next finer grid results in minimal numerical dispersion (although the grid-orientation effect is not eliminated). Further reducing the grid spacing (fig. 31d) does not significantly further reduce numerical dispersion, thereby indicating the desired characteristic of grid convergence.

Table 14. Parameters used in *MOC3D* simulation of three-dimensional transport from a point source with flow in the *x*-direction and flow at 45 degrees to *x* and *y*

Parameter	Value
$T_{xx} = T_{yy}$	10.0 m^2/day
ε	0.1
α_L	1.0 m
α_{T_H}	0.1 m
α_{T_V}	0.1 m
PERLEN (length of stress period)	90 days
V_x	1.0 m/day
V_y	0.0 m/day*
V_z	0.0 m/day
Source concentration (C')	1×10^6
Source location	x = 30 m, y = 120 m, z = 40 m**
Number of rows	24, 48, and 72
Number of columns	24, 48, and 72
Number of layers	24
DELR (Δx)	10.0 m, 5.0 m, and 3.33 m
DELC (Δy)	10.0 m, 5.0 m, and 3.33 m
Thickness (b)	10.0 m
NPTPND (Initial number of particles per cell)	8
CELDIS	0.5
INTRPL (Interpolation scheme)	2

* For flow at 45 degrees to x and y, V_y = 1.0 m/day

** For flow at 45 degrees to x and y, the source location is x = 30 m, y = 30 m, z = 120 m.

Each of the *MOC3D* results also shows a slight asymmetry in the shape of the plume in the direction of flow (that is, there is slightly less forward spreading compared to backward spreading), which is inconsistent with symmetrical spreading indicated by the analytical solution. This is caused by the sequence in which the dispersive and advective terms of the transport equation are solved.

The numerical errors are exaggerated in figures 30 and 31 because the concentrations are contoured on a logarithmic scale. Some of

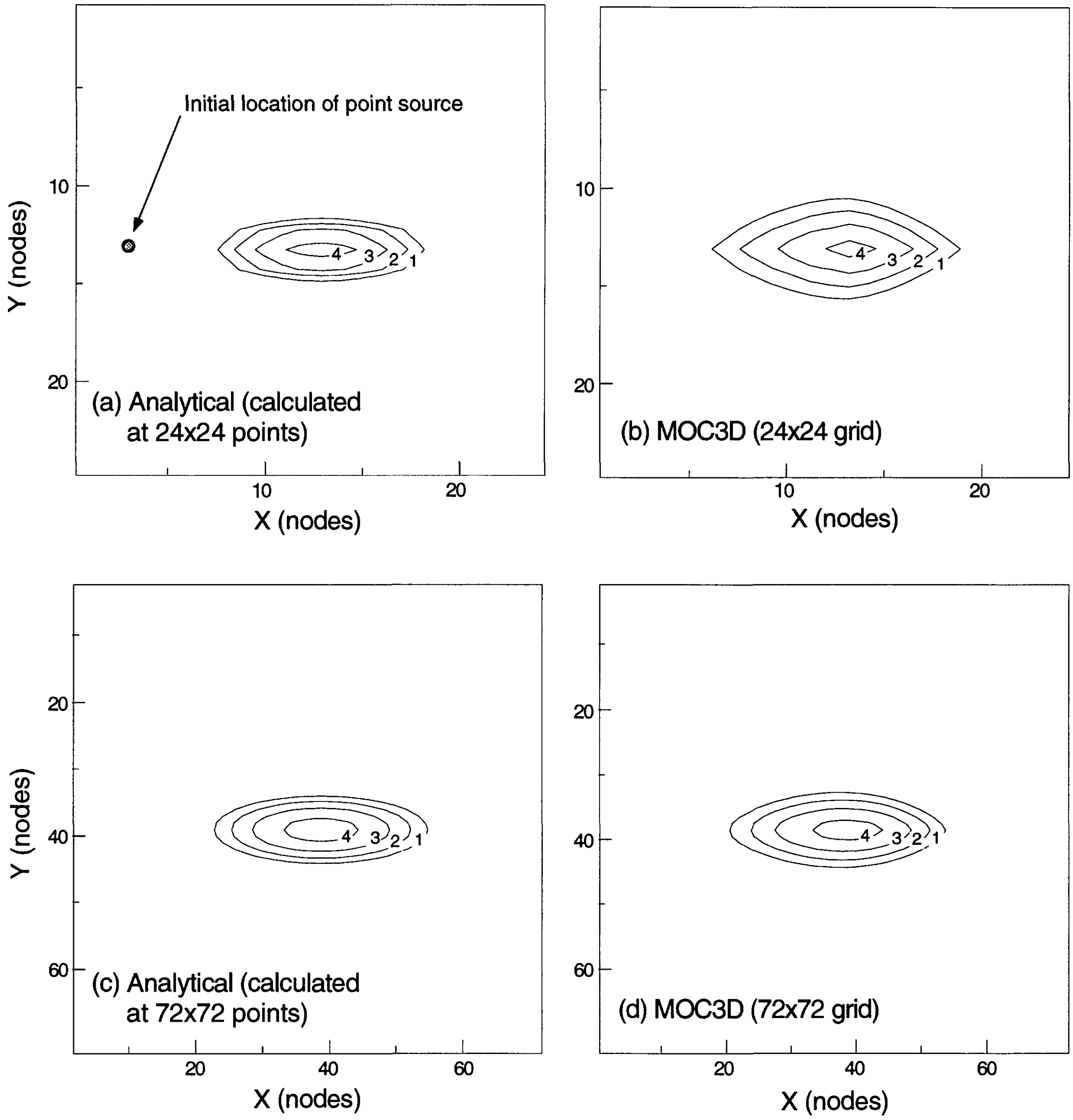


Figure 30. Concentration contours for (a, c) analytical and (b, d) numerical solutions for transport of a point initial condition in uniform flow in the *x*-direction. The *z*-component of flow is zero, but there is dispersion in all three directions. Contour values are the log of the concentrations.

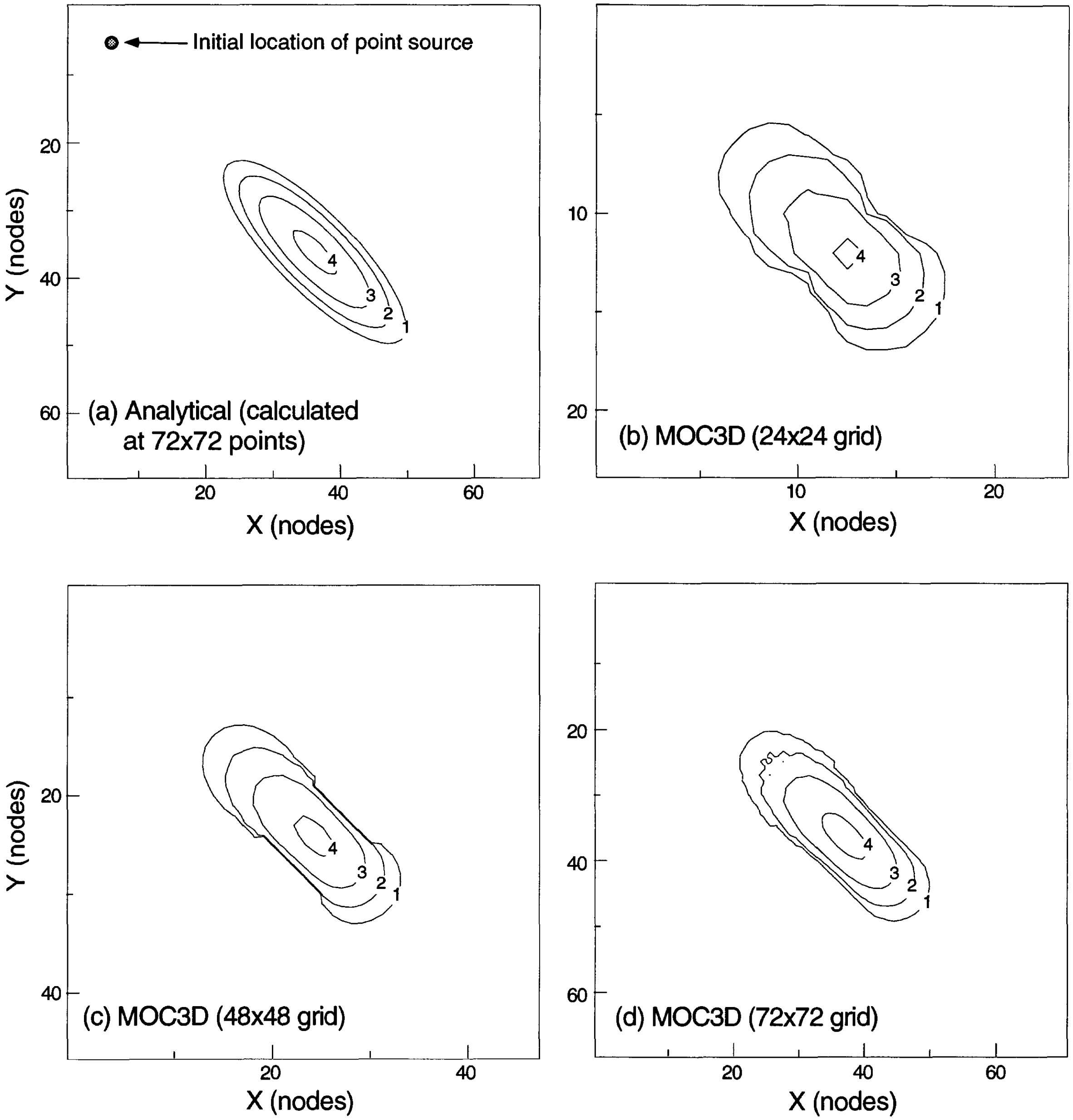


Figure 31. Concentration contours for (a) analytical and (b, c, d) numerical solutions for transport of a point initial condition in uniform flow at 45 degrees to *x* and *y*. Contour values are the log of the concentrations.

the discrepancies may also be due to the contouring program used to visually represent the solutions.

When the flow is at an angle to the grid, as for the case illustrated in fig. 31, then negative concentrations are most likely to occur. In this case, some small areas of slightly negative concentrations were calculated, but are not evident in fig. 31 because they were filtered out during the contouring process to allow a clear depiction of the position of the plume. However, to indicate the extent of the area of negative concentrations, we have replotted the central part of the domain illustrated in fig. 31b (for the 24x24 grid) in fig. 32, in which all areas where the relative concentration is less

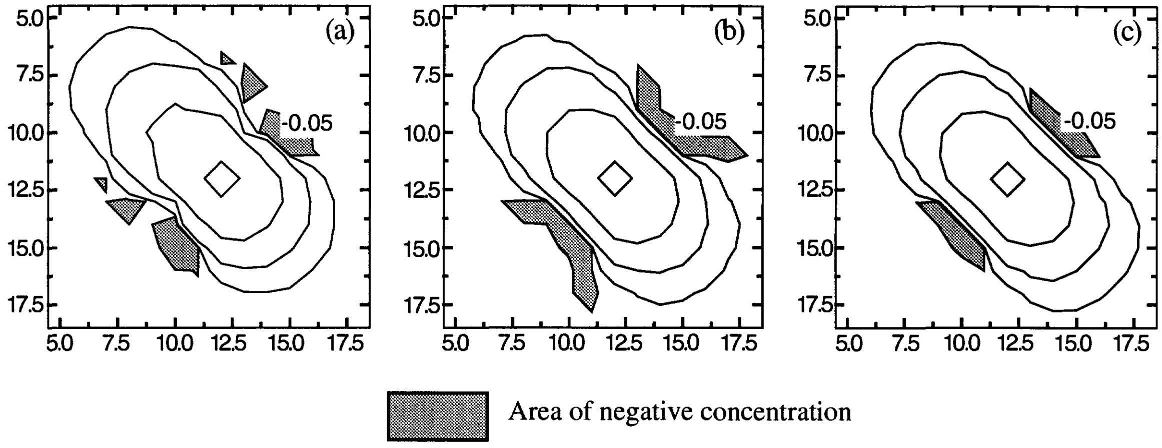


Figure 32. Concentration contours showing effects on areas of negative concentrations related to decreasing CELDIS factor in *MOC3D* in simulation of flow at 45 degrees to grid having 24 rows and 24 columns of nodes: (a) CELDIS = 0.50; (b) CELDIS = 0.25; and (c) CELDIS = 0.10.

than -0.05 are shaded. Figure 32a represents the same solution as shown in fig. 31b. We tested the sensitivity of the solution and of the extent of negative concentrations to the size of the transport time increment by adjusting the value of CELDIS. The area in which negative concentrations occurred at the same elapsed simulation time was slightly smaller for the smallest value of CELDIS. In all three cases the mass balance errors were about the same and always less than 0.2 percent relative to the initial solute mass stored.

Constant Source in Nonuniform Flow

Burnett and Frind (1987) used a numerical model to analyze a hypothetical problem having a constant source of solute in a finite area at the surface of an aquifer having homogeneous properties, but nonuniform boundary conditions, which result in nonuniform flow. Because an analytical solution is not available for such a complex system, we use their results for this test case as a benchmark for comparison with the results of applying *MOC3D* to the same problem. Burnett and Frind (1987) used an alternating-direction Galerkin finite-element technique to solve the solute-transport equation in both two and three dimensions. Their model also includes the capability to vary α_T as a function of direction, thereby allowing that feature of *MOC3D* to be evaluated in the same problem set.

A simplified diagram of the problem is illustrated in figure 33. The left (x = 0) and bottom surfaces are no-flow boundaries, representing a ground-water divide and an impermeable base, respectively. The top and right surfaces are constant-head boundaries, representing the water table and a discharge boundary, respectively. The front and back vertical faces are no-flow boundaries, representing streamlines or flow paths parallel to those surfaces. The length of the domain is 200 m and the saturated thickness varies from 21 m on the left no-flow boundary to 20 m on the right constant-head boundary. The heads on the upper surface are specified as a one-quarter cosine from 1 m on the left to 0 m on the right. Heads are fixed at a slightly negative value (-0.00736 m) at all nodes in the right column of cells, so that the head on the right side of the transport domain (at x = 200 m) will almost exactly equal 0 m, and the heads and

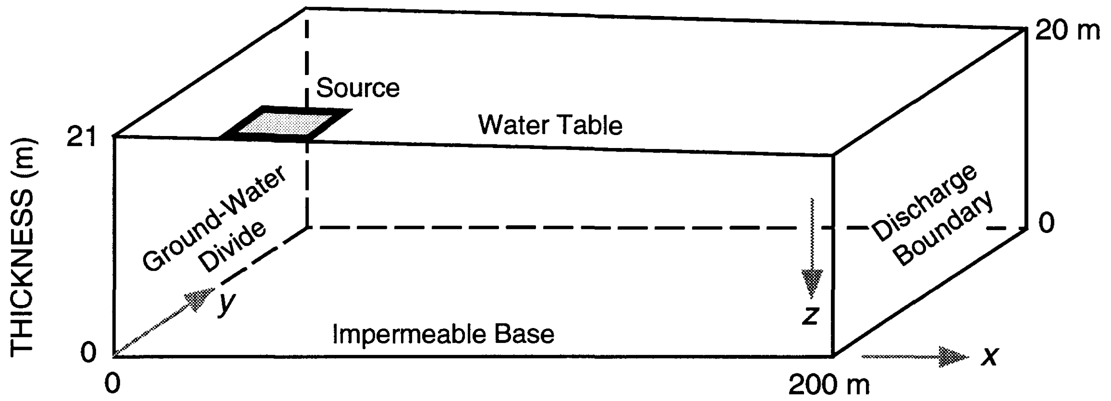


Figure 33. Transport domain and boundary conditions for nonuniform-flow test problem of Burnett and Frind (1987); front surface represents a plane of symmetry (modified from Burnett and Frind, 1987, fig. 4).

hydraulic gradients throughout the domain calculated using *MOC3D* would be consistent with those in the analysis of Burnett and Frind (1987). The aquifer properties are assumed to be homogeneous and isotropic. These boundary conditions yield a two-dimensional flow field, which has components of flow in the x- and z-directions only. Therefore, the specified width of the domain is varied for each particular simulation to accommodate the plume, and depends on the dimensionality of the simulation (whether two- or three-dimensions) and on the values of the dispersivity coefficients. The solute source is located between 18.25 and 32.50 m from the left side of the aquifer and has a width of 10 m, extending 5 m on either side of the plane of symmetry at $y = 0$.

Cases of both two- and three-dimensional transport were examined for this basic problem. The grids used in the *MOC3D* simulations were designed to maximize compatibility with the results of the finite-element models used by Burnett and Frind (1987), so that comparisons of results would represent a reasonable benchmarking exercise. However, some differences in discretization could not be avoided because the finite-element methods allow specifications of values at nodes, which can be placed directly on boundaries. Nodes in *MOC3D* are located at the centers of cells, and values specified at nodes are always one-half of the grid spacing away from the edge of the model domain. Among the small differences arising from the alternative discretization schemes is that (1) the modeled location of the 14.25 m long source area is offset by 0.225 m towards the right in the *MOC3D* grid, and (2) the total length of the domain is 199.5 m in the *MOC3D* grid.

The first analysis of this test case focused on the simplest one—a two-dimensional analysis. The input data values for this analysis are listed in Table 15. The *MOC3D* grid consisted of 141 columns and 91 layers. However, the top layer and right column of cells are devoted to assuring consistent boundary conditions between the two models, and they were considered to lie outside of the domain of the transport problem for benchmarking purposes. That is, 140 columns at a spacing of 1.425 m yields a transport domain length of 199.5 m, which is approximately equal to the desired total horizontal distance of 200 m. Similarly, the appropriate height of the domain is assured by setting the thickness of each cell within a given column equal to 1/90th of the height of the column, where the height (or total saturated thickness) varies from 21 m on the left to 20 m on the right. (In comparison, Burnett and Frind used a variable spacing in which Δx ranged from 2.5 to 6.0 m and Δz ranged from 0.75 to 1.25 m.) The top flow layer consisted of constant-head nodes and the solute source. Because of the symmetry in the flow field, we were able to increase the efficiency of the simulation by using a custom initial particle placement of only three particles in each cell, as shown in figure 34, and still achieve reasonably accurate results. Burnett and Frind (1987) report that their solution yielded an average areal recharge rate of about 8×10^{-4} m/day; the *MOC3D* solution yielded an average areal recharge rate of about 7.8×10^{-4} m/day. This agreement is evidence that the parameters and boundary conditions for the two model analyses are similar enough to permit a benchmarking comparison.

Table 15. Parameters used in *MOC3D* simulation of transport in a vertical plane from a continuous point source in a nonuniform, steady-state, two-dimensional flow system described by Burnett and Frind (1987)

Parameter	Value
K	1.0 m/day
ε	0.35
α_L	3 m
α_{TV}	0.01 m
D_m	10^{-4} m^2/day
PERLEN (length of stress period)	12,000 days
Source concentration (C')	1.0
Number of rows[1]	141
Number of columns	1
Number of layers[1]	91
DELR (Δx)	1.425 m
DELC (Δy)	1.0 m
Thickness (b)	0.2222-0.2333 m
NPTPND (Initial number of particles per cell)	3
CELDIS	1.0
INTRPL (Interpolation scheme)	1

[1] One row and layer were allocated to defining boundary conditions, so concentrations calculated in only 140 rows and 90 layers were used for benchmarking.

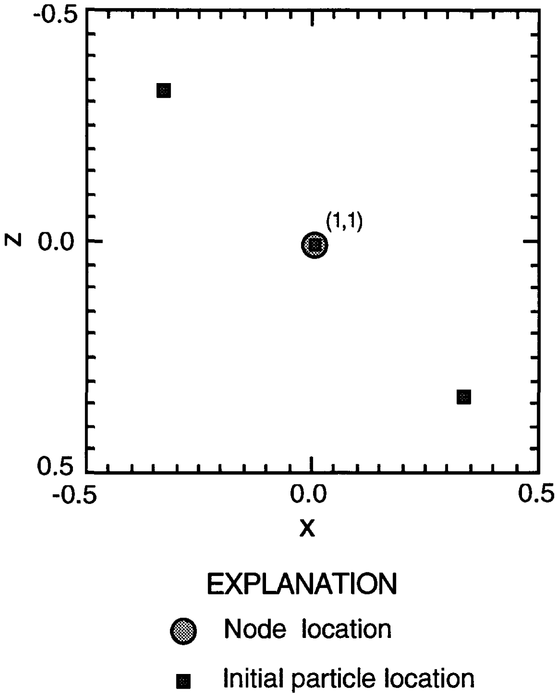


Figure 34. Initial particle positions within a cell for the Burnett-Frind (1987) test case (based on custom particle placement and NPTPND = 3). The relative coordinates on the *x*- and *z*-axes shown for the cell (1,1) are the same for any cell of the grid; this relative coordinate system is used for the custom definition of particle locations in the input file.

Results for the two-dimensional case from the *MOC3D* model closely match those of Burnett and Frind (see fig. 35). Both models represent the solution on the plane of symmetry (that is, on the front face of the block shown in fig. 33). The concentration contours are located in almost exactly the same positions for both models. However, in the *MOC3D* results, the contours lag slightly behind those of Burnett and Frind (1987). This may be

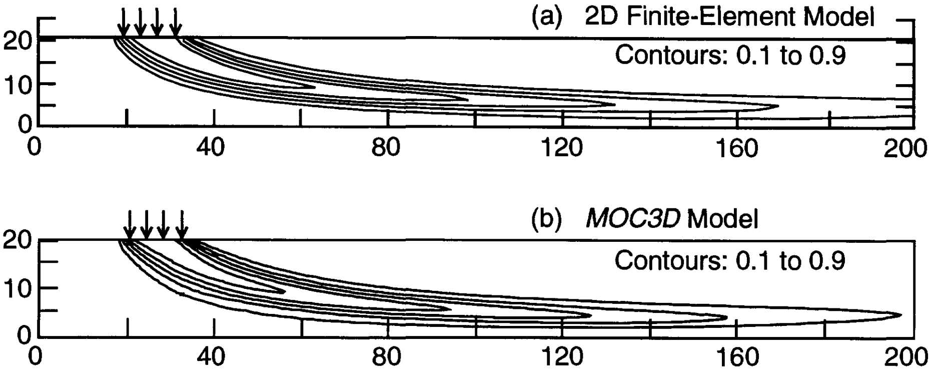


Figure 35. Two-dimensional simulation results for nonuniform-flow test case showing plume positions as contours of relative concentration: (a) finite-element model (modified from Burnett and Frind, 1987, fig. 8a), and (b) *MOC3D*. Contour interval is 0.2 relative concentration.

attributable to small differences in the numerical treatment of the source between the two models and (or) to the slightly lower flux (and velocity) in the *MOC3D* solution. Note that the solution to the two-dimensional case in the *x*-*z* plane does not depend on the value of α_{T_H}.

For the three-dimensional analyses, the *MOC3D* grid is expanded to 15 rows having Δy of 1.0 m. The source is applied over the first 5 rows, taking advantage of the symmetry along the *y*-axis to account for the 10 m width of the source. Note that because of symmetry, the flow fields are identical in the two- and three-dimensional cases. Figure 36 shows the transport results for both models for the case in which α_{T_V} = 0.01 m and α_{T_H} = 0.1 m. In the *MOC3D* results (fig. 36b) the vertical plane in the first row is contoured. Note that there will be a slight discrepancy in the basis of comparison because concentrations from *MOC3D* are evaluated at the center of the block (1/2 of a cell width from the plane of

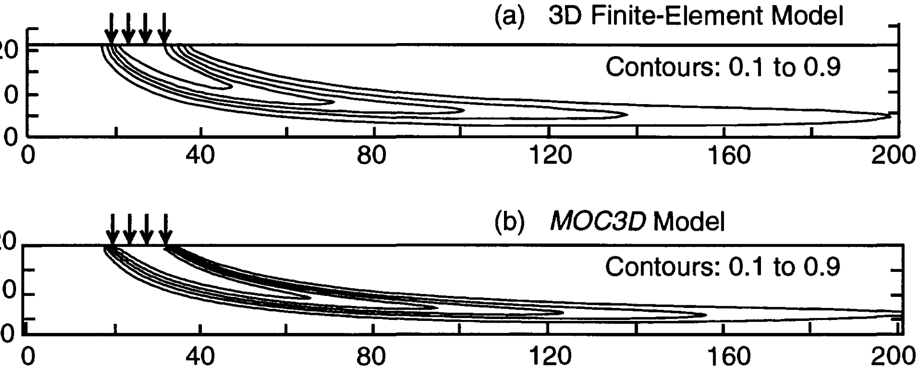


Figure 36. Three-dimensional simulation results for nonuniform-flow test case in which α_{T_H} = 0.1 m and α_{T_V} = 0.01 m: (a) finite-element model (modified from Burnett and Frind, 1987, fig. 8c), and (b) *MOC3D*. Plume positions are represented by contours of relative concentration; contour interval is 0.2 relative concentration.

symmetry), whereas those from Burnett and Frind (1987) are evaluated on the cell faces (directly on the plane of symmetry). A three-dimensional visualization of the *MOC3D* results are presented in fig. 37, which was generated from the concentrations on particles (as opposed to nodal concentrations in fig. 36b) using a three-dimensional visualization software package. (The plume definition from particle concentrations and locations is inherently more precise than can be defined from averaged nodal concentrations.) This perspective shows more clearly the entire plume and the magnitude of lateral spreading of the plume. It also illustrates the impact of specifying unequal transverse dispersivity coefficients, in that the transverse spreading in the horizontal direction increases more rapidly than does the transverse spreading in the vertical direction, as evident in the "chair" cut near the source location and on the face of the plume at the downgradient boundary. Figure 38 shows the results for the case in which the vertical transverse dispersivity is increased by a factor of ten, so that $\alpha_{T_H} = \alpha_{T_V} = 0.1$ m. Overall, the *MOC3D* results (figs. 35b, 36b and 38b) agree closely with those of Burnett and Frind (1987) (figs. 35a, 36a and 38a).

Comparison of the three-dimensional results with the two-dimensional analysis shows that all contours are closer to the source of solute in the three-dimensional cases. This is expected because the contaminant source has a finite length and consideration of the additional dimension for the dispersion process allows spreading of solute in the *y*-direction, which means that less solute mass will remain in the vertical plane being contoured. This

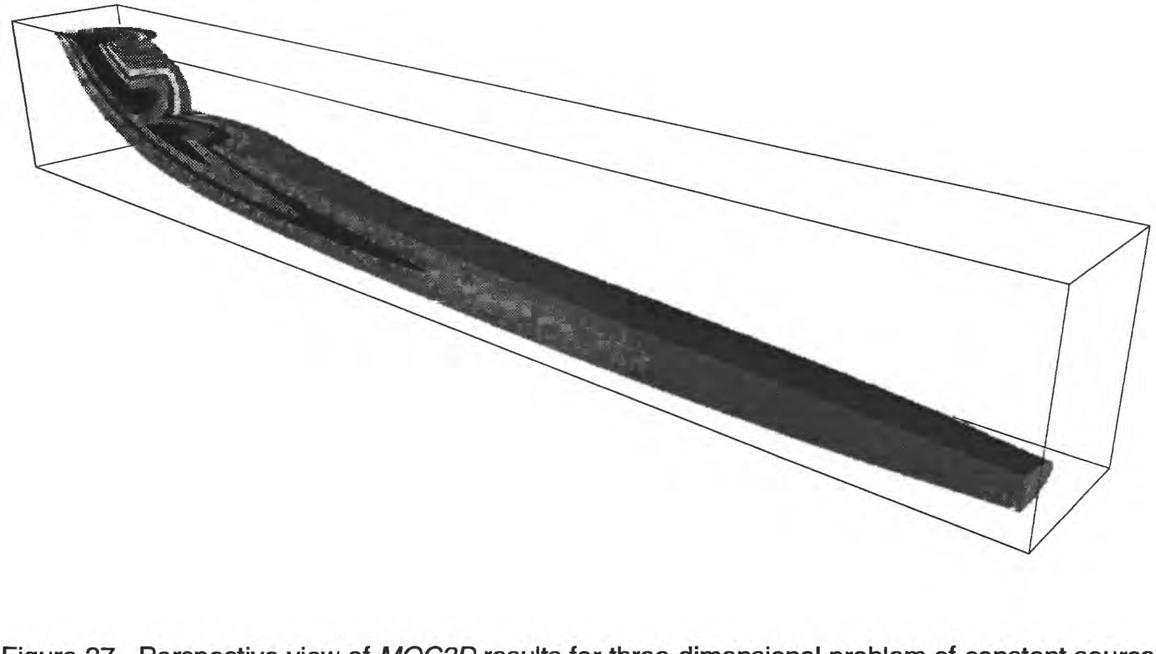


Figure 37. Perspective view of *MOC3D* results for three-dimensional problem of constant source in nonuniform flow and unequal transverse dispersivity coefficients. This visualization of the plume was generated from particle concentrations using a three-dimensional visualization software package and is derived from the same simulation that is the basis of fig. 36b. Note that a piece of the plume near the source is cut away (a "chair" cut) to expose a clearer view of the degree of transverse spreading in the selected vertical and horizontal planes. Shading increments are in relative-concentration intervals of 0.20, and interval bounds range between 0.9 and 0.1. Concentrations less than 0.10 are transparent.

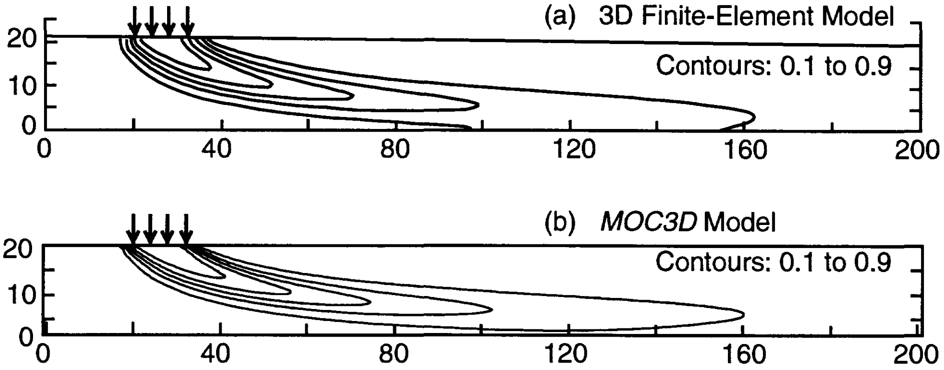


Figure 38. Three-dimensional simulation results for nonuniform-flow test case in which $\alpha_{T_H} = \alpha_{T_V} = 0.01$ m: (a) finite-element model (modified from Burnett and Frind, 1987, fig. 9b), and (b) *MOC3D*. Plume positions are represented by contours of relative concentration; contour interval is 0.2 relative concentration.

makes it appear that the plume has not spread as far in the x-direction during the same elapsed time. In the three-dimensional simulations, the lower-value and higher-value contours are slightly closer to the 0.5 contour in the *MOC3D* results, and the contours defining the lateral edges of the plume are spaced closer together. These characteristics indicate that the *MOC3D* results may include less numerical dispersion than the finite-element results. However, this minor difference may simply be an outcome of having used a finer grid spacing in *MOC3D*.

Relative Computational Efficiency

The computational effort required by the *MOC3D* code is strongly dependent on the size of the problem being solved, as reflected primarily by the total number of nodes, total number of particles, and total number of time increments. Analyses indicate that the greatest computational effort, as measured by *CPU* time, is typically expended in the particle tracking routines. For a given problem, the efficiency of the code may vary significantly as a function of the characteristics of the particular computer on which the simulation is performed, and to some extent on which FORTRAN compiler (and which compiler options) were used to generate the executable code.

To provide a qualitative indication of these relations, we have run all of the sample problems described in this report on a variety of computers. The relative running times for each problem on a variety of different computers are presented in table 16. The run times are measured as *CPU* time in seconds. As indicated in table 16, the efficiency for a given problem may vary by more than a factor of ten, depending on which of the tested computers were used. However, for the given test problems, the efficiency was much more sensitive to the overall size of the problem, and the *CPU* time on a given computer varied by about four orders of magnitude between the simple one-dimensional problem and the more complex three-dimensional problem. The model user should be aware that in some cases, model efficiency may be a serious constraint.

Table 16. Comparison of *MOC3D* simulation times for selected test cases on various computer platforms

Problem Description	Number of Nodes	Number of Moves	Maximum Number of Particles	Run Time in CPU-seconds					
				Silicon Graphics[1]	Data General Server[2]	Data General Workstation[3]	PC (90 MHz, Pentium)[4]	PC (133 MHz, Pentium)[5]	Macintosh PowerPC (9500/120)[6]
One-Dimensional Steady Flow	120	201	360	0.71	2.6	5.2	3.5	1.5	NT
Three-Dimensional Steady Flow	14,400	207	43,200	82	350	650	320	172	217
Two-Dimensional Radial Flow and Dispersion	900	596	52,666	240	750	1,450	620	440	596
Point Initial Condition in Uniform Flow (flow at 45 degrees to grid)	13,824	19	110,729	35	240	450	140	65	84
Constant Source in Nonuniform Flow (Two-Dimensional)	12,831	4,218	68,027	2,290	13,450	18,310	9,060	7,470	6,294
Constant Source in Nonuniform Flow (Three-Dimensional)	192,465	4,218	881,260	51,720	139,300	NT	NT	NT	NT

NT = Not Tested

[1] Silicon Graphics server with an R8000 chip running Irix 6.0.1 with 576MB RAM and a 90 MHz processor. MIPSpro F77 was used to compile *MOC3D*.

[2] Data General server with a Motorola 88110 chip running DG Unix 5.4R3.10 with 256MB RAM and a 45 MHz processor. Green Hills Software FORTRAN-88000 was used to compile *MOC3D*.

[3] Data General AVIION 530 running DG Unix 5.4R3.10 with 32MB RAM and a 33MHz processor. Green Hills Software FORTRAN-88000 was used to compile *MOC3D*.

[4] IBM-compatible Pentium PC running MS-DOS 6.2 with 8MB RAM and a 90 MHz processor. Powerstation 1.0 FORTRAN was used to compile *MOC3D*.

[5] IBM-compatible Pentium PC running Windows95 with 32MB RAM and a 133 MHz processor. Powerstation 1.0 FORTRAN was used to compile *MOC3D*.

[6] Macintosh 9500 PC with 120 MHz PowerPC 604 processor and 32MB of RAM. *MOC3D* was compiled using Fortner Research LS FORTRAN Version 1.1 for Power Mac.

CONCLUSIONS

The *MOC3D* model described in this report can simulate the transient, three-dimensional, transport and dispersion of a solute subject to decay and retardation. The solute-transport model is integrated with the *MODFLOW* model, which is used to solve the ground-water flow equation for either steady-state or transient flow. The numerical methods used to solve the governing equations allow their application to systems having heterogeneous properties and complex boundary conditions. The package thus has broad general application and flexibility for application to a wide range of hydrogeological problems.

The accuracy and precision of the numerical results were tested and evaluated by comparison of the *MOC3D* results with analytical solutions for several relatively simple and idealized problems and by benchmarking comparisons against the results of other numerical codes for more complex problems for which no analytical solutions are available. These tests indicate that the model can successfully and accurately simulate the three-dimensional transport and dispersion of a solute in flowing ground water. This implementation of the method of characteristics is not strictly mass conservative and the method of calculating a solute mass balance is inherently an approximation. Therefore, calculated mass-balance errors may be nonzero, but are generally less than 10 percent and often 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. The efficiency of the solution is sensitive to the total number of particles used and to the size of the transport time increment, as determined by the stability criteria for the solute-transport equation. An advantage of this method is that, in general, its accuracy and efficiency are greatest for advection-dominated problems, which is a characteristic of many ground-water contamination problems that pose serious environmental risks.

REFERENCES

Appel, C.A., Reilly, T.E., 1994, Summary of computer programs produced by the U.S. Geological Survey for simulation of ground-water flow and quality, 1994: U.S. Geological Survey Circular 1104, 98 p.

Bear, Jacob, 1979, Hydraulics of Groundwater: McGraw-Hill, New York, 567 p.

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

Burnett, R.D., and Frind, E.O., 1987, Simulation of contaminant transport in three dimensions, 2. Dimensionality effects: Water Resources Research, v. 23, no. 4, p. 695-705.

Domenico, P.A., and Schwartz, F.W., 1990, Physical and Chemical Hydrogeology: John Wiley & Sons, New York, 824 p.

Engineering Technologies Associates, Inc., 1989, Three dimensional solute transport using the random-walk algorithm and MODFLOW, User's manual for RAND3D: Engineering Technologies Associates, Inc., Ellicott City, MD, 102 p.

Freeze, R.A., and Cherry, J.A., 1979, Groundwater: Prentice-Hall, Englewood Cliffs, N.J., 588 p.

Garabedian, S.P., LeBlanc, D.R., Gelhar, L.W., and Celia, M.A., 1991, Large-scale natural gradient tracer test in sand and gravel, Cape Cod, Massachusetts, 2, Analysis of spatial moments for a nonreactive tracer: Water Resources Research, v. 27, no. 5, p. 911-924.

Garder, A.O., Peaceman, D.W., Pozzi, A.L., 1964, Numerical calculation of multidimensional miscible displacement by the method of characteristics: Soc. Petroleum Eng. Jour., v. 4, no.1, p. 26-36.

Gelhar, L.W., Welty, C., and Rehfeldt, K.R., 1992, A critical review of data on field-scale dispersion in aquifers: Water Resources Research, v. 28, no. 7, p. 1955-1974.

Goode, D.J., 1990a, Particle velocity interpolation in block-centered finite difference groundwater flow models: Water Resources Research, v. 26, no. 5, p. 925-940.

Goode, D.J., 1990b, Governing equations and model approximation errors associated with the effects of fluid-storage transients on solute transport in aquifers: U.S. Geological Survey Water-Resources Investigations Report 90-4156, 20 p.

Goode, D.G., and Appel, C.A., 1992, Finite-difference interblock transmissivity for unconfined aquifers and for aquifers having smoothly varying transmissivity: U.S. Geological Survey Water-Resources Investigations Report 92-4124, 79 p.

Goode, D.J., and Konikow, L.F., 1989, Modification of a method-of-characteristics solute-transport model to incorporate decay and equilibrium-controlled sorption or ion exchange: U.S. Geological Survey Water-Resources Investigations Report 89-4030, 65 p.

Goode, D.J., and Konikow, L.F., 1991, Testing a method-of-characteristics model of three-dimensional solute transport in ground water: *in* Lennon, G. P., ed., Symposium on Ground Water, Proceedings of the International Symposium, Nashville, Tenn., American Society of Civil Engineers, New York, p. 21-27.

Harbaugh, A.W., and McDonald, M.G., 1996a, User's documentation for MODFLOW-96, an update to the U.S. Geological Survey modular finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 96-485, 56 p.

Harbaugh, A.W., and McDonald, M.G., 1996b, Programmer's documentation for MODFLOW-96, an update to the U.S. Geological Survey modular finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 96-486, 220 p.

Hill, M.C., 1990, Preconditioned Conjugate-Gradient 2 (PCG2)—A computer program for solving ground-water flow equations: U.S. Geological Survey Water-Resources Investigations Report 90-4048, 43 p.

Hsieh, P. A., 1986, A new formula for the analytical solution of the radial dispersion problem: Water Resources Research, v. 22, no. 11, p. 1597-1605.

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

Konikow, L.F., and Grove, D.B., 1977, Derivation of equations describing solute transport in ground water: U.S. Geological Survey Water-Resources Investigations Report 77-19, [Revised 1984], 30 p.

McDonald, M.G., and Harbaugh, A.W., 1988, A modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 6, Chapter A1, 586 p.

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

Pinder, G.F., and Gray, W.G., 1977, Finite Element Simulation in Surface and Subsurface Hydrology: Academic Press, New York, N.Y, 288 p.

Prickett, T.A., Naymik, T.G., and Lonnquist, C.G., 1981, A "random-walk" solute transport model for selected groundwater quality evaluations: Illinois State Water Survey Bulletin 65, 103 p.

Reddell, D.L., and Sunada, D.K., 1970, Numerical simulation of dispersion in groundwater aquifers: Colorado State University, Ft. Collins, CO, Hydrology Paper 41, 79 p.

Robson, S.G., 1974, Feasibility of digital water-quality modeling illustrated by application at Barstow, California: U.S. Geological Survey Water-Resources Investigations Report 46-73, 66 p.

Robson, S.G., 1978, Application of digital profile modeling techniques to ground-water solute-transport at Barstow, California: U.S. Geological Survey Water-Supply Paper 2050, 28 p.

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

Wexler, E.J., 1992, Analytical solutions for one-, two-, and three-dimensional solute transport in ground-water systems with uniform flow: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 3, Chapter B7, 190 p.

Zheng, C., 1990, MT3D: A modular three-dimensional transport model: S.S. Papadopulos and Associates, Inc., Bethesda, MD.

APPENDIX A: FINITE-DIFFERENCE APPROXIMATIONS

Applying finite-difference approximations that are centered-in-space and explicit (forward-in-time), the component of the dispersive flux in the x-direction across the cell face at $(j+1/2,i,k)$ (equivalent to $M_{f(j+1/2,i,k)}/\Delta t\,\Delta y$ from eq. 57) may be written:

$$\begin{aligned}
-\left(\varepsilon b D_{1m}\frac{\partial C}{\partial x_m}\right)^{*}_{j+1/2,i,k} &= -\left(\varepsilon b D_{xx}\right)^{t+1}_{j+1/2,i,k}\frac{\left(C^{*}_{j+1,i,k}-C^{*}_{j,i,k}\right)}{\Delta x}\\
&\quad -\left(\varepsilon b D_{xy}\right)^{t+1}_{j+1/2,i,k}\frac{\left(C^{*}_{j,i+1,k}+C^{*}_{j+1,i+1,k}-C^{*}_{j,i-1,k}-C^{*}_{j+1,i-1,k}\right)}{4\Delta y}\\
&\quad -\left(\varepsilon b D_{xz}\right)^{t+1}_{j+1/2,i,k}\frac{1}{2}\left(\frac{C^{*}_{j+1,i,k+1}-C^{*}_{j+1,i,k-1}}{2B^{t+1}_{j+1,i,k}}+\frac{C^{*}_{j,i,k+1}-C^{*}_{j,i,k-1}}{2B^{t+1}_{j,i,k}}\right)
\end{aligned} \tag{A1}$$

where $2B_{j,i,k} \equiv b_{j,i,k} + 1/2\,(b_{j,i,k-1} + b_{j,i,k+1})$ is the vertical distance between nodes $(j,i,k+1)$ and $(j,i,k-1)$. The superscript "*" indicates the use of an average concentration, as defined in eq. 55.

Similarly, the y-component of the dispersive flux vector at $j,i+1/2,k$ is approximated by:

$$\begin{aligned}
-\left(\varepsilon b D_{2m}\frac{\partial C}{\partial x_m}\right)^{*}_{j,i+1/2,k} &= -\left(\varepsilon b D_{yy}\right)^{t+1}_{j,i+1/2,k}\frac{\left(C^{*}_{j,i+1,k}-C^{*}_{j,i,k}\right)}{\Delta y}\\
&\quad -\left(\varepsilon b D_{yx}\right)^{t+1}_{j,i+1/2,k}\frac{\left(C^{*}_{j+1,i,k}+C^{*}_{j+1,i+1,k}-C^{*}_{j-1,i,k}-C^{*}_{j-1,i+1,k}\right)}{4\Delta x}\\
&\quad -\left(\varepsilon b D_{yz}\right)^{t+1}_{j,i+1/2,k}\frac{1}{2}\left(\frac{C^{*}_{j,i+1,k+1}-C^{*}_{j,i+1,k-1}}{2B^{t+1}_{j,i+1,k}}+\frac{C^{*}_{j,i,k+1}-C^{*}_{j,i,k-1}}{2B^{t+1}_{j,i,k}}\right).
\end{aligned} \tag{A2}$$

The z-component does not include the saturated thickness and is approximated by:

$$\begin{aligned}
-\left(\varepsilon D_{3m}\frac{\partial C}{\partial x_m}\right)^{*}_{j,i,k+1/2} &= -\left(\varepsilon D_{zz}\right)^{t+1}_{j,i,k+1/2}\frac{\left(C^{*}_{j,i,k+1}-C^{*}_{j,i,k}\right)}{b^{t+1}_{j,i,k+1/2}}\\
&\quad -\left(\varepsilon D_{zx}\right)^{t+1}_{j,i,k+1/2}\frac{\left(C^{*}_{j+1,i,k}+C^{*}_{j+1,i,k+1}-C^{*}_{j-1,i,k}-C^{*}_{j-1,i,k+1}\right)}{4\Delta y}\\
&\quad -\left(\varepsilon D_{zy}\right)^{t+1}_{j,i,k+1/2}\frac{\left(C^{*}_{j,i+1,k}+C^{*}_{j,i+1,k+1}-C^{*}_{j,i-1,k}-C^{*}_{j,i-1,k+1}\right)}{4\Delta y}.
\end{aligned} \tag{A3}$$

Applying centered finite-difference approximations, the change in concentration due to dispersion, neglecting the sink/source term, can be written:

$$
\begin{aligned}
\left.\frac{\mathrm{d}C}{\mathrm{d}t}\right|_{\mathrm{disp}} &= \frac{1}{\left(R_f\right)_k (\varepsilon b)_{j,i,k}^{t+1}} \\
&\left\{ \frac{1}{\Delta x^2}\left[\left(\varepsilon b D_{xx}\right)_{j+1/2,i,k}^{t+1}\left(C_{j+1,i,k}^{*}-C_{j,i,k}^{*}\right)-\left(\varepsilon b D_{xx}\right)_{j-1/2,i,k}^{t+1}\left(C_{j,i,k}^{*}-C_{j-1,i,k}^{*}\right)\right]\right. \\
&+\frac{1}{2\Delta x\Delta y}\left[\left(\varepsilon b D_{xy}\right)_{j+1/2,i,k}^{t+1}\left(C_{j+1,i+1,k}^{*}+C_{j,i+1,k}^{*}-C_{j+1,i-1,k}^{*}-C_{j,i-1,k}^{*}\right)\right. \\
&\qquad\left.-\left(\varepsilon b D_{xy}\right)_{j-1/2,i,k}^{t+1}\left(C_{j,i+1,k}^{*}+C_{j-1,i+1,k}^{*}-C_{j,i-1,k}^{*}-C_{j-1,i-1,k}^{*}\right)\right] \\
&+\frac{1}{\Delta x}\left[\frac{\left(\varepsilon b D_{xz}\right)_{j+1/2,i,k}^{t+1}}{2}\left(\frac{C_{j+1,i,k+1}^{*}-C_{j+1,i,k-1}^{*}}{2B_{j+1,i,k}^{t+1}}+\frac{C_{j,i,k+1}^{*}-C_{j,i,k-1}^{*}}{2B_{j,i,k}^{t+1}}\right)\right. \\
&\qquad\left.-\frac{\left(\varepsilon b D_{xz}\right)_{j-1/2,i,k}^{t+1}}{2}\left(\frac{C_{j,i,k+1}^{*}-C_{j,i,k-1}^{*}}{2B_{j,i,k}^{t+1}}+\frac{C_{j-1,i,k+1}^{*}-C_{j-1,i,k-1}^{*}}{2B_{j\text{-}1,i,}^{t+1}k}\right)\right] \\
&+\frac{1}{\Delta y^2}\left[\left(\varepsilon b D_{yy}\right)_{j,i+1/2,k}^{t+1}\left(C_{j,i+1,k}^{*}-C_{j,i,k}^{*}\right)-\left(\varepsilon b D_{yy}\right)_{j,i-1/2,k}^{t+1}\left(C_{j,i,k}^{*}-C_{j,i-1,k}^{*}\right)\right] \\
&+\frac{1}{2\Delta x\Delta y}\left[\left(\varepsilon b D_{yx}\right)_{j,i+1/2,k}^{t+1}\left(C_{j+1,i+1,k}^{*}+C_{j+1,i,k}^{*}-C_{j-1,i+1,k}^{*}-C_{j-1,i,k}^{*}\right)\right. \\
&\qquad\left.-\left(\varepsilon b D_{yx}\right)_{j,i-1/2,k}^{t+1}\left(C_{j+1,i,k}^{*}+C_{j+1,i-1,k}^{*}-C_{j-1,i,k}^{*}-C_{j-1,i-1,k}^{*}\right)\right] \\
&+\frac{1}{\Delta y}\left[\frac{\left(\varepsilon b D_{yz}\right)_{j,i+1/2,k}^{t+1}}{2}\left(\frac{C_{j,i+1,k+1}^{*}-C_{j,i+1,k-1}^{*}}{2B_{j,i+1,k}^{t+1}}+\frac{C_{j,i,k+1}^{*}-C_{j,i,k-1}^{*}}{2B\ t+1}j,i,k\right)\right. \\
&\qquad\left.-\frac{\left(\varepsilon b D_{yz}\right)_{j,i-1/2,k}^{t+1}}{2}\left(\frac{C_{j,i,k+1}^{*}-C_{j,i,k-1}^{*}}{2B_{j,i,k}^{t+1}}+\frac{C_{j,i-1,k+1}^{*}-C_{j,i-1,k-1}^{*}}{2B_{j,i\text{-}1,k}^{t+1}}\right)\right] \\
&+\left[\left(\varepsilon D_{zz}\right)_{j,i,k+1/2}^{t+1}\frac{\left(C_{j,i,k+1}^{*}-C_{j,i,k}^{*}\right)}{b_{j,i,k+1/2}^{t+1}}-\left(\varepsilon D_{zz}\right)_{j,i,k-1/2}^{t+1}\frac{\left(C_{j,i,k}^{*}-C_{j,i,k-1}^{*}\right)}{b_{j,i,k-1/2}^{t+1}}\right]
\end{aligned}
$$

$$+\frac{1}{2\Delta x}\left[\left(\varepsilon D_{zx}\right)^{t+1}_{j,i,k+1/2}\left(C^*_{j+1,i,k+1}+C^*_{j+1,i,k}-C^*_{j-1,i,k+1}-C^*_{j-1,i,k}\right)\right.$$

$$\left.-\left(\varepsilon D_{zx}\right)^{t+1}_{j,i,k-1/2}\left(C^*_{j+1,i,k}+C^*_{j+1,i,k-1}-C^*_{j-1,i,k}-C^*_{j-1,i,k-1}\right)\right]$$

$$+\frac{1}{2\Delta y}\left[\left(\varepsilon D_{zy}\right)^{t+1}_{j,i,k+1/2}\left(C^*_{j,i+1,k+1}+C^*_{j,i+1,k}-C^*_{j,i-1,k+1}-C^*_{j,i-1,k}\right)\right.$$

$$\left.\left.-\left(\varepsilon D_{zy}\right)^{t+1}_{j,i,k-1/2}\left(C^*_{j,i+1,k}+C^*_{j,i+1,k-1}-C^*_{j,i-1,k}-C^*_{j,i-1,k-1}\right)\right]\right\} \tag{A4}$$

where the dispersion coefficient terms *εbD* and *εD* are given by:

$$\left(\varepsilon bD_{xx}\right)_{j+1/2,i,k}=\alpha_{L(k)}\frac{(bq)^2_{x(j+1/2,i,k)}}{|bq|_{j+1/2,i,k}}+\alpha_{T_H(k)}\frac{(bq)^2_{y(j+1/2,i,k)}}{|bq|_{j+1/2,i,k}}+\alpha_{T_V(k)}\frac{(bq)^2_{z(j+1/2,i,k)}}{|bq|_{j+1/2,i,k}} \tag{A5}$$

$$\left(\varepsilon bD_{xx}\right)_{j-1/2,i,k}=\alpha_{L(k)}\frac{(bq)^2_{x(j-1/2,i,k)}}{|bq|_{j-1/2,i,k}}+\alpha_{T_H(k)}\frac{(bq)^2_{y(j-1/2,i,k)}}{|bq|_{j-1/2,i,k}}+\alpha_{T_V(k)}\frac{(bq)^2_{z(j-1/2,i,k)}}{|bq|_{j-1/2,i,k}} \tag{A6}$$

$$\left(\varepsilon bD_{xy}\right)_{j+1/2,i,k}=\left(\alpha_{L(k)}-\alpha_{T_H(k)}\right)\frac{bq_{x(j+1/2,i,k)}bq_{y(j+1/2,i,k)}}{|bq|_{j+1/2,i,k}} \tag{A7}$$

$$\left(\varepsilon bD_{xy}\right)_{j-1/2,i,k}=\left(\alpha_{L(k)}-\alpha_{T_H(k)}\right)\frac{bq_{x(j-1/2,i,k)}bq_{y(j-1/2,i,k)}}{|bq|_{j-1/2,i,k}} \tag{A8}$$

$$\left(\varepsilon bD_{xz}\right)_{j+1/2,i,k}=\left(\alpha_{L(k)}-\alpha_{T_V(k)}\right)\frac{bq_{x(j+1/2,i,k)}bq_{z(j+1/2,i,k)}}{|bq|_{j+1/2,i,k}} \tag{A9}$$

$$\left(\varepsilon bD_{xz}\right)_{j-1/2,i,k}=\left(\alpha_{L(k)}-\alpha_{T_V(k)}\right)\frac{bq_{x(j-1/2,i,k)}bq_{z(j-1/2,i,k)}}{|bq|_{j-1/2,i,k}} \tag{A10}$$

$$\left(\varepsilon bD_{yy}\right)_{j,i+1/2,k}=\alpha_{T_H(k)}\frac{(bq)^2_{x(j,i+1/2,k)}}{|bq|_{j,i+1/2,k}}+a_{L(k)}\frac{(bq)^2_{y(j,i+1/2,k)}}{|bq|_{j,i+1/2,k}}+\alpha_{T_V(k)}\frac{(bq)^2_{z(j,i+1/2,k)}}{|bq|_{j,i+1/2,k}} \tag{A11}$$

$$\left(\varepsilon bD_{yy}\right)_{j,i-1/2,k}=\alpha_{T_H(k)}\frac{(bq)^2_{x(j,i-1/2,k)}}{|bq|_{j,i-1/2,k}}+\alpha_{L(k)}\frac{(bq)^2_{y(j,i-1/2,k)}}{|bq|_{j,i-1/2,k}}+\alpha_{T_V(k)}\frac{(bq)^2_{z(j,i-1/2,k)}}{|bq|_{j,i-1/2,k}} \tag{A12}$$

$$\left(\varepsilon b D_{yx}\right)_{j,i+1/2,k} = \left(\alpha_{L(k)} - \alpha_{T_H(k)}\right)\frac{bq_{x(j,i+1/2,k)}bq_{y(j,i+1/2,k)}}{\left|bq\right|_{j,i+1/2,k}} \tag{A13}$$

$$\left(\varepsilon b D_{yx}\right)_{j,i-1/2,k} = \left(\alpha_{L(k)} - \alpha_{T_H(k)}\right)\frac{bq_{x(j,i-1/2,k)}bq_{y(j,i-1/2,k)}}{\left|bq\right|_{j,i-1/2,k}} \tag{A14}$$

$$\left(\varepsilon b D_{yz}\right)_{j,i+1/2,k} = \left(\alpha_{L(k)} - \alpha_{T_V(k)}\right)\frac{bq_{y(j,i+1/2,k)}bq_{z(j,i+1/2,k)}}{\left|bq\right|_{j,i+1/2,k}} \tag{A15}$$

$$\left(\varepsilon b D_{yz}\right)_{j,i-1/2,k} = \left(\alpha_{L(k)} - \alpha_{T_V(k)}\right)\frac{bq_{y(j,i-1/2,k)}bq_{z(j,i-1/2,k)}}{\left|bq\right|_{j,i-1/2,k}} \tag{A16}$$

$$\left(\varepsilon D_{zz}\right)_{j,i,k+1/2} = \alpha_{T_V(k+1/2)}\frac{q^2_{x(j,i,k+1/2)}}{\left|q\right|_{j,i,k+1/2}} + \alpha_{T_V(k+1/2)}\frac{q^2_{y(j,i,k+1/2)}}{\left|q\right|_{j,i,k+1/2}} + \alpha_{L(k+1/2)}\frac{q^2_{z(j,i,k+1/2)}}{\left|q\right|_{j,i,k+1/2}} \tag{A17}$$

$$\left(\varepsilon D_{zz}\right)_{j,i,k-1/2} = \alpha_{T_V(k-1/2)}\frac{q^2_{x(j,i,k-1/2)}}{\left|q\right|_{j,i,k-1/2}} + \alpha_{T_V(k-1/2)}\frac{q^2_{y(j,i,k-1/2)}}{\left|q\right|_{j,i,k-1/2}} + \alpha_{L(k-1/2)}\frac{q^2_{z(j,i,k-1/2)}}{\left|q\right|_{j,i,k-1/2}} \tag{A18}$$

$$\left(\varepsilon D_{zx}\right)_{j,i,k+1/2} = \left(\alpha_{L(k+1/2)} - \alpha_{T_V(k+1/2)}\right)\frac{q_{x(j,i,k+1/2)}q_{z(j,i,k+1/2)}}{\left|q\right|_{j,i,k+1/2}} \tag{A19}$$

$$\left(\varepsilon D_{zx}\right)_{j,i,k-1/2} = \left(\alpha_{L(k-1/2)} - \alpha_{T_V(k-1/2)}\right)\frac{q_{x(j,i,k-1/2)}q_{z(j,i,k-1/2)}}{\left|q\right|_{j,i,k-1/2}} \tag{A20}$$

$$\left(\varepsilon D_{zy}\right)_{j,i,k+1/2} = \left(\alpha_{L(k+1/2)} - \alpha_{T_V(k+1/2)}\right)\frac{q_{y(j,i,k+1/2)}q_{z(j,i,k+1/2)}}{\left|q\right|_{j,i,k+1/2}} \tag{A21}$$

$$\left(\varepsilon D_{zy}\right)_{j,i,k-1/2} = \left(\alpha_{L(k-1/2)} - \alpha_{T_V(k-1/2)}\right)\frac{q_{y(j,i,k-1/2)}q_{z(j,i,k-1/2)}}{\left|q\right|_{j,i,k-1/2}} \tag{A22}$$

where all terms are at time level t+1. The flux terms normal to the finite-difference block (or cell faces) are known directly from the solution to the flow equation. For the horizontal terms, bq, the volumetric flux per unit width is known, whereas the vertical flux is specific discharge, or volumetric flux per unit area.

To compute the dispersion coefficients at a block interface, the flux must be computed at this location. The flux normal to the block face is known from the finite-difference solution of the flow equation. However, the other two components must be interpolated from nearby values. Horizontal fluxes are averaged for x fluxes at y block interfaces, and vice versa, by:

$$bq_{x(j,i+1/2,k)} = \frac{bq_{x(j-1/2,i,k)} + bq_{x(j+1/2,i,k)} + bq_{x(j-1/2,i+1,k)} + bq_{x(j+1/2,i+1,k)}}{4} \tag{A23}$$

$$bq_{x(j,i-1/2,k)} = \frac{bq_{x(j-1/2,i,k)} + bq_{x(j+1/2,i,k)} + bq_{x(j-1/2,i-1,k)} + bq_{x(j+1/2,i-1,k)}}{4} \tag{A24}$$

$$bq_{y(j-1/2,i,k)} = \frac{bq_{y(j,i-1/2,k)} + bq_{y(j,i+1/2,k)} + bq_{y(j-1,i-1/2,k)} + bq_{y(j-1,i+1/2,k)}}{4} \tag{A25}$$

$$bq_{y(j+1/2,i,k)} = \frac{bq_{y(j,i-1/2,k)} + bq_{y(j,i+1/2,k)} + bq_{y(j+1,i-1/2,k)} + bq_{y(j+1,i+1/2,k)}}{4}. \tag{A26}$$

Horizontal fluxes per unit width are averaged and normalized by corresponding layer thicknesses to convert to fluxes per unit area at layer interfaces:

$$q_{x(j,i,k-1/2)} = \frac{bq_{x(j-1/2,i,k-1)} + bq_{x(j+1/2,i,k-1)}}{4b_{j,i,k-1}} + \frac{bq_{x(j-1/2,i,k)} + bq_{x(j+1/2,i,k)}}{4b_{j,i,k}} \tag{A27}$$

$$q_{x(j,i,k+1/2)} = \frac{bq_{x(j-1/2,i,k+1)} + bq_{x(j+1/2,i,k+1)}}{4b_{j,i,k+1}} + \frac{bq_{x(j-1/2,i,k)} + bq_{x(j+1/2,i,k)}}{4b_{j,i,k}} \tag{A28}$$

$$q_{y(j,i,k+1/2)} = \frac{bq_{y(j,i-1/2,k+1)} + bq_{y(j,i+1/2,k+1)}}{4b_{j,i,k+1}} + \frac{bq_{y(j,i-1/2,k)} + bq_{y(j,i+1/2,k)}}{4b_{j,i,k}} \tag{A29}$$

$$q_{y(j,i,k-1/2)} = \frac{bq_{y(j,i-1/2,k-1)} + bq_{y(j,i+1/2,k-1)}}{4b_{j,i,k-1}} + \frac{bq_{y(j,i-1/2,k)} + bq_{y(j,i+1/2,k)}}{4b_{j,i,k}}. \tag{A30}$$

Vertical fluxes per unit area are averaged and multiplied by corresponding layer thicknesses to convert to fluxes per unit width at row and column interfaces:

$$bq_{z(j,i+1/2,k)} = \frac{b_{j,i,k}}{4}\left[q_{z(j,i,k-1/2)} + q_{z(j,i,k+1/2)}\right] + \frac{b_{j,i+1,k}}{4}\left[q_{z(j,i+1,k-1/2)} + q_{z(j,i+1,k+1/2)}\right] \tag{A31}$$

$$bq_{z(j,i-1/2,k)} = \frac{b_{j,i,k}}{4}\left[q_{z(j,i,k-1/2)} + q_{z(j,i,k+1/2)}\right] + \frac{b_{j,i-1,k}}{4}\left[q_{z(j,i-1,k-1/2)} + q_{z(j,i-1,k+1/2)}\right] \tag{A32}$$

$$bq_{z(j+1/2,i,k)} = \frac{b_{j,i,k}}{4}\left[q_{z(j,i,k-1/2)} + q_{z(j,i,k+1/2)}\right] + \frac{b_{j+1,i,k}}{4}\left[q_{z(j+1,i,k-1/2)} + q_{z(j+1,i,k+1/2)}\right] \tag{A33}$$

$$bq_{z(j-1/2,i,k)} = \frac{b_{j,i,k}}{4}\left[q_{z(j,i,k-1/2)} + q_{z(j,i,k+1/2)}\right] + \frac{b_{j-1,i,k}}{4}\left[q_{z(j-1,i,k-1/2)} + q_{z(j-1,i,k+1/2)}\right]. \tag{A34}$$

The magnitudes of fluid flux at block interfaces are:

$$|bq|_{j+1/2,i,k} = \left[(bq)^2_{x(j+1/2,i,k)} + (bq)^2_{y(j+1/2,i,k)} + (bq)^2_{z(j+1/2,i,k)}\right]^{1/2} \tag{A35}$$

$$|bq|_{j-1/2,i,k} = \left[(bq)^2_{x(j-1/2,i,k)} + (bq)^2_{y(j-1/2,i,k)} + (bq)^2_{z(j-1/2,i,k)}\right]^{1/2} \tag{A36}$$

$$|bq|_{j,i+1/2,k} = \left[(bq)^2_{x(j,i+1/2,k)} + (bq)^2_{y(j,i+1/2,k)} + (bq)^2_{z(j,i+1/2,k)}\right]^{1/2} \tag{A37}$$

$$|bq|_{j,i-1/2,k} = \left[(bq)^2_{x(j,i-1/2,k)} + (bq)^2_{y(j,i-1/2,k)} + (bq)^2_{z(j,i-1/2,k)}\right]^{1/2} \tag{A38}$$

$$|q|_{j,i,k+1/2} = \left[q^2_{x(j,i,k+1/2)} + q^2_{y(j,i,k+1/2)} + q^2_{z(j,i,k+1/2)}\right]^{1/2} \tag{A39}$$

$$|q|_{j,i,k-1/2} = \left[q^2_{x(j,i,k-1/2)} + q^2_{y(j,i,k-1/2)} + q^2_{z(j,i,k-1/2)}\right]^{1/2}. \tag{A40}$$

As before, all terms that are determined from the flow solution are at time level *t*+1.

APPENDIX B: DATA INPUT INSTRUCTIONS FOR *MOC3D*

MODFLOW Name File

Transport simulation is activated by including a record in the *MODFLOW* name file using the file type (Ftype) "CONC" to link to the transport name file. The transport name file specifies the files to be used when simulating solute transport in conjunction with a simulation of ground-water flow using *MODFLOW*. The transport name file works in the same way as the *MODFLOW* name file.

MODFLOW Source and Sink Packages

Except for recharge, concentrations associated with fluid sources (C') are read as auxiliary parameters in the *MODFLOW* source package. The source concentration is read from a new column appended to the end of each line of the data file describing a fluid sink/source (see documentation for revised *MODFLOW* model; Harbaugh and McDonald, 1996a and 1996b). For example, concentrations associated with well nodes should be appended to the line in the WEL Package where the well's location and pumping rate are defined. These concentrations will be read if the auxiliary parameter "CONCENTRATION" (or "CONC") appears on the first line of the well input data file. The concentration in recharge is defined separately, as described in following section "Source Concentration in Recharge File."

To simulate solute transport the *MODFLOW* option enabling storage of cell-by-cell flow rates for each fluid source or sink is required in all fluid packages except recharge. The key word "CBCALLOCATE" (or "CBC") must appear on the first line of each input data file for a fluid package (see Harbaugh and McDonald, 1996a and 1996b).

MOC3D Input Data Files

All input variables are read using free formats, except as specifically indicated. In free format, variables are separated by one or more spaces or by a comma and optionally one or more spaces. Blank spaces are not read as zeros.

MOC3D Transport Name File (CONC)

FOR EACH SIMULATION:

1. Data: FTYPE NUNIT FNAME

The name file consists of records defining the names and units numbers of the files. Each "record" consists of a separate line of data. There must be a record for the listing file and for the main *MOC3D* input file.

The listing (or output) file ("CLST") must be the first record. The other files may be in any order. Each record can be no more than 79 characters.

FTYPE The file type, which may be one of the following character strings:

CLST *MOC3D* listing file (separate from the *MODFLOW* listing file) *[required].*

MOC Main *MOC3D* input data file *[required].*

CRCH Concentrations in recharge *[optional].*

CNCA Separate output file containing concentration data in ASCII (text-only) format. Frequency and format of printing controlled by NPNTCL and ICONFM *[optional].*

CNCB Separate output file containing concentration data in binary format *[optional].*

VELA Separate output file with velocity data in ASCII format. Frequency and format of printing controlled by NPNTVL and IVELFM *[optional].*

VELB Separate output file with velocity data in binary format *[optional].*

PRTA Separate output file with particle locations printed in ASCII format. Frequency and format of printing controlled by NPNTPL *[optional].*

PRTB Separate output file with particle locations printed in binary format *[optional].*

OBS Observation wells input file *[optional].*

DATA For formatted files such as those required by the OBS package and for array data separate from the main *MOC3D* input data file *[optional].*

DATA(BINARY) For formatted input/output files *[optional].*

NUNIT The FORTRAN unit number used to read from and write to files. Any legal unit number other than 97, 98, and 99 (which are reserved by *MODFLOW*) can be used provided that it is not previously specified in the *MODFLOW* name file.

FNAME The name of the file.

Main *MOC3D* Package Input (MOC)

Input for the method-of-characteristics (*MOC3D*) solute-transport package is read from the unit specified in the transport name file. The input consists of 18 separate records or data sets, as described in detail below. These data are used to specify information about the transport subgrid, physical and chemical transport parameters, numerical solution variables, and output formats. Output file controls for the *MOC3D* package are specified in the transport name file, described previously.

FOR EACH SIMULATION:

1. Data: HEDMOC — A two-line character-string title describing the simulation (80 text characters per line).

2. Data: HEDMOC (continued)

3. Data: ISLAY1 ISLAY2 ISROW1 ISROW2 ISCOL1 ISCOL2

ISLAY1 Number of first (uppermost) layer for transport.

ISLAY2 Last layer for transport.

ISROW1 First row for transport.

ISROW2 Last row for transport.

ISCOL1 First column for transport.

ISCOL2 Last column for transport.

Notes:

Transport may be simulated within a subgrid, which is a "window" within the primary *MODFLOW* grid used to simulate flow. Within the subgrid, the row and column spacing must be uniform, but thickness can vary from cell to cell and layer to layer. However, as discussed in the section reviewing *MOC3D* assumptions, the range in thickness values (or product of thickness and porosity) should be as small as possible.

4. Data: NODISP DECAY DIFFUS

NODISP Flag for no dispersion (set NODISP=1 if no dispersion in problem; this will reduce storage allocation).

DECAY First-order decay rate [1/T] (DECAY=0.0 indicates no decay occurs).

DIFFUS Effective molecular diffusion coefficient $[L^2/T]$.

Notes:

The decay rate (λ) is related to the half life ($t_{1/2}$) of a constituent by $\lambda = (\ln 2)/t_{1/2}$.

The effective molecular diffusion coefficient (D_m) includes the effect of tortuosity.

5. Data: NPMAX NPTPND

NPMAX Maximum number of particles available for particle tracking of advective transport in *MOC3D*. If set to zero, the model will calculate NPMAX according to the following equation:

$$\mathrm{NPMAX} = 2 \times \mathrm{NPTPND} \times \mathrm{NSROW} \times \mathrm{NSCOL} \times \mathrm{NSLAY}.$$

NPTPND Initial number of particles per cell in transport simulation (that is, at $t = 0.0$). Valid options for default geometry of particle placement include 1, 2, 3, or 4 for one-dimensional transport simulation; 1, 4, 9, or 16 for two-dimensional transport simulation; and 1, 8, or 27 for three-dimensional transport simulation. The user can also customize initial placement of particles by specifying NPTPND as a negative number, in which case the minus sign is recognized as a flag to indicate custom placement is desired. In this case, the user must input local particle coordinates as described below.

IF *NPTPND* IS NEGATIVE IN SIGN:

6. Data: PNEWL PNEWR PNEWC

PNEWL Relative position in the layer (z) direction for initial placement of particle within any finite-difference cell.

PNEWR Relative position in the row (y) direction for initial placement of particle.

PNEWC Relative position in the column (x) direction for initial placement of particle.

Notes:

The three new (or initial) particle coordinates are entered sequentially for each of the NPTPND particles. Each line contains the three relative local coordinates for the new particles, in order of layer, row, and column. There must be NPTPND lines of data, one for each particle. The local coordinate system range is from -0.5 to 0.5, and represents the relative distance within the cell about the node location at the center of the cell, so that the node is located at 0.0 in each direction.

FOR EACH SIMULATION:

7. Data: CELDIS FZERO INTRPL

CELDIS Maximum fraction of cell dimension that particle may move in one step (typically, $0.5 \le \mathrm{CELDIS} \le 1.0$).

FZERO If the fraction of active cells having no particles exceeds FZERO, the program will automatically regenerate an initial particle distribution before continuing the simulation (typically, $0.01 \le \mathrm{FZERO} \le 0.05$).

INTRPL Flag for interpolation scheme used to estimate velocity of particles. The default (INTRPL=1) will use a linear interpolation routine; if INTRPL=2, a scheme will be implemented that uses bilinear interpolation in the row and column (j and i) directions only (linear interpolation will still be applied in the k, or layer, direction). (See section “Discussion—Choosing appropriate interpolation scheme.”)

FOR EACH SIMULATION:

```
 8.  Data:    NPNTCL  ICONFM  NPNTVL  IVELFM  NPNTDL  IDSPFM  NPNTPL
```

NPNTCL Flag for printing concentration data. If NPNTCL=-2, concentration data will be printed at the end of every stress period; if NPNTCL=-1, data will be printed at the end of every flow time step; if NPNTCL=0, data will be printed at the end of the simulation; if NPNTCL=N>0, data will be printed every Nth particle moves, and at the end of the simulation. Initial concentrations are always printed.

ICONFM Specification for format of concentration data in main output file (see Table 17 and *MODFLOW* documentation on array-reading utility modules).

NPNTVL Flag for printing velocity data. If NPNTVL=-1, velocity data will be printed at the end of every stress period; if NPNTVL=0, data will be printed at the end of the simulation; if NPNTVL=N>0, data will be printed every Nth flow time steps, and at the end of the simulation.

IVELFM Specification for format of velocity data, if being printed in main output file (see Table 17).

NPNTDL Flag for printing dispersion equation coefficients that include cell dimension factors (see section "Program Segments"). If NPNTDL=-2, coefficients will be printed at the end of every stress period; if NPNTDL=-1, coefficients will be printed at the end of the simulation; if NPNTDL=0, coefficients will not be printed; if NPNTDL=N>0, coefficients will be printed every Nth flow time step.

IDSPFM Specification for format of dispersion equation coefficients (see Table 17).

NPNTPL Flag for printing particle locations in a separate output file (only used if file types "PRTA" or "PRTB" appear in the *MOC3D* name file). If neither "PRTA" or "PRTB" is entered in the name file, NPNTPL will be read but ignored (so you must always have some value specified here). If either "PRTA" or "PRTB" is entered in the name file, initial particle locations will be printed to the separate file first, followed by particle data at intervals determined by the value of NPNTPL. If NPNTPL=-2, particle data will be printed at the end of every stress period; if NPNTPL=-1, data will be printed at the end of every flow time step; if NPNTPL=0, data will be printed at the end of the simulation; if NPNTPL=N>0, data will be printed every Nth particle moves, and at the end of the simulation.

Table 17. Formats associated with *MOC3D* print flags

Print flag	Format	Print flag	Format	Print flag	Format
0	10G11.4	7	20F5.0	14	10F6.1
1	11G10.3	8	20F5.1	15	10F6.2
2	9G13.6	9	20F5.2	16	10F6.3
3	15F7.1	10	20F5.3	17	10F6.4
4	15F7.2	11	20F5.4	18	10F6.5
5	15F7.3	12	10G11.4		
6	15F7.4	13	10F6.0		

FOR EACH SIMULATION:

9. Data: CNOFLO Concentration associated with inactive cells of subgrid (used for output purposes only).

FOR EACH LAYER OF THE TRANSPORT SUBGRID:

10. Data: CINT(NSCOL,NSROW) Initial concentration.
Module: U2DREL*

FOR EACH SIMULATION, ONLY IF TRANSPORT SUBGRID DIMENSIONS ARE SMALLER THAN FLOW GRID DIMENSIONS:

11. Data: CINFL(ICINFL) C' to be associated with fluid inflow across the boundary of the subgrid.
Module: U1DREL*

Notes:

The model assumes that the concentration outside of the subgrid is the same within each layer, so only one value of CINFL is specified for each layer within and adjacent to the subgrid. That is, the size of the array (ICINFL) is determined by the position of the subgrid with respect to the entire (primary) *MODFLOW* grid. If the transport subgrid has the same dimensions as the flow grid, this parameter should not be included in the input data set. If the subgrid and flow grid have the same number of layers, but the subgrid has fewer rows or fewer columns, ICINFL=NSLAY. Values are also required if there is a flow layer above the subgrid and/or below the subgrid. The order of input is: C' for first (uppermost) transport layer (if required); C' for each successive (deeper) transport layer (if required); C' for layer above subgrid (if required); and C' for layer below subgrid (if required).

FOR EACH SIMULATION

12. Data: NZONES Number of zone codes among fixed-head nodes in transport subgrid.

IF *NZONES* > 0:

Data: IZONE ZONCON

IZONE Value identifying a particular zone.

ZONCON Source concentration associated with nodes in the zone defined by IZONE above.

Notes:

Zones are defined within the IBOUND array in the BAS Package of *MODFLOW* by specifying unique negative values for fixed-head nodes to be associated with separate fluid source concentrations. Each zone is defined by a unique value of IZONE and a concentration associated with it (ZONCON). There must be NZONES lines of data, one for each zone. Note that values of IZONE in this list must be negative for consistency with the definitions of fixed-head nodes in the IBOUND array in the BAS Package. If a negative value of IBOUND is defined in the BAS package but is not assigned a concentration value here, *MOC3D* will assume that the source concentrations associated with those nodes equal 0.0.

* Module is a standard *MODFLOW* input/output module.

FOR EACH LAYER OF THE TRANSPORT SUBGRID:

13. Data: IGENPT(NSCOL,NSROW) Flag to treat fluid sources and sinks as either “strong” or “weak.”
Module: U2DINT*

Notes:

Where fluid source is “strong,” new particles are added to replace old particles as they are advected out of that cell. Where a fluid sink is “strong,” particles are removed after they enter that cell and their effect accounted for. Where sources or sinks are weak, particles are neither added nor removed, and the source/sink effects are incorporated directly into appropriate changes in particle positions and concentrations. If IGENPT=0, the node will be considered a weak source or sink; if IGENPT=1, it will be a strong source or sink. See section on “Special Problems” and discussion by Konikow and Bredehoeft (1978).

IF *NODISP* ≠ 1 (If dispersion is included in simulation):

14. Data: ALONG(NSLAY) Longitudinal dispersivity.
Module: U1DREL*

15. Data: ATRANH(NSLAY) Horizontal transverse dispersivity.
Module: U1DREL*

16. Data: ATRANV(NSLAY) Vertical transverse dispersivity.
Module: U1DREL*

FOR EACH SIMULATION:

17. Data: RF(NSLAY) Retardation factor (RF=1 indicates no retardation).
Module: U1DREL*

Notes:

If RF=0.0 in input, the code automatically resets it as RF=1.0 to indicate no retardation.

FOR EACH LAYER OF TRANSPORT SUBGRID:

18a. Data: THCK(NSCOL,NSROW) Cell thickness.
Module: U2DREL*

18b. Data: POR(NSCOL,NSROW) Cell porosity.
Module: U2DREL*

Notes:

The thickness and porosity are input as separate arrays for each layer of the transport subgrid. The sequence used in data set 18 is to first define the thickness of the first layer of the transport subgrid, and then define the porosity of that same layer. Next, that sequence is repeated for all succeeding layers. The product of thickness and porosity should not be allowed to vary greatly among cells in the transport subgrid.

* Module is a standard *MODFLOW* input/output module.

Source Concentration in Recharge File (CRCH)

Concentrations in recharge, if the recharge package is used, are read from a separate unit specified in the *MOC3D* name file. This is defined using the file type (Ftype) "CRCH."

FOR EACH STRESS PERIOD, IF RECHARGE PACKAGE USED:

1. Data: INCRCH Flag to reuse or read new recharge concentrations.

Notes:

Read new recharge concentrations if INCRCH ≥ 0. Reuse recharge concentrations from the last stress period if INCRCH < 0.

2. Data: CRECH(NSCOL,NSROW) Source concentration associated with fluid entering the aquifer in recharge.

Module: U2DREL*

Observation Well File (OBS)

Nodes of the transport subgrid can be designated as "observation wells." At each such node, the time, head, and concentration after each move increment will be written to a separate output file to facilitate graphical postprocessing of the calculated data. The input file for specifying observation wells is read if the file type (Ftype) "OBS" is included in the *MOC3D* name file.

FOR EACH SIMULATION, IF *OBS* PACKAGE USED:

1. Data: NUMOBS IOBSFL

NUMOBS Number of observation wells.

IOBSFL If IOBSFL = 0, well data are saved in NUMOBS separate files. If IOBSFL>0, all observation well data will be written to one file, and the file name and unit number used for this file will be that of the first observation well in the list.

FOR EACH OBSERVATION WELL:

2. Data: LAYER ROW COLUMN UNIT

LAYER Layer of observation well node.

ROW Row of observation well node.

COLUMN Column of observation well node.

UNIT Unit number for output file.

Notes:

If NUMOBS>1 and IOBSFL = 0, you must specify a unique unit number for each observation well and match those unit numbers to DATA file types and file names in the *MOC3D* name file. If IOBSFL>0, you must specify a unique unit number for the first observation well and match that unit number to a DATA file type and file name in the *MOC3D* name file.

* Module is a standard *MODFLOW* input/output module.

APPENDIX C: ANNOTATED EXAMPLE INPUT DATA SET FOR SAMPLE PROBLEM

This example input data set is the one used to generate the solution for the base case in the one-dimensional steady-state flow problem. Parameter values are indicated in Table 11 and selected results are shown in fig. 18. Several of the following data files (*modflow.nam*, *bas95.dat*, *bcf11.dat*, and *sip19.dat*) are those required for *MODFLOW-96*, and their formats are described by Harbaugh and McDonald (1996a).

In several of the data files shown below, the right side of some data lines includes a semi-colon followed by text that describes the parameters for which values are given. These comments (including the semi-colon) are not read by the program because in free format the code will only read the proper number of variables and ignore any subsequent information on that line. This style of commenting data files is optional, but users may find it helpful when viewing the content of data files.

Following (enclosed in a border) are the contents of the *MODFLOW* name file for the sample problem; explanations are noted outside of border:

File name: `modflow.nam`

```
list     16      flow.out     ← Designates main output file for MODFLOW
bas      95      bas95.dat    ← Basic input data for MODFLOW
bcf      11      bcf11.dat    ← Block-centered flow package
sip      19      sip19.dat    ← Input for numerical solution of flow equation
conc     33      moc.nam      ← Transport name file (turns transport "on")
 ↑        ↑        ↑
 1        2        3
```

[1] Ftype (that is, the type of file)

[2] Unit number

[3] File name (name chosen to reflect contents of file)

Following (enclosed in a border) are the contents of the basic package input data file for the *MODFLOW* simulation of the sample problem; explanations are noted outside of border:

File name: `bas95.dat`

```
Finite:  Compare to Wexler program and MOC3D                        BAS Input      ← 1
      NLAY       NROW       NCOL       NPER     ITMUNI                             ← 1
         1          1        122          1          1                             ← 2
FREE                                                                               ← 3
         0          1      ; IAPART,ISTRT                                          ← 4
        95          1(25I3)                                   3    ; IBOUND        ← 5
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1        ← 5
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1        ← 5
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1        ← 5
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1        ← 5
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -2                 ← 5
      0.00                                                 ; HNOFLO                ← 6
        95         1.0(122F5.0)                      1     ; HEAD                  ← 7
  12.1                                                                             ← 7
  120.0             1          1.      ; PERLEN,NSTP,TSMULT                        ← 8
end                                                                                ← 9
```

[1] Two header lines of comments. For convenience and clarity, the second line is used to label names of parameters on subsequent line of file.

[2] Flow grid dimensions, number of periods, and time units.

[3] Options line (new in *MODFLOW-96*)

[4] Flags for buffer array and drawdown calculations.

[5] *IBOUND* identifiers (first line) and array

[6] Head value assigned to inactive cells

[7] Initial head information

[8] *MODFLOW* time-step information

[9] Final comment line

Following (enclosed in a border) are the contents of the block-centered flow package input data file; explanations are noted outside of border:

File name: `bcf11.dat`

```
  1  0    0.0 0 0.0 0 0     ; ISS, flags      BCF Input   ← 1
          0                 ; LAYCON                      ← 2
          0        1.0      ; TRPY                        ← 3
          0        0.1      ; DELR                        ← 4
          0        0.1      ; DELC                        ← 4
          0       0.01      ; TRAN                        ← 5
```

[1] Flag for steady-state flow, flag for cell-by-cell flow terms, five flags related to wetting

[2] Layer type

[3] Anisotropy factor

[4] Grid spacing information

[5] Transmissivity data

Following (enclosed in a border) are the contents of the strongly implicit procedure package input data file; explanations are noted outside of border:

File name: `sip19.dat`

```
     500          5      ; MXITER,NPARM            SIP Input          ← 1
       1. 0.0000001       0       0.001 ; ACCL,ERR,IPCALC,WSEED       ← 2
```

[1] Maximum iterations, number of iteration parameters

[2] Acceleration parameter, head change criterion, flag for seed, seed

Following (enclosed in a border) are the contents of the *MOC3D* name file for the sample problem; explanations are noted outside of border:

File name: `moc.nam`

```
clst     97       moc.out      ← Designates main output file for MOC3D
moc      96       moc96.dat    ← Main input data file for MOC3D
obs      44       obs44.dat    ← Input data file for observation wells
data     45       obs.out      ← Output file for observation well data
cnca     22       conc.txt     ← Separate output file for conc. data (ASCII)
cncb     23       conc.bin     ← Separate output file for conc. data (binary)
 ↑        ↑         ↑
 1        2         3
```

[1] Ftype

[2] Unit number

[3] File name

Following (enclosed in a border) are the contents of the main input data file for the *MOC3D* simulation for the sample problem; selected explanations are noted outside of border:

File name: `moc96.dat`

```
One-dimensional, Steady Flow, No Decay, Low Dispersion: BASE CASE  MOC3D Input   ← 1
   ISLAY1     ISLAY2     ISROW1     ISROW2     ISCOL1     ISCOL2                 ← 1
        1          1          1          1          2        121                 ← 2
        0        0.0        0.0 ; NODISP, DECAY, DIFFUS                          ← 3
     2000          3            ; NPMAX, NPTPND                                  ← 4
      0.5       0.05          2 ; CELDIS, FZERO, INTRPL                          ← 4
  0 0 0 -1 0 0 0 ; NPNTCL, ICONFM, NPNTVL, IVELFM, NPNTDL, IDSPFM, NPRTPL         ← 5
      0.0        ; CNOFLO                                                        ← 6
        0        0.0 (122F3.0)                    ; initial concentration
        0        1.                               ; C' inflow
        2                                         ; NZONES to follow             ← 7
       -1        1.0                              ; IZONE, ZONCON                ← 7
       -2        0.0                              ; IZONE, ZONCON                ← 7
        0          0                              ; IGENPT                       ← 8
        0        0.1                              ; longitudinal disp.
        0        0.1                              ; transverse disp. horiz.
        0        0.1                              ; transverse disp. vert.
        0        1.0                              ; retardation factor
        0        1.0                              ; thickness
        0        0.1                              ; porosity
```

[1] Two header lines of comments. For convenience and clarity, the second line is used to label names of parameters on subsequent line of file.

[2] Indices for transport subgrid

[3] Flag for no dispersion, decay rate, diffusion coefficient

[4] Particle information for advective transport

[5] Print flags

[6] Value of concentration associated with inactive cells

[7] Concentrations associated with fixed-head nodes (fixed head nodes are defined in the IBOUND array in the *MODFLOW* BAS package)

[8] Flag for "strong" sources or sinks

Following (enclosed in a border) are the contents of the observation well input data file for the sample problem; explanations are noted outside of border:

File name: `obs44.dat`

```
 3  1                ;NUMOBS   IOBSFL   Observation well data   ← 1
 1  1   2  45        ;layer, row, column, unit number           ← 2
 1  1  42            ;layer, row, column                        ← 2
 1  1 112            ;layer, row, column                        ← 2
```

[1] Number of observation wells, flag to print to one file or separate files

[2] Node location and unit number for output file (linked to the Ftype DATA in *MOC3D* name file)

APPENDIX D: SELECTED OUTPUT FOR SAMPLE PROBLEM

This example output was generated from the input data sets listed in Appendix C for the base case of the one-dimensional steady-state flow problem. The line spacing and font sizes of the output files have been modified in places to enhance the clarity of reproduction in this report. Some repetitive lines of output have been deleted where indicated by an ellipsis (...).

Some brief annotations were added in a few places in this sample output listing to help the reader understand the purpose of various sections of output. These annotations are written in bold italics to clarify that they are not part of the output file.

Following are the contents of the *MOC3D* main output file for the sample problem.

```
              U.S. GEOLOGICAL SURVEY
 METHOD-OF-CHARACTERISTICS SOLUTE TRANSPORT MODEL
           MOC3D (Version 1.0) 11/08/96

 MOC BASIC INPUT READ FROM UNIT
 LISTING FILE: moc.out    UNIT  97

 OPENING moc96.dat
 FILE TYPE: MOC    UNIT  96

 OPENING obs44.dat
 FILE TYPE: OBS    UNIT  44

 OPENING obs.out
 FILE TYPE: DATA    UNIT  45

 OPENING conc.txt
 FILE TYPE: CNCA    UNIT  22

 OPENING conc.bin
 FILE TYPE: CNCB    UNIT  23
```

FILE INFORMATION

```
 MOC BASIC INPUT READ FROM UNIT  96
```

2 TITLE LINES:

```
 One-dimensional, Steady Flow, No Decay, Low Dispersion: BASE CASE  MOC3D Input
     ISLAY1     ISLAY2     ISROW1     ISROW2     ISCOL1     ISCOL2
```

PROBLEM DESCRIPTORS, INCLUDING GRID CHARACTERISTICS AND PARTICLE INFORMATION:

```
      MAPPING OF SOLUTE TRANSPORT SUBGRID IN FLOW GRID:
 FIRST LAYER FOR SOLUTE TRANSPORT =   1       LAST LAYER FOR SOLUTE TRANSPORT  =   1
 FIRST ROW FOR SOLUTE TRANSPORT   =   1       LAST ROW FOR SOLUTE TRANSPORT    =   1
 FIRST COLUMN FOR SOLUTE TRANSPORT=   2       LAST COLUMN FOR SOLUTE TRANSPORT = 121

 UNIFORM DELCOL AND DELROW IN SUBGRID FOR SOLUTE TRANSPORT

 NO. OF LAYERS =    1    NO. OF ROWS =    1    NO. OF COLUMNS =  120
 NO SOLUTE DECAY
 NO MOLECULAR DIFFUSION
 MAXIMUM NUMBER OF PARTICLES (NPMAX) =     2000
```

```
    14006 ELEMENTS IN X ARRAY ARE USED BY MOC
       12 ELEMENTS IN X ARRAY ARE USED BY OBS

 NUMBER OF PARTICLES INITIALLY IN EACH ACTIVE CELL (NPTPND) =   3
 PARTICLE MAP ("o" indicates particle location; shown as
              fractions of cell distances relative to node location):

       o------o------o

     -1/3     0     1/3

 INITIAL RELATIVE PARTICLE COORDINATES
   1    0.00000   0.00000  -0.33333
   2    0.00000   0.00000   0.00000
   3    0.00000   0.00000   0.33333

 CELDIS=     0.500
 FZERO =     0.050


INTRPL= 2;  BILINEAR INTERPOLATION SCHEME
```

OUTPUT CONTROL

```
NPNTCL=  0:     CONCENTRATIONS WILL BE WRITTEN AT THE END OF THE SIMULATION
MODFLOW FORMAT SPECIFIER FOR CONCENTRATION DATA: ICONFM=  0

NPNTVL=  0:         VELOCITIES WILL BE WRITTEN AT THE END OF THE SIMULATION
MODFLOW FORMAT SPECIFIER FOR      VELOCITY DATA: IVELFM= -1

NPNTDL=  0: DISP. COEFFICIENTS WILL NOT BE WRITTEN

NPNTPL=  0: PARTICLE LOCATIONS WILL NOT BE WRITTEN

CONCENTRATION WILL BE SET TO 0.00000E+00 AT ALL NO-FLOW NODES (IBOUND=0).
```

INITIAL AND BOUNDARY CONDITIONS FOR SOLUTE

```
    INITIAL CONCENTRATION =  0.0000000E+00 FOR LAYER   1


VALUES OF C' REQUIRED FOR SUBGRID BOUNDARY ARRAY =    1
ONE FOR EACH LAYER IN TRANSPORT SUBGRID

ORDER OF C' VALUES: FIRST LAYER IN SUBGRID, EACH SUBSEQUENT LAYER,
LAYER ABOVE SUBGRID, LAYER BELOW SUBGRID:

  SUBGRID BOUNDARY ARRAY  =   1.000000


 NUMBER OF ZONES FOR CONCENTRATIONS AT FIXED HEAD CELLS =    2

 ZONE FLAG =   -1     INFLOW CONCENTRATION =   1.0000E+00
 ZONE FLAG =   -2     INFLOW CONCENTRATION =   0.0000E+00
```

```
         SINK-SOURCE FLAG =              0 FOR LAYER   1

 LONGITUDNL. DISPERSIVITY =  0.1000000

  HORIZ. TRANSVERSE DISP. =  0.1000000

   VERT. TRANSVERSE DISP. =  0.1000000

        RETARDATION FACTOR =   1.000000

         INITIAL THICKNESS =   1.000000     FOR LAYER   1

          INITIAL POROSITY =  0.1000000     FOR LAYER   1
```

```
COORDINATES FOR   3 OBSERVATION WELLS:

  WELL #   LAYER     ROW  COLUMN    UNIT
       1       1       1       2      45
       2       1       1      42      45
       3       1       1     112      45
ALL OBSERVATION WELL DATA WILL BE WRITTEN ON UNIT  45

CONCENTRATION DATA WILL BE SAVED ON UNIT  22 IN ASCII FORMAT
CONCENTRATION DATA WILL BE SAVED ON UNIT  23 IN BINARY FORMAT

 TOTAL NUMBER OF PARTICLES GENERATED (GENPT) =       360
 TOTAL NUMBER OF ACTIVE NODES (NACTIV) =       120
 MAX. NUMBER OF CELLS THAT CAN BE VOID OF PARTICLES (NZCRIT) =       6
     (IF NZCRIT EXCEEDED, PARTICLES ARE REGENERATED)
```

CALCULATED VELOCITIES (INCLUDING EFFECTS OF RETARDATION, IF PRESENT):

```
EFFECTIVE MEAN SOLUTE VELOCITIES IN COLUMN DIRECTION
                          AT NODES
1
  VELOCITY (COL)   IN LAYER  1 AT END OF TIME STEP  1 IN STRESS PERIOD  1
  -----------------------------------------------------------------------

         1         2         3         4         5         6         7         8         9   ...
 ..............................................................................................
   1  9.917E-02 9.917E-02 9.917E-02 9.917E-02 9.917E-02 9.917E-02 9.917E-02 9.917E-02 9.917E-02 ...

...

       111       112       113       114       115       116       117       118       119       120
 ..............................................................................................
   1  9.917E-02 9.917E-02 9.917E-02 9.917E-02 9.917E-02 9.917E-02 9.917E-02 9.917E-02 9.917E-02 9.917E-02

EFFECTIVE MEAN SOLUTE VELOCITIES IN ROW DIRECTION
                          AT NODES
1
  VELOCITY (ROW)   IN LAYER  1 AT END OF TIME STEP  1 IN STRESS PERIOD  1
  -----------------------------------------------------------------------

         1         2         3         4         5         6         7         8         9   ...
 ..............................................................................................
   1  0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 ...

...

       111       112       113       114       115       116       117       118       119       120
 ..............................................................................................
   1  0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00

EFFECTIVE MEAN SOLUTE VELOCITIES IN LAYER DIRECTION
                          AT NODES
1
  VELOCITY (LAYER) IN LAYER  1 AT END OF TIME STEP  1 IN STRESS PERIOD  1
  -----------------------------------------------------------------------

         1         2         3         4         5         6         7         8         9   ...
 ..............................................................................................
   1  0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 ...

...

       111       112       113       114       115       116       117       118       119       120
 ..............................................................................................
   1  0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
```

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

     MAXIMUM FLUID VELOCITIES:  C-VEL =  9.92E-02     R-VEL =  1.00E-20     L-VEL =  1.00E-18
MINIMUM TIME TO TRAVEL THCK =  1.00E+18

TIMV =  5.04E-01     NTIMV  =     239

     MAX. C-VEL. IS CONSTRAINT AND OCCURS BETWEEN NODES (    2,    1,    1) AND (    1,    1,    1)

TIMD =  5.04E-01     NTIMD  =     239

   THERE ARE NO FLUID SOURCES IN THE TRANSPORT SUBGRID

NUMBER OF MOVES FOR ALL STABILITY CRITERIA:
   CELDIS  DISPERSION   INJECTION
      239         239           1

CELDIS IS LIMITING
DISPERSION IS LIMITING

         NO. OF PARTICLE MOVES REQUIRED TO COMPLETE THIS TIME STEP  =  239
           MOVE TIME STEP (TIMV)=  5.020920634270E-01
```

(NUMERICAL SOLUTION TO TRANSPORT EQUATION STARTS AT THIS POINT)

```
 NP      =     360 AT START OF MOVE        IMOV     =          1
 NP      =     360 AT START OF MOVE        IMOV     =          2
 NP      =     360 AT START OF MOVE        IMOV     =          3
 NP      =     360 AT START OF MOVE        IMOV     =          4
 NP      =     360 AT START OF MOVE        IMOV     =          5
 NP      =     360 AT START OF MOVE        IMOV     =          6
...
 NP      =     360 AT START OF MOVE        IMOV     =        234
 NP      =     360 AT START OF MOVE        IMOV     =        235
 NP      =     360 AT START OF MOVE        IMOV     =        236
 NP      =     360 AT START OF MOVE        IMOV     =        237
 NP      =     360 AT START OF MOVE        IMOV     =        238
 NP      =     360 AT START OF MOVE        IMOV     =        239
```

ONE LINE PRINTED FOR EACH MOVE TO TRACK PROGRESS AND NUMBER OF ACTIVE PARTICLES

```
        SOLUTE BUDGET AND MASS BALANCE FOR TRANSPORT SUBGRID

   VALUES CALCULATED AT END OF:
            STRESS PERIOD    1  OUT OF    1
           FLOW TIME STEP    1  OUT OF    1
 TRANSPORT TIME INCREMENT  239  OUT OF  239

   ELAPSED TIME =  1.2000E+02

   CHEMICAL MASS IN STORAGE:
       INITIAL:   MASS DISSOLVED =  0.0000E+00     MASS SORBED =  0.0000E+00
       PRESENT:   MASS DISSOLVED =  1.1341E-01     MASS SORBED =  0.0000E+00

            CHANGE IN MASS STORED = -1.1341E-01



  CUMULATIVE SOLUTE MASS  (L**3)(M/VOL)
  ----------------------

      IN:
      ---
                  DECAY =  0.0000E+00
          CONSTANT HEAD =  0.0000E+00
       SUBGRID BOUNDARY =  1.1901E-01
               RECHARGE =  0.0000E+00
                  WELLS =  0.0000E+00
                 RIVERS =  0.0000E+00
                 DRAINS =  0.0000E+00
  GENL. HEAD-DEP. BDYS. =  0.0000E+00
     EVAPOTRANSPIRATION =  0.0000E+00

               TOTAL IN =  1.1901E-01


     OUT:
     ----
                  DECAY =  0.0000E+00
          CONSTANT HEAD =  0.0000E+00
       SUBGRID BOUNDARY = -5.6659E-03
               RECHARGE =  0.0000E+00
                  WELLS =  0.0000E+00
                 RIVERS =  0.0000E+00
                 DRAINS =  0.0000E+00
  GENL. HEAD-DEP. BDYS. =  0.0000E+00
     EVAPOTRANSPIRATION =  0.0000E+00

              TOTAL OUT = -5.6659E-03


      SOURCE-TERM DECAY =  0.0000E+00


               RESIDUAL = -6.7927E-05

    PERCENT DISCREPANCY = -0.5708E-01 RELATIVE TO MASS FLUX IN
```

ITEMIZED BUDGETS FOR SOLUTE FLUXES

Following (enclosed in a border) are the abridged contents of the observation well output file for the sample problem. This output file was generated using the option to write all observation well data to a single file (IOBSFL = 1).

File name: *obs.out*

```
"OBSERVATION WELL DATA"
"TIME, THEN HEAD AND CONC. FOR EACH OBS. WELL AT NODE (K,I,J)"
"  TIME:       H & C AT    1,  1,  2    H & C AT    1,  1, 42    H & C AT    1,  1,112 "
 5.0209E-01   1.190E+01   2.503E-01   7.934E+00   0.000E+00   9.917E-01   0.000E+00
 1.0042E+00   1.190E+01   6.539E-01   7.934E+00   0.000E+00   9.917E-01   0.000E+00
 1.5063E+00   1.190E+01   5.994E-01   7.934E+00   0.000E+00   9.917E-01   0.000E+00
 2.0084E+00   1.190E+01   7.914E-01   7.934E+00   0.000E+00   9.917E-01   0.000E+00
 2.5105E+00   1.190E+01   7.747E-01   7.934E+00   0.000E+00   9.917E-01   0.000E+00
 3.0126E+00   1.190E+01   8.578E-01   7.934E+00   0.000E+00   9.917E-01   0.000E+00
...
...
 1.1849E+02   1.190E+01   1.000E+00   7.934E+00   1.000E+00   9.917E-01   6.835E-01
 1.1900E+02   1.190E+01   1.000E+00   7.934E+00   1.000E+00   9.917E-01   6.900E-01
 1.1950E+02   1.190E+01   1.000E+00   7.934E+00   1.000E+00   9.917E-01   7.045E-01
 1.2000E+02   1.190E+01   1.000E+00   7.934E+00   1.000E+00   9.917E-01   7.125E-01
```

Following (enclosed in a border) are the partial contents of the separate output file for concentration in ASCII format. Initial concentrations are abridged; complete set of final concentrations are shown.

File name: *conc.txt*

```
CONCENTRATIONS AT NODES IN SUBGRID.  IMOV=     0, NSTP=     0, NPER=     1, SUMTCH=0.0000E+00
 SUBGRID LAYER           1
 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
...
 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

CONCENTRATIONS AT NODES IN SUBGRID.  IMOV=   239, NSTP=     1, NPER=     1, SUMTCH=1.2000E+02
 SUBGRID LAYER           1
 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 9.9999E-01
 9.9999E-01 9.9999E-01 9.9999E-01 9.9998E-01 9.9998E-01 9.9997E-01 9.9997E-01 9.9996E-01 9.9994E-01 9.9993E-01
 9.9990E-01 9.9988E-01 9.9984E-01 9.9980E-01 9.9975E-01 9.9968E-01 9.9960E-01 9.9950E-01 9.9938E-01 9.9922E-01
 9.9904E-01 9.9882E-01 9.9854E-01 9.9821E-01 9.9782E-01 9.9735E-01 9.9679E-01 9.9611E-01 9.9532E-01 9.9439E-01
 9.9330E-01 9.9201E-01 9.9050E-01 9.8877E-01 9.8678E-01 9.8449E-01 9.8182E-01 9.7876E-01 9.7534E-01 9.7149E-01
 9.6710E-01 9.6207E-01 9.5647E-01 9.5035E-01 9.4358E-01 9.3594E-01 9.2738E-01 9.1810E-01 9.0818E-01 8.9737E-01
 8.8536E-01 8.7219E-01 8.5826E-01 8.4369E-01 8.2811E-01 8.1104E-01 7.9266E-01 7.7364E-01 7.5433E-01 7.3423E-01
 7.1254E-01 6.8920E-01 6.6520E-01 6.4163E-01 6.1868E-01 5.9552E-01 5.7147E-01 5.4711E-01 5.2527E-01 5.1194E-01
```