Contract Report SERDP-99-1
December 1999

US Army Corps of Engineers
Engineer Research and Development Center

Strategic Environmental Research and Development Program

MT3DMS: A Modular Three-Dimensional Multispecies Transport Model for Simulation of Advection, Dispersion, and Chemical Reactions of Contaminants in Groundwater Systems; Documentation and User's Guide

by Chunmiao Zheng, P. Patrick Wang
University of Alabama

SERDP
Strategic Environmental Research and Development Program
Improving Mission Readiness through Environmental Research

Prepared for Headquarters, U.S. Army Corps of Engineers

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.


PRINTED ON RECYCLED PAPER

Strategic Environmental Research and Development Program

Contract Report SERDP-99-1
December 1999

MT3DMS: A Modular Three-Dimensional Multispecies Transport Model for Simulation of Advection, Dispersion, and Chemical Reactions of Contaminants in Groundwater Systems; Documentation and User's Guide

by Chunmiao Zheng, P. Patrick Wang

Department of Geological Sciences
University of Alabama
Tuscaloosa, AL 35487

Final report

Approved for public release; distribution is unlimited

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under Work Unit No. CU-1062

Monitored by Environmental Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

U.S. Army Engineer Research and Development Center Cataloging-in-Publication Data

Zheng, Chunmiao, 1962-.

MT3DMS: a modular three-dimensional multispecies transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems : documentation and user's guide / by Chunmiao Zheng, P. Patrick Wang ; prepared for U.S. Army Corps of Engineers ; monitored by U.S. Army Engineer Research and Development Center.

221 p. : ill. ; 28 cm — (Contract report ; SERDP-99-1)

Includes bibliographic references.

1. MT3DMS (Computer program) 2. Groundwater flow — Computer programs. 3. Groundwater — Mathematical models. I. Wang, P. Patrick. II. United States. Army. Corps of Engineers. III. U.S. Army Engineer Research and Development Center. IV. Strategic Environmental Research and Development Program (U.S.) V. Title. VI. Title: Modular three-dimensional multispecies transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems : documentation and user's guide. VII. Series: Contract report SERDP ; 99-1.
TA7 E8c no.SERDP-99-1

Contents

List of Figures

List of Tables

Preface

The work reported herein was conducted under the Cleanup Pillar of the Strategic Environmental Research and Development Program (SERDP) as part of Work Unit CU-1062, "Development of Simulators for In-Situ Remediation Evaluation, Design, and Operation." SERDP is sponsored by the U.S. Department of Defense, the U.S. Environmental Protection Agency, and the U.S. Department of Energy. Work Unit CU-1062 was conducted by the U.S. Army Engineer Research and Development Center (ERDC), Waterways Experiment Station (WES), under the purview of the Environmental Laboratory (EL). The WES involvement in SERDP was coordinated by Dr. John Cullinane, Program Manager for the EL Installation Restoration Research Program (IRRP). Mr. Charles Miller was Assistant Manager for the IRRP. Dr. Femi Ayorinde and Ms. Catherine Vogel were the Cleanup Program Managers for SERDP.

The Principal Investigator of Work Unit CU-1062 was Dr. Mark S. Dortch, Chief, Water Quality and Contaminant Modeling Branch (WQCMB), Environmental Processes and Effects Division (EPED), EL. The work reported herein was conducted by Drs. Chunmiao Zheng and P. Patrick Wang, University of Alabama, for WES under contract DACA 39-96-K-0016. This report was prepared by Drs. Zheng and Wang. Dr. Dortch monitored the contract. Dr. Dortch and Drs. Carlos Ruiz and Mansour Zakakihani, WQCMB, reviewed this report.

The study was conducted under the direct supervision of Dr. Richard E. Price, Chief, EPED, and under the general supervision of Dr. John Keeley, Director, EL.

At the time of publication of this report, Dr. Lewis E. Link was Acting Director of ERDC, and COL Robin R. Cababa, EN, was Commander.

This report should be cited as follows:

Zheng, Chunmiao, and Wang, P. Patrick. (1999). "MT3DMS: A modular three-dimensional multispecies transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems; documentation and user's guide," Contract Report SERDP-99-1, U.S. Army Engineer Research and Development Center, Vicksburg, MS.

1 Introduction

Purpose and Scope

The modular three-dimensional (3-D) transport model referred to as MT3D was originally developed by Zheng (1990) at S. S. Papadopulos & Associates, Inc., and subsequently documented for the Robert S. Kerr Environmental Research Laboratory of the U.S. Environmental Protection Agency. In the past several years, various versions of the MT3D code have been commonly used in contaminant transport modeling and remediation assessment studies. This manual describes the next generation of MT3D with significantly expanded capabilities, including the addition of (a) a third-order total-variation-diminishing (TVD) scheme for solving the advection term that is mass conservative but does not introduce excessive numerical dispersion and artificial oscillation; (b) an efficient iterative solver based on generalized conjugate gradient methods to remove stability constraints on the transport time stepsize; (c) options for accommodating nonequilibrium sorption and dual-domain advection-diffusion mass transport; and (d) a multicomponent program structure that can accommodate add-on reaction packages for modeling general biological and geochemical reactions.

Key Features

The new mass transport model documented in this manual is referred to as MT3DMS, where MT3D stands for the Modular 3-Dimensional Transport model, and MS denotes the Multi-Species structure for accommodating add-on reaction packages. MT3DMS has a comprehensive set of options and capabilities for simulating advection, dispersion/diffusion, and chemical reactions of contaminants in groundwater flow systems under general hydrogeologic conditions. This section summarizes the key features of MT3DMS.

MT3DMS is unique in that it includes three major classes of transport solution techniques (the standard finite-difference method, the particle-tracking-based Eulerian-Lagrangian methods, and the higher-order finite-volume TVD method) in a single code. Since no single numerical technique has been effective for all

transport conditions, the combination of these solution techniques, each having its own strengths and limitations, is believed to offer the best approach for solving the most wide-ranging transport problems with efficiency and accuracy.

In addition to the explicit formulation of the original MT3D code, MT3DMS includes an implicit formulation that is solved with an efficient and versatile solver. The iterative solver is based on generalized conjugate gradient (GCG) methods with three preconditioning options and the Lanczos/ORTHOMIN acceleration scheme for nonsymmetrical matrices. If the GCG solver is selected, dispersion, sink/source, and reaction terms are solved implicitly without any stability constraints. For the advection term, the user has the option to select any of the solution schemes available, including the standard finite-difference method, the particle-tracking-based Eulerian-Lagrangian methods, and the third-order TVD method. The finite-difference method can be fully implicit without any stability constraint to limit transport step sizes, but the particle-tracking-based Eulerian-Lagrangian methods and the third-order TVD method still have time-step constraints associated with particle tracking and TVD methodology. If the GCG solver is not selected, the explicit formulation is automatically used in MT3DMS with the usual stability constraints. The explicit formulation is efficient for solving advection-dominated problems in which the transport step sizes are restricted by accuracy considerations. It is also useful when the implicit solver requires a large number of iterations to converge or when the computer system does not have enough memory to use the implicit solver.

MT3DMS is implemented with an optional, dual-domain formulation for modeling mass transport. With this formulation, the porous medium is regarded as consisting of two distinct domains, a mobile domain where transport is predominately by advection and an immobile domain where transport is predominately by molecular diffusion. Instead of a single "effective" porosity for each model cell, two porosities, one for the mobile domain and the other for the immobile domain, are used to characterize the porous medium. The exchange between the mobile and immobile domains is specified by a mass transfer coefficient. The dual-domain advective-diffusive model may be more appropriate for modeling transport in fractured media or extremely heterogeneous porous media than the single-porosity advective-dispersive model, provided the porosities and mass transfer coefficients can be properly characterized.

MT3DMS retains the same modular structure of the original MT3D code which is similar to that implemented in the U.S. Geological Survey modular three-dimensional finite-difference groundwater flow model, MODFLOW (McDonald and Harbaugh 1988; Harbaugh and McDonald 1996). The modular structure of the transport model makes it possible to simulate advection, dispersion/diffusion, source/sink mixing, and chemical reactions separately without reserving computer memory space for unused options; furthermore, new packages involving other transport processes and reactions can be added to the model readily without having to modify the existing code.

As in the original MT3D code, MT3DMS is developed for use with any block-centered finite-difference flow model such as MODFLOW and is based on

the assumption that changes in the concentration field will not affect the flow field significantly. After a flow model is developed and calibrated, the information needed by the transport model can be saved in disk files which are then retrieved by the transport model. Since most potential users of a transport model are likely to have been familiar with one or more flow models, MT3DMS provides an opportunity to simulate contaminant transport without having to learn a new flow model or modify an existing flow model to fit the transport model. In addition, separate flow simulation and calibration outside the transport model can result in substantial savings in computer memory. The model structure also saves execution time when many transport runs are required while the flow solution remains the same. Although this report only describes the use of MT3DMS in conjunction with MODFLOW, MT3DMS can be linked to any other block-centered finite-difference flow model in a simple and straightforward fashion.

MT3DMS can be used to simulate changes in concentrations of miscible contaminants in groundwater considering advection, dispersion, diffusion, and some basic chemical reactions, with various types of boundary conditions and external sources or sinks. The chemical reactions included in the model are equilibrium-controlled or rate-limited linear or nonlinear sorption and first-order irreversible or reversible kinetic reactions. It should be noted that the basic chemical reaction package included in MT3DMS is intended for single-species systems. An add-on reaction package such as RT3D (Clement 1997) or SEAM3D (Widdowson and Waddill 1997) must be used to model more sophisticated multispecies reactions. MT3DMS can accommodate very general spatial discretization schemes and transport boundary conditions, including: (a) confined, unconfined, or variably confined/unconfined aquifer layers; (b) inclined model layers and variable cell thickness within the same layer; (c) specified concentration or mass flux boundaries; and (d) the solute transport effects of external hydraulic sources and sinks such as wells, drains, rivers, areal recharge, and evapotranspiration.

Organization of the Report

This report covers the theoretical, numerical, and application aspects of the MT3DMS transport model. Following this introduction, Chapter 2 gives a brief overview of the physical-mathematical basis and various functional relationships underlying the transport model. Chapter 3 explains the basic ideas behind the various solution schemes implemented in MT3DMS. Chapter 4 discusses the computer implementation of the numerical solution schemes. Chapter 5 describes the program structure and design of the MT3DMS code, which has been divided into a main program and a number of packages, each dealing with a single aspect of the transport simulation. Chapter 6 provides detailed model input instructions and discusses how to set up a simulation. Chapter 7 describes the benchmark and example problems that were used to test the MT3DMS code and illustrate its applications. The appendices include information on the iterative solver (Appendix A), the computer memory requirements of the MT3DMS model (Appendix B), the interface between MT3DMS and a flow model (Appendix C), several postprocessing programs (Appendix D), and tables of abbreviated input instructions (Appendix E).

2 Fundamentals of Contaminant Transport Modeling

Governing Equations

The partial differential equation describing the fate and transport of contaminants of species k in 3-D, transient groundwater flow systems can be written as follows:

$$\frac{\partial\left(\theta C^k\right)}{\partial t} = \frac{\partial}{\partial x_i}\left(\theta D_{ij}\frac{\partial C^k}{\partial x_j}\right) - \frac{\partial}{\partial x_i}\left(\theta v_i C^k\right) + q_s C_s^k + \sum R_n \qquad (1)$$

where

θ = porosity of the subsurface medium, dimensionless

C^k = dissolved concentration of species k, ML^{-3}

t = time, T

$x_{i,j}$ = distance along the respective Cartesian coordinate axis, L

D_{ij} = hydrodynamic dispersion coefficient tensor, L^2T^{-1}

v_i = seepage or linear pore water velocity, LT^{-1}; it is related to the specific discharge or Darcy flux through the relationship, $v_i = q_i/\theta$

q_s = volumetric flow rate per unit volume of aquifer representing fluid sources (positive) and sinks (negative), T^{-1}

C_s^k = concentration of the source or sink flux for species k, ML^{-3}

$\sum R_n$ = chemical reaction term, $ML^{-3}T^{-1}$

The left-hand side of Equation 1 can be expanded into two terms, i.e.,

$$\frac{\partial(\theta C^k)}{\partial t} = \theta\frac{\partial C^k}{\partial t} + C^k\frac{\partial \theta}{\partial t} = \theta\frac{\partial C^k}{\partial t} + q_s' C^k \tag{2}$$

where $q_s' = \partial\theta/\partial t$ is the rate of change in transient groundwater storage (unit, T^{-1}).

The chemical reaction term in Equation 1 can be used to include the effect of general biochemical and geochemical reactions on contaminant fate and transport. Considering only two basic types of chemical reactions, i.e., aqueous-solid surface reaction (sorption) and first-order rate reaction, the chemical reaction term can be expressed as follows:

$$\sum R_n = -\rho_b\frac{\partial \overline{C}^k}{\partial t} - \lambda_1\theta C^k - \lambda_2\rho_b\overline{C}^k \tag{3}$$

where

ρ_b = bulk density of the subsurface medium, ML^{-1}

$\overline{C}^k$ = concentration of species k sorbed on the subsurface solids, MM^{-1}

λ_1 = first-order reaction rate for the dissolved phase, T^{-1}

λ_2 = first-order reaction rate for the sorbed (solid) phase, T^{-1}

Substituting Equations 2 and 3 into Equation 1 and dropping the species index for simplicity of presentation, Equation 1 can be rearranged and rewritten as

$$\theta\frac{\partial C}{\partial t} + \rho_b\frac{\partial \overline{C}}{\partial t} = \frac{\partial}{\partial x_i}\left(\theta D_{ij}\frac{\partial C}{\partial x_j}\right) - \frac{\partial}{\partial x_i}(\theta v_i C) + q_s C_s - q_s' C - \lambda_1\theta C - \lambda_2\rho_b\overline{C} \tag{4}$$

Equation 4 is essentially a mass balance statement, i.e., the change in the mass storage (both dissolved and sorbed phases) at any given time is equal to the difference in the mass inflow and outflow due to dispersion, advection, sink/source, and chemical reactions.

Local equilibrium is often assumed for the various sorption processes (i.e., sorption is sufficiently fast compared to the transport time scale). When the local equilibrium assumption (LEA) is invoked, it is customary to express Equation 4 in the following form:

$$R\theta\frac{\partial C}{\partial t}=\frac{\partial}{\partial x_i}\left(\theta D_{ij}\frac{\partial C}{\partial x_j}\right)-\frac{\partial}{\partial x_i}(\theta v_i C)+q_s C_s-q_s' C-\lambda_1\theta C-\lambda_2\rho_b\overline{C} \qquad (5)$$

where R is referred to as the retardation factor, which is a dimensionless factor defined as:

$$R=1+\frac{\rho_b}{\theta}\frac{\partial\overline{C}}{\partial C} \qquad (6)$$

When the LEA is not appropriate, the sorption processes are typically represented through a first-order kinetic mass transfer equation as discussed in the section on Chemical Reactions.

Note that in the transport-governing equations described above, only a single porosity has been assumed. This porosity has been commonly referred to as "effective" porosity, which is generally smaller than the total porosity of the porous medium, reflecting the fact that some pore spaces may contain immobile water with zero groundwater seepage velocity. However, as discussed in some detail by Zheng and Bennett (1995), this so-called effective porosity cannot be readily measured in the field due to the complexity of the pore structure. Rather, it generally must be interpreted as the lumped parameter which, in model calibration, gives the closest representation both of plume movement and of observed solute accumulation effects. In some cases, such as transport in fractured media or extremely heterogeneous porous media, it may be more appropriate to use a dual-porosity approach, defining a primary porosity for those pore spaces filled with mobile water where advection is the predominant means of transport and a secondary porosity for those pore spaces filled with immobile water where transport is primarily by molecular diffusion. The exchange between the mobile and immobile domains can be defined through a kinetic mass transfer equation similar to that used to describe nonequilibrium sorption.

The transport equation is related to the flow equation through the Darcy's Law

$$v_i=\frac{q_i}{\theta}=-\frac{K_i}{\theta}\frac{\partial h}{\partial x_i} \qquad (7)$$

where

K_i = principal component of the hydraulic conductivity tensor, LT^{-1}

h = hydraulic head, L

The hydraulic head is obtained from the solution of the three-dimensional groundwater flow equation

$$\frac{\partial}{\partial x_i}\left(K_i \frac{\partial h}{\partial x_i}\right) + q_s = S_s \frac{\partial h}{\partial t} \tag{8}$$

where S_s is the specific storage of the aquifer, L^{-1}, and q_s is the fluid sink/source term as defined in Equation 1.

Implied in Equations 7 and 8 is the assumption that the principal components of the hydraulic conductivity tensor, K_x, K_y, and K_z, are aligned with the x, y, and z coordinate axes so that all nonprincipal components (cross terms) become zero. This assumption is incorporated in most commonly used finite-difference groundwater flow models, including MODFLOW (McDonald and Harbaugh 1988; Harbaugh and McDonald 1996).

Advection

The advection term of the transport equation, $\partial(\theta v_i C)/\partial x_i$, describes the transport of miscible contaminants at the same velocity as the groundwater. For many field-scale contaminant transport problems, the advection term dominates over other terms. To measure the degree of advection domination, a dimensionless Peclet number is usually used, which is defined as

$$P_e = \frac{|v|L}{D} \tag{9}$$

where

$|v|$ = magnitude of the seepage velocity vector, LT^{-1}

L = characteristic length, commonly taken as the grid cell width, L

D = dispersion coefficient, L^2T^{-1}

In advection-dominated problems, also referred to as sharp front problems, the Peclet number has a large value. For pure advection problems, the Peclet number approaches infinity.

For advection-dominated problems, the solution of the transport equation is often plagued to some degree by two types of numerical problems as illustrated in Figure 1. The first type is numerical dispersion, which has an effect similar to that of physical dispersion, but is caused by truncation error. When physical dispersion is small or negligible, numerical dispersion becomes a serious problem, leading to the smearing of concentration fronts which should have a sharp appearance (Figure 1a). The second type of numerical problem is artificial

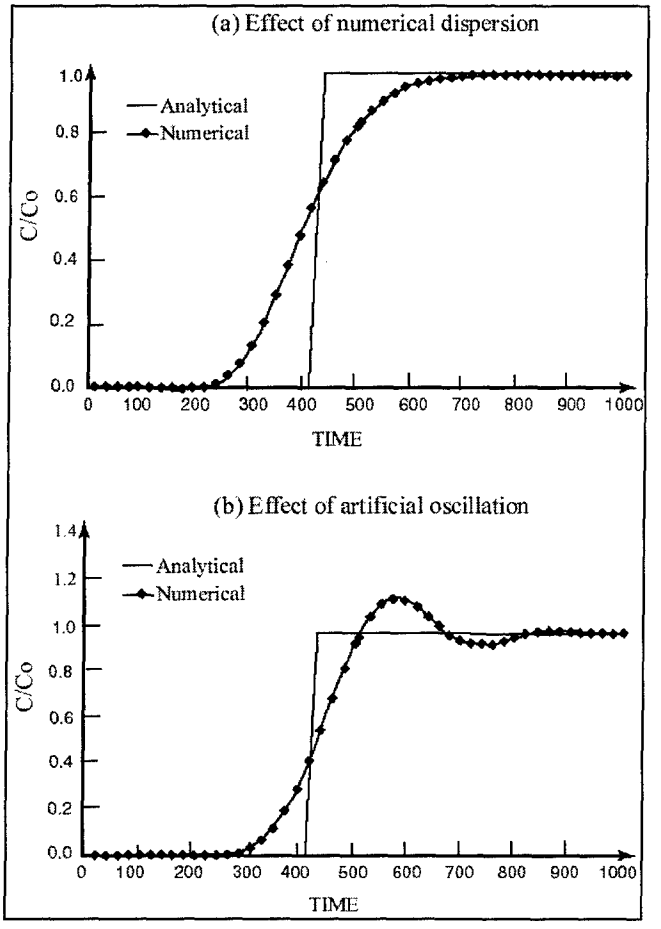


Figure 1. Illustration of common numerical errors in contaminant transport modeling

oscillation, also referred to as overshoot and undershoot, as illustrated in Figure 1b. Artificial oscillation is typical of some higher-order schemes designed to eliminate numerical dispersion and tends to become more severe as the concentration front becomes sharper.

The MT3DMS code has several solution options designed to overcome both the numerical dispersion and artificial oscillation problems. The method of characteristics is highly effective for eliminating numerical dispersion in strongly advection-dominated problems. The third-order TVD method, implemented with a universal flux limiter, minimizes both numerical dispersion and artificial oscillation. When numerical dispersion is not a significant problem, as in cases where a fine model grid is used or physical dispersion is large, the standard finite-difference method can be used for greater computational efficiency.

Dispersion

Dispersion mechanism

Dispersion in porous media refers to the spreading of contaminants over a greater region than would be predicted solely from the average groundwater velocity vectors (Anderson 1979 and 1984). Dispersion is caused by mechanical dispersion, a result of deviations of actual velocity on a microscale from the average groundwater velocity, and by molecular diffusion driven by concentration gradients. Molecular diffusion is generally secondary and negligible, compared with the effects of mechanical dispersion, and only becomes important when groundwater velocity is very low. The sum of mechanical dispersion and molecular diffusion is termed hydrodynamic dispersion, or simply dispersion.

Although the dispersion mechanism is generally understood, the representation of dispersion phenomena in a transport model is the subject of intense continuing research. The treatment of mechanical dispersion as a Fickian process (in effect, addictive to diffusion) represents a pragmatic approach through which realistic transport calculations can be made without fully describing the heterogeneous velocity field, which, of course, is impossible to do in practice. While many different approaches and theories have been developed to represent the dispersion process, Equation 1 is still the basis for most field-scale simulations (Zheng and Bennett 1995). In MT3DMS, the newly implemented dual-domain advection-diffusion formulation may be used as an alternative to the standard advection-dispersion formulation (see Dual-Domain Mass Transfer section).

Dispersion coefficient

The hydrodynamic dispersion tensor, D_{ij}, for an isotropic porous medium, is defined, according to Bear (1972 and 1979), in the following component forms

$$D_{xx} = \alpha_L \frac{v_x^2}{|v|} + \alpha_T \frac{v_y^2}{|v|} + \alpha_T \frac{v_z^2}{|v|} + D^* \tag{10a}$$

$$D_{yy} = \alpha_L \frac{v_y^2}{|v|} + \alpha_T \frac{v_x^2}{|v|} + \alpha_T \frac{v_z^2}{|v|} + D^* \tag{10b}$$

$$D_{zz} = \alpha_L \frac{v_z^2}{|v|} + \alpha_T \frac{v_x^2}{|v|} + \alpha_T \frac{v_y^2}{|v|} + D^* \tag{10c}$$

$$D_{xy} = D_{yx} = (\alpha_L - \alpha_T) \frac{v_x v_y}{|v|} \tag{10d}$$

$$D_{xz} = D_{zx} = (\alpha_L - \alpha_T)\frac{v_x v_z}{|v|} \tag{10e}$$

$$D_{yz} = D_{zy} = (\alpha_L - \alpha_T)\frac{v_y v_z}{|v|} \tag{10f}$$

where

D_{xx}, D_{yy}, D_{zz} = principal components of the dispersion tensor, L^2T^{-1}

$D_{xy}, D_{xz}, D_{yx}, D_{yz}, D_{zx}, D_{zy}$ =.cross terms of the dispersion tensor, L^2T^{-1}

α_L = longitudinal dispersivity, L

α_T = transverse dispersivity, L

D^* = effective molecular diffusion coefficient, L^2T^{-1}

v_x, v_y, v_z = components of the velocity vector along the x, y, and z axes, LT^{-1}

$|v| = \sqrt{v_x^2 + v_y^2 + v_z^2}$ = magnitude of the velocity vector, LT^{-1}

When the velocity vector is aligned with one of the coordinate axes, all the cross terms become zero.

Strictly speaking, the dispersion tensor defined by two independent dispersivities for isotropic media as in Equations 10a to 10f is not valid for anisotropic porous media, which require five independent dispersivities (Bear 1979). However, it is generally not feasible to obtain all five dispersivities in the field. As a practical alternative, the usual practice in transport modeling is to assume that the isotropic dispersion coefficient is also applicable to anisotropic porous media. In addition to the isotropic dispersion described above, the MT3DMS transport model supports an alternative form which allows the use of two transverse dispersivities, a horizontal transverse dispersivity, α_{TH}, and a vertical transverse dispersivity, α_{TV}, as proposed by Burnett and Frind (1987):

$$D_{xx} = \alpha_L \frac{v_x^2}{|v|} + \alpha_{TH} \frac{v_y^2}{|v|} + \alpha_{TV} \frac{v_z^2}{|v|} + D^* \tag{11a}$$

$$D_{yy} = \alpha_L \frac{v_y^2}{|v|} + \alpha_{TH} \frac{v_x^2}{|v|} + \alpha_{TV} \frac{v_z^2}{|v|} + D^* \tag{11b}$$

$$D_{zz} = \alpha_L \frac{v_z^2}{|v|} + \alpha_{TV} \frac{v_x^2}{|v|} + \alpha_{TV} \frac{v_y^2}{|v|} + D^* \tag{11c}$$

$$D_{xy} = D_{yx} = (\alpha_L - \alpha_{TH}) \frac{v_x v_y}{|v|} \tag{11d}$$

$$D_{xz} = D_{zx} = (\alpha_L - \alpha_{TV}) \frac{v_x v_z}{|v|} \tag{11e}$$

$$D_{yz} = D_{zy} = (\alpha_L - \alpha_{TV}) \frac{v_y v_z}{|v|} \tag{11f}$$

Equations 11a to 11f become equivalent to Equations 10a to 10f when the two transverse dispersivities are set equal.

In Equations 10 and 11, the components of the dispersion coefficient tensor are defined in terms of the groundwater seepage velocity, v. When the porosity is spatially varying, it is more convenient to define an apparent dispersivity tensor as

$$\hat{D}_{ij} = \theta D_{ij} \tag{12}$$

where the apparent dispersion coefficient, $\hat{D}$, can be obtained by using the components of the specific discharge q, instead of the linear velocity v, and replacing D^* with θD^* in Equations 10 and 11.

Sinks and Sources

The fluid sink/source term of the governing equation, $q_s C_s$, represents solute mass entering the model domain through sources or leaving the model domain through sinks. The term $q'_s C$ may be viewed as the "internal" sink/source term which represents the change in solute mass storage caused by the change in transient groundwater storage. It does not cause mass to leave or enter the model domain.

Sinks or sources may be classified as areally distributed sinks or sources or as point sinks or sources. The areally distributed sinks or sources include recharge and evapotranspiration. The point sinks or sources include wells, drains, and rivers. Constant-head and general head dependent boundaries in the flow model are also treated as point sinks or sources because they function in exactly the same fashion as wells, drains, or rivers in the transport model.

For sources, it is necessary to specify the concentration of source water. For sinks, the concentration of sink water is generally equal to the concentration of

groundwater in the aquifer at the sink location and cannot be specified. However, there is one exception where the concentration of sinks may differ from that of groundwater. The exception is evapotranspiration, which may be assumed to take only pure water away from the aquifer so that the concentration of the evapotranspiration flux is zero.

Chemical Reactions

The MT3DMS code is capable of handling equilibrium-controlled linear or nonlinear sorption, nonequilibrium (rate-limited) sorption, and first-order reaction that can represent radioactive decay or provide an approximate representation of biodegradation. The general formulation designed to model rate-limited sorption can also be used to model kinetic mass transfer between the mobile and immobile domains in a dual-domain advection-diffusion model. More sophisticated chemical reactions can be modeled through add-on reaction packages.

Equilibrium-controlled linear or nonlinear sorption

Sorption refers to the mass transfer process between the contaminants dissolved in groundwater (aqueous phase) and the contaminants sorbed on the porous medium (solid phase). It is generally assumed that equilibrium conditions exist between the aqueous-phase and solid-phase concentrations and that the sorption reaction is fast enough, relative to groundwater velocity, to be treated as instantaneous. The functional relationship between the dissolved and sorbed concentrations under a constant temperature is referred to as the sorption isotherm. Equilibrium-controlled sorption isotherms are generally incorporated into the transport model through the use of the retardation factor as defined in Equation 6. Three types of equilibrium-controlled sorption isotherms (linear, Freundlich, and Langmuir) are considered in the MT3DMS transport model.

a. The linear sorption isotherm assumes that the sorbed concentration ($\overline{C}$) is directly proportional to the dissolved concentration (C):

$$\overline{C} = K_d C \tag{13}$$

where K_d is the distribution coefficient, L^3M^{-1}. The retardation factor is thus given by

$$R = 1 + \frac{\rho_b}{\theta}\frac{\partial \overline{C}}{\partial C} = 1 + \frac{\rho_b}{\theta}K_d \tag{14}$$

b. The Freundlich isotherm is a nonlinear isotherm which can be expressed in the following form,

$$\overline{C} = K_f C^a \tag{15}$$

where

K_f = Freundlich constant, $\left(L^3 M^{-1}\right)^a$

a = Freundlich exponent, dimensionless

Both K_f and a are empirical coefficients. When the exponent a is equal to unity, the Freundlich isotherm is equivalent to the linear isotherm. The retardation factor for the Freundlich isotherm is defined accordingly as

$$R = 1 + \frac{\rho_b}{\theta}\frac{\partial \overline{C}}{\partial C} = 1 + \frac{\rho_b}{\theta} a K_f C^{a-1} \tag{16}$$

c. Another nonlinear sorption isotherm is the Langmuir isotherm, which is described by the equation,

$$\overline{C} = \frac{K_l \overline{S} C}{1 + K_l C} \tag{17}$$

where

K_l = Langmuir constant, L^3M^{-1}

$\overline{S}$ = total concentration of sorption sites available, MM^{-1}

The retardation factor defined for the Langmuir isotherm is then

$$R = 1 + \frac{\rho_b}{\theta}\frac{\partial \overline{C}}{\partial C} = 1 + \frac{\rho_b}{\theta}\left[\frac{K_l \overline{S}}{(1 + K_l C)^2}\right] \tag{18}$$

Nonequilibrium sorption

When the local equilibrium assumption is not valid, it is assumed that the sorption process can be represented through a first-order reversible kinetic reaction as follows:

$$\rho_b \frac{\partial \overline{C}}{\partial t} = \beta\left(C - \frac{\overline{C}}{K_d}\right) \tag{19}$$

where

β = first-order mass transfer rate between the dissolved and sorbed phases, T^{-1}

K_d = distribution coefficient for the sorbed phase as defined previously for the linear sorption

Equation 19 needs to be solved simultaneously with the transport governing equation to obtain solutions of solute transport affected by nonequilibrium

sorption. As the mass transfer rate β increases (i.e., the sorption process becomes increasingly faster), the nonequilibrium sorption approaches the equilibrium-controlled linear sorption as defined in Equation 13. For very small values of β, the exchange between the aqueous and solid phases is so slow that sorption becomes essentially negligible.

Radioactive decay or biodegradation

The first-order irreversible rate reaction term included in the governing equation, $-\left(\lambda_1 \theta C + \lambda_2 \rho_b \overline{C}\right)$, represents the mass loss of both the dissolved phase (C) and the sorbed phase $(\overline{C})$. The rate constant is usually given in terms of the half-life,

$$\lambda = (ln\,2) / t_{1/2} \tag{20}$$

where $t_{1/2}$ is the half-life of radioactive or biodegradable materials (i.e., the time required for the concentration to decrease to one-half of the original value).

For radioactive decay, the reaction generally occurs at the same rate in both phases. For biodegradation, however, it has been observed that certain reactions occur only in the dissolved phase. That is why two different rate constants may be needed. It should be noted that various biodegradation processes in the subsurface are usually more complex than that described by the first-order irreversible rate reaction. Other reaction packages for MT3DMS are either currently available or under development for modeling more complex types of biochemical and geochemical reactions.

Dual-Domain Mass Transfer

As discussed previously, solute transport in fractured media or extremely heterogeneous porous media may be conceptualized as a dual-domain system; transport is primarily by advection through the fractures or zones of high hydraulic conductivity filled with mobile water (mobile domain) whereas transport is primarily by diffusion through the nonfractured matrix or zones of low hydraulic conductivity filled with immobile or relatively stagnant water (immobile domain). The governing equations for the dual-domain system can be expressed, without explicit consideration of sorption, as follows:

$$\theta_m \frac{\partial C_m}{\partial t} + \theta_{im} \frac{\partial C_{im}}{\partial t} = \frac{\partial}{\partial x_i}\left(\theta_m D_{ij} \frac{\partial C_m}{\partial x_j}\right) - \frac{\partial}{\partial x_i}\left(\theta_m v_i C_m\right) + q_s C_s - q'_s C_m - \lambda_{1,m} \theta_m C_m - \lambda_{1,im} \theta_{im} C_{im} \tag{21a}$$

$$\theta_{im} \frac{\partial C_{im}}{\partial t} = \zeta\left(C_m - C_{im}\right) - \lambda_{1,im}\, \theta_{im}\, C_{im} \tag{21b}$$

where

C_m = dissolved concentration in the mobile domain (referred to as the mobile liquid phase), ML^{-3}

C_{im} = dissolved concentration in the immobile domain (referred to as the immobile liquid phase), ML^{-3}

θ_m = porosity of the mobile domain, dimensionless

θ_{im} = porosity of the immobile domain, or the difference between the total and mobile porosities, dimensionless $\theta_{im} = \theta - \theta_m$, dimensionless

$\lambda_{1,m}$ = first-order reaction rate for the mobile liquid phase, T^{-1}

$\lambda_{1,im}$ = first-order reaction rate for the immobile liquid phase, T^{-1}

ζ = first-order mass transfer rate between the mobile and immobile domains, T^{-1}

Equation 21a is a statement of mass conservation for the total mass, while Equation 21b is for the mass in the immobile domain. If sorption must be considered for solute transport in a dual-domain system, Equations 21a and 21b can be rewritten as

$$\theta_m \frac{\partial C_m}{\partial t} + f\rho_b \frac{\partial \overline{C}_m}{\partial t} + \theta_{im} \frac{\partial C_{im}}{\partial t} + (1-f)\rho_b \frac{\partial \overline{C}_{im}}{\partial t}$$

$$= \frac{\partial}{\partial x_i}\left(\theta_m D_{ij} \frac{\partial C_m}{\partial x_j}\right) - \frac{\partial}{\partial x_i}(\theta_m v_i C_m) + q_s C_s - q'_s C_m \qquad (21c)$$

$$- \lambda_{1,m}\theta_m C_m - \lambda_{1,im}\theta_{im} C_{im} - \lambda_{2,m} f\rho_b \overline{C}_m - \lambda_{2,im}(1-f)\rho_b \overline{C}_{im}$$

$$\theta_{im} \frac{\partial C_{im}}{\partial t} + (1-f)\rho_b \frac{\partial \overline{C}_{im}}{\partial t} \qquad (21d)$$

$$= \zeta(C_m - C_{im}) - \lambda_{1,im}\theta_{im} C_{im} - \lambda_{2,im}(1-f)\rho_b \overline{C}_{im}$$

where

$\overline{C}_m$ = sorbed concentration in the mobile domain (referred to as the mobile sorbed phase), MM^{-1}

$\overline{C}_{im}$ = sorbed concentration in the immobile domain (referred to as the immobile sorbed phase), MM^{-1}

f = fraction of sorption sites that are in contact with the mobile liquid phase dimensionless (f may be approximately set equal to θ_m/θ)

$\lambda_{2,m}$ = first-order reaction rate for the mobile sorbed phase, T^{-1}

$\lambda_{2,im}$ = first-order reaction rate for the immobile sorbed phase, T^{-1}.

If a linear equilibrium-controlled sorption isotherm can be assumed, Equations 21c and 21d are simplified as follows, assuming that the sorption coefficient is the same in the mobile and immobile domains:

$$\theta_m R_m \frac{\partial C_m}{\partial t} + \theta_{im} R_{im} \frac{\partial C_{im}}{\partial t} = \frac{\partial}{\partial x_i}\left(\theta_m D_{ij} \frac{\partial C_m}{\partial x_j}\right) - \frac{\partial}{\partial x_i}(\theta_m v_i C_m) + q_s C_s \quad . \text{(21e)}$$
$$- q_s' C_m - \lambda_{1,m}\theta_m C_m - \lambda_{1,im}\theta_{im} C_{im} - \lambda_{2,m} f \rho_b K_d C_m - \lambda_{2,im}(1-f)\rho_b K_d C_{im}$$

$$\theta_{im} R_{im} \frac{\partial C_{im}}{\partial t} = \zeta(C_m - C_{im}) - \lambda_{1,im}\theta_{im} C_{im} - \lambda_{2,im}(1-f)\rho_b K_d C_{im} \quad \text{(21f)}$$

where

$R_m = 1 + f\rho_b K_d/\theta_m$ = retardation factor for the mobile domain, dimensionless

$R_{im} = 1 + (1-f)\rho_b K_d/\theta_{im}$ = retardation factor for the immobile domain, dimensionless

With the dual-domain conceptual model, the dispersion coefficient D_{ij} in Equations 21a, 21c, and 21e may be set equal or close to molecular diffusion coefficient, since mechanical dispersion is introduced through mass transfer between the mobile and immobile domains. As the mass transfer rate ζ increases, i.e., the exchange between the mobile and immobile domains becomes increasingly fast, the dual-domain model functions more and more like a single-domain model whose porosity approaches the total porosity of the porous medium. On the other end of the spectrum, as the mass transfer rate ζ approaches zero, the dual-domain model also becomes equivalent to a single-domain model but with the porosity of the single-domain model approaching the porosity of the mobile domain.

Initial Conditions

The governing equation of the transport model describes the transient changes of solute concentration in groundwater. Therefore, initial conditions are necessary to obtain a solution of the governing equation. The initial condition in general form is written as

$$C(x,y,z,t) = c_o(x,y,z) \quad on\,\Omega; \quad t = 0 \tag{22}$$

where $c_o(x,y,z)$ is a known concentration distribution and Ω denotes the entire model domain. If nonequilibrium sorption or dual-domain mass transfer is simulated, it may also be necessary to define initial concentration for the sorbed or immobile phase.

Boundary Conditions

The solution of the governing equation also requires specification of boundary conditions. Three general types of boundary condition are considered in the MT3DMS transport model: (a) concentration known along a boundary (Dirichlet Condition), (b) concentration gradient known across a boundary (Neumann Condition); and (c) a combination of (a) and (b) (Cauchy Condition).

a. For the Dirichlet boundary condition, the concentration is specified along the boundary for the entire duration of the simulation,

$$C(x,y,z,t) = c(x,y,z,t) \quad \text{on } \Gamma_1, \quad t \geq 0 \tag{23}$$

where Γ_1 denotes the specified-concentration boundary, and $c(x,y,z,t)$ is the specified concentration along Γ_1. The specified concentration may be set to vary with time.

In a flow model, a Dirichlet boundary is a specified-head boundary which acts as a source or sink of water entering or leaving the model domain. Similarly, a specified-concentration boundary in a transport model acts as a source providing solute mass to the model domain or as a sink taking solute mass out of the model domain. A specified-head boundary in the flow model may or may not be a specified-concentration boundary in the transport model.

b. For the Neumann boundary condition, the concentration gradient is specified across the boundary, or

$$\theta D_{ij} \frac{\partial C}{\partial x_j} = f_i(x,y,z,t) \quad \text{on } \Gamma_2, t \geq 0 \tag{24}$$

where $f_i(x,y,z,t)$ is a known function representing the dispersive flux normal to the boundary Γ_2. A special case is a no-dispersive-mass-flux boundary where $f_i(x,y,z,t) = 0$.

c. For the Cauchy boundary condition, both the concentration value and the concentration gradient are specified, i.e.,

$$\theta D_{ij} \frac{\partial C}{\partial x_j} - q_i C = g_i(x, y, z, t) \quad \text{on } \Gamma_3, \quad t \geq 0 \tag{25}$$

where $g_i(x, y, z, t)$ is a known function representing the total flux (dispersive and advective) normal to the boundary Γ_3. For a physically impermeable boundary, both dispersive and advective fluxes are equal to zero so that $g_i(x, y, z, t) = 0$. It is customary to assume that the advective flux dominates over the dispersive flux so that the above equation can be simplified as

$$-q_i C = g_i(x, y, z, t) \tag{26}$$

Equation 26 can be handled readily in the transport model, similarly to the sink/source term.

3 Overview of Solution Techniques

Introduction

Numerical solution of the advection-dispersion equation as described in Chapter 2 has been referred to as an "embarrassingly" difficult problem (Mitchell 1984; Leonard 1988). The fundamental difficulty stems from the fact that the spatial first derivative term (advection) and the spatial second derivative term (mechanical dispersion and molecular diffusion) co-exist in the transport-governing equation. While numerous techniques have been developed within and outside the groundwater modeling community in the last 3 decades, there is still not a single technique that can yield completely satisfactory solutions under general hydrogeologic conditions. Zheng and Bennett (1995) provide an introduction to some of the more commonly used transport solution techniques and discuss their relative strengths and limitations.

Most numerical methods for solving the advection-dispersion-reaction equation can be classified as Eulerian, Lagrangian, and mixed Eulerian-Lagrangian (Neuman 1984). In the Eulerian approach, the transport equation is solved with a fixed grid method such as the standard finite-difference or finite-element method. The Eulerian approach is mass conservative, offers the advantage and convenience of a fixed grid, and handles dispersion/reaction dominated problems effectively. For advection-dominated problems which exist under many field conditions, however, an Eulerian method may be susceptible to excessive numerical dispersion or artificial oscillation. To overcome these problems, restrictively small grid spacing and time-steps may be required.

In the Lagrangian approach, the transport equation (both the advection and dispersion terms) is solved in either a deforming grid or deforming coordinate in a fixed grid through particle tracking, as in the random walk method. The Lagrangian approach provides a highly efficient solution to advection-dominated problems virtually free of numerical dispersion. However, without a fixed grid or coordinate system, a Lagrangian method can lead to numerical instability and computational difficulties in nonuniform media with multiple sinks/sources and complex boundary conditions (Yeh 1990). Velocity interpolation needed in particle tracking can also lead to local mass balance errors and solution anomalies (LaBolle, Fogg, and Tompson 1996). In addition, the concentration

solution obtained generally has a "rough" appearance that requires posterior smoothing and interpretation.

The mixed Eulerian-Lagrangian approach attempts to combine the advantages of both the Eulerian and the Lagrangian approaches by solving the advection term with a Lagrangian method (particle tracking) and the dispersion and reaction terms with a Eulerian method (finite-difference or finite-element). However, some commonly used Eulerian-Lagrangian procedures, such as the method of characteristics, do not guarantee mass conservation. Since particle tracking is required, mixed Eulerian-Lagrangian methods also suffer from some of the same numerical difficulties that plague the Lagrangian methods. In addition, the mixed Eulerian-Lagrangian methods may not be as computationally efficient as either the purely Eulerian or the purely Lagrangian methods.

In recent years, a class of transport solution techniques, sometimes collectively referred to as the total-variation-diminishing (TVD) methods (Harten 1983, 1984), have been developed, mainly in the computational fluid dynamics (CFD) community. The term TVD refers to the property shared by these methods that the sum of concentration differences between adjacent nodes diminishes over successive transport steps, a necessary condition if the transport solution is to remain largely free of spurious oscillations. The TVD methods are essentially higher-order finite-difference (or finite-volume) methods, and as such, they belong to the Eulerian family of solution techniques and are inherently mass conservative. Because a higher-order method usually minimizes numerical dispersion at the expense of introducing spurious oscillations, TVD schemes are typically implemented with numerical procedures (termed flux limiters) to suppress or eliminate spurious oscillations while preserving the sharp concentration fronts. Compared with the standard finite-difference method with either upstream or central-in-space weighting, TVD schemes are generally much more accurate in solving advection-dominated problems, albeit with a greater computational burden. Compared with some Lagrangian or mixed Eulerian-Lagrangian methods such as the method of characteristics, TVD schemes are not as effective in eliminating numerical dispersion while preserving concentration "peaks", but their mass conservation property, smaller memory requirements, and some other advantages make TVD schemes arguably the best compromise between the standard finite-difference method and the particle tracking based Lagrangian or mixed Eulerian-Lagrangian methods.

MT3DMS includes the standard finite-difference method, several mixed Eulerian-Lagrangian methods, and a third-order TVD method with a universal flux limiter. These solution options treat the dispersion, sink/source, and reaction terms in exactly the same fashion, using the block-centered finite-difference method, either explicitly or implicitly. They differ, however, in the way the advection term is solved. Thus, the remainder of this chapter will focus on the different approaches used to solve the advection term; the finite-difference solution of dispersion, sink/source, and reaction terms, similar to that of the flow equation, is discussed in detail in the next chapter.

Standard Finite-Difference Method

The general transport equation described in Chapter 2 can be written as

$$R\theta\frac{\partial C}{\partial t} = -\frac{\partial}{\partial x}(\theta v_x C) - \frac{\partial}{\partial y}(\theta v_y C) - \frac{\partial}{\partial z}(\theta v_z C) + L(C) \tag{27}$$

where $L(C)$ denotes the operator for various nonadvection terms including dispersion/diffusion, fluid sink/source, and chemical reactions. Applying the finite difference algorithm, the first partial derivatives representing the three components of the advection term at any finite difference cell (e.g., (i, j, k) (see Figure 2)) can be approximated by the concentration values at the cell interfaces, as given below:

$$\begin{aligned}
&\frac{\partial}{\partial x}(\theta v_x C) + \frac{\partial}{\partial y}(\theta v_y C) + \frac{\partial}{\partial z}(\theta v_z C) \\
&= \frac{q_{x(i,j+1/2,k)} C_{i,j+1/2,k} - q_{x(i,j-1/2,k)} C_{i,j-1/2,k}}{\Delta x_j} \\
&+ \frac{q_{y(i+1/2,j,k)} C_{i+1/2,j,k} - q_{y(i-1/2,j,k)} C_{i-1/2,j,k}}{\Delta y_i} \\
&+ \frac{q_{z(i,j,k+1/2)} C_{i,j,k+1/2} - q_{z(i,j,k-1/2)} C_{i,j,k-1/2}}{\Delta z_k}
\end{aligned} \tag{28}$$

where Δx_j, Δy_i, Δz_k are the dimensions of cell (i, j, k) in the x, y, and z directions, respectively; and $j+½$, $i+½$, and $k+½$ denote the cell interfaces normal to the x, y, and z directions, respectively.

How the interface concentration is determined is what distinguishes one solution technique from another. In the standard finite-difference method, the interface concentration is evaluated using either the upstream or the central-in-space weighting scheme. For the upstream weighting scheme, the interface concentration between two neighboring nodes in a particular direction, (e.g., x) is set equal to the concentration at the upstream node along the same direction, i.e.,

$$C_{i,j-1/2,k} = \begin{cases} C_{i,j-1,k}, & \text{if } q_{x(i,j-1/2,k)} > 0 \\ C_{i,j,k}, & \text{if } q_{x(i,j-1/2,k)} < 0 \end{cases} \tag{29}$$

The upstream weighting scheme results in oscillation-free solutions. However, the solution of the advection term is only accurate to the first order, and as a result, it can lead to significant numerical dispersion when applied to advection-dominated problems, since the truncation error resulting from the advection solution is of the same order and could overwhelm the second-derivative physical dispersion term (Zheng and Bennett 1995).

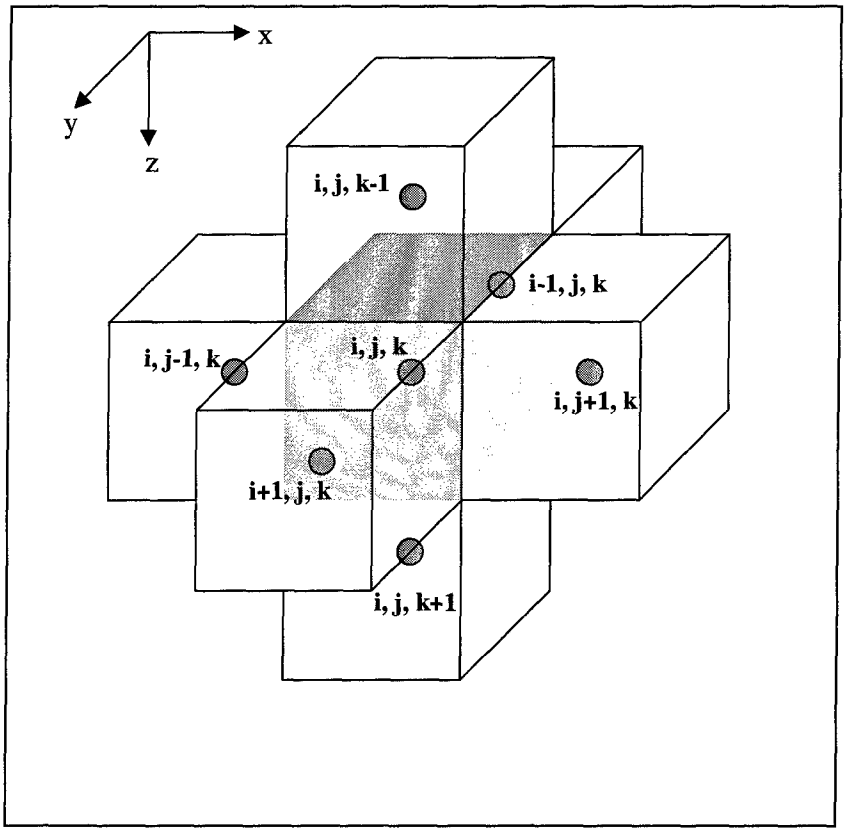


Figure 2. Index system used for the finite-difference (finite-volume) grid (the interfaces between node (i, j, k) and the six adjacent nodes are shown as shaded surfaces)

For the central-in-space weighting scheme, the interface concentration is set equal to the weighted average of the concentrations on the two sides of the interface, i.e.,

$$C_{i,j-1/2,k} = \alpha_x C_{i,j-1,k} + (1-\alpha_x) C_{i,j,k} \tag{30}$$

where $\alpha_x = \Delta x_j / (\Delta x_{j-1} + \Delta x_j)$ is the spatial weighting factor along the x direction. This weighting is equivalent to linear interpolation. When the grid spacing is regular, $\alpha_x = 0.5$. With the central-in-space weighting scheme, the solution of the advection term is accurate to the second order, and as a result, it does not lead to any numerical dispersion since the solution of the dispersion term is also accurate to the second order. However, if the transport problem is advection dominated, the central-in-space weighting scheme can lead to excessive artificial oscillation, which is typical of higher-order truncation errors.

Because of the dual problems of numerical dispersion and artificial oscillation, the standard finite-difference method is suitable only for solving transport models not dominated by advection (i.e., when either the physical dispersivity is large or the grid spacing is made sufficiently fine). Several

sources have indicated that when the grid Peclet number is smaller than four (see Zheng and Bennett 1995), the standard finite-difference method with either upstream or central-in-space weighting is reasonably accurate and thus can be used with confidence. In addition, because the standard finite-difference method is generally more computationally efficient than other methods, it may be useful for obtaining first approximations in the initial stages of a modeling study.

Third-Order TVD Method

A large number of TVD schemes for solving advection-dominated transport problems can be found in the literature (Harten 1983, 1984; Osher and Chakravarthy 1984, 1986; Yee 1987; Cox and Nishikawa 1991). The MT3DMS code is implemented with a third-order TVD scheme based on the ULTIMATE algorithm (Universal Limiter for Transient Interpolation Modeling of the Advective Transport Equations) (Leonard 1988; Leonard and Niknafs 1990, 1991) which is in turn a descendant of the earlier QUICKEST algorithm (Quadratic Upstream Interpolation for Convective Kinematics with Estimated Streaming Terms) (Leonard 1979). With the ULTIMATE scheme, the interface concentrations are determined through a third-order polynomial interpolation of nodal concentrations, supplemented by a universal flux limiting procedure to minimize unphysical oscillations which may occur if sharp concentration fronts are involved. This third-order ULTIMATE scheme is mass conservative, without excessive numerical dispersion, and essentially oscillation-free. The ULTIMATE scheme was significantly superior to some popular second-order TVD schemes (Leonard 1988) and was considered to be possibly the most accurate practical method available (Roache 1992). The basic ideas of the ULTIMATE scheme in one dimension are presented below; general 3-D equations are provided in the next chapter.

Consider a one-dimensional (1-D) form of Equation 27 in a uniform flow field (Figure 3),

$$\frac{\partial C}{\partial t} = -v\frac{\partial C}{\partial x} + L(C) \tag{31}$$

Assume a finite-difference grid with regular spacing superimposed by a local coordinate system with the origin at nodal point j. Further assume that the velocity is positive from left to right (Figure 3). The concentration at nodal point j ($x=0$) at time level $n+1$ due to advection alone can be directly written as

$$C_j^{n+1} = C(0,\Delta t) = C(-v\Delta t,0) \tag{32}$$

where Δt is the transport step size between the old time level n ($t=0$) and the new time level $n+1$ ($t = \Delta t$). Equation 32 can be understood by imagining a particle originating at a distance of $v\Delta t$ upstream (to the left) of the nodal point j. After a time increment Δt, the particle reaches the nodal point j. The concentration carried by the particle is thus the concentration at the nodal point j.

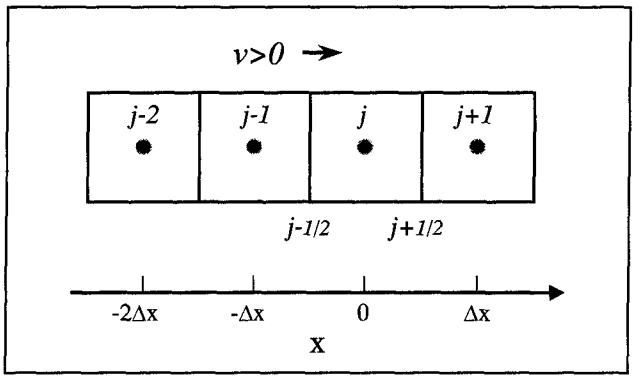


Figure 3. Illustration of the nodal points involved in the ULTIMATE scheme in one dimension

Because $C(-v\Delta t,0)$ at the old time level generally does not coincide with a nodal point, it must be determined through interpolation from the concentrations at the nearby nodes. A general third-order polynomial formula can be used for this purpose, which, in 1-D form, can be written as

$$C(x,0) = a + bx + cx^2 + dx^3 \tag{33}$$

where a, b, c, and d are coefficients which can be related to the nodal concentrations by noting that

$$\begin{aligned} &C(0,0) = C_j, \; at \; x = 0 \\ &C(-2\Delta x,0) = C_{j-2}, \; at \; x = -2\Delta x \\ &C(-\Delta x,0) = C_{j-1}, \; at \; x = -\Delta x \\ &C(\Delta x,0) = C_{j+1}, \; at \; x = \Delta x \end{aligned} \tag{34}$$

Also note that four nodal points are needed to determine the four coefficients in Equation 33. Among the four nodal points selected are the node j under consideration, the two nodes in the immediate vicinity of node j, and one additional node on the upstream side. If the velocity were from right to left, the node $j+2$ instead $j-2$ would have been selected. Thus, the polynomial interpolation as used in the ULTIMATE scheme is third order, biased toward the upstream direction. Substituting Equation 34 into 33, we can determine the coefficients in Equation 33 as follows:

$$
\begin{aligned}
a &= C_j \\
b &= \frac{1}{\Delta x}\left(\frac{C_{j+1}}{3}+\frac{C_j}{2}-C_{j-1}+\frac{C_{j-2}}{6}\right) \\
c &= \frac{1}{2(\Delta x)^2}\left(C_{j+1}-2C_j+C_{j-1}\right) \\
d &= \frac{1}{6(\Delta x)^3}\left(C_{j+1}-3C_j+3C_{j-1}-C_{j-2}\right)
\end{aligned}
\tag{35}
$$

The solution of Equation 32 can thus be obtained from Equations 33 and 35 as

$$
\begin{aligned}
C_j^{n+1} &= C(0,\Delta t) = C(-v\Delta t,0) = a + b(-v\Delta t) + c(-v\Delta t)^2 + d(-v\Delta t)^3 \\
&= C_j^n - C_r\left[\left(\frac{C_{j+1}^n}{3}+\frac{C_j^n}{2}-C_{j-1}^n+\frac{C_{j-2}^n}{6}\right)\right. \\
&\quad \left. -C_r\left(\frac{C_{j+1}^n-2C_j^n+C_{j-1}^n}{2}\right)+C_r{}^2\left(\frac{C_{j+1}^n-3C_j^n+3C_{j-1}^n-C_{j-2}^n}{6}\right)\right]
\end{aligned}
\tag{36}
$$

where $C_r = v\Delta t/\Delta x$. Compared with the explicit finite-difference solution in terms of interface concentrations,

$$
C_j^{n+1} = C_j^n - C_r\left(C_{j+1/2}^n - C_{j-1/2}^n\right) \tag{37}
$$

it can be seen that in the ULTIMATE scheme, the interface concentrations, in one dimensional form, are determined by

$$
\begin{aligned}
C_{j+1/2} &= \left(C_{j+1}+C_j\right)/2 - C_r\left(C_{j+1}-C_j\right)/2 \\
&\quad -\left(1-C_r{}^2\right)\left(C_{j+1}-2C_j+C_{j-1}\right)/6
\end{aligned}
\tag{38a}
$$

and

$$
\begin{aligned}
C_{j-1/2} &= \left(C_j+C_{j-1}\right)/2 - C_r\left(C_j-C_{j-1}\right)/2 \\
&\quad -\left(1-C_r{}^2\right)\left(C_j-2C_{j-1}+C_{j-2}\right)/6
\end{aligned}
\tag{38b}
$$

where the first term corresponds to the standard finite-difference solution with central-in-space weighting, the second term is referred to as the GRADIENT term, and the third term as the CURVATURE term. Note that in two-dimensional (2-D) or 3-D space, the equations for the ULTIMATE scheme are much more complex than Equation 38, with more than one GRADIENT and CURVATURE term and additional TWIST terms (Chapter 4). However, the basic ideas remain the same and the more complex equations can be derived following the same procedures as outlined above.

The interface concentrations as determined in Equation 38 can lead to unphysical oscillations if sharp concentration fronts are involved, as in advection-dominated problems. To circumvent this problem, the ULTIMATE scheme employs a universal flux limiter to make adjustments to the interface concentrations, after they are first determined from polynomial interpolation, according to certain rules. The following is a brief description of how the universal flux limiter works, based on Leonard and Niknafs (1990).

Consider Figure 4 which shows three nodes (the two nodes j and $j+1$ straddling the interface in question, $j+½$, and the adjacent upstream node $j-1$). If the concentration at interface $j+½$ lies between the two neighboring nodal concentrations, the concentration profile is locally monotonic and no spurious oscillation has resulted from interpolation. On the other hand, if the concentration profile is not monotonic, it is likely due to spurious oscillation, and some numerical procedure (i.e., a flux limiter) may be needed to remove the oscillation. Thus, the first step in implementing a flux limiter is to check the local monotonicity after the interface concentration is determined. For this purpose, we normalize all concentrations as shown in Figure 4 according to

$$\tilde{C} = \frac{C - C_{j-1}}{C_{j+1} - C_{j-1}} \tag{39}$$

Note from Equation 39, in terms of normalized concentrations,

$$\tilde{C}_{j-1} = 0 \quad and \quad \tilde{C}_{j+1} = 1$$

If the concentration profile across the three nodes as shown in Figure 4 is monotonic, the following conditions must hold:

$$\tilde{C}_j^n \le \tilde{C}_{j+1/2} \le 1 \tag{40}$$

and

$$0 \le \tilde{C}_{j-1/2} \le \tilde{C}_j^n \tag{41}$$

Equations 40 and 41 are a necessary condition for the concentration profile to be locally monotonic, but they alone are not enough to guarantee overall computational monotonicity. For this, consider the explicit finite-difference solution at node j as shown in Equation 37 but now in terms of normalized variables,

$$\tilde{C}_j^{n+1} = \tilde{C}_j^n - C_r\left(\tilde{C}_{j+1/2}^n - \tilde{C}_{j-1/2}^n\right) \tag{42}$$

In order to ensure local monotonicity, $\tilde{C}_j^{n+1}$ must satisfy

$$\tilde{C}_{j-1}^{n+1} \le \tilde{C}_j^{n+1} \le \tilde{C}_{j+1}^{n+1} \tag{43}$$

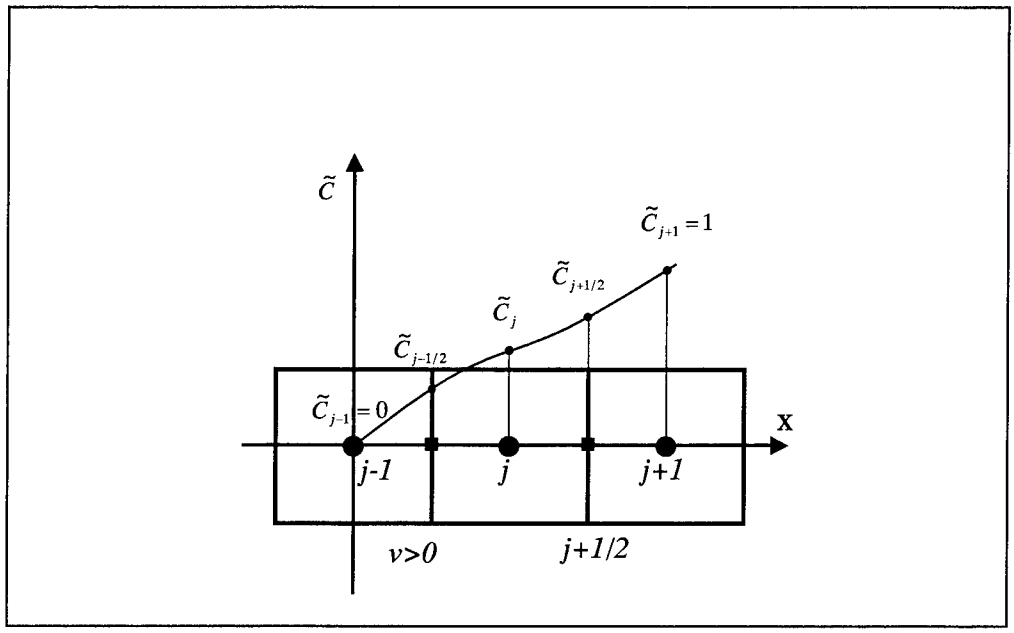


Figure 4. Diagram showing a monotonic concentration distribution in the immediate vicinity of the interface j+1/2 under consideration (concentrations are normalized according to Equation 39)

The worst-case condition is

$$0 \le \tilde{C}_j^{n+1} \le 1 \tag{44}$$

The right-hand inequality in Equation 44 is ensured because, as seen in Equation 42 when $\tilde{C}_{j+1/2}^n \ge \tilde{C}_{j-1/2}^n$, we have $\tilde{C}_j^{n+1} \le \tilde{C}_j^n \le 1$. The left-hand inequality in Equation 44 implies, using Equation 42,

$$\tilde{C}_{j+1/2} \le \tilde{C}_{j-1/2} + \tilde{C}_j^n / C_r \tag{45}$$

Again, a worst-case estimate for $\tilde{C}_{j-1/2}$ (=0) results in additional constraints on $\tilde{C}_{j+1/2}$:

$$\tilde{C}_{j+1/2} \le \tilde{C}_j^n / C_r \tag{46}$$

Equations 40 and 46 constitute the universal limiter constraints on the interface concentration $\tilde{C}_{j+1/2}$ with respect to the nodal concentration $\tilde{C}_j^n$, when $\tilde{C}_j^n$ is within the monotonic range:

$$0 \le \tilde{C}_j^n \le 1 \tag{47}$$

Figure 5 is a graphical illustration of the universal limiter constraints. The shaded region is bounded by the constraints as given in Equations 40 and 46. If the normalized interface concentration $\tilde{C}_{j+1/2}$ lies within the shaded region, the universal limiter constraints are satisfied (i.e., the concentration profile is locally monotonic without spurious oscillations). If $\tilde{C}_{j+1/2}$ lies outside the shaded region, some strategies may be devised to adjust $\tilde{C}_{j+1/2}$ in order to suppress the spurious oscillation and maintain the computational monotonicity. The simplest strategy is to set $\tilde{C}_{j+1/2}$ equal to the concentration at the closest upstream node,

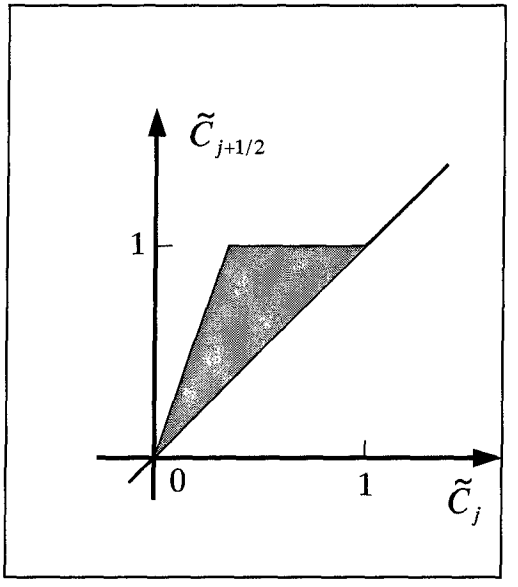


Figure 5. Illustration of the universal limiter constraints used in the ULTIMATE scheme

$$\tilde{C}_{j+1/2} = \tilde{C}_j^n \tag{48}$$

In summary, the ULTIMATE scheme as described above consists of the following basic steps:

a. Construct an explicit estimate for the normalized interface concentration value $C_{j+1/2}$ using Equation 38.

b. Compute the corresponding normalized value $\tilde{C}_{j+1/2}$ according to Equation 39, and also normalize concentrations at the two nodes straddling the interface and the adjacent upstream node.

c. If the point $(\tilde{C}_j^n, \tilde{C}_{j+1/2})$ lies within the shaded region of Figure 5, proceed with the unadjusted $C_{j+1/2}$.

d. If the point $(\tilde{C}_j^n, \tilde{C}_{j+1/2})$ lies outside this region, adjust $\tilde{C}_{j+1/2}$ according to Equation 48.

e. Reconstruct (unnormalize) $C_{j+1/2} = C_{j-1} + \tilde{C}_{j+1/2}(C_{j+1} - C_{j-1})$.

f. Repeat for each finite-difference cell (control-volume) interface.

Mixed Eulerian-Lagrangian Methods

MT3DMS is implemented with several mixed Eulerian-Lagrangian methods as in the original MT3D code, including the forward-tracking method of characteristics (MOC), the backward-tracking modified method of characteristics (MMOC), and a hybrid of these two methods (HMOC). This section provides a brief description of the basic principles and associated equations behind each of these methods. More detailed information on computer implementation of these methods is presented in the next chapter.

To use the Eulerian-Lagrangian approach, the transport governing equation, presented as Equation 1 in Chapter 2, needs to be expressed in Lagrangian form. First, the advection term can be expanded as follows:

$$\frac{\partial}{\partial x_i}\left(\theta v_i C\right) = \theta v_i \frac{\partial C}{\partial x_i} + C\frac{\partial\left(\theta v_i\right)}{\partial x_i} = \theta v_i \frac{\partial C}{\partial x_i} + C q_s \tag{49}$$

Substituting Equation 49 into Equation 1 and dividing both sides by the retardation factor, the governing equation becomes

$$\frac{\partial C}{\partial t} = \frac{1}{R\theta}\frac{\partial}{\partial x_i}\left(\theta D_{ij}\frac{\partial C}{\partial x_j}\right) - \bar{v}_i\frac{\partial C}{\partial x_i} - \frac{q_s}{R\theta}\left(C - C_s\right) - \frac{\lambda_1}{R}C - \frac{\lambda_2}{R}\frac{\rho_b}{\theta}\overline{C} \tag{50}$$

where $\bar{v}_i = v_i/R$ represents the "retarded" velocity of a contaminant particle.

Equation 50 is an Eulerian expression in which the partial derivative, $\partial C/\partial t$, represents the rate of change in solute concentration at a fixed point in space. Equation 50 can also be expressed in Lagrangian form as

$$\frac{DC}{Dt} = \frac{1}{R\theta}\frac{\partial}{\partial x_i}\left(\theta D_{ij}\frac{\partial C}{\partial x_j}\right) - \frac{q_s}{R\theta}\left(C - C_s\right) - \frac{\lambda_1}{R}C - \frac{\lambda_2}{R}\frac{\rho_b}{\theta}\overline{C} \tag{51}$$

where the substantial derivative, $DC/Dt = \partial C/\partial t + \bar{v}_i\,\partial C/\partial x_i$, represents the rate of change in solute concentration along the pathline of a contaminant particle (or a characteristic curve of the velocity field).

By introducing the finite-difference algorithm, the substantial derivative in Equation 51 can be approximated as

$$\frac{DC}{Dt} = \frac{C_m^{n+1} - C_m^{n^*}}{\Delta t} \tag{52}$$

so that Equation 51 becomes

$$C_m^{n+1} = C_m^{n^*} + \Delta t \times RHS \tag{53}$$

where

C_m^{n+1} = solute concentration for node m at new time level $n+1$;

C_m^{n*} = solute concentration for node m at new time level $n+1$ due to advection alone, also referred to as intermediate time level $n*$

Δt = time increment between old time level n and new time level $n+1$

RHS = finite-difference approximation to the terms on the right-hand side of Equation 51. The finite-difference approximation is explicit if the concentration at the old time level C^n is used in the calculation of *RHS*; it is implicit if the concentration at the new time level C^{n+1} is used

Equation 53 constitutes the basic algorithm of the mixed Eulerian-Lagrangian methods as implemented in the MT3DMS code. In these methods, the term C_m^{n*} in Equation 53, which accounts for the effect of advection, is solved with a Lagrangian method in a moving coordinate system, while the second term in Equation 53, which accounts for the effects of dispersion, sink/source mixing, and chemical reactions, is solved with either the explicit or implicit finite-difference method on a fixed Eulerian grid.

Depending on the use of different Lagrangian techniques to approximate the advection term, the mixed Eulerian-Lagrangian methods may be loosely classified as the front-tracking MOC (Garder, Peaceman, and Pozzi 1964; Konikow and Bredehoeft 1978; Zheng 1993; Konikow, Goode, and Hornberger 1996); the backward-tracking MMOC (Russell and Wheeler 1983; Cheng, Casulli, and Milford 1984; Roache 1992; Healy and Russell 1993; Yeh et al. 1993); and a combination of these two (Neuman 1981 and 1984; Farmer 1987; Yeh, Chang, Short 1992). Numerical schemes representative of each of these methods are implemented in the MT3DMS code. The concepts and the fundamental ideas behind these methods are briefly discussed below; their numerical implementation is discussed in detail in Chapter 4.

Method of characteristics (MOC)

The MOC was originally applied to transport in porous media by Garder, Peaceman, Pozzi (1964) for calculation of miscible displacement in reservoir simulation. This method was later made popular by the U.S. Geological Survey 2-D transport model (Konikow and Bredehoeft 1978) and has been widely used in field studies. The MOC uses a conventional particle-tracking technique for solving the advection term. At the beginning of the simulation, a set of moving particles is distributed in the flow field either randomly or with a fixed pattern. A concentration and a position in the Cartesian coordinate system are associated with each of these particles. Particles are tracked forward through the flow field using a small time increment. At the end of each time increment, the average

intermediate concentration at cell m due to advection alone over the time increment, C_m^{n*}, is evaluated from the concentrations of moving particles which are located within that cell (Figure 6). This intermediate concentration is used to calculate changes in concentration due to dispersion and other processes over that time increment. If a simple arithmetical averaging algorithm is used, this average intermediate concentration is expressed by the following equation:

$$C_m^{n*} = \frac{1}{NP_m} \sum_{p=1}^{NP_m} C_p^n \quad if \quad NP_m > 0 \tag{54}$$

where

NP_m = number of particles within cell m

C_p^n = concentration of the p^{th} particle at the old time level (n)

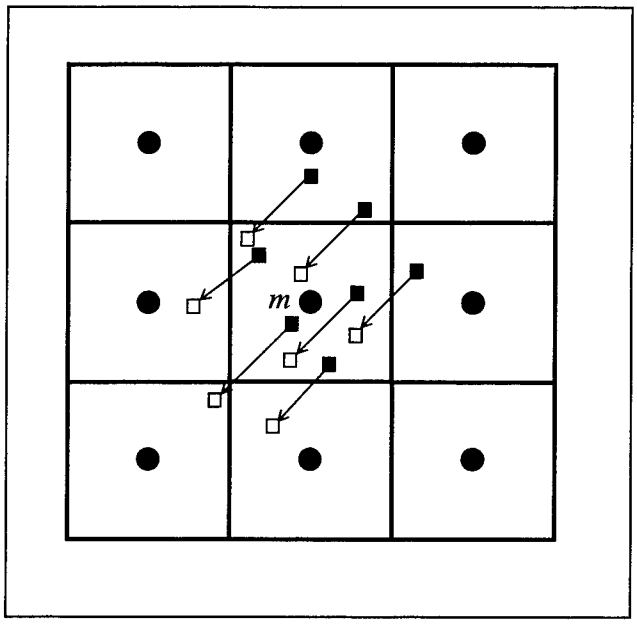


Figure 6. Illustration of the MOC

Strictly speaking, Equation 54 is applicable only if the model grid is regular. Other averaging algorithms can be used to handle an irregular grid more accurately. Zheng (1993) presents a volume-based averaging algorithm as

$$C_m^{n*} = \sum_{p=1}^{NP_m} V_p C_p^n \Big/ \sum_{p=1}^{NP_m} V_p \quad if \quad NP_m > 0 \tag{55}$$

where V_p is the volume of the cell in which the p^{th} particle is first generated. In a regular grid, Equation 55 reduces to Equation 54.

After completing the evaluation of C_m^{n*} for all cells, a weighted concentration, $C_m^{\hat{n}}$, is calculated based on C_m^{n*} and the concentration at the old time level C_m^n,

$$C_m^{\hat{n}} = \omega C_m^{n*} + (1-\omega)C_m^n \tag{56}$$

where ω is a weighting factor between 0.5 and 1. $C_m^{\hat{n}}$ is then used to calculate the second term in Equation 53, or the changes in concentration due to dispersion, sink/source mixing, and chemical reactions (the terms on the right-hand side of Equation 51) with either the explicit or implicit finite-difference method. If the explicit finite-difference method is used, then

$$\Delta C_m^{n+1} = \Delta t \times RHS\left(C_m^{\hat{n}}\right) \tag{57}$$

The use of the weighted concentration in Equation 57 represents an averaged approach because the processes of dispersion, sink/source mixing, and/or chemical reactions occur throughout the time increment.

The concentration for cell m at the new time level *(n+1)* is then the sum of the C_m^{n*} and ΔC_m^{n+1} terms. The concentrations of all moving particles are also updated to reflect the change due to dispersion, sink/source mixing, and chemical reactions. This completes the calculation of one transport step for the MOC. The procedure is repeated until the end of a desired time period is reached.

One of the most desirable features of the MOC technique is that it is virtually free of numerical dispersion caused by spatial truncation errors. The major drawback of the MOC technique is that it can be slow and requires a large amount of computer memory when it is necessary to track a large number of moving particles, especially in three dimensions. The MOC technique can also lead to large mass balance discrepancies under certain situations because the discrete nature of the particle-tracking-based mixed Eulerian-Lagrangian solution techniques does not guarantee local mass conservation at a particular time-step. In the MT3DMS code, the computer memory requirement for the MOC technique is dramatically reduced through the use of a dynamic approach for particle distribution. The mass balance discrepancy problem is also mitigated to some degree through the use of consistent velocity interpolation schemes and higher-order particle-tracking algorithms. However, it should be pointed out that the MOC technique can still result in significant mass balance errors if the model grid is highly irregular. When this happens, it is more appropriate to the use the TVD scheme or the standard finite-difference method if numerical dispersion is not a concern.

Modified method of characteristics (MMOC)

The MMOC was originally developed to approximate the advection term accurately without sacrificing a great deal of computational efficiency (Russell and Wheeler 1983; Cheng, Casulli, and Milford 1984). The MMOC technique is similar to the MOC technique except in the treatment of the advection term. Unlike the MOC technique, which tracks a large number of moving particles forward in time and keeps track of the concentration and position of each particle, the MMOC technique places one fictitious particle at the nodal point of the fixed grid at each new time level *(n+1)*. The particle is tracked backward to find its position at the old time level *(n)*. The concentration associated with that position is used to approximate the C_m^{n*} term, is the intermediate concentration due to the effect of advection during the period since the preceding time level (Figure7):

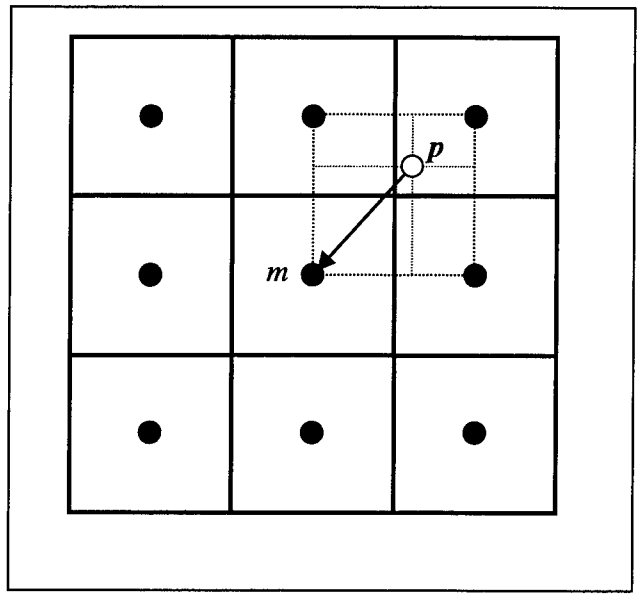


Figure 7. Illustration of the MMOC

$$C_m^{n*} = C^n(x_p) = C^n(x_m - d) \tag{58}$$

where

x_p = position which a particle starting from nodal point m reaches when it is tracked backward along the reverse pathline over the time increment Δt

x_m = position vector of nodal point m

d = characteristic nodal displacement, or the distance along a particle path from x_m to x_p

$C^n(x_p)$ = concentration at position x_p at the old time level *(n)*, generally interpolated from concentrations at neighboring nodal points

The MMOC technique uses one particle for each finite-difference cell, whereas the MOC technique generally requires several particles per cell. Therefore, the MMOC technique, used in conjunction with a simple lower-order interpolation scheme, is normally faster than the MOC technique. Furthermore, because the MMOC technique starts particles at nodal points at each new time level, there is no need to store the particle identities in computer memory. Hence, for problems in which the MOC technique requires a large number of particles, the MMOC technique requires much less computer memory. The MMOC technique is also free of artificial oscillations if implemented with a lower-order interpolation scheme such as linear interpolation (also referred to as bilinear in two dimensions or trilinear in three dimensions). However, with a lower-order interpolation scheme, the MMOC technique introduces considerable numerical dispersion, especially for sharp front problems. Higher-order interpolation schemes can be used to eliminate or reduce numerical dispersion. For example, Cheng, Casulli, and Milford (1984) used a quadratic interpolation scheme in 2-D simulations and pointed out that it is free of numerical dispersion. However, it is computationally less efficient than the linear scheme and can lead to severe artificial oscillations for sharp front problems. Healy and Russell (1989) tested several interpolation schemes for 1-D problems and concluded that a mixed linear/quadratic scheme can minimize both numerical dispersion and artificial oscillations. The drawback is a much higher computational requirement, especially in a multidimensional simulation, than the linear scheme, and it does not conserve mass as well, thereby offsetting much of the computational efficiency of the MMOC technique. For these reasons, the MMOC technique in the MT3DMS transport model is implemented only with a lower-order interpolation scheme and is intended only for use in situations where sharp fronts are not present so that any numerical dispersion error resulting from the solution scheme is insignificant.

Hybrid method of characteristics (HMOC)

As described in the preceding discussions, either the MOC or the MMOC scheme may be utilized to solve the mixed Eulerian-Lagrangian equation. The selection of the methods is based on such considerations as field conditions (whether the concentration field has sharp or smooth fronts) and computer resources available (generally the MOC solution requires more memory space and longer execution time). A third option is to use a hybrid of the two methods; this option is referred to here as the HMOC.

The HMOC technique attempts to combine the strengths of the MOC and the MMOC techniques by using an automatic adaptive scheme conceptually similar to the one proposed by Neuman (1984). The fundamental idea behind this

scheme is automatic adaptation of the solution process to the nature of the concentration field. When sharp concentration fronts are present, the advection term is solved by the MOC technique through the use of moving particles distributed dynamically around each front. Away from such fronts, the advection term is solved by the MMOC technique, with fictitious particles placed at the nodal points tracked directly backward in time. When a front dissipates due to dispersion and chemical reactions, the forward tracking stops automatically and the corresponding particles are removed. By selecting an appropriate criterion for controlling the switch between the MOC and MMOC schemes, the adaptive procedure can provide accurate solutions to transport problems over the entire range of Peclet numbers from 0 to ∞, with virtually no numerical dispersion, while at the same time using far fewer particles than would be required by the MOC scheme alone.

Under certain circumstances, the choice for the adaptive criterion used in the HMOC scheme may not be obvious, and the adaptive procedure may not lead to an optimal solution. In these cases, manual selection of either the MOC or the MMOC scheme may be more efficient. Thus, all three solution schemes are included in the current version of the MT3DMS code.

Summary of Available Solution Options

A unique aspect of MT3DMS is the multitude of solution options available in a single code as listed in Table 1. As discussed in the preceding sections, the advection term can be solved with the particle-tracking-based Eulerian-Lagrangian methods, the standard finite-difference method, or the third-order TVD (ULTIMATE) method. In the current version, the TVD option is always explicit, subject to a stability constraint. Whether the standard finite-difference solution is explicit or implicit depends on whether the Generalized Conjugate Gradient Solver (GCG) package is selected by the user (Chapter 4). If the GCG package is selected, the implicit finite-difference method is used to solve the advection term, without any stability constraints, using either the upstream or the central-in-space weighting. If the GCG package is not selected, the explicit finite-difference method is used instead, subject to the usual stability constraint. When the explicit finite-difference method is used, only the upstream weighting is available since the explicit scheme with central-in-space weighting is unconditionally unstable. The dispersion, sink/source, and reaction terms are solved together implicitly if the GCG package is selected, or explicitly if the GCG package is not selected. The MT3DMS code is structured in such a way that the developers of the add-on reaction packages for MT3DMS can implement their packages either implicitly or explicitly.

Table 1
Solution Options Available in the MT3DMS Code

Group	Solution Options for Advection[1]	Solution Options for Dispersion, Sink/Source, and Reaction[1]
A	Particle-tracking-based Eulerian-Lagrangian methods • MOC • MMOC • HMOC	Explicit finite-difference method
B	Particle Tracking Based Eulerian-Lagrangian Methods • MOC • MMOC • HMOC	*Implicit finite-difference method*
C	Explicit Finite-Difference Method • Upstream weighting	Explicit finite-difference method
D	*Implicit Finite-Difference Method* • *Upstream weighting* • *Central-in-space weighting*	*Implicit finite-difference method*
E	*Explicit 3rd-order TVD (ULTIMATE)*	Explicit finite-difference method
F	*Explicit 3rd-order TVD (ULTIMATE)*	*Implicit finite-difference method*

[1] New options are showing in italics.

4 Numerical Implementation

Spatial Discretization

The MT3DMS transport model follows the same spatial discretization convention as used by the United States Geological Survey modular 3-D finite-difference groundwater flow model, referred to as MODFLOW (McDonald and Harbaugh 1988; Harbaugh and McDonald 1996). An aquifer system is discretized into a mesh of blocks, or cells, the locations of which are described in terms of rows (I), columns (J), and layers (K) as illustrated in Figure 8. Following the convention used in Figure 8, the width of cells in the row direction, at a given column, J, is designated Δr_j; the width of cells in the column direction, at a given row, I, is designated Δc_i; and the thickness of cells in a given layer, K, is designated Δv_k. Thus a cell with indices (i, j, k) has a volume of $\Delta r_j \Delta c_i \Delta v_k$.

While the flow model does not require the designation of x, y, and z coordinate axes, the transport model does. In the MT3DMS model, an assumption is made that the x, y, and z coordinate axes are oriented along the row, column, and layer directions, respectively. The origin of the Cartesian coordinate system is located at the upper, left corner of the cell at the first row, first column, and first layer, or cell (1,1,1), as illustrated in Figure 9. Because the convention followed in both the MT3DMS and MODFLOW models is to number layers from the top down, the z axis is pointed downward in the direction of decreasing elevation. With the Cartesian coordinate system, Δr_j along the row direction is equivalent to Δx_j along the x axis; Δc_i along the column direction is equivalent to Δy_i along the y axis; and Δv_k along the layer direction is equivalent to Δz_k along the z axis.

The fixed grid system of the transport model is based on the block-centered formulation as illustrated in Figure 10. The block-centered formulation places a point, called a node, at the center of the cell, where the concentration or hydraulic head is calculated. The chemical and hydraulic parameters such as dispersivities or hydraulic conductivities are assumed to be uniform over the extent of a cell.

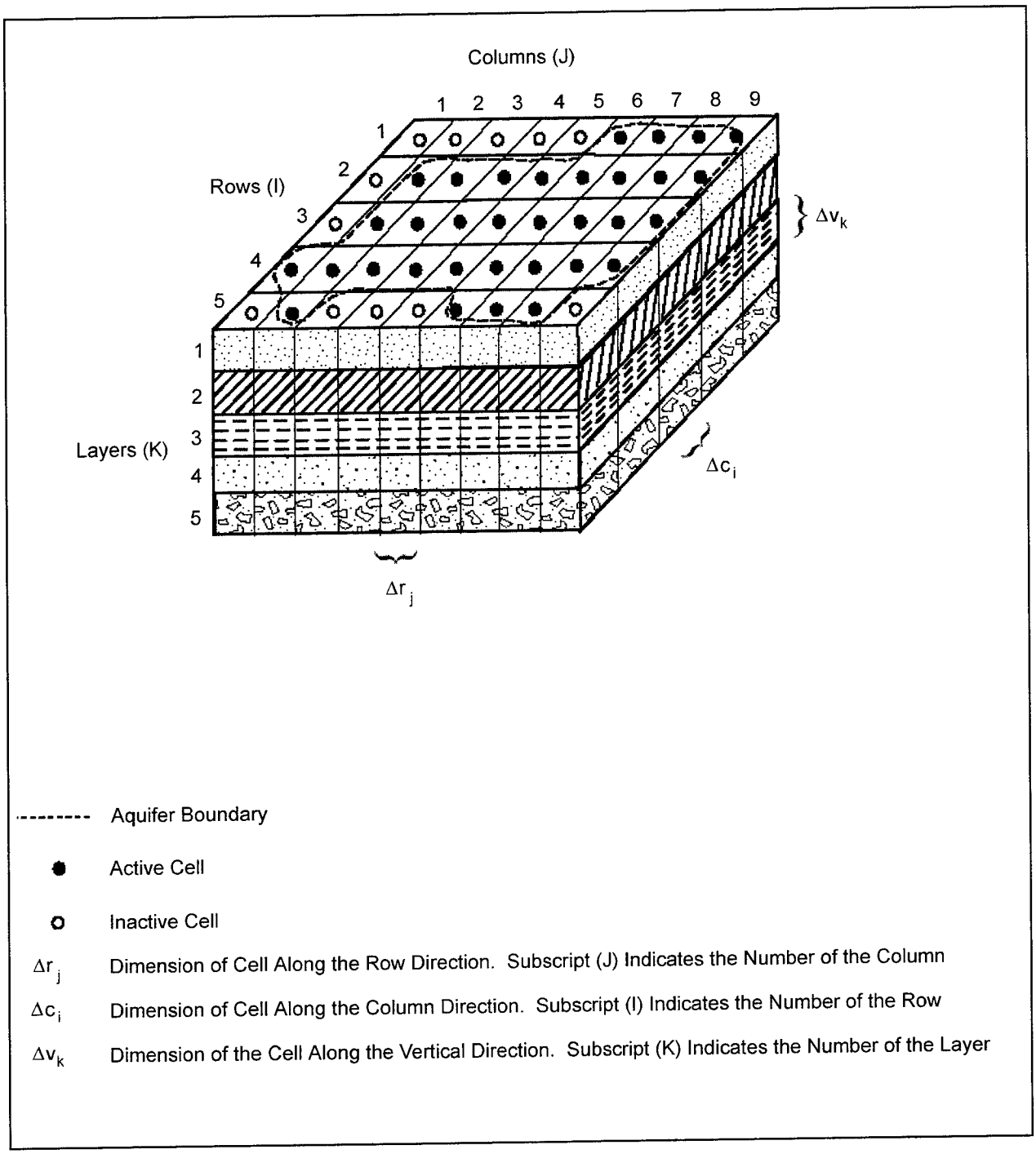


Figure 8. Spatial discretization of an aquifer system (after McDonald and Harbaugh 1988)

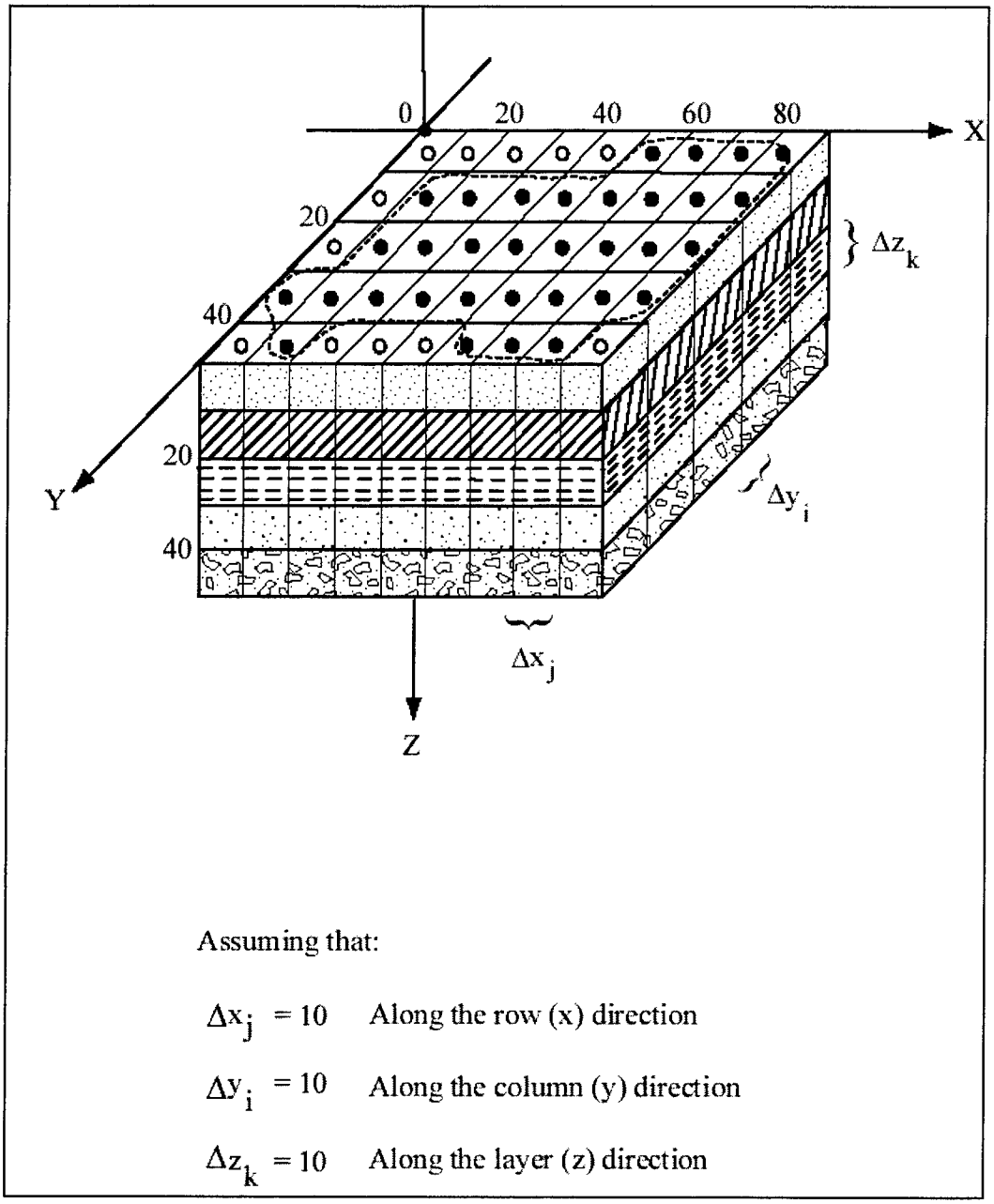


Figure 9. Cartesian coordinate system used in the MT3DMS transport model

As shown in Figures 8 and 9, an aquifer system is normally divided areally by two sets of parallel, orthogonal lines and vertically by parallel, horizontal planes so that each cell formed by the discretization is a rectangular block. To allow flexibility in handling geologic units of varying thickness, the MT3DMS transport model, as in MODFLOW, permits the use of a deformed mesh in the vertical direction as illustrated in Figure 11. The deformed vertical discretization, however, can introduce some numerical discretization error, especially in the transport simulation. Therefore, when the MT3DMS model is used with highly deformed vertical discretization, the simulation results should be evaluated carefully to ensure their accuracy.

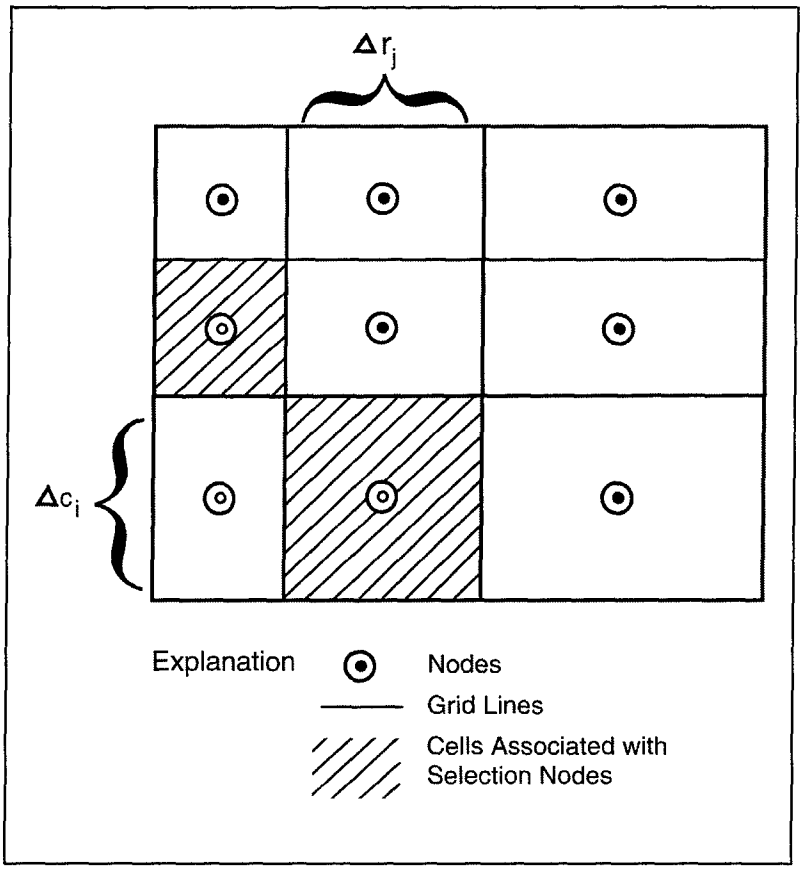


Figure 10. Diagram showing the block-centered grid system (after McDonald and Harbaugh 1988)

Temporal Discretization

In most flow models, such as MODFLOW, simulation time is usually divided into "stress periods" -- time intervals during which all external stress parameters (i.e., sink/source) are constant. Stress periods are, in turn, divided into time-steps if the simulation is transient. The time-steps within each stress period usually form a geometric progression. The length of each time-step is normally calculated by the program using the user-specified length of the stress period, the number of time-steps, and a time-step multiplier.

In the MT3DMS model, transport simulation is based on the head solution provided by a separate flow model. The length of the time-step used for the head solution is generally too large to be used as the length of the time-step for the transport solution, because the transport solution has either stability constraints and/or accuracy requirements that are more restrictive than those for the flow solution. Each time-step of the head solution is, therefore, divided further into smaller time increments, called transport steps, during which heads are considered constant. The length of each transport step can be specified in the model input, or determined by the model with an automatic step-size control procedure. The discretization of time is illustrated in Figure 12.

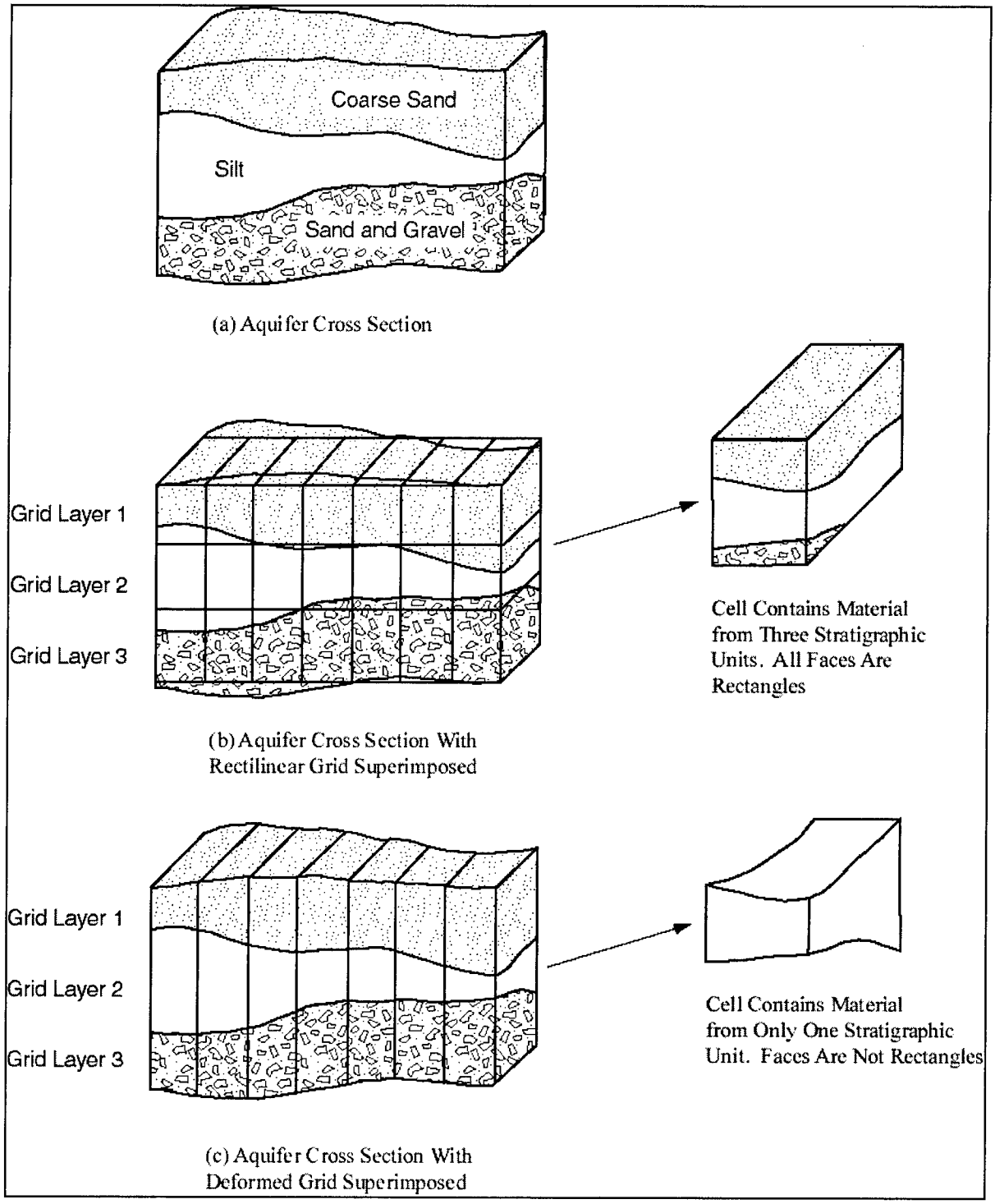


Figure 11. Schemes of vertical discretization (after McDonald and Harbaugh 1988)

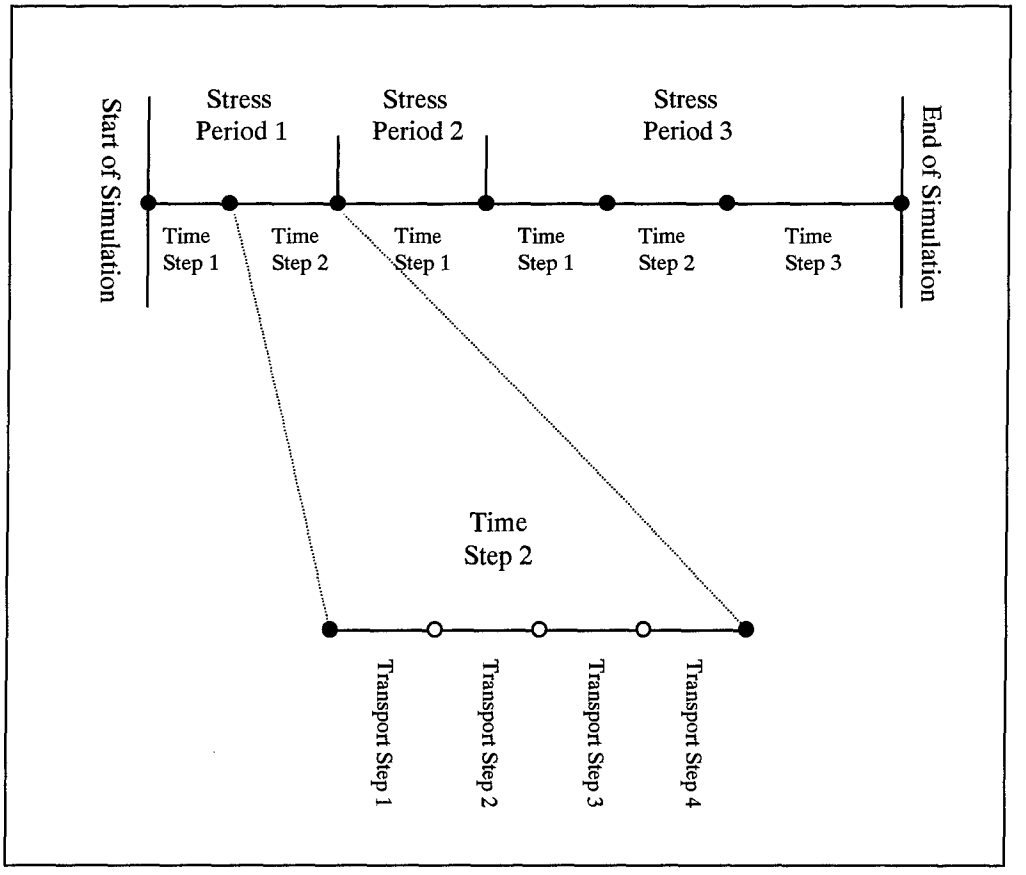


Figure 12. Discretization of simulation time in the transport model (top: simulation is divided into stress periods and time steps; bottom: time step is further divided into transport steps)

Generally, the flow and transport models should have the same number of stress periods. The number of time steps used in obtaining the flow solution in each stress period, referred to as "flow time step", should be specified in the transport model so that the velocity field and sink/source information can be updated in the transport model properly. Only when the flow model is steady-state with only one stress period and one time step, does the MT3DMS code allow the flow and transport models to have different numbers of stress periods. This is because, when the flow model is steady state, the velocity and sink/source information needs to be updated only once at the beginning of the simulation. In this case, the transport model is allowed to have as many stress periods as necessary to accommodate time-varying sink/source concentrations, while the flow rates of sinks/sources remain unchanged.

Implementation of the Finite-Difference Method

Implicit finite-difference formulation

As derived in Chapter 2, the Eulerian form of the transport equation, under the local equilibrium assumption, can be written as,

$$R\theta\frac{\partial C}{\partial t}=\frac{\partial}{\partial x_i}\left(\hat{D}_{ij}\frac{\partial C}{\partial x_j}\right)-\frac{\partial}{\partial x_i}(q_i C)+q_s C_s-q_s' C-\lambda_1\theta C-\lambda_2\rho_b\overline{C} \qquad (59)$$

where $\hat{D}_{ij}=\theta D_{ij}$ is the apparent dispersion coefficient tensor, and all other terms have been defined in Chapter 2. If nonequilibrium sorption or dual-porosity mass transfer is considered, a second equation is required to define the rate of change in the sorbed or immobile concentration as discussed later in subparagraph *e* "Chemical Reaction Component".

Applying the implicit finite-difference algorithm for node (*i,j,k*), the time derivative in Equation 59 can be approximated as follows,

$$R\theta\frac{\partial C}{\partial t}=R_{i,j,k}\theta_{i,j,k}\frac{C_{i,j,k}^{n+1}-C_{i,j,k}^{n}}{\Delta t} \qquad (60)$$

where index *n* represents the old time level with known concentrations, and index *n+1* represents the new time-level with unknown concentrations.

The advection term in Equation 59, expressed in component form, can be approximated as

$$\begin{aligned}
\frac{\partial}{\partial x_i}(q_i C)&=\frac{\partial}{\partial x}(q_x C)+\frac{\partial}{\partial y}(q_y C)+\frac{\partial}{\partial z}(q_z C)\\
&=q_{xi,j+1/2,k}\frac{\alpha_{xj+1/2}C_{i,j,k}^{n+1}+(1-\alpha_{xj+1/2})C_{i,j+1,k}^{n+1}}{\Delta x_j}-q_{xi,j-1/2,k}\frac{\alpha_{xj-1/2}C_{i,j-1,k}^{n+1}+(1-\alpha_{xj-1/2})C_{i,j,k}^{n+1}}{\Delta x_j}\\
&+q_{yi+1/2,j,k}\frac{\alpha_{yi+1/2}C_{i,j,k}^{n+1}+(1-\alpha_{yi+1/2})C_{i+1,j,k}^{n+1}}{\Delta y_i}-q_{yi-1/2,j,k}\frac{\alpha_{yi-1/2}C_{i-1,j,k}^{n+1}+(1-\alpha_{yi-1/2})C_{i,j,k}^{n+1}}{\Delta y_i}\\
&+q_{zi,j,k+1/2}\frac{\alpha_{zk+1/2}C_{i,j,k}^{n+1}+(1-\alpha_{zk+1/2})C_{i,j,k+1}^{n+1}}{\Delta z_k}-q_{zi,j,k-1/2}\frac{\alpha_{zk-1/2}C_{i,j,k-1}^{n+1}+(1-\alpha_{zk-1/2})C_{i,j,k}^{n+1}}{\Delta z_k}
\end{aligned} \qquad (61)$$

where $\alpha_{xj\pm1/2}$, $\alpha_{yi\pm1/2}$, and $\alpha_{zk\pm1/2}$ are the spatial weighting factors for the advection term. If the upstream weighting scheme is used, the weighting factors are defined by

$$\alpha_{xj+1/2}=\begin{cases}1 & q_{xi,j+1/2,k}>0\\ 0 & q_{xi,j+1/2,k}<0\end{cases}$$

$$\alpha_{yi+1/2}=\begin{cases}1 & q_{yi+1/2,j,k}>0\\ 0 & q_{yi+1/2,j,k}<0\end{cases} \qquad (62)$$

$$\alpha_{zk+1/2}=\begin{cases}1 & q_{zi,j,k+1/2}>0\\ 0 & q_{zi,j,k+1/2}<0\end{cases}$$

If the central-in-space weighting scheme is used, the weighting factors are defined by

$$\alpha_{xj+1/2}=\frac{\Delta x_{j+1}}{\Delta x_j+\Delta x_{j+1}}$$

$$\alpha_{yi+1/2}=\frac{\Delta y_{i+1}}{\Delta y_i+\Delta y_{i+1}} \qquad (63)$$

$$\alpha_{zk+1/2}=\frac{\Delta z_{k+1}}{\Delta z_k+\Delta z_{k+1}}$$

The dispersion term in Equation 59, expressed in component form, can be approximated as follows

$$\begin{aligned}
&\frac{\partial}{\partial x_i}\left(\hat{D}_{ij}\frac{\partial C}{\partial x_j}\right)=\frac{\partial}{\partial x}\left(\hat{D}_{xx}\frac{\partial C}{\partial x}\right)+\frac{\partial}{\partial x}\left(\hat{D}_{xy}\frac{\partial C}{\partial y}\right)+\frac{\partial}{\partial x}\left(\hat{D}_{xz}\frac{\partial C}{\partial z}\right)\\
&+\frac{\partial}{\partial y}\left(\hat{D}_{yx}\frac{\partial C}{\partial x}\right)+\frac{\partial}{\partial y}\left(\hat{D}_{yy}\frac{\partial C}{\partial y}\right)+\frac{\partial}{\partial y}\left(\hat{D}_{yz}\frac{\partial C}{\partial z}\right)\\
&+\frac{\partial}{\partial z}\left(\hat{D}_{zx}\frac{\partial C}{\partial x}\right)+\frac{\partial}{\partial z}\left(\hat{D}_{zy}\frac{\partial C}{\partial y}\right)+\frac{\partial}{\partial z}\left(\hat{D}_{zz}\frac{\partial C}{\partial z}\right)\\
&=\hat{D}_{xx_{i,j+1/2,k}}\frac{C^{n+1}_{i,j+1,k}-C^{n+1}_{i,j,k}}{\Delta x_j\left(0.5\Delta x_j+0.5\Delta x_{j+1}\right)}-\hat{D}_{xx_{i,j-1/2,k}}\frac{C^{n+1}_{i,j,k}-C^{n+1}_{i,j-1,k}}{\Delta x_j\left(0.5\Delta x_{j-1}+0.5\Delta x_j\right)}\\
&+\hat{D}_{xy_{i,j+1/2,k}}\frac{\omega_{x_{j+1/2}}C^{n+1}_{i+1,j,k}+\left(1-\omega_{x_{j+1/2}}\right)C^{n+1}_{i+1,j+1,k}-\omega_{x_{j+1/2}}C^{n+1}_{i-1,j,k}-\left(1-\omega_{x_{j+1/2}}\right)C^{n+1}_{i-1,j+1,k}}{\Delta x_j\left(0.5\Delta y_{i-1}+\Delta y_i+0.5\Delta y_{i+1}\right)}\\
&-\hat{D}_{xy_{i,j-1/2,k}}\frac{\omega_{x_{j-1/2}}C^{n+1}_{i+1,j-1,k}+\left(1-\omega_{x_{j-1/2}}\right)C^{n+1}_{i+1,j,k}-\omega_{x_{j-1/2}}C^{n+1}_{i-1,j-1,k}-\left(1-\omega_{x_{j-1/2}}\right)C^{n+1}_{i-1,j,k}}{\Delta x_j\left(0.5\Delta y_{i-1}+\Delta y_i+0.5\Delta y_{i+1}\right)}\\
&+\hat{D}_{xz_{i,j+1/2,k}}\frac{\omega_{x_{j+1/2}}C^{n+1}_{i,j,k+1}+\left(1-\omega_{x_{j+1/2}}\right)C^{n+1}_{i,j+1,k+1}-\omega_{x_{j+1/2}}C^{n+1}_{i,j,k-1}-\left(1-\omega_{x_{j+1/2}}\right)C^{n+1}_{i,j+1,k-1}}{\Delta x_j\left(0.5\Delta z_{k-1}+\Delta z_k+0.5\Delta z_{k+1}\right)}
\end{aligned} \qquad (64)$$

$$
\begin{aligned}
&-\hat{D}_{xz_{i,j-1/2,k}}\frac{\omega_{x_{j-1/2}}C^{n+1}_{i,j-1,k+1}+\left(1-\omega_{x_{j-1/2}}\right)C^{n+1}_{i,j,k+1}-\omega_{x_{j-1/2}}C^{n+1}_{i,j-1,k-1}-\left(1-\omega_{x_{j-1/2}}\right)C^{n+1}_{i,j,k-1}}{\Delta x_j\left(0.5\Delta z_{k-1}+\Delta z_k+0.5\Delta z_{k+1}\right)}\\
&+\hat{D}_{yx_{i+1/2,j,k}}\frac{\omega_{y_{i+1/2}}C^{n+1}_{i,j+1,k}+\left(1-\omega_{y_{i+1/2}}\right)C^{n+1}_{i+1,j+1,k}-\omega_{y_{i+1/2}}C^{n+1}_{i,j-1,k}-\left(1-\omega_{y_{i+1/2}}\right)C^{n+1}_{i+1,j-1,k}}{\Delta y_i\left(0.5\Delta x_{j-1}+\Delta x_j+0.5\Delta x_{j+1}\right)}\\
&-\hat{D}_{yx_{i-1/2,j,k}}\frac{\omega_{y_{i-1/2}}C^{n+1}_{i-1,j+1,k}+\left(1-\omega_{y_{i-1/2}}\right)C^{n+1}_{i,j+1,k}-\omega_{y_{i-1/2}}C^{n+1}_{i-1,j-1,k}-\left(1-\omega_{y_{i-1/2}}\right)C^{n+1}_{i,j-1,k}}{\Delta y_i\left(0.5\Delta x_{j-1}+\Delta x_j+0.5\Delta x_{j+1}\right)}\\
&+\hat{D}_{yy_{i+1/2,j,k}}\frac{C^{n+1}_{i+1,j,k}-C^{n+1}_{i,j,k}}{\Delta y_i\left(0.5\Delta y_i+0.5\Delta y_{i+1}\right)}-\hat{D}_{yy_{i-1/2,j,k}}\frac{C^{n+1}_{i,j,k}-C^{n+1}_{i-1,j,k}}{\Delta y_i\left(0.5\Delta y_{i-1}+0.5\Delta y_i\right)}\\
&+\hat{D}_{yz_{i+1/2,j,k}}\frac{\omega_{y_{i+1/2}}C^{n+1}_{i,j,k+1}+\left(1-\omega_{y_{i+1/2}}\right)C^{n+1}_{i+1,j,k+1}-\omega_{y_{i+1/2}}C^{n+1}_{i,j,k-1}-\left(1-\omega_{y_{i+1/2}}\right)C^{n+1}_{i+1,j,k-1}}{\Delta y_i\left(0.5\Delta z_{k-1}+\Delta z_k+0.5\Delta z_{k+1}\right)}\\
&-\hat{D}_{yz_{i-1/2,j,k}}\frac{\omega_{y_{i-1/2}}C^{n+1}_{i-1,j,k+1}+\left(1-\omega_{y_{i-1/2}}\right)C^{n+1}_{i,j,k+1}-\omega_{y_{i-1/2}}C^{n+1}_{i-1,j,k-1}-\left(1-\omega_{y_{i-1/2}}\right)C^{n+1}_{i,j,k-1}}{\Delta y_i\left(0.5\Delta z_{k-1}+\Delta z_k+0.5\Delta z_{k+1}\right)}\\
&+\hat{D}_{zx_{i,j,k+1/2}}\frac{\omega_{z_{k+1/2}}C^{n+1}_{i,j+1,k}+\left(1-\omega_{z_{k+1/2}}\right)C^{n+1}_{i,j+1,k+1}-\omega_{z_{k+1/2}}C^{n+1}_{i,j-1,k}-\left(1-\omega_{z_{k+1/2}}\right)C^{n+1}_{i,j-1,k+1}}{\Delta z_k\left(0.5\Delta x_{j-1}+\Delta x_j+0.5\Delta x_{j+1}\right)}\\
&-\hat{D}_{zx_{i,j,k-1/2}}\frac{\omega_{z_{k-1/2}}C^{n+1}_{i,j+1,k-1}+\left(1-\omega_{z_{k-1/2}}\right)C^{n+1}_{i,j+1,k}-\omega_{z_{k-1/2}}C^{n+1}_{i,j-1,k-1}-\left(1-\omega_{z_{k-1/2}}\right)C^{n+1}_{i,j-1,k}}{\Delta z_k\left(0.5\Delta x_{j-1}+\Delta x_j+0.5\Delta x_{j+1}\right)}\\
&+\hat{D}_{zy_{i,j,k+1/2}}\frac{\omega_{z_{k+1/2}}C^{n+1}_{i+1,j,k}+\left(1-\omega_{z_{k+1/2}}\right)C^{n+1}_{i+1,j,k+1}-\omega_{z_{k+1/2}}C^{n+1}_{i-1,j,k}-\left(1-\omega_{z_{k+1/2}}\right)C^{n+1}_{i-1,j,k}}{\Delta z_k\left(0.5\Delta y_{i-1}+\Delta y_i+0.5\Delta y_{i+1}\right)}\\
&-\hat{D}_{zy_{i,j,k-1/2}}\frac{\omega_{z_{k-1/2}}C^{n+1}_{i+1,j,k-1}+\left(1-\omega_{z_{k-1/2}}\right)C^{n+1}_{i+1,j,k}-\omega_{z_{k-1/2}}C^{n+1}_{i-1,j,k-1}-\left(1-\omega_{z_{k-1/2}}\right)C^{n+1}_{i-1,j,k}}{\Delta z_k\left(0.5\Delta y_{i-1}+\Delta y_i+0.5\Delta y_{i+1}\right)}\\
&+\hat{D}_{zz_{i,j,k+1/2}}\frac{C^{n+1}_{i,j,k+1}-C^{n+1}_{i,j,k}}{\Delta z_k\left(0.5\Delta z_k+0.5\Delta z_{k+1}\right)}-\hat{D}_{zz_{i,j,k-1/2}}\frac{C^{n+1}_{i,j,k}-C^{n+1}_{i,j,k-1}}{\Delta z_k\left(0.5\Delta z_{k-1}+0.5\Delta z_k\right)}
\end{aligned}
\tag{64}
$$

where ω_x, ω_y, and ω_z are the spatial weighting factors used to compute the concentration value at a cell interface when evaluating a dispersion cross term, defined as

$$
\begin{aligned}
\omega_{x_{j+1/2}} &= \frac{\Delta x_{j+1}}{\Delta x_j+\Delta x_{j+1}}\\
\omega_{y_{i+1/2}} &= \frac{\Delta y_{i+1}}{\Delta y_i+\Delta y_{i+1}}\\
\omega_{z_{k+1/2}} &= \frac{\Delta z_{k+1}}{\Delta z_k+\Delta z_{k+1}}
\end{aligned}
\tag{65}
$$

Substituting Equations 60, 61, and 64 into 59 and multiplying both sides by $\Delta x_j \Delta y_i \Delta z_k$ (i.e., the volume of cell (*i,j,k*)) yields the following general finite-difference equation for cell (*i,j,k*),

$$\begin{aligned}
&A^1_{i,j,k}C^{n+1}_{i,j,k} + A^2_{i,j,k}C^{n+1}_{i,j,k-1} + A^3_{i,j,k}C^{n+1}_{i,j,k+1} + A^4_{i,j,k}C^{n+1}_{i-1,j,k} + A^5_{i,j,k}C^{n+1}_{i+1,j,k} + A^6_{i,j,k}C^{n+1}_{i,j-1,k} \\
&+ A^7_{i,j,k}C^{n+1}_{i,j+1,k} + A^8_{i,j,k}C^{n+1}_{i-1,j,k-1} + A^9_{i,j,k}C^{n+1}_{i,j-1,k-1} + A^{10}_{i,j,k}C^{n+1}_{i,j+1,k-1} + A^{11}_{i,j,k}C^{n+1}_{i+1,j,k-1} \\
&+ A^{12}_{i,j,k}C^{n+1}_{i-1,j,k+1} + A^{13}_{i,j,k}C^{n+1}_{i,j-1,k+1} + A^{14}_{i,j,k}C^{n+1}_{i,j+1,k+1} + A^{15}_{i,j,k}C^{n+1}_{i+1,j,k+1} + A^{16}_{i,j,k}C^{n+1}_{i,j-1,k-1} \\
&+ A^{17}_{i,j,k}C^{n+1}_{i-1,j+1,k} + A^{18}_{i,j,k}C^{n+1}_{i+1,j-1,k} + A^{19}_{i,j,k}C^{n+1}_{i+1,j+1,k} = b_{i,j,k}
\end{aligned} \tag{66}$$

where A is the coefficient matrix and b is the right-hand-side vector containing all the known quantities. The 19 coefficients in the coefficient matrix A and the entries in vector b are listed below in five groups; basic, advection, dispersion, sink/source, and chemical reaction.

a. Basic component. The basic components of matrix A and right-hand-side vector b stem from the approximation of the time derivative, i.e.,

$$A^1_{i,j,k} = -R_{i,j,k}\theta_{i,j,k}\Delta x_j \Delta y_i \Delta z_k / \Delta t \tag{67}$$

$$b_{ijk} = -R_{ijk}\theta_{ijk}\Delta x_j \Delta y_i \Delta z_k C^n_{ijk} / \Delta t \tag{68}$$

where the retardation factor $R_{i,j,k}$ is set equal to one if sorption is not present or is not controlled by local equilibrium. When nonlinear sorption is simulated, $R_{i,j,k}$ is updated with the concentration values calculated in the preceding iteration.

For inactive and constant-concentration cells, we always have $C^{n+1}_{i,j,k} = C^n_{i,j,k}$ which is obtained by setting

$$A^1_{i,j,k} = -1 \tag{69}$$

$$b_{i,j,k} = -C^n_{i,j,k} \tag{70}$$

b. Advection component. The contributions to the coefficient matrix A from advection are listed below:

$$\begin{aligned}
A^1_{i,j,k} = &-\alpha x_{j+1/2} Q_{xi,j+1/2,k} + \left(1-\alpha x_{j-1/2}\right) Q_{xi,j-1/2,k} - \alpha y_{i+1/2} Q_{yi+1/2,j,k} \\
&+ \left(1-\alpha y_{i-1/2}\right) Q_{yi-1/2,j,k} - \alpha z_{k+1/2} Q_{zi,j,k+1/2} + \left(1-\alpha z_{k-1/2}\right) Q_{zi,j,k-1/2}
\end{aligned} \tag{71a}$$

$$A^2_{i,j,k} = \alpha z_{k-1/2} Q_{zi,j,k-1/2} \tag{71b}$$

$$A^3_{i,j,k} = -\left(1-\alpha_{zk+1/2}\right) Q_{zi,j,k+1/2} \tag{71c}$$

$$A^4_{i,j,k} = \alpha_{y_{i-1/2}} Q_{y_{i-1/2},j,k} \tag{71d}$$

$$A^5_{i,j,k} = -\left(1-\alpha_{y_{i+1/2}}\right) Q_{yi+1/2,j,k} \tag{71e}$$

$$A^6_{i,j,k} = \alpha_{x_{j-1/2}} Q_{xi,j-1/2,k} \tag{71f}$$

$$A^7_{i,j,k} = -\left(1-\alpha_{x_{j+1/2}}\right) Q_{xi,j+1/2,k} \tag{71g}$$

where Q is the volumetric flow rate across a cell interface between two neighboring cells, equal to the specific discharge times the interface area. For example, $Q_{xi,j+1/2,k} = q_{xi,j+1/2,k}\Delta y_i \Delta z_k$ is the volumetric flow rate across the interface (i,j+1/2,k) between node (i,j,k) and (i,j+1,k). The volumetric flow rates across each cell interface are known from the flow model.

c. *Dispersion component.* The contributions to the coefficient matrix A from dispersion are list below:

$$A^1_{i,j,k} = -\tilde{D}_{xxijk} - \tilde{D}_{xxij-1k} - \tilde{D}_{yy_{ijk}} - \tilde{D}_{yy_{i-1jk}} - \tilde{D}_{zzijk} - \tilde{D}_{zzijk-1} \tag{72a}$$

$$\begin{aligned} A^2_{i,j,k} = {} & \tilde{D}_{zz_{i,j,k-1}} - \tilde{D}_{xzi,j,k}\omega_{xj+1/2} + \tilde{D}_{xzi,j-1,k}(1-\omega_{xj-1/2}) \\ & - \tilde{D}_{yzi,j,k}\omega_{yi+1/2} + \tilde{D}_{yzi-1,j,k}(1-\omega_{yi-1/2}) \end{aligned} \tag{72b}$$

$$\begin{aligned} A^3_{i,j,k} = {} & \tilde{D}_{zz_{i,j,k}} + \tilde{D}_{xzi,j,k}\omega_{xj+1/2} - \tilde{D}_{xzi,j-1,k}(1-\omega_{xj-1/2}) \\ & + \tilde{D}_{yzi,j,k}\omega_{yi+1/2} - \tilde{D}_{yzi-1,j,k}(1-\omega_{yi-1/2}) \end{aligned} \tag{72c}$$

$$\begin{aligned} A^4_{i,j,k} = {} & \tilde{D}_{yy_{i-1,j,k}} - \tilde{D}_{xyi,j,k}\omega_{xj+1/2} + \tilde{D}_{xyi,j-1,k}(1-\omega_{xj-1/2}) \\ & - \tilde{D}_{zyi,j,k}\omega_{zk+1/2} + \tilde{D}_{zyi-1,j,k}(1-\omega_{zk-1/2}) \end{aligned} \tag{72d}$$

$$\begin{aligned} A^5_{i,j,k} = {} & \tilde{D}_{yy_{i,j,k}} + \tilde{D}_{xyi,j,k}\omega_{xj+1/2} - \tilde{D}_{xyi,j-1,k}(1-\omega_{xj-1/2}) \\ & + \tilde{D}_{zyi,j,k}\omega_{zk+1/2} - \tilde{D}_{zyi-1,j,k}(1-\omega_{zk-1/2}) \end{aligned} \tag{72e}$$

$$\begin{aligned} A^6_{i,j,k} = {} & \tilde{D}_{xx_{i,j-1,k}} - \tilde{D}_{yxi,j,k}\omega_{yi+1/2} + \tilde{D}_{yxi-1,j,k}(1-\omega_{yi-1/2}) \\ & - \tilde{D}_{zxi,j,k}\omega_{zk+1/2} + \tilde{D}_{zxi,j,k-1}(1-\omega_{zk-1/2}) \end{aligned} \tag{72f}$$

$$A^{7}_{i,j,k} = \tilde{D}_{xx_{i,j,k}} + \tilde{D}_{yxi,j,k}\omega_{yi+1/2} - \tilde{D}_{yxi-1,j,k}(1-\omega_{yi-1/2}) + \tilde{D}_{zxi,j,k}\omega_{zk+1/2} - \tilde{D}_{zxi,j,k-1}(1-\omega_{zk-1/2}) \quad (72g)$$

$$A^{8}_{i,j,k} = \tilde{D}_{yzi-1,j,k}\omega_{yi-1/2} + \tilde{D}_{zyi,j,k-1}\omega_{zk-1/2} \quad (72h)$$

$$A^{9}_{i,j,k} = \tilde{D}_{xzi,j-1,k}\omega_{xj-1/2} + \tilde{D}_{zxi,j,k-1}\omega_{zk-1/2} \quad (72i)$$

$$A^{10}_{i,j,k} = -\tilde{D}_{xzi,j,k}\left(1-\omega_{yi-1/2}\right) - \tilde{D}_{zxi,j,k-1}\omega_{zk-1/2} \quad (72j)$$

$$A^{11}_{i,j,k} = -\tilde{D}_{yzi,j,k}\left(1-\omega_{yi+1/2}\right) - \tilde{D}_{zyi,j,k-1}\omega_{zk-1/2} \quad (72k)$$

$$A^{12}_{i,j,k} = -\tilde{D}_{yzi-1,j,k}\omega_{yi-1/2} - \tilde{D}_{zyi,j,k}\left(1-\omega_{zk+1/2}\right) \quad (72l)$$

$$A^{13}_{i,j,k} = -\tilde{D}_{xyi,j-1,k}\omega_{xj-1/2} - \tilde{D}_{zxi,j,k}\left(1-\omega_{zk+1/2}\right) \quad (72m)$$

$$A^{14}_{i,j,k} = \tilde{D}_{xzi,j,k}\left(1-\omega_{xj+1/2}\right) + \tilde{D}_{zxi,j,k}\left(1-\omega_{zk+1/2}\right) \quad (72n)$$

$$A^{15}_{i,j,k} = \tilde{D}_{yzi,j,k}\left(1-\omega_{yi+1/2}\right) + \tilde{D}_{zyi,j,k}\left(1-\omega_{zk+1/2}\right) \quad (72o)$$

$$A^{16}_{i,j,k} = \tilde{D}_{xyi,j-1,k}\omega_{xj-1/2} + \tilde{D}_{yxi-1,j,k}\omega_{yi-1/2} \quad (72p)$$

$$A^{17}_{i,j,k} = -\tilde{D}_{xyi,j,k}\left(1-\omega_{xj+1/2}\right) - \tilde{D}_{yxi-1,j,k}\omega_{yi-1/2} \quad (72q)$$

$$A^{18}_{i,j,k} = -\tilde{D}_{xyi,j-1,k}\omega_{xj-1/2} - \tilde{D}_{yxi,j,k}\left(1-\omega_{yi+1/2}\right) \quad (72r)$$

$$A^{19}_{i,j,k} = \tilde{D}_{xyi,j,k}\left(1-\omega_{xj+1/2}\right) + \tilde{D}_{yxi,j,k}\left(1-\omega_{yi+1/2}\right) \quad (72s)$$

where $\tilde{D}$ represents the "dispersion conductance", defined as the apparent dispersion coefficient $\hat{D}$ times the cross section area across which the dispersion flux is evaluated, divided by the distance over which the concentration gradient is calculated. For example, the dispersion conductances for the dispersive flux across the interface (i,j+1/2,k) are defined as follows:

$$\tilde{D}_{xx_{i,j+1/2,k}} = \hat{D}_{xx_{i,j+1/2,k}} \frac{\Delta y_i \Delta z_k}{\left(0.5\Delta x_j + 0.5\Delta x_{j+1}\right)} \quad (73a)$$

$$\tilde{D}_{xy_{i,j+1/2,k}} = \hat{D}_{xy_{i,j+1/2,k}} \frac{\Delta y_i \Delta z_k}{\left(0.5\Delta y_{i-1} + \Delta y_i + 0.5\Delta y_{i+1}\right)} \quad (73b)$$

$$\tilde{D}_{xz_{i,j+1/2,k}} = \hat{D}_{xz_{i,j+1/2,k}} \frac{\Delta y_i \Delta z_k}{(0.5\Delta z_{k-1} + \Delta z_k + 0.5\Delta z_{k+1})} \quad (73c)$$

Similar expressions for the dispersion conductances at other interfaces can be defined.

The components of the apparent dispersion coefficient are calculated from the Darcy flux (specific discharge) components and the dispersivities according to equations given in Chapter 2. For example, the values of $\hat{D}_{xx}, \hat{D}_{xy}$, and $\hat{D}_{xz}$ are evaluated at the cell interfaces between (i, j, k) and (i, j+1, k) along the x direction. The x-component of the Darcy flux is known directly from the flow model. The y- and z-components are interpolated from Darcy fluxes at the interfaces along the y- and z-directions (Figure 13):

$$q_{y_{i,j+1/2,k}} = \frac{1}{2}\left[\left(q_{y_{i-1/2,j,k}} + q_{y_{i+1/2,j,k}}\right)\omega_{x_{j+1/2}} + \left(q_{y_{i-1/2,j+1,k}} + q_{y_{i+1/2,j+1,k}}\right)\left(1-\omega_{x_{j-1/2}}\right)\right] \quad (74a)$$

$$q_{z_{i,j+1/2,k}} = \frac{1}{2}\left[\left(q_{z_{i,j,k-1/2}} + q_{z_{i,j,k+1/2}}\right)\omega_{x_{j+1/2}} + \left(q_{z_{i,j+1,k-1/2}} + q_{z_{i,j+1,k+1/2}}\right)\left(1-\omega_{x_{j+1/2}}\right)\right] \quad (74b)$$

$$q_{i,j+1/1,k} = \sqrt{q^2_{x_{i,j+1/2,k}} + q^2_{y_{i,j+1/2,k}} + q^2_{z_{i,j+1/2,k}}} \quad (74c)$$

The longitudinal and transverse dispersivities are entered into the model on a cell-by-cell basis. Their values at the cell interfaces are interpolated accordingly using the cell weighting factors:

$$\alpha_{L_{i,j+1/2,k}} = \alpha_{L_{i,j,k}}\omega_{x_{j+1/2}} + \alpha_{L_{i,j+1,k}}\left(1-\omega_{x_{j+1/2}}\right) \quad (74d)$$

$$\alpha_{TH_{i,j+1/2,k}} = \alpha_{TH_{i,j,k}}\omega_{x_{j+1/2}} + \alpha_{TH_{i,j+1,k}}\left(1-\omega_{x_{j+1/2}}\right) \quad (74e)$$

$$\alpha_{TV_{i,j+1/2,k}} = \alpha_{TV_{i,j,k}}\omega_{x_{j+1/2}} + \alpha_{TV_{i,j+1,k}}\left(1-\omega_{x_{j+1/2}}\right) \quad (74f)$$

The values of $\hat{D}_{xx}, \hat{D}_{xy}$, and $\hat{D}_{xz}$ at the cell interface $(i, j+1/2, k)$ can be computed as

$$\hat{D}_{xx_{i,j+1/2,k}} = \alpha_{L_{i,j+1/2,k}} q^2_{x_{i,j+1/2,k}} / q_{i,j+1/2,k} + \alpha_{TH_{i,j+1/2,k}} q^2_{y_{i,j+1/2,k}} / q_{i,j+1/2,k} + \alpha_{TV_{i,j+1/2,k}} q^2_{z_{i,j+1/2,k}} / q_{i,j+1/2,k} \quad (74g)$$

$$\hat{D}_{xy_{i,j+1/2,k}} = \left(\alpha_{L_{i,j+1/2,k}} - \alpha_{TH_{i,j+1/2,k}}\right) q_{x_{i,j+1/2,k}} q_{y_{i,j+1/2,k}} / q_{i,j+1/2,k} \quad (74h)$$

$$\hat{D}_{xz_{i,j+1/2,k}} = \left(\alpha_{L_{i,j+1/2,k}} - \alpha_{TV_{i,j+1/2,k}}\right) q_{x_{i,j+1/2,k}} q_{y_{i,j+1/2,k}} / q_{i,j+1/2,k} \quad (74i)$$

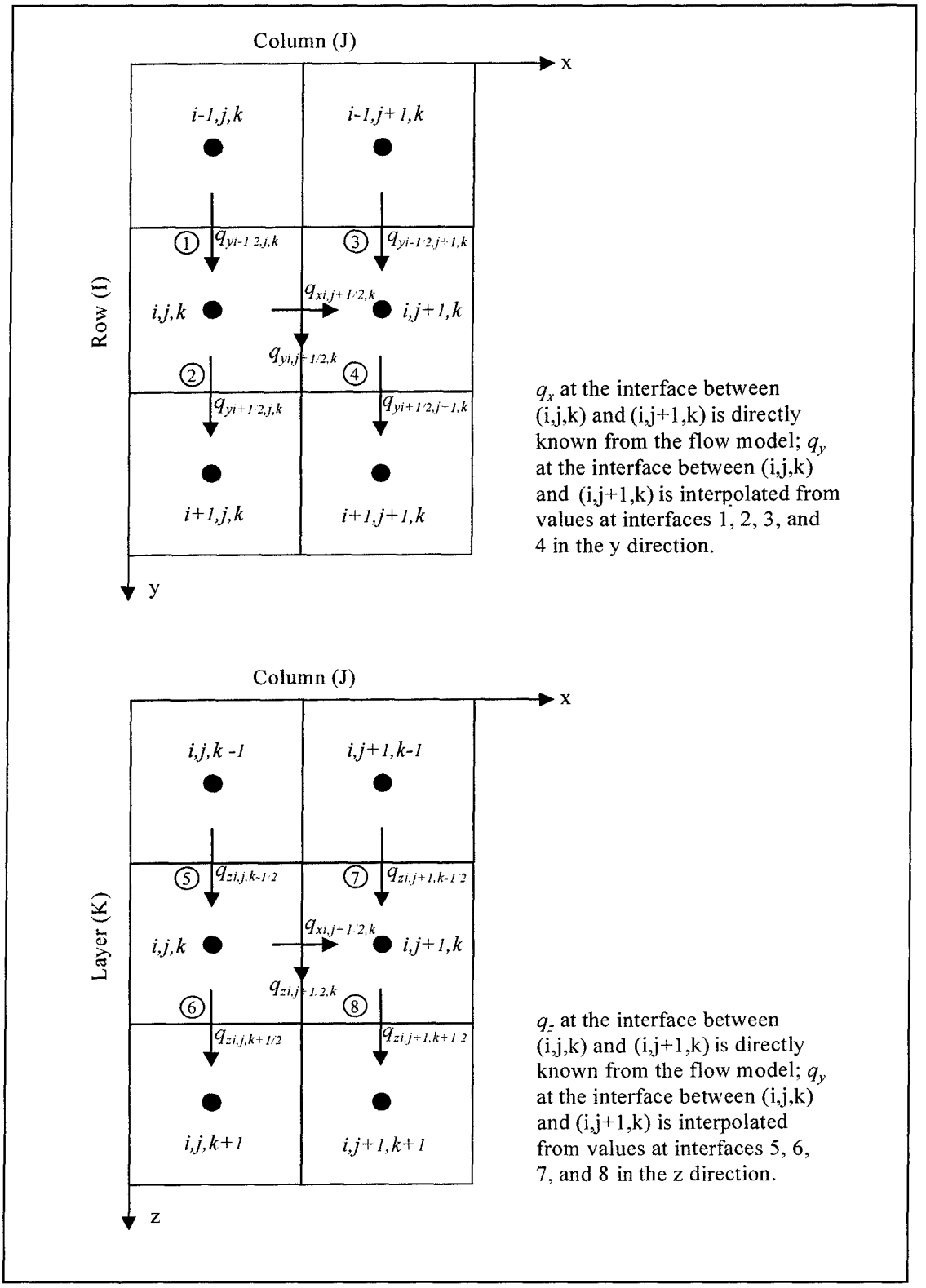


Figure 13. Evaluation of the velocity components at the cell interfaes in the x-direction for calculating components of the dispersion coefficients D_{xx}, D_{xy}, and D_{xz}

Similarly, the components of $\hat{D}_{yx}, \hat{D}_{yy}$, and $\hat{D}_{yz}$ are calculated at the cell interfaces along the y-direction and $\hat{D}_{zx}, \hat{D}_{zy}$, and $\hat{D}_{zz}$ at the cell interfaces along the z-direction, using the same procedure as listed from Equations 74a through 74i.

d. Sink/source component. The contributions to the coefficient matrix A and the right-hand-side vector from sinks/sources are listed below:

$$A^1_{i,j,k} = Q^-_{s(i,j,k)} - Q'_{s(i,j,k)} \tag{75}$$

$$b_{i,j,k} = -Q^+_{s(i,j,k)} C_{s(i,j,k)} \tag{76}$$

where

$Q^+_{s(i,j,k)} = q_{s(i,j,k)} \Delta x_j \Delta y_i \Delta z_k$ $(q_{s(i,j,k)} > 0)$ is the volumetric flow rate of a fluid source

$Q^-_{s(i,j,k)} = q_{s(i,j,k)} \Delta x_j \Delta y_i \Delta z_k$ $(q_{s(i,j,k)} < 0)$ is the volumetric flow rate of a fluid sink

$Q'_{s(i,j,k)} = q'_{s(i,j,k)} \Delta x_j \Delta y_i \Delta z_k$ is the rate of transient groundwater storage change

Both Q_s and Q'_s are known from the flow model.

e. Chemical reaction component. The contributions to the coefficient matrix A and the right-hand-side vector b from chemical reactions depend on how the sorbed concentration is handled.

For equilibrium-controlled linear sorption, $\overline{C}^{n+1}_{i.j.k} = K_d C^{n+1}_{i,j,k}$, and it can be incorporated into the coefficient matrix:

$$A^1_{i,j,k} = -\lambda_{1(i,j,k)} \theta_{i,j,k} \Delta x_j \Delta y_i \Delta z_k - \lambda_{2(i,j,k)} \rho_{b(i,j,k)} \Delta x_j \Delta y_i \Delta z_k K_d \tag{77}$$

For equilibrium-controlled nonlinear sorption, the sorbed concentration is approximated by the solute concentration calculated in the preceding iteration so that it can be lumped into the right-hand-side vector:

$$A^1_{i,j,k} = -\lambda_{1(i,j,k)} \theta_{i,j,k} \Delta x_j \Delta y_i \Delta z_k \tag{78}$$

$$b_{i,j,k} = \lambda_{2(i,j,k)} \rho_{b(i,j,k)} \Delta x_j \Delta y_i \Delta z_k \overline{C}^{n+1}_{i,j,k}$$

For nonequilibrium sorption modeled as first-order kinetic mass transfer, a second equation is needed to define the sorbed, as discussed in Chapter 2:

$$\theta\frac{\partial C}{\partial t}+\rho_b\frac{\partial \overline{C}}{\partial t}=L(C)-\lambda_1\theta C-\lambda_2\rho_b\overline{C} \tag{79}$$

$$\rho_b\frac{\partial \overline{C}}{\partial t}=\beta\left(C-\overline{C}/K_d\right)-\lambda_2\rho_b\overline{C} \tag{80}$$

where $L(C)$ is the operator for all nonreaction terms.

The implicit finite-difference approximation for Equations 79 and 80 at cell (i,j,k), considering only the reaction terms, can be expressed as

$$\theta_{i,j,k}\frac{C^{n+1}_{i,j,k}-C^{n}_{i,j,k}}{\Delta t}+\rho_{b(i,j,k)}\frac{\overline{C}^{n+1}_{i,j,k}-\overline{C}^{n}_{i,j,k}}{\Delta t}=-\lambda_1\theta_{i,j,k}C^{n+1}_{i,j,k}-\lambda_2\rho_{b(i,j,k)}\overline{C}^{n+1}_{i,j,k} \tag{81}$$

$$\rho_{b(i,j,k)}\frac{\overline{C}^{n+1}_{i,j,k}-\overline{C}^{n}_{i,j,k}}{\Delta t}=\beta_{i,j,k}\left(C^{n+1}_{i,j,k}-\overline{C}^{n+1}_{i,j,k}/K_{d(i,j,k)}\right)-\lambda_2\rho_{b(i,j,k)}\overline{C}^{n+1}_{i,j,k} \tag{82}$$

Coupling of the two equations leads to the following terms in the coefficient matrix and the right-hand-side vector:

$$\begin{aligned}A^1_{i,j,k}=&-\left(\lambda_{1(i,j,k)}\theta_{i,j,k}+\beta_{i,j,k}\right)\Delta x_j\Delta y_i\Delta z_k\\&+\beta^2_{i,j,k}/K_{d(i,j,k)}\,\Delta x_j\Delta y_i\Delta z_k/\left(\rho_{b(i,j,k)}/\Delta t+\beta_{i,j,k}/K_{d(i,j,k)}+\lambda_{2(i,j,k)}\rho_{b(i,j,k)}\right)\end{aligned} \tag{83}$$

$$b_{i,j,k}=-\beta_{i,j,k}/K_{d(i,j,k)}\,\Delta x_j\Delta y_i\Delta z_k\rho_{b(i,j,k)}\overline{C}^{n}_{i,j,k}/\left(\rho_{b(i,j,k)}+\beta_{i,j,k}\Delta t/K_{d(i,j,k)}+\lambda_{2(i,j,k)}\rho_{b(i,j,k)}\Delta t\right) \tag{84}$$

The dual-domain mass transfer between the mobile and immobile domains can be implemented in a similar fashion.

f. Solution of the matrix equations. Equation 66 can be written for each active node in the model, and the result is a large system of linear equations, in the form of

$$AC^{n+1}=b \tag{85}$$

The MT3DMS code includes a general-purpose iterative solver based on the generalized conjugate gradient methods for solving the matrix Equation 85. The matrix solver is implemented in a new package named Generalized Conjugate Gradient (GCG) Solver. The basic features of the GCG solver are outlined below; a more detailed description of the iterative algorithm behind the GCG package is provided in Appendix A.

The basic iterative algorithm implemented in the GCG package has three preconditioning options, Jacobi, Symmetric Successive Over Relaxation (SSOR), and the Modified Incomplete Cholesky (MIC). The MIC preconditioner usually takes fewer iterations to converge than Jacobi or SSOR, but it requires significantly more memory to operate. The basic iterative algorithm is accelerated by the Lanczos/ORTHOMIN acceleration scheme (Jea and Young 1983; Young and Mai 1988) if the coefficient matrix A is nonsymmetric, as in the case of fully implicit finite-difference formulation. The use of the Lanczos/ORTHOMIN acceleration scheme can greatly increase the convergence speed of an iterative solution. If the coefficient matrix A is symmetric, which is the case when the advection term is solved by the third-order TVD method or a particle-tracking-based method, the Lanczos/ORTHOMIN acceleration scheme is equivalent to the standard conjugate gradient (CG) acceleration, similar to that implemented in the PCG2 package for MODFLOW (Hill 1992).

The GCG solver has two iteration loops, an inner loop and an outer loop. Within the inner loop, the solver will continue to iterate toward the solution until the user-specified convergence criterion is satisfied or the user-specified maximum number of inner iterations allowed is reached. All the coefficients in matrix A and the right-hand-side vector remain unchanged during inner iterations. If some of these coefficients are dependent on the concentrations being solved, as in the case of nonlinear sorption, they must be updated after a certain number of inner iterations are completed. This updating is accomplished by setting the maximum of outer iterations to a value greater than one. When a new outer iteration begins, all the coefficients that are concentration-dependent are updated with the newly calculated concentrations. When it takes only one inner iteration to converge in an outer cycle, the solution is considered to be globally converged and the simulation proceeds to the next transport step.

The coefficient matrix A is a sparse matrix with 19 nonzero diagonals, for a 3-D problem, if all cross terms of the dispersion tensor are kept in the matrix, as shown in Equation 66. The GCG package is implemented with an option to lump all dispersion cross terms to the right-hand-side vector. In other words, the concentrations associated with the dispersion cross terms are assigned to the values calculated in the previous iteration so that they become known quantities and thus can be moved to the right-hand-side vector. With this option, the coefficient matrix contains only nine nonzero diagonals for a 3-D problem. The use of this option of lumped dispersion cross terms reduces the memory requirement of the GCG solver by nearly two-thirds and generally leads to faster convergence. The loss of accuracy is generally insignificant. Although usually not necessary, the number of outer iterations can be set to be greater than one so that the lumped dispersion cross terms are updated as the converged solution is obtained.

Explicit finite-difference formulation

In the original MT3D code, dispersion, sink/source, and reactions are solved explicitly without using an iterative solver. The advection term can be solved by either a particle-tracking-based method or the explicit finite-difference method. The explicit finite-difference method is retained in MT3DMS as an alternative to the implicit method. The explicit finite-difference method can be expressed as

$$R_{i,j,k}\theta_{i,j,k}\frac{C_{i,j,k}^{n+1}-C_{i,j,k}^{n}}{\Delta t}=L_{ADV}\left(C^{n}\right)+L_{DSP}\left(C^{n}\right)+L_{SSM}\left(C^{n}\right)+L_{RCT}\left(C^{n}\right) \quad (86)$$

where L_{ADV}, L_{DSP}, L_{SSM}, and L_{RCT} represent the finite-difference operators for the advection, dispersion, sink/source, and chemical reaction terms, respectively. Because the known concentrations at the old time level n are used to evaluate the finite-difference operators, the concentration at the new time level $n+1$ at cell (i,j,k) can be directly calculated in one step as

$$C_{i,j,k}^{n+1}=C_{i,j,k}^{n}+\frac{\Delta t}{R_{i,j,k}\theta_{i,j,k}}\left[L_{ADV}\left(C^{n}\right)+L_{DSP}\left(C^{n}\right)+L_{SSM}\left(C^{n}\right)+L_{RCT}\left(C^{n}\right)\right]. \quad (87)$$

The explicit solution is subject to various stability constraints on the size (length) of the transport time-step. These stability constraints are listed below for different transport components,

a. Advection

$$\Delta t \le \frac{R}{\left|v_x\right|/\Delta x+\left|v_y\right|/\Delta y+\left|v_z\right|/\Delta z} \quad (88a)$$

b. Dispersion

$$\Delta t \le \frac{0.5R}{\left(\frac{D_{xx}}{\Delta x^2}+\frac{D_{yy}}{\Delta y^2}+\frac{D_{zz}}{\Delta z^2}\right)} \quad (88b)$$

c. Sink/source

$$\Delta t \le \frac{R\theta}{\left|q_s\right|} \quad (88c)$$

d. Chemical reaction

$$\Delta t \le \frac{1}{\left|\lambda_1\right|+\left|\lambda_2\right|} \quad (88d)$$

When the explicit method is used, the stability constraint for each transport component is calculated for each active cell, and the minimum is taken as the maximum allowed step size for solving that particular transport component. The actual transport step size used in the simulation is then determined from the maximum step sizes allowed for individual transport components.

The explicit method is simple and does not need the significant amount of memory required by an iterative solver. It is also computationally efficient for certain advection-dominated problems in which the transport step sizes are limited not by stability constraints but by accuracy considerations. However, for transport problems with fine grid spacing, large dispersion coefficients, and/or strong sink/source terms, the stability constraints could lead to exceedingly small transport steps for the explicit method. Under these circumstances, the GCG solver described above should be used for greater computational efficiency. In addition, for problems of strongly nonlinear sorption, the explicit method may require such a small step size that the implicit method is much more effective.

Implementation of the Third-Order TVD Method

General equations

The basic ideas behind the third-order TVD scheme (ULTIMATE) have been presented in Chapter 3. This section describes the implementation of the general, three-dimensional form of the ULTIMATE scheme.

In the ULTIMATE scheme, the advection term is solved independently of other terms which are solved either implicitly or explicitly with the standard finite-difference method as described in the previous discussions. If the implicit finite-difference method is used for solving the nonadvective terms, the concentration change due to advection, after being solved with the ULTIMATE scheme, is lumped to the right-hand-side vector b, and the contributions to the coefficient matrix from the advection term are set equal to zero. On the other hand, if the explicit finite-difference method is used for solving the nonadvective terms, the concentration change due to advection, after being solved with the ULTIMATE scheme is added to the total concentration change at the cell in question.

The three-dimensional transport equation considering advection alone is as follows:

$$\theta \frac{\partial C}{\partial t} = -\frac{\partial}{\partial x}(q_x C) - \frac{\partial}{\partial y}(q_y C) - \frac{\partial}{\partial z}(q_z C) \tag{89}$$

Applying the explicit finite-difference algorithm, Equation 89 can be written, at cell (i,j,k), as follows (Figure 14):

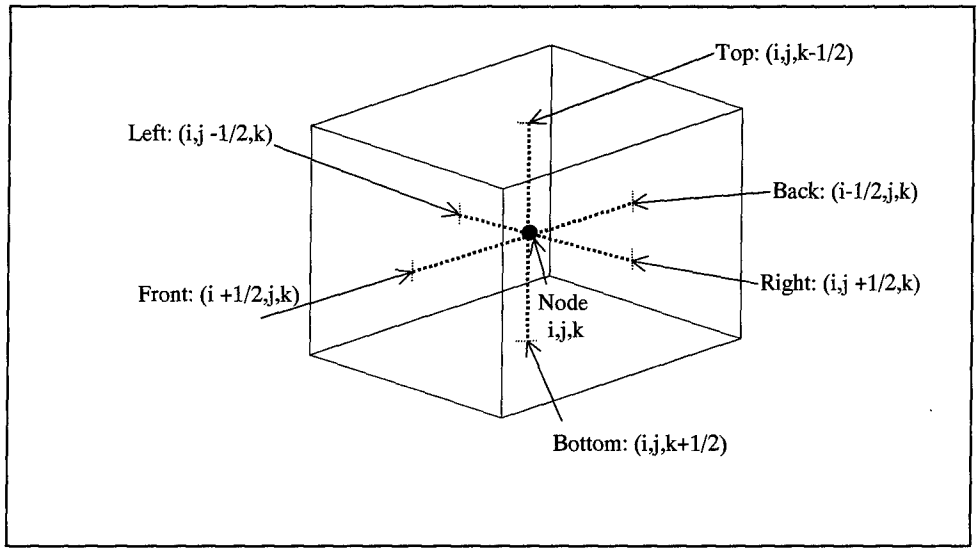


Figure 14. Definition of six surrounding interfaces of cell (i, j, k)

$$\theta_{i,j,k}\frac{C_{i,j,k}^{n+1}-C_{i,j,k}^{n}}{\Delta t}=-\frac{q_{xi,j+1/2,k}C_{i,j+1/2,k}^{n}-q_{xi,j-1/2,k}C_{i,j-1/2,k}^{n}}{\Delta x_j}$$
$$-\frac{q_{yi+1/2,j,k}C_{i+1/2,j,k}^{n}-q_{xi-1/2,j,k}C_{i-1/2,j,k}^{n}}{\Delta y_i} \quad (90)$$
$$-\frac{q_{zi,j,k+1/2}C_{i,j,k+1/2}^{n}-q_{zi,j,k-1/2}C_{i,j,k-1/2}^{n}}{\Delta z_k}$$

Rearranging terms, we obtain the new solution at new time level $n+1$ for cell (i, j, k)

$$C_{i,j,k}^{n+1}=C_{i,j,k}^{n}-(\gamma)_{right}C_{i,j+1/2,k}^{n}+(\gamma)_{left}C_{i,j-1/2,k}^{n}$$
$$-(\gamma)_{front}C_{i+1/2,j,k}^{n}+(\gamma)_{back}C_{i-1/2,j,k}^{n} \quad (91)$$
$$-(\gamma)_{bottom}C_{i,j,k+1/2}^{n}+(\gamma)_{top}C_{i,j,k-1/2}^{n}$$

where the Courant number, γ, is defined as

$$
\begin{aligned}
(\gamma)_{left} &= \frac{q_{xi,j-1/2,k}\Delta t}{\theta_{i,j,k}\Delta x_j} = \frac{v_{xi,j-1/2,k}\Delta t}{\Delta x_j} \\
(\gamma)_{right} &= \frac{q_{xi,j+1/2,k}\Delta t}{\theta_{i,j,k}\Delta x_j} = \frac{v_{xi,j+1/2,k}\Delta t}{\Delta x_j} \\
(\gamma)_{front} &= \frac{q_{yi+1/2,j,k}\Delta t}{\theta_{i,j,k}\Delta y_i} = \frac{v_{yi+1/2,j,k}\Delta t}{\Delta y_i} \\
(\gamma)_{back} &= \frac{q_{yi-1/2,j,k}\Delta t}{\theta_{i,j,k}\Delta y_i} = \frac{v_{yi-1/2,j,k}\Delta t}{\Delta y_i} \\
(\gamma)_{top} &= \frac{q_{zi,j,k-1/2}\Delta t}{\theta_{i,j,k}\Delta z_k} = \frac{v_{zi,j,k-1/2}\Delta t}{\Delta z_k} \\
(\gamma)_{bottom} &= \frac{q_{zi,j,k+1/2}\Delta t}{\theta_{i,j,k}\Delta z_k} = \frac{v_{zi,j,k+1/2}\Delta t}{\Delta z_k}
\end{aligned}
\tag{92}
$$

Applying the same procedure for third-order polynomial interpolation of nodal concentrations as described in Chapter 3, the concentration value at the left interface can be derived as

$$
\begin{aligned}
C^n_{i,j-1/2,k} = {} & \overline{C}_{i,j-1/2,k} - \frac{d_x}{2} f_x - \frac{d_y}{2} f_y - \frac{d_z}{2} f_z \\
& - \frac{\left(\overline{\Delta x_j^2} - d_x^2\right)}{6} f_{xx} + \left(\frac{d_y^2}{6} - \frac{\overline{\Delta y_i} d_y}{4}\right) f_{yy} + \left(\frac{d_z^2}{6} - \frac{\overline{\Delta z_k} d_z}{4}\right) f_{zz} \\
& + \left(\frac{d_x d_y}{3} - \frac{\overline{\Delta x_j} d_y}{4}\right) f_{xy} + \left(\frac{d_x d_z}{3} - \frac{\overline{\Delta x_j} d_z}{4}\right) f_{xz} + \frac{d_y d_z}{3} f_{yz}
\end{aligned}
\tag{93}
$$

where d_x, d_y, and d_z (unit, L) are all defined at the left interface face (i, j-1/2, k) as,

$$
\begin{aligned}
d_x &= v_{xi,j-1/2,k}\Delta t \\
d_y &= v_{yi,j-1/2,k}\Delta t \\
d_z &= v_{zi,j-1/2,k}\Delta t
\end{aligned}
\tag{94}
$$

The left face value in Equation 93 has 10 terms; one average term, three first-order derivatives, and six mixed second-order derivative terms, together constituting the third-order approximation. The first term on the right-hand side of Equation 93 is a weighted average face value

$$
\overline{C}_{i,j-1/2,k} = \omega C^n_{i,j-1,k} + (1-\omega) C^n_{i,j,k} \tag{95}
$$

where ω is a weighting factor given by

$$\omega = \frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \tag{96}$$

The normal gradient across the left face is defined as

$$f_x = \frac{C^n_{i,j,k} - C^n_{i,j-1,k}}{\overline{\Delta x_j}} \tag{97}$$

where $\overline{\Delta x_j}$ is the distance between the centers of the two cells (i, j-1, k) and (i, j, k)

$$\overline{\Delta x_j} = \frac{\Delta x_{j-1} + \Delta x_j}{2} \tag{98}$$

The upwinding gradient along the y-direction depends on the directions of the velocities:

$$\begin{aligned}
f_y &= \left(C^n_{i,j-1,k} - C^n_{i-1,j-1,k} \right) / \overline{\Delta y_i} && v_{xi,j-1/2,k} > 0, v_{yi,j-1/2,k} > 0 \\
&= \left(C^n_{i+1,j-1,k} - C^n_{i,j-1,k} \right) / \overline{\Delta y_{i+1}} && v_{xi,j-1/2,k} > 0, v_{yi,j-1/2,k} < 0 \\
&= \left(C^n_{i,j,k} - C^n_{i-1,j,k} \right) / \overline{\Delta y_i} && v_{xi,j-1/2,k} < 0, v_{yi,j-1/2,k} > 0 \\
&= \left(C^n_{i+1,j,k} - C^n_{i,j,k} \right) / \overline{\Delta y_{i+1}} && v_{xi,j-1/2,k} < 0, v_{yi,j-1/2,k} < 0
\end{aligned} \tag{99}$$

with

$$\overline{\Delta y_i} = \frac{\Delta y_{i-1} + \Delta y_i}{2} \tag{100}$$

Similarly, the upwinding gradient along the z-direction is given by

$$\begin{aligned}
f_z &= \left(C^n_{i,j-1,k} - C^n_{i,j-1,k-1} \right) / \overline{\Delta z_k} && v_{xi,j-1/2,k} > 0, v_{z_{xi,j-1/2,k}} > 0 \\
&= \left(C^n_{i,j-1,k+1} - C^n_{i,j-1,k} \right) / \overline{\Delta z_{k+1}} && v_{x_{xi,j-1/2,k}} > 0, v_{z_{xi,j-1/2,k}} < 0 \\
&= \left(C^n_{i,j,k} - C^n_{i,j,k-1} \right) / \overline{\Delta z_k} && v_{x_{xi,j-1/2,k}} < 0, v_{z_{xi,j-1/2,k}} > 0 \\
&= \left(C^n_{i,j,k+1} - C^n_{i,j,k} \right) / \overline{\Delta z_{k+1}} && v_{x_{xi,j-1/2,k}} < 0, v_{zi,j-1/2,k} < 0
\end{aligned} \tag{101}$$

with

$$\overline{\Delta z_k} = \frac{\Delta z_{k-1} + \Delta z_k}{2} \tag{102}$$

Based on the first-order derivatives, various second-order derivatives can be calculated. The computation of the upwinding second-order derivatives (curvatures) along the x direction requires three cells: two neighboring cells and one upstream cell. It is written in the form of first-order derivatives:

$$\begin{aligned} f_{xx} &= \left[(f_x)_{i,j-1/2,k} - (f_x)_{i,j-1-1/2,k}\right] / \Delta x_{j-1} \qquad v_{xi,j-1/2,k} > 0 \\ &= \left[(f_x)_{i,j+1/2,k} - (f_x)_{i,j-1/2,k}\right] / \Delta x_j \qquad v_{xi,j-1/2,k} < 0 \end{aligned} \tag{103}$$

The other two curvatures along the y and z directions are given as

$$\begin{aligned} f_{yy} &= \left[(f_y)_{i-1/2,j,k} - (f_y)_{i-1-1/2,j,k}\right] / \Delta y_{i-1} \qquad v_{yi,j-1/2,k} > 0 \\ &= \left[(f_y)_{i+1/2,j,k} - (f_y)_{i-1/2,j,k}\right] / \Delta y_i \qquad v_{yi,j-1/2,k} < 0 \end{aligned} \tag{104}$$

$$\begin{aligned} f_{zz} &= \left[(f_z)_{i,j,k-1/2} - (f_z)_{i,j,k-1-1/2}\right] / \Delta z_{k-1} \qquad v_{zi,j-1/2,k} > 0 \\ &= \left[(f_z)_{i,j,k+1/2} - (f_z)_{i,j,k-1/2}\right] / \Delta z_k \qquad v_{zi,j-1/2,k} < 0 \end{aligned} \tag{105}$$

Finally, there are three upwinding mixed second-order derivatives (twist terms). The two along the x axis are given by

$$\begin{aligned} f_{xy} &= \left[(f_y)_{i-1/2,j,k} - (f_y)_{i-1/2,j-1,k}\right] / \overline{\Delta x_j} \qquad v_{yi,j-1/2,k} > 0 \\ &= \left[(f_y)_{i+1/2,j,k} - (f_y)_{i+1/2,j-1,k}\right] / \overline{\Delta x_j} \qquad v_{yi,j-1/2,k} < 0 \end{aligned} \tag{106}$$

$$\begin{aligned} f_{xz} &= \left[(f_z)_{i,j,k-1/2} - (f_z)_{i,j-1,k-1/2}\right] / \overline{\Delta x_j} \qquad v_{zi,j-1/2,k} > 0 \\ &= \left[(f_z)_{i,j,k+1/2} - (f_z)_{i,j-1,k+1/2}\right] / \overline{\Delta x_j} \qquad v_{zi,j-1/2,k} < 0 \end{aligned} \tag{107}$$

The one that is normal to the x-axis is given by

$$f_{yz} = \left[(f_y)_{i,j-1/2,k+1/2} - (f_y)_{i,j-1/2,k-1/2}\right] / \overline{\Delta z_k} \tag{108}$$

The computation of each twist term requires four cells. Other interface values at right, back, front, top, and bottom can be evaluated in a similar manner.

After all the six interface concentration values are evaluated using the above equations, the universal flux limiter as described in Chapter 3 is applied to adjust these values, if necessary, in order to limit spurious oscillations that may have arisen from higher-order interpolation. The interface concentrations, including any necessary adjustments, are then used in Equation 91 to compute the concentration at node (i,j,k) for the new time level (n+1) due to advection alone.

Mass balance considerations and stability constraint

To conserve mass, the continuity of interface values is assumed. That is, the concentration at the left interface of one cell is identical to that at the right interface of the adjacent cell. Therefore, at each cell only three interface values are computed and stored. The other three interface values can be obtained from previous calculations.

Special attention must be addressed to the concern that mass balance errors could arise when cell thicknesses in the same model layer vary spatially. The Courant numbers must be adjusted in this case. In emphasizing the spatially variable cell thickness, we use $\Delta z_{i,j,k}$ instead of Δz_k. The Courant numbers at left and front interfaces are adjusted as

$$(\gamma)_{left} = \frac{\Delta z_{i,j-1,k}}{\Delta z_{i,j,k}} \frac{v_{xi,j-1/2,k}\Delta t}{\Delta x_j}$$
$$(\gamma)_{front} = \frac{\Delta z_{i-1,j,k}}{\Delta z_{i,j,k}} \frac{v_{yi-1/2,j,k}\Delta t}{\Delta y_i} \qquad (109)$$

The other four Courant numbers remain unchanged as given in Equation 92.

The ULTIMATE scheme as implemented in MT3DMS is explicit, thus, the transport step size is subject to a stability constraint, which is given as

$$\Delta t \leq \frac{R}{|v_x|/\Delta x + |v_y|/\Delta y + |v_z|/\Delta z} \qquad (110)$$

Equation 110 is applied to every active cell in the model grid and the minimum Δt is used as the maximum allowed step size for the ULTIMATE scheme.

Implementation of the Eulerian-Lagrangian Methods

Velocity interpolation

Both the MOC and the MMOC involve the use of a particle-tracking technique to approximate the advective component of the transport process. Since any particle-tracking technique requires the evaluation of velocity at an arbitrary point from hydraulic heads calculated at nodal points, it is necessary to use a velocity interpolation scheme in the particle-tracking calculations.

The velocity interpolation scheme used in this transport model is simple piecewise linear interpolation (Pollock 1988; Zheng 1988). This scheme assumes that a velocity component varies linearly within a finite-difference cell with respect to the direction of that component. Thus, the x-component of the Darcy flux at an arbitrary point within cell (i,j,k) can be expressed in terms of the fluxes on cell interfaces in the same direction (Figure 15):

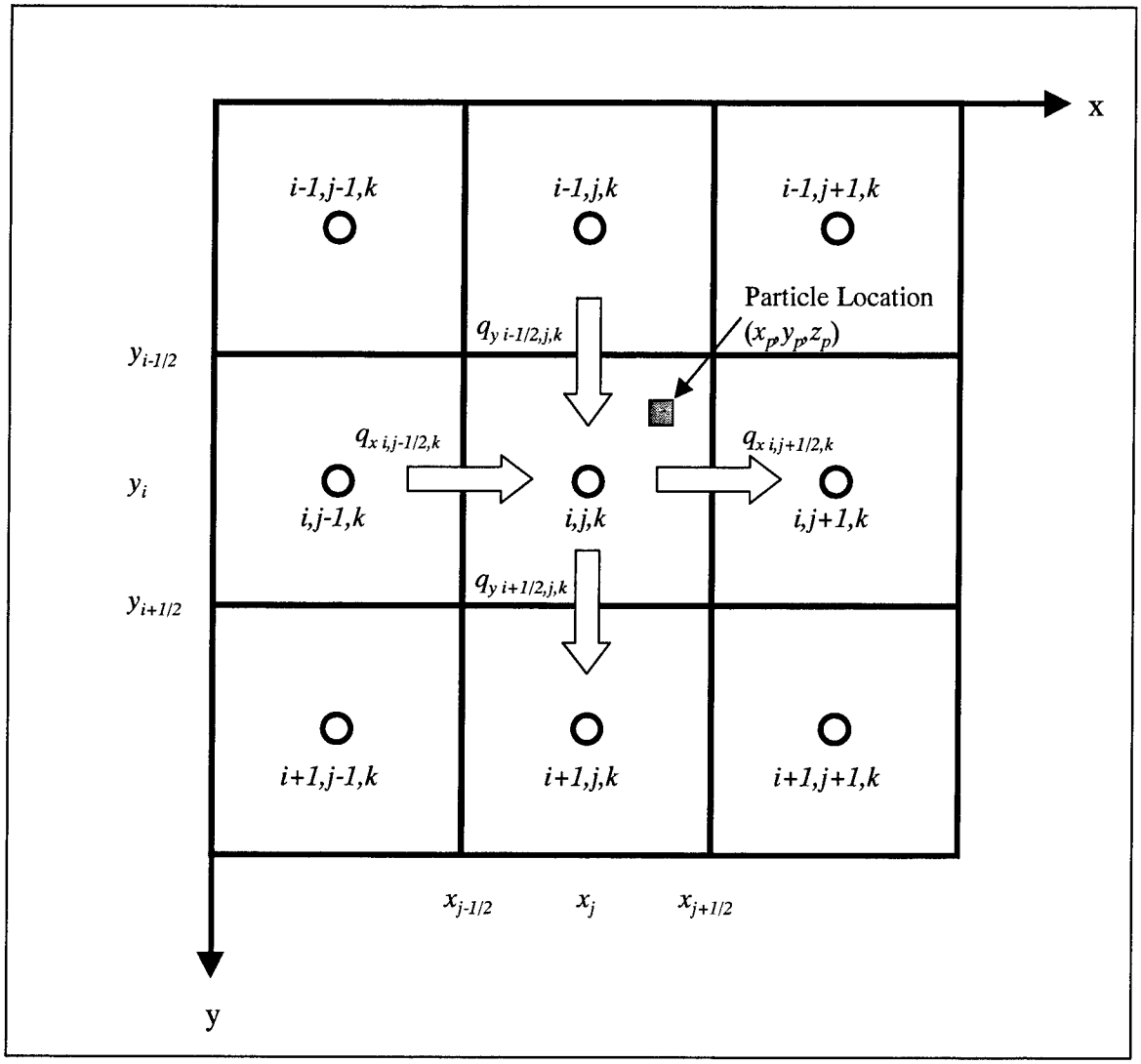


Figure 15. Velocity interpolation scheme used in particle tracking

$$q_x(x_P, y_P, z_P) = (1-\alpha_x) q_{i,j-1/2,k} + \alpha_x q_{i,j+1/2,k} \tag{111}$$

where

$q_{i,j-1/2,k} = -K_{i,j-1/2,k}\left(h_{i,j,k} - h_{i,j-1/2,k}\right)/\left(x_j - x_{j-1}\right)$, the flux, or the specific discharge, through the interface between cells (i, j-1, k) and (i, j, k), and $K_{i,j-1/2,k}$ is the harmonic mean of hydraulic conductivity between the two cells. The flux at the cell interface is calculated in the flow model and directly used in the transport model

$q_{i,j+1/2,k} = -K_{i,j+1/2,k}\left(h_{i,j+1/2,k} - h_{i,j,k}\right)/\left(x_{j+1} - x_j\right)$, the flux through the interface between cells (i,j,k) and (i, j+1, k)

$\alpha_x = (x_p - x_{j-1/2})/\Delta x_j$, the linear interpolation factor for the x component

x_P, y_P, z_P = Cartesian coordinates of the particle location

$x_{j\pm1/2}$ = .x coordinates of the left and right interfaces of the cell (i, j, k)

x_j = x coordinates of the node (i, j, k)

Δx_j = cell width along the x-axis at cell (i, j, k)

The x-component of the linear or pore water velocity, v_x, is then obtained from:

$$v_x(x_P, y_P, z_P) = \frac{q_x(x_P, y_P, z_P)}{\theta_{i,j,k}} \tag{112}$$

where $\theta_{i,j,k}$ is the porosity value at cell (i,j,k).

Similarly, the y- and z-components of the velocity are calculated as

$$q_y(x_P, y_P, z_P) = (1-\alpha_y)\, q_{i-1/2,j,k} + \alpha_y q_{i+1/2,j,k} \tag{113}$$

$$v_y(x_P, y_P, z_P) = \frac{q_y(x_P, y_P, z_P)}{\theta_{i,j,k}} \tag{114}$$

where $\alpha_y = \dfrac{y_P - y_{i-1/2}}{\Delta y_i}$ is the linear interpolation factor for the y-component; and

$$q_z(x_P, y_P, z_P) = (1-\alpha_z)\, q_{i,j,k-1/2} + \alpha_z q_{i,j,k+1/2} \tag{115}$$

$$v_z(x_P, y_P, z_P) = \frac{q_z(x_P, y_P, z_P)}{\theta_{i,j,k}} \tag{116}$$

where $\alpha_z = (z_P - z_{k-1/2})/\Delta z_k$ is the linear interpolation factor for the z-component.

The velocity field generated with this scheme is consistent with the block-centered finite-difference formulation of the three-dimensional flow equation and thus conserves mass locally within each finite-difference block. It also preserves the velocity discontinuities caused by changes in hydraulic conductivities present in heterogeneous media.

It is noted that this velocity scheme differs from the multilinear scheme used in earlier MOC models (Garder, Peaceman, and Pozzi 1964; Konikow and Bredehoeft 1978). The multilinear scheme in a three-dimensional flow field assumes that velocity components vary linearly in all three directions and thus generates a continuous velocity field in every direction. Goode (1990) notes that the multilinear scheme may result in more satisfactory results in homogeneous media. However, the multilinear scheme is not consistent with the cell-by-cell mass balance described by the block-centered finite-difference formulation and does not preserve the velocity discontinuities present in heterogeneous media, unlike the piecewise linear scheme. Because of this and because the piecewise linear scheme is computationally much more efficient, the piecewise linear scheme has been utilized in the MT3DMS model.

Particle tracking

With the velocity field known, a numerical tracking scheme can be used to move particles from one position to another to approximate the advection of the contaminant front. Traditionally, the first-order Euler algorithm has been used for particle tracking (Konikow and Bredehoeft 1978; Goode and Konikow 1989):

$$\begin{cases} x^{n+1} = x^n + \dfrac{\Delta t}{R} v_x\left(x^n, y^n, z^n\right) \\ y^{n+1} = y^n + \dfrac{\Delta t}{R} v_y\left(x^n, y^n, z^n\right) \\ z^{n+1} = z^n + \dfrac{\Delta t}{R} v_z\left(x^n, y^n, z^n\right) \end{cases} \tag{117}$$

where

$x^{n+1}, y^{n+1}, z^{n+1}$ = particle coordinates at the new time level $(n+1)$

x^n, y^n, z^n = coordinates at the old time level (n)

v_x, v_y, v_z = linear velocities evaluated at $\left(x^n, y^n, z^n\right)$

R = retardation factor resulting from the incorporation of sorption isotherms into the transport equation

Δt = size of the transport step, which is generally expressed in terms of the Courant number, γ

$$|\Delta t| \le \gamma MIN\left(\frac{R\Delta x}{v_x}, \frac{R\Delta y}{v_y}, \frac{R\Delta z}{v_z}\right) \tag{118}$$

where the Courant number represents the number of cells a particle will be allowed to move in any direction in one transport step. The particle tracking is forward if the sign of Δt is positive, and backward if the sign of Δt is negative.

A uniform step size, Δt, is used for all moving particles during each transport step in the particle-tracking calculations. For particles located in areas of relatively uniform velocity, the first-order Euler algorithm may have sufficient accuracy. However, for particles located in areas of strongly converging or diverging flows, for example, near sources or sinks, the first-order algorithm may not be sufficiently accurate unless Δt is very small. In these cases, a higher-order algorithm such as the fourth-order Runge-Kutta method may be used. The basic idea of the fourth-order Runge-Kutta method is to evaluate the velocity four times for each tracking step: once at the initial point, twice at two trial midpoints, and once at a trial end point (Figure 16). A weighted velocity based on values evaluated at these four points is used to move the particle to the new position $(x^{n+1}, y^{n+1}, z^{n+1})$. This process may be expressed as follows:

$$\begin{cases} x^{n+1} = x^n + \dfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ y^{n+1} = y^n + \dfrac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4) \\ z^{n+1} = z^n + \dfrac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4) \end{cases} \tag{119}$$

where

$$k_1 = \Delta t v_x\left(x^n, y^n, z^n, t^n\right)$$
$$k_2 = \Delta t v_x\left(x^n + k_1/2, y^n + l_1/2, z^n + m_1/2, t^n + \Delta t/2\right)$$
$$k_3 = \Delta t v_x\left(x^n + k_2/2, y^n + l_2/2, z^n + m_2/2, t^n + \Delta t/2\right)$$
$$k_4 = \Delta t v_x\left(x^n + k_3, y^n + l_3, z^n + m_3, t^n + \Delta t\right)$$
$$l_1 = \Delta t v_y\left(x^n, y^n, z^n, t^n\right)$$
$$l_2 = \Delta t v_y\left(x^n + k_1/2, y^n + l_1/2, z^n + m_1/2, t^n + \Delta t/2\right)$$
$$l_3 = \Delta t v_y\left(x^n + k_2/2, y^n + l_2/2, z^n + m_2/2, t^n + \Delta t/2\right)$$
$$l_4 = \Delta t v_y\left(x^n + k_3, y^n + l_3, z^n + m_3, t^n + \Delta t\right)$$
$$m_1 = \Delta t v_z\left(x^n, y^n, z^n, t^n\right)$$
$$m_2 = \Delta t v_z\left(x^n + k_1/2, y^n + l_1/2, z^n + m_1/2, t^n + \Delta t/2\right)$$
$$m_3 = \Delta t v_z\left(x^n + k_2/2, y^n + l_2/2, z^n + m_2/2, t^n + \Delta t/2\right)$$
$$m_4 = \Delta t v_z\left(x^n + k_3, y^n + l_3, z^n + m_3, t^n + \Delta t\right)$$

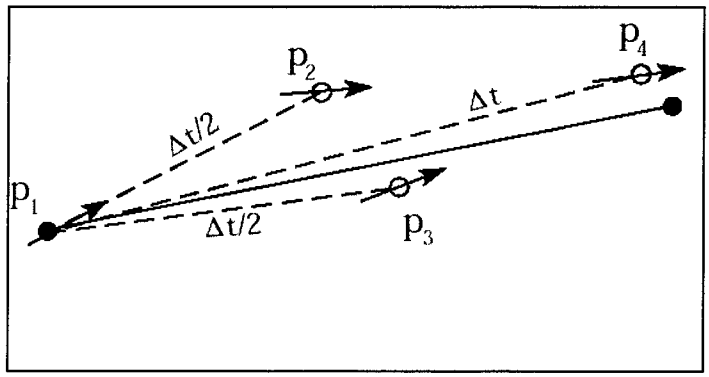


Figure 16. Fourth-order Runge-Kutta method (In each step, the velocity is evaluated four times: once at the initial point, twice at trial midpoints, and once at a trial endpoint. From these velocities a weighted velocity is calculated which is used to compute the final position of the particle (shown as a filled dot))

The fourth-order Runge-Kutta algorithm is more accurate than the Euler algorithm and permits the use of larger tracking steps. However, the computational effort required by the fourth-order Runge-Kutta algorithm is considerably more than that required by the Euler algorithm, making the former less efficient than the latter for three-dimensional simulations when a very large number of particles are used. For these reasons, the MT3DMS model provides three options: a first-order Euler algorithm, a fourth-order Runge-Kutta algorithm, and a combination of these two. These options, when used properly, allow sufficient accuracy throughout the finite-difference grid without using exceedingly small step sizes.

MOC procedure

The first step in the MOC is to generate representative particles in the finite-difference grid. Instead of placing a uniform number of particles in every cell of the grid, a dynamic approach is used in the MT3DMS transport model to control the distribution of moving particles. The number of particles placed at each cell is normally set either at a high level or at a low level, according to the so-called "relative cell concentration gradient," or, DCCELL, defined as:

$$DCCELL_{i,j,k} = \frac{CMAX_{i,j,k} - CMIN_{i,j,k}}{CMAX - CMIN} \tag{120}$$

where

$CMAX_{i,j,k} = \max_{kk=k-1}^{k+1}\left(\max_{jj=j-1}^{j+1}\left(\max_{ii=i-1}^{i+1}\left(C_{ii,jj,kk}\right)\right)\right)$ is the maximum concentration in the immediate vicinity of the cell (i, j, k)

$$CMIN_{i,j,k} = \min_{kk=k-1}^{k+1}\left(\min_{jj=j-1}^{j+1}\left(\min_{ii=i-1}^{i+1}\left(C_{ii,jj,kk}\right)\right)\right)$$ is the minimum concentration in the immediate vicinity of the cell (i, j, k)

$CMAX$ = maximum concentration in the entire grid

$CMIN$ = minimum concentration in the entire grid

With the dynamic approach, the user defines the criterion, DCEPS, which is a small number near zero; the higher number of particles, NPH, is placed in cells where the relative concentration gradient is greater than DCEPS, and the lower number of particles, NPL, in cells where the relative concentration gradient is less than DCEPS.

$$\begin{cases} NP_{i,j,k} = NPH, & \text{if } DCCELL_{i,j,k} > DCEPS; \\ NP_{i,j,k} = NPL, & \text{if } DCCELL_{i,j,k} \le DCEPS. \end{cases} \tag{121}$$

where $NP_{i,j,k}$ is number of particles placed in cell (i, j, k).

Initially, if the concentration gradient at a cell is zero or small (i.e., the concentration field is relatively constant near that cell), the number of particles placed in that cell is NPL, which may be zero or some small integer number; this is done because the concentration change due to advection between that cell and the neighboring cells will be insignificant. If the concentration gradient at a cell is large, which indicates that the concentration field near that cell is changing rapidly, then the number of particles placed in that cell is NPH.

As particles leave source cells or accumulate at sink cells, it becomes necessary to insert new particles at sources or remove particles at sinks. At nonsource or nonsink cells, it also becomes necessary to insert or remove particles as the cell concentration gradient changes with time. This is done in the dynamic insertion-deletion procedure by specifying the minimum and maximum numbers of particles allowed per cell, called NPMIN and NPMAX, respectively. When the number of particles in any cell, (source or nonsource), becomes smaller than the specified minimum, NPMIN, new particles equal to NPL or NPH are inserted into that cell without affecting the existing particles. On the other hand, when the number of particles in any cell, (sink or nonsink), exceeds the specified maximum, or NPMAX, all particles are removed from that cell and replaced by a new set of NPH particles to maintain mass balance. To save computer storage, memory space occupied by the deleted particles is reused by newly inserted particles.

Figure 17 illustrates the dynamic particle distribution approach in contrast with the uniform approach in simulating two-dimensional solute transport from a continuous point source in a uniform flow field. Whereas the uniform approach inserts and maintains an approximately uniform particle distribution throughout the simulated domain, the dynamic approach adjusts the distribution of moving particles dynamically, adapting to the changing nature of the concentration field.

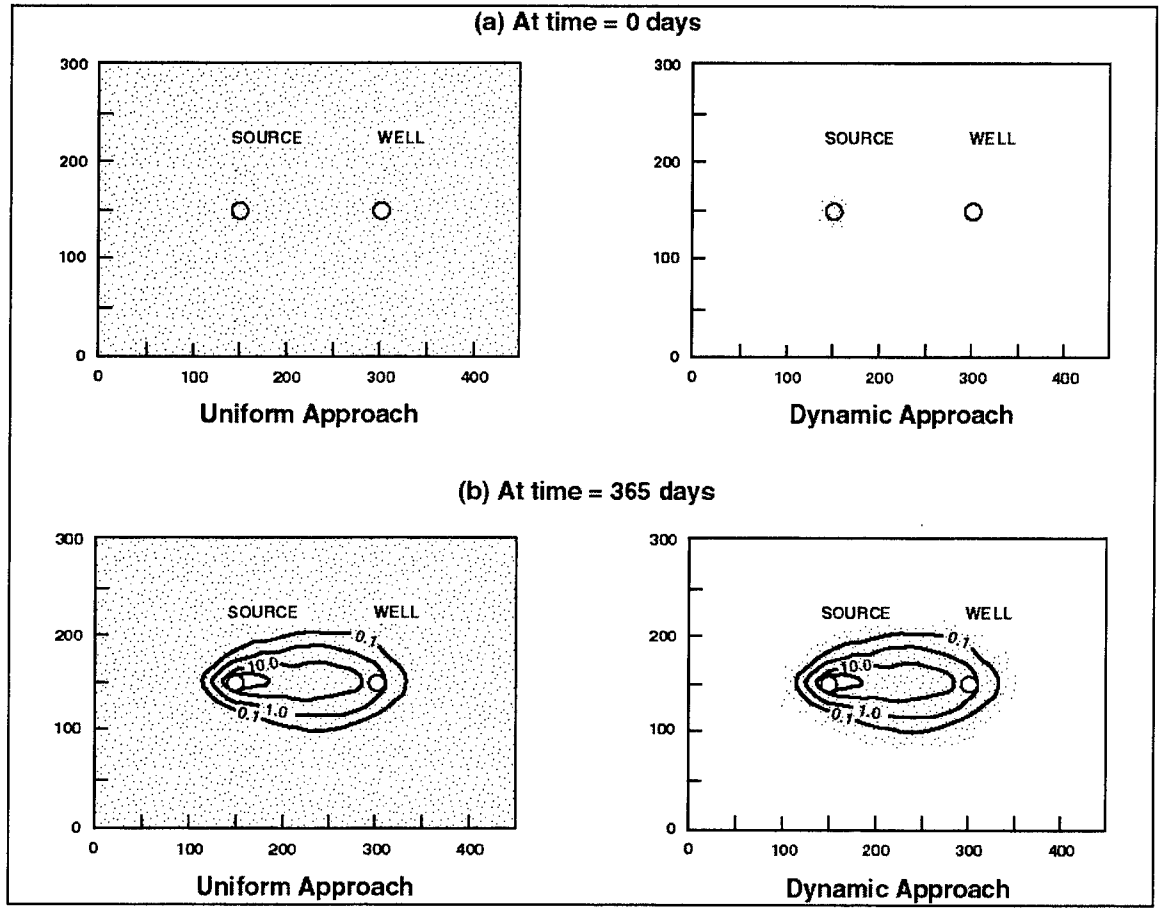


Figure 17. Comparison of the uniform and dynamic approaches in controlling the distribution of moving particles

In many practical problems involving contaminant transport modeling, the contaminant plumes may occupy only a small fraction of the finite-difference grid, and the concentrations may be changing rapidly only at sharp fronts. In these cases, the number of total particles used is much smaller than that required in the uniform particle distribution approach, thereby dramatically increasing the efficiency of the MOC model with little loss in accuracy.

Particles can be distributed either with a fixed pattern or randomly, as controlled by the user-specified option (Figure 18). If the fixed pattern is chosen, the user determines not only the number of particles to be placed per cell, but also the pattern of the particle placement in plan view and the number of vertical planes on which particles are placed within each cell block. If the random pattern is chosen, the user only needs to specify the number of particles to be placed per cell. The program then calls a random number generator and distributes the required number of particles randomly within each cell block. (The selection of these options is discussed in Chapter 6: Input Instructions). The fixed pattern may work better if the flow field is relatively uniform. On the other hand, if the flow field is highly nonuniform with many sinks or sources in largely heterogeneous media, the random pattern may capture the essence of the flow field better than the fixed pattern does.

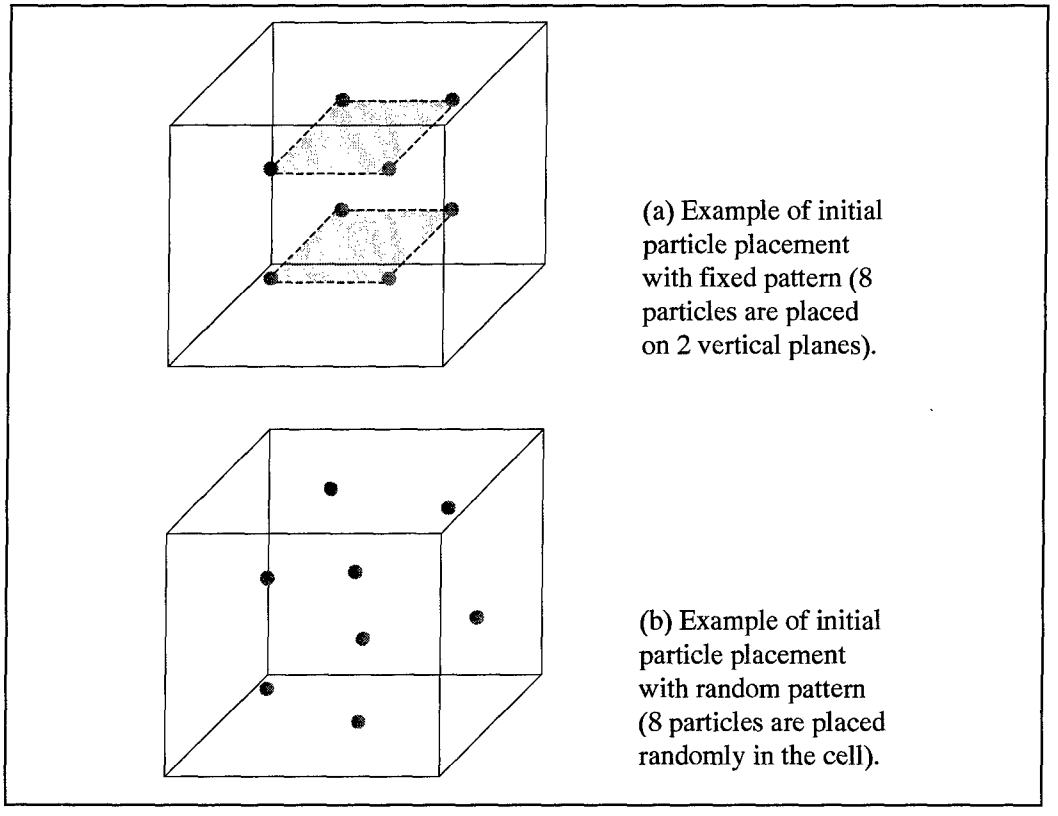


Figure 18. Initial placement of moving particles

Each particle is associated with a set of attributes, that is, the x-, y-, and z-coordinates and the concentration. The initial concentration of the particle is assigned as the concentration of the cell where the particle is initialized. At the beginning of each transport step, all particles are moved over the time increment, Δt, using the particle-tracking techniques described previously. The x-, y-, and z-coordinates of the moving particles are then updated to reflect their new positions at the end of the transport step. The average concentration of a finite-difference cell at the end of the transport step due to advection alone, $C^{n}_{i,j,k}$, can be obtained from the concentrations of all particles that are located at that cell. If a simple arithmetic averaging algorithm is used, the average concentration is expressed by the following equation:

$$C^{n}_{i,j,k} = \frac{1}{NP_{i,j,k}} \sum_{l=1}^{NP_{i,j,k}} C^{n}_{l}, \text{ if } NP_{i,j,k} > 0 \qquad (122)$$

If the grid spacing is irregular, the volume-based averaging algorithm of Zheng (1993) is used as follows:

$$C^{n*}_{i,j,k} = \sum_{l=1}^{NP_{i,j,k}} V_l C_l{}^{n} \Bigg/ \sum_{l=1}^{NP_{i,j,k}} V_l \quad \text{if } NP_{i,j,k} > 0 \qquad (123)$$

where V_l is the volume of the cell in which the l^{th} particle is first generated. If the number of particles at the cell is zero, then the average concentration after particle tracking is set equal to the cell concentration at the previous time level because the concentration change at that cell over the time increment is either negligible or dominated by an external source:

$$C^{n^*}_{i,j,k} = C^n_{i,j,k}, \text{ if } NP_{i,j,k} = 0 \tag{124}$$

It is necessary to locate the cell indices of any particle in the tracking and averaging calculations as described above. If the finite-difference grid is regular, it is straightforward to convert particle coordinates (x_P, y_P, z_P) to cell indices (JP, IP, KP) according to the following formulas:

$$\begin{cases} JP = INT(x_P / \Delta x) + 1 \\ IP = INT(y_P / \Delta y) + 1 \\ KP = INT(z_P / \Delta z) + 1 \end{cases} \tag{125}$$

where *INT(x)* is a FORTRAN function, equal to the truncated value of x; and $\Delta x, \Delta y, \Delta z$ are the uniform grid spacings along the x-, y-, and z-axes. If the finite-difference grid is irregular, the coordinates of the particle in any direction (e.g., x) must be compared with those of cell interfaces in the same direction to determine in which cell the particle is located.

After the $C^{n^*}_{i,j,k}$ term is evaluated at every cell, it is used to calculate the concentration change due to dispersion, sink/source mixing, and/or chemical reactions $\left(\Delta C^{n+1}_{i,j,k}\right)$ using the finite-difference method as discussed previously. The concentration of all active particles is then updated by adding the concentration change $\left(\Delta C^{n+1}_{i,j,k}\right)$ calculated at the cell where each particle is located. Therefore, for moving particles located at cell (i, j, k),

$$C^{n+1}_l = C^n_l + \Delta C^{n+1}_{i,j,k} \tag{126}$$

where C^{n+1}_l is the concentration of the l^{th} particle which is located at cell (i, j, k) at the new time level. If $\Delta C^{n+1}_{i,j,k}$ is positive, Equation 126 is applied directly. However, if $\Delta C^{n+1}_{i,j,k}$ is negative, the concentration of the moving particle may become negative if its concentration at the old time level, C^n_l, is zero or small. When this happens, all particles in the cell are removed and replaced by a new set of particles which are assigned the concentration of the cell.

MMOC procedure

The first step in the MMOC procedure is to move a particle located at the nodal point of the cell backward in time using the particle-tracking techniques. The purpose of this backward tracking is to find the position from which a particle would have originated at the beginning of the time-step so as to reach the nodal point at the end of the time-step. The concentration associated with that position, denoted as $(\hat{x}, \hat{y}, \hat{z})$, is the concentration of the cell due to advection alone over the time increment Δt.

The position $(\hat{x}, \hat{y}, \hat{z})$ generally does not coincide with a nodal point. Thus, it is necessary to interpolate the concentration at $(\hat{x}, \hat{y}, \hat{z})$ from concentrations at neighboring nodal points. The interpolation scheme used in the MT3DMS transport model is first-order polynomial interpolation, also referred to as bilinear in two dimensions or trilinear in three dimensions. The general equation for first-order polynomial interpolation is as follows, assuming that $\hat{x}$ is located between nodes x_{j-1} and x_j, $\hat{y}$ is between y_{i-1} and y_i, and $\hat{z}$ is between z_{k-1} and z_k (Figure 19):

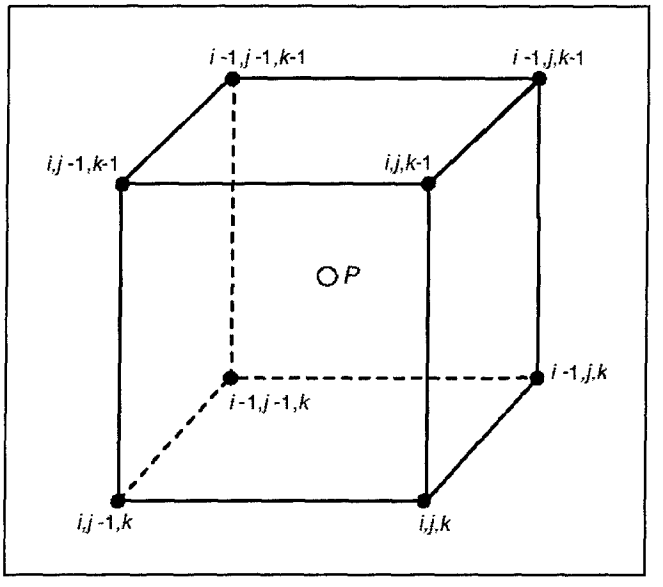


Figure 19. Interpolation of the concentration at point *P* from the concentrations at neighboring nodes using the trilinear scheme in three dimensions

$$
\begin{aligned}
C(\hat{x},\hat{y},\hat{z}) = {} & (1-\omega_x)(1-\omega_y)(1-\omega_z)\, C_{i-1,j-1,k-1} + (1-\omega_x)\,\omega_y(1-\omega_z)\, C_{i,j-1,k-1} \\
& + \omega_x(1-\omega_y)(1-\omega_z)\, C_{i-1,j,k-1} + \omega_x\omega_y(1-\omega_z)\; C_{i,j,k-1} \\
& + (1-\omega_x)(1-\omega_y)\,\omega_z C_{i-1,j-1,k} + (1-\omega_x)\,\omega_y\omega_z C_{i,j-1,k} \\
& + \omega_x(1-\omega_y)\,\omega_z C_{i-1,j,k} + \omega_x\omega_y\omega_z C_{i,j,k}
\end{aligned}
\qquad (127)
$$

where ω_x, ω_y, and ω_z are interpolation factors as given below:

$$\begin{cases} \omega_x = \dfrac{\hat{x} - x_{j-1}}{0.5\Delta x_j + 0.5\Delta x_{j-1}} \\ \omega_y = \dfrac{\hat{y} - y_{i-1}}{0.5\Delta y_i + 0.5\Delta y_{i-1}} \\ \omega_z = \dfrac{\hat{z} - z_{k-1}}{0.5\Delta z_k + 0.5\Delta z_{k-1}} \end{cases} \tag{128}$$

If the x-, y-, or z-dimension is not simulated, the weighting factor in the respective direction is zero. If any cell is inactive, the cell is skipped in the calculation. The low-order interpolation represented by Equation 127 is computationally very efficient and has small mass balance error. It is also virtually free of artificial oscillation. However, this linear scheme leads to numerical dispersion. As the concentration fronts become sharper, the amount of numerical dispersion increases. However, in the MT3DMS transport model, the MMOC scheme is only intended for problems with relatively smooth concentration fronts (sharp front problems are handled by the MOC or TVD technique). When the concentration field is relatively smooth, the numerical dispersion resulting from the MMOC technique is insignificant.

Sinks or sources create special problems for the MMOC scheme and thus have to be treated differently. First, examine a sink cell with inward hydraulic gradients on all of the cell interfaces as illustrated in Figure 20. If the sink is symmetric, the velocity at the nodal point is zero. Therefore, instead of placing one particle at the nodal point, the MT3DMS program places multiple particles within the cell. The number and distribution of these particles are controlled by the user-specified options in a manner similar to those described in the MOC procedure. Each particle is tracked backward over Δt, and its concentration is interpolated for neighboring nodes. The cell concentration is then averaged from the concentrations of all particles, based on the inverse-distance algorithm:

$$C^{n*}_{i,j,k} = \frac{\sum_{l=1}^{NP} C^n_l / d_l}{\sum_{l=1}^{NP} 1/d_l} \tag{129}$$

where d_l is the distance between the nodal point and the position where the l^{th} particle is initially placed. The inverse-distance algorithm differs from the simple average algorithm used in the MOC scheme in that the former gives more weight to particles that are located closer to the node whereas the latter gives the same weight to all particles in the same cell.

Next, consider a source cell with outward hydraulic gradients on all cell interfaces. Backward tracking will cause particles placed in the source cell to converge toward the nodal point so that:

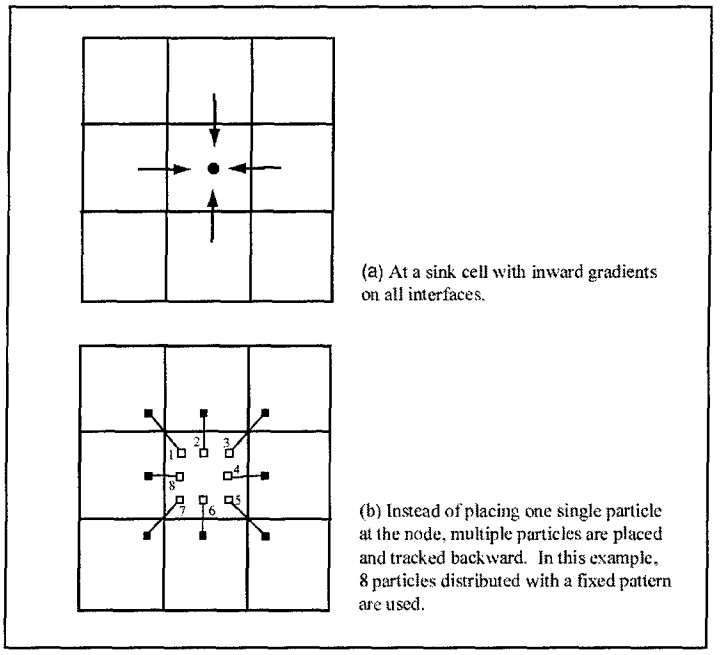


Figure 20. Special treatment of sink cells in the MMOC scheme

$$C^{n*}_{i,j,k} = C^{n}_{i,j,k} \tag{130}$$

With the MMOC scheme, particles are restarted at each time-step, so there is no need to store the particle locations and concentrations in computer memory. Thus, the MMOC solution normally requires far less computer memory and is generally more efficient computationally than the MOC solution.

HMOC procedure

The forward-tracking MOC scheme is well suited for sharp front problems (pure advection or largely advection-dominated problems) because it virtually eliminates numerical dispersion. The MOC scheme implemented with dynamic particle distribution is also very efficient computationally for many practical problems where the contaminant plume occupies only a small fraction of the finite-difference grid, and the concentration field is changing rapidly only at sharp fronts. However, as the degree of advection domination over dispersion and chemical reactions decreases, the advantage of the MOC scheme is less obvious because as physical dispersion increases, numerical dispersion becomes less of a problem. Furthermore, as large physical dispersion causes the contaminant plume to spread through a large portion of the simulated domain, the number of moving particles needed by the MOC scheme can become very large for a three-dimensional simulation, pushing the memory requirement beyond the limits of many personal computers. The backward-tracking MMOC scheme tends to complement the MOC scheme for smooth front problems because the MMOC scheme requires far less computer memory and is generally more efficient computationally.

If the flow field and the dispersivity parameters are relatively constant, and the spatial discretization is fairly regular, it may be straightforward to select either the MOC or the MMOC scheme to be used in the simulation based on the grid Peclet number:

$$Pe_i = \frac{|v_i|\Delta x_i}{D_{ii}}; \qquad i = x,\ y,\ or\ z \tag{131}$$

where all the terms have been defined previously. The MOC scheme is suitable for problems with large grid Peclet numbers while the MMOC scheme can be used for problems with small grid Peclet numbers. As a rule of thumb, the MOC scheme may be used effectively for problems with a grid Peclet number greater than 10 while the MMOC scheme may be used for problems with a grid Peclet number smaller than 0.1 without introducing any significant amount of numerical dispersion. It should be pointed out that this rule of thumb is based on a limited number of numerical experiments and may not be true for all situations.

Under certain circumstances, the use of the MOC scheme alone may require too much computer memory and execution time while the use of the MMOC scheme may lead to noticeable numerical dispersion, so that neither provides a satisfactory solution. In these cases, a hybrid scheme, HMOC, combining the MOC and MMOC schemes may work best. The fundamental idea behind the HMOC scheme is to combine the strengths of the MOC and the MMOC techniques by using an automatic adaptive procedure which applies MOC in areas of steep concentration gradients and MMOC in areas of low concentration gradients. The automatic selection is based on the sharpness of the concentration fronts, measured by the relative concentration gradient between the cell being considered and its neighboring cells as defined in Equation 120. The implementation of the hybrid scheme is through the use of a user-specified criterion *DCHMOC*:

$$\begin{cases} MOC \text{ is used}, & \text{if } DCCELL_{i,j,k} > DCHMOC \\ MMOC \text{ is used}, & \text{if } DCCELL_{i,j,k} \le DCHMOC \end{cases} \tag{132}$$

At the beginning of each transport step, the value of *DCCELL* calculated at each cell is compared to *DCHMOC*. If $DCCELL_{i,j,k} > DCHMOC$, the advection term at that cell is solved using the MOC technique with the aid of moving particles. If there are no particles present at that cell, new particles are inserted. On the other hand, if $DCCELL_{i,j,k} \le DCHMOC$, the advection term at that cell is solved using the MMOC technique. If there are still particles present at that cell, these particles are removed. The *DCHMOC* criterion is empirical, but values between 0.001 and 0.01 have been found to be generally adequate for the test problems discussed in Chapter 7. By selecting an appropriate valve for *DCHMOC*, the adaptive procedure can provide accurate solutions to the transport problems over the entire range of mesh Peclet numbers from 0 to ∞ with virtually no numerical dispersion, while at the same time using far fewer moving particles than would be required by the MOC scheme alone.

Mass Budget Calculations

A mass budget is calculated at the end of each transport step and accumulated to provide summarized information on the total mass into or out of the groundwater flow system. The discrepancy between the total mass in and out also serves as an indicator of the accuracy of the simulation results.

Sources that release mass into the aquifer system, and sinks that remove mass from the aquifer system, include constant-concentration boundaries, constant-head boundaries, general-head-dependent boundaries, wells, drains, rivers, recharge, evapotranspiration, and chemical reactions. In addition, mass release from aquifer storage and mass accumulation in aquifer storage are also considered as sources and sinks, inasmuch as release from mass storage effectively adds mass to the groundwater flow system and accumulation in mass storage effectively removes mass from the groundwater flow system. The difference is that the changes in mass storage do not involve mass entering or leaving the aquifer as with other types of sources or sinks. The mass accumulated in or released from mass storage includes the solution phase (solute mass) and the solid phase (sorbed mass) if sorption is simulated.

The difference between the total mass in and out is calculated, as a percent error, using the following formula:

$$\text{DISCREPANCY}(\%) = \frac{(\text{IN} - \text{OUT})}{0.5(\text{IN} + \text{OUT})} \times 100 \qquad (133)$$

where IN is total mass into the groundwater flow system from external sources plus mass release from storage as a result of the decrease in solute and sorbed concentrations, and OUT is total mass out of the groundwater flow system through sinks plus mass accumulation in storage as a result of the increase in solute and sorbed concentrations. DISCREPANCY is the percentage discrepancy between IN and OUT. Generally, the mass balance error is an indication of the validity of a numerical solution, and it should be small for the numerical solution to be acceptable.

Equation 133 is a robust estimator of mass discrepancy errors in a flow or transport model and is applicable under any circumstance. However, because the total mass in the aquifer at any particular time is not explicitly included in Equation 133, it is possible that in some cases Equation 133 fails to provide a true measure of mass conservation (i.e., the total mass in the aquifer at any particular time should be equal to the initial mass in the aquifer, plus the mass gained through all sources, minus the mass lost through all sinks. For this reason, an alternative mass discrepancy indicator is also used in MT3DMS, which is defined as

$$\text{DISCREPANCY}(\%) = \frac{(\text{SOURCE} + M_o) - (\text{SINK} + M_t)}{0.5[(\text{SOURCE} + M_o) + (\text{SINK} + M_t)]} \times 100 \qquad (134)$$

where

M_t = total mass (dissolved and sorbed) in the aquifer at time t

M_o = initial mass (dissolved and sorbed) in the aquifer at the beginning of the simulation

SOURCE = total mass into the aquifer from all external sources

SINK = total mass out of the aquifer from all external sinks

If the standard finite-difference method or the third-order TVD method is used, the mass balance discrepancy should be small, generally much less than 1 percent. If this is not the case, it is likely that either the solution has not converged or some errors have occurred. This is because the standard finite-difference method or the third-order TVD method starts from a mass balance statement for each model cell, and thus for the solution to be valid, the mass balance discrepancy must be small.

Due to the discrete nature of moving particles, however, a particle-tracking-based solution does not guarantee that the mass into a cell will equal the mass out of that cell in a particular transport step. This can best be illustrated in the following example. Assume that a well, as shown in Figure 21 is continuously injecting water of a certain concentration at a fixed rate into Cell 1, which has the same concentration as the injected water. All cells except Cell 1 have zero initial concentration. Suppose that four moving particles are placed at Cell 1. It is obvious that it takes some time before the particles move into Cell 2. Until that time (denoted as t_0), there is no change in the aquifer mass storage, but there is flow of mass into the aquifer from the injection well. Therefore, if the transport step size selected for the simulation is smaller than t_0, the mass balance discrepancy is 200 percent for the first step, because the *OUT* term is equal to 0 while the *IN* term is not. This, of course, does not mean the solution is incorrect; it is simply a characteristic of moving particle methods and of the selection of the step sizes. As the simulation goes on, the mass balance error should decrease to near zero as the increase in mass storage, due to the increase in aquifer concentrations, approximates the mass entering the aquifer from the injection well. Therefore, it is generally not very meaningful to look at the mass balance at a particular step with moving particle methods; rather, it is the averaged and accumulated mass balance information that is more indicative of the overall acceptability of the simulation.

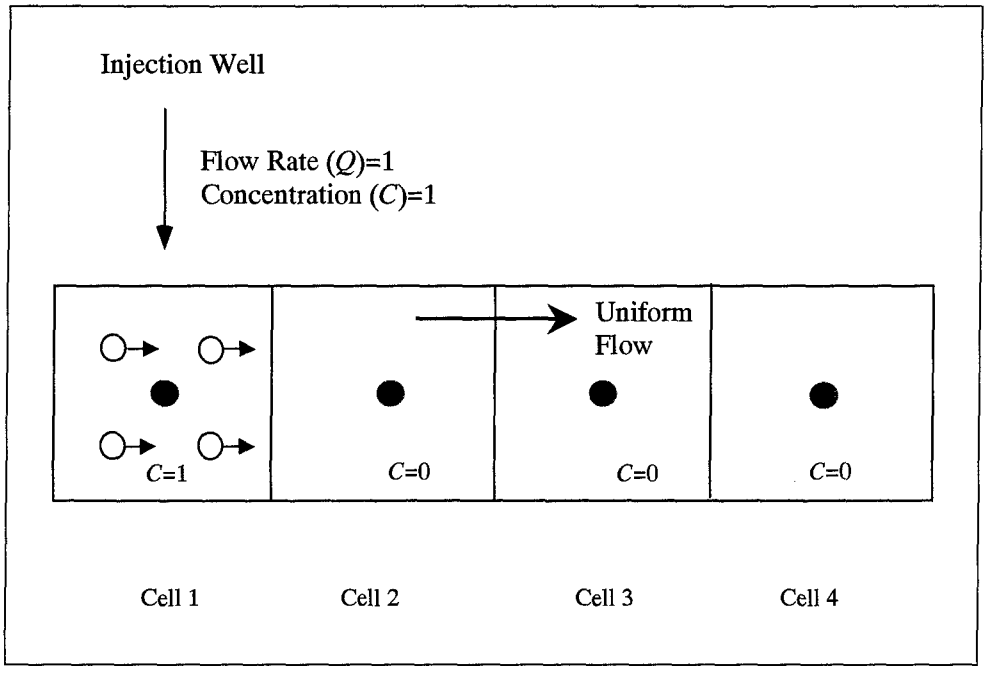


Figure 21. Illustration of the mass balance problem associated with an Eulerian-Lagrangian technique due to the discrete nature of moving particle methods

5 Program Structure and Design

Overall Structure

The computer program of the MT3DMS transport model uses a modular structure similar to that implemented in the U.S. Geological Survey modular three-dimensional finite-difference groundwater flow model, or MODFLOW (McDonald and Harbaugh 1988; Harbaugh and McDonald 1996). Like the MODFLOW model, the MT3DMS model consists of a main program and a large number of highly independent subroutines, called modules, which are grouped into a series of "packages." Each of these packages deals with a single aspect of the transport simulation. The similarity between MT3DMS and MODFLOW in the program structure and design is intended to facilitate the use of the MT3DMS transport model in conjunction with MODFLOW, one of the most widely used flow models.

The general procedures performed in the transport model for a typical simulation run are illustrated in Figure 22. The simulation time is divided into "stress periods", also referred to as "pumping periods", within which the stress (i.e., the sink/source) parameters are constant. Each stress period, in turn, is divided into a series of time-steps. The hydraulic heads and fluxes at each time step are solved by the flow model and used by the transport model. Since the transport model generally has more stringent requirements on the time-step size than the flow model does, the length of each time-step for the flow solution may exceed the limitation required for accuracy or stability in the transport solution. Thus, each time-step used in obtaining the flow solution is further divided into a number of smaller time increments, termed transport steps, within which the hydraulic heads and fluxes are assumed to be constant.

Prior to entering the stress period loop, the program executes three procedures which pertain to the simulation as a whole (Figure 22). In the Define procedure, the simulation problem is defined, that is, the size of the model, the number of stress periods, the number of species if the simulation is multispecies, and the various transport options to be used in the simulation are specified. In the Allocate procedure, computer memory is allocated for the data arrays whose dimensions depend on the parameters specified in the Define procedure. In the Read & Prepare procedure, input data that are constant throughout the simulation

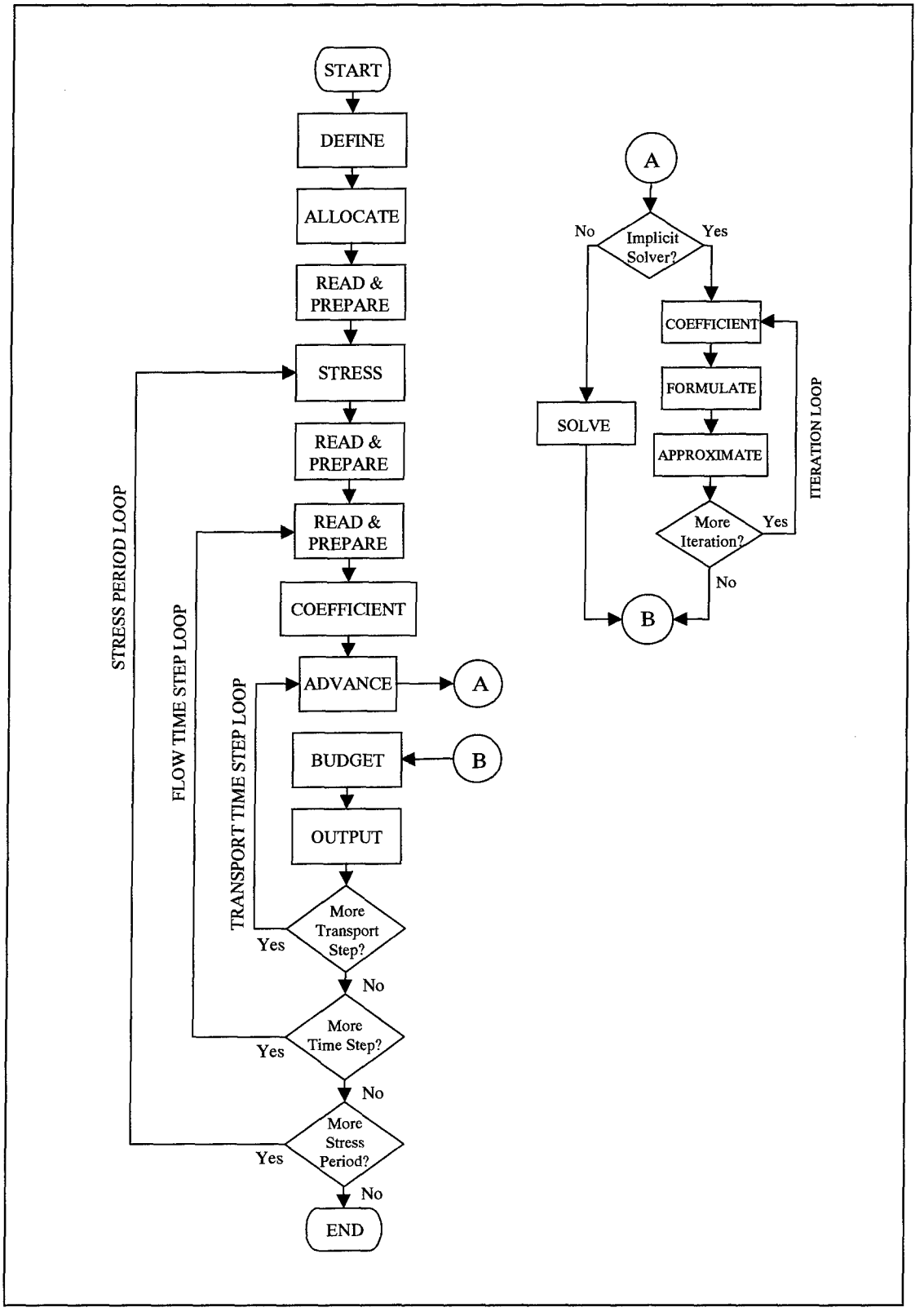


Figure 22. General procedures for a typical transport simulation

are read and processed. These input data include spatial and temporal discretization information, boundary conditions, initial conditions, transport parameters, solution parameters, and output control options.

The first procedure within the Stress Period Loop is the Stress procedure which obtains timing information for the current stress period: the length of the stress period, the number of time-steps within the stress period, and the length of each time-step. A second Read & Prepare procedure then reads and prepares input data which are constant within the current stress period (i.e., the concentrations of those sources or sinks that need to be specified). The transport model obtains the location, type, and flow rates of all sources and sinks simulated in the flow model from an unformatted flow-transport link file saved by the flow model. Source concentrations are automatically set equal to zero unless the user specifies a different concentration through this Read & Prepare procedure. Sink concentrations are always set equal to the concentration in the aquifer at the sink location, except for evapotranspiration, whose concentration can be specified through this "Read & Prepare" procedure.

Within the Flow Time-Step Loop, a third Read & Prepare procedure reads and processes the hydraulic heads and flow terms saved by the flow model, automatically incorporating the specified hydrologic boundary conditions. The Coefficient procedure then calculates certain coefficients that are constant within each time step of the head solution, such as the dispersion coefficient.

Within the Transport Time-Step Loop, the Advance procedure determines an appropriate step size for use in the current transport time-step. Depending on the choice of explicit or implicit solution schemes, different procedures follow the Advance procedure. If an explicit scheme is chosen, the Solve procedure solves each transport component directly in one step without any iteration. If an implicit scheme is chosen, the program enters the Iteration Loop. Within the Iteration Loop, another Coefficient procedure updates nonlinear coefficients that vary with each iteration, the Formulate procedure updates and prepares the coefficient matrices for the iterative solver, and the Approximate procedure solves the linear equations. The Budget procedure calculates and prepares global mass balance information, and the Output procedure saves simulation results as needed according to the user-specified output control options.

The general procedures outlined in the preceding discussion are implemented for each of the four components in the transport equation: advection, dispersion, sink/source mixing, and chemical reactions. The implementation is done through the use of individual modules, or highly independent subroutines, each of which performs one particular procedure. For example, the advection component is implemented through five modules named ADV3AL, ADV3RP, ADV3SV, ADV3FM, and ADV3BD, respectively, where ADV indicates the transport component (advection) for which these modules are implemented. The number 3 indicates the current version number of the computer code. AL (abbreviation for Allocate), RP (abbreviation for Read & Prepare), SV (abbreviation for Solve explicitly), FM (abbreviation for Formulate), and BD (abbreviation for Budget) indicate the procedures these modules perform. These modules, which are called by the main program, are termed primary modules to distinguish them from

secondary modules, which are used only inside the primary modules to which they belong. The primary modules ADV3AL, ADV3RP, ADV3SV, ADV3FM, and ADV3BD and their associated secondary modules are grouped into a "package", called the Advection Package (abbreviated as ADV) (Figure 23). Dispersion, sink/source mixing, and chemical reactions are similarly implemented and grouped into the Dispersion Package (DSP), Sink & Source Mixing Package (SSM), and Chemical Reaction Package (RCT).

In addition to the four transport component packages, the MT3DMS model program includes four additional packages: the Basic Transport Package (BTN), the Flow Model Interface Package (FMI), the Generalized Conjugate Gradient Package (GCG), and the Utility Package (UTL). The BTN handles basic tasks that are required by the transport model as a whole. Among these tasks are definition of the simulation problem, specification of the initial and boundary conditions, determination of appropriate transport step size, preparation of global mass balance information, and output of simulation results. The FMI interfaces with a flow model to obtain the flow solution from the flow model. Currently, the interfacing is done through an unformatted disk file containing hydraulic heads and various flow and sink/source terms solved by the flow model. This file is read and processed in the form needed by the transport model. The GCG solves the matrix equations resulting from the implicit finite-difference solution based on the generalized conjugate gradients methods. The UTL contains several utility modules which are called upon by other modules to perform general computer input and output tasks.

All of the primary modules contained in the MT3DMS transport model as organized by package and procedure are shown in Figure 23. All the packages documented in this manual are listed in Table 2. A significant difference between MT3DMS and the original MT3D code is the addition of the GCG solver package and the Formulate modules into other transport packages to formulate the matrix coefficients needed by the GCG solver.

Memory Allocation

The amount of computer memory required to run a specific model is dynamically allocated in MT3DMS at run time. MT3DMS uses two one-dimensional arrays called the "X" array (for real variables) and the "IX" array (for integer variables) to store individual data arrays whose dimensions depend on the problems to be simulated. The sizes of individual arrays are calculated and accumulated by the Allocate procedure of their respective packages; the accumulated sizes serve as pointers indicating the locations of individual arrays within the X or IX array. At the end of all Allocate procedures, the accumulated sizes of the individual arrays are set equal to the required dimensions for the X and IX arrays which are then allocated dynamically using the ALLOCATE statement of FORTRAN-90. If the computer system does not have sufficient memory, an error message is issued and the program execution is terminated.

Procedures	Packages						
	BTN	FMI	ADV	DSP	SSM	RCT	GCG
Define (DF)	BTN3DF						
Allocate (AL)	BTN3AL	FMI3AL	ADV3AL	DSP3AL	SSM3AL	RCT3AL	GCG3AL
Read & Prepare (RP)[1]	BTN3RP		ADV3RP	DSP3RP		RCT3RP	GCG3RP
Stress (ST)	BTN3ST						
Read & Prepare (RP)[2]					SSM3RP		
Read & Prepare (RP)[3]		FMI3RP					
Coefficient (CF)[4]				DSP3CF			
Advance (AD)	BTN3AD						
Solve (SV)	BTN3SV		ADV3SV	DSP3SV	SSM3SV	RCT3SV	
Coefficient (CF)[5]						RCT3CF	
Formulate (FM)	BTN3FM		ADV3FM	DSP3FM	SSM3FM	RCT3FM	
Approximate (AP)							GCG3AP
Budget (BD)	BTN3BD		ADV3BD	DSP3BD	SSM3BD	RCT3BD	
Output (OT)	BTN3OT						

Note:
[1]This Read & Prepare Procedure reads and processes input data that are constant throughout the entire simulation.
[2]This Read & Prepare Procedure reads and processes input data that are constant within each stress period.
[3]This Read & Prepare Procedure reads and processes input data that are constant within each time step of the flow solution.
[4]This Coefficient Procedure calculates coefficients that are constant within each time step of the flow solution.
[5]This Coefficient Procedure updates nonlinear reaction coefficients during each outer iteration of the matrix solver.

Figure 23. Primary modules of MT3DMS as organized by procedures and packages

Table 2
Packages Included in the MT3DMS Transport Model

Package Name	Abbreviation	Package Description
Basic Transport	BTN	Handles basic tasks that are required by the entire transport model. Among these tasks are definition of the problem, specification of the boundary and initial conditions, determination of the step size, preparation of mass balance information, and printout of the simulation results.
Flow Model Interface	FMI	Interfaces with a flow model. Currently, the interfacing is done through an unformatted disk file containing heads and flow terms. The FMI Package reads the contents of this file and prepares heads and flow terms in the form needed by the transport model.
Advection	ADV	Solves the concentration change due to advection with an explicit scheme or formulates the coefficient matrix of the advection term for the matrix solver.
Dispersion	DSP	Solves the concentration change due to dispersion explicitly or formulates the coefficient matrix of the dispersion term for the matrix solver.
Sink & Source Mixing	SSM	Solves the concentration change due to sink/source mixing explicitly or formulates the coefficient matrix of all sink/source terms for the matrix solver.
Chemical Reactions	RCT	Solves the concentration change due to reaction explicitly or formulates the coefficient matrix of the reaction term for the matrix solver.
Generalized Conjugate Gradient Solver	GCG	Solves the matrix equations resulting from the implicit solution of the transport equation.
Utility	UTL	Contains utility modules that are called upon by primary modules to perform such general-purpose tasks as input/output of data arrays.

The dimensions of the X and IX arrays required for a specific problem depend on the type and number of packages used. The exact sizes of the X and IX arrays can be calculated according to the instructions in Appendix B. They can also be known by simply running the program, which will always print out the required X and IX array sizes, even if the program stops execution because the computer system does not have enough memory for the required X and IX array sizes. As a general rule, a simulation using the implicit matrix solver requires additional memory ranging from 15 to 46 times the total number of model nodes, compared with the same simulation without using the implicit matrix solver. The additional memory required by the particle-tracking-based solution schemes is approximately equal to 4 to 8 times the maximum total number of moving particles and the number of mobile species.

The MT3DMS model program is written for 3-D simulation. When the program is used for 1- or 2-D applications some arrays are not needed, and, to save memory storage, these arrays are not allocated space in the X or IX array.

For example, array ZP is intended for storing the vertical coordinates of moving particles. If the program is used to simulate a two-dimensional problem in plan view, ZP is not needed and thus is not allocated storage space, and, as a result, it is never used or operated upon in any way in the program. This is important to keep in mind when the user wants to modify the computer code.

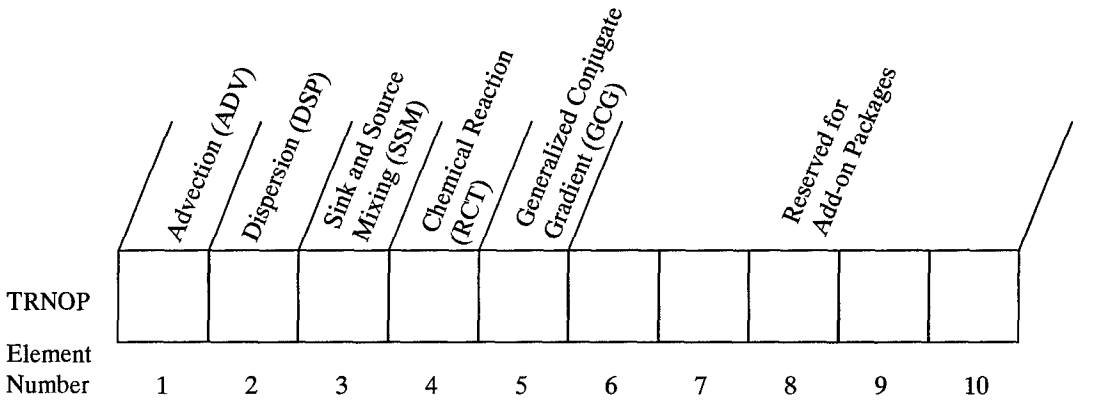
Input Structure

Like the original MT3D code, the input structure of the MT3DMS code is designed to gather input data from as many different files as needed in a simulation. This structure is similar in many ways to that of MODFLOW. Therefore, the users who are familiar with MODFLOW should find it very easy and straightforward to understand and prepare input for the MT3DMS model. In addition, the input structure of MT3DMS is compatible with that of MT3D so that any input files prepared previously for MT3D can still be used by MT3DMS without any modification. However, it should be pointed out that without making any changes to the old input files, most of the new features added to MT3DMS cannot be utilized.

The BTN is always used in every simulation. Thus, an input file for the BTN is required every time the program is run. In the input file to the BTN, there is a record containing the logical TRNOP array. Each element of the TRNOP array corresponds to a major option, generally a package (Figure 24). An option is invoked by setting the value of its corresponding TRNOP element to T (for True) and turned off by setting the value to F (for False). When an option is used, an input file containing data exclusively for that option is then required. For example, the second element of the TRNOP array corresponds to the Dispersion Option (or Package). If it is necessary to simulate dispersion, the second TRNOP array element must be entered as T. An input file containing dispersion parameters must be created and read by the program.

In addition to the main input file for the BTN and the input files for the various transport component packages which are used, the transport model always requires another input file which contains saturated thickness, fluxes across cell interfaces, and the location and flow rate of sinks and/or sources, including transient groundwater storage. This file is generated by a flow model used in conjunction with the transport model, and read by the FMI of the MT3DMS model. The structure and form of this file are described in Chapter 6 and Appendix C.

In the standard ANSI FORTRAN language, a data file must be opened with a FORTRAN unit number and a file name before it can be accessed by the program. This is done in the subroutine OPENFL which is included in the Utility Package. The file names can be entered interactively from the monitor screen or through a response file in a batch mode as discussed in the next chapter. The unit numbers associated with major input files are preset in the main program. They can be changed, if necessary, by modifying the following parameter statement in the main program:



Sample TRNOP Input Record

TRNOP	T	T	T	F	T	F	F	F	F	F
Element Number	1	2	3	4	5	6	7	8	9	10

Note:
1. The Advection Option is used; an input file is needed for the Advection (ADV) Package;
2. The Dispersion Option is used; an input file is needed for the Dispersion (DSP) Package;
3. The Sink & Source Option is used; an input file is needed for the Sink & Source Mixing (SSM) Package;
4. The Chemical Reaction Option is not used; an input file is not needed for the Chemical Reaction (RCT) Package;
5. The Generalized Conjugate Gradient Solver is used; an input file is needed for the Generalized Conjugate Gradient (GCG) Package;
6-10. Reserved for future add-on packages.

Figure 24. Specification of the transport components and solver to be included using the logical TRNOP array (the TRNOP array is entered in the input file to the BTN)

```
PARAMETER (INBTN=1,INADV=2,INDSP=3,INSSM=4,INRCT=8,
          INGCG=9,INUHF=10)
```

where

INBTN = unit number for the Basic Transport Package

INADV = unit number for the Advection Package

INDSP = unit number for the Dispersion Package

INSSM = unit number for the Sink & Source Mixing Package

INRCT = unit number for the Chemical Reaction Package

INGCG = unit number for the Generalized Conjugate Gradient Package

INUHF = unit number for the unformatted flow-transport link file, read by the Flow Model Interface Package

Once the unit number associated with a package is used in the simulation, it must not be used again elsewhere.

Output Structure

The MT3DMS program generates a standard output file and several optional output files. The standard output file is generated every time the model is run. The optional output files are generated only if they are requested. The amount, type, and frequency of information to be written on the output files are controlled by the user-specified options in the input file to the Basic Transport Package. The functions of these output files are listed below:

a. The Standard Output File contains echo of input data (so that they can be checked to ensure they have been read in properly), printout of calculated concentrations for each species, and some other useful information such as model-calculated parameters and mass budgets, at user-specified times and at the end of each stress period.

b. The Unformatted Concentration File, one for each species, (default name: MT3Dnnn.UCN where nnn is the species index number) contains concentrations saved at user-selected times in the unformatted (binary) form for postprocessing purposes or for a continuation run.

c. The Observation File, one for each species (default name: MT3Dnnn.OBS where nnn is the species index number) contains concentrations versus total elapsed time at user-specified observation points at every transport step or at a user-specified interval.

d. The Mass Budget Summary File, one for each species, (default name: MT3Dnnn.MAS where nnn is the species index number) contains a one-line summary of mass budgets at every transport step or at a user-specified interval.

e. The Model Grid Configuration File (default name: MT3D.CNF) contains model spatial discretization information to be used by the postprocessor for graphic presentation (Appendix D).

Note that the formats and structures of the default files, MT3Dnnn.UCN, MT3Dnnn.OBS, and MT3Dnnn.MAS, are identical to those of single-species counterparts, MT3D.UCN, MT3D.OBS, and MT3D.MAS, as saved by the original MT3D code. Thus, they can be processed by the same existing graphical interface software packages and other postprocessing programs without any changes.

The unit numbers associated with these output files are preset in the following parameter statement in the main program:

```
PARAMETER (IOUT=16,ICNF=17,IUCN=200,IOBS=400,IMAS=600)
```

where

IOUT = unit number for the Standard Output File

ICNF = unit number for the Model Grid Configuration File, MT3D.CNF

IUCN = starting value for the unit number of the Unformatted Concentration File, MT3Dnnn.UCN. The unit number for Species #nnn is set equal to (IUCN+nnn). For example, the Unformatted Concentration File for Species #001 is MT3D001.UCN, saved on unit 201

IOBS = starting value for the unit number of the Observation File, MT3Dnnn.OBS. The unit number for Species #nnn is set equal to (IOBS+nnn). For example, the Observation File for Species #001 is MT3D001.OBS, saved on unit 401

IMAS = starting value for the unit number of the Mass Balance Summary File, MT3D001.MAS. The unit number for Species #nnn is set equal to (IMAS+nnn). For example, the Mass Budget Summary File for Species #001 is MT3D001.MAS, saved on unit 601

These unit numbers can be changed, if necessary, by modifying the above parameter statement.

Computer Program Description (Version 3.5)

The computer program of the MT3DMS transport model is written in the standard FORTRAN 77 language as defined by the American National Standards Institute (ANSI). However, several useful FORTRAN-90 extensions have been utilized in the code, including the following statements:

IMPLICIT NONE
ALLOCATABLE
ALLOCATE
DO...ENDDO
CYCLE
EXIT

Thus, the source code is intended for compiling by a standard FORTRAN 90 compiler, although some FORTRAN 77 compilers, such as the Lahey F77L3 compiler, support all of these extensions.

All real data are declared single precision in the source code. If it is necessary to run the model in double precision, simply change the statement `IMPLICIT NONE` in every subroutine of the MT3DMS code to `IMPLICIT DOUBLE PRECISION (A-H,O-Z)`. Also, the user must modify the LKMT3 package added to MODFLOW or other flow models to save the flow terms in double precision.

The main program and all the subroutines of the MT3DMS program (including primary and secondary modules) are described briefly in the following sections. For more information, the user may refer to the extensively commented source code.

Main program – *MAIN350*

The main program controls the overall execution of the entire program. A flowchart showing all the primary modules called by the main program is provided in Figure 25. The basic steps of the main program for each simulation are:

a. Assign unit numbers to major input and output files and open these files.

b. Define the simulation problem in terms of layers, rows, columns, stress periods, and major transport options to be used.

c. Calculate the required sizes of the X and IX arrays to store all individual data arrays.

d. Allocate memory for the required sizes of the X and IX arrays. If the computer does not have enough memory, stop.

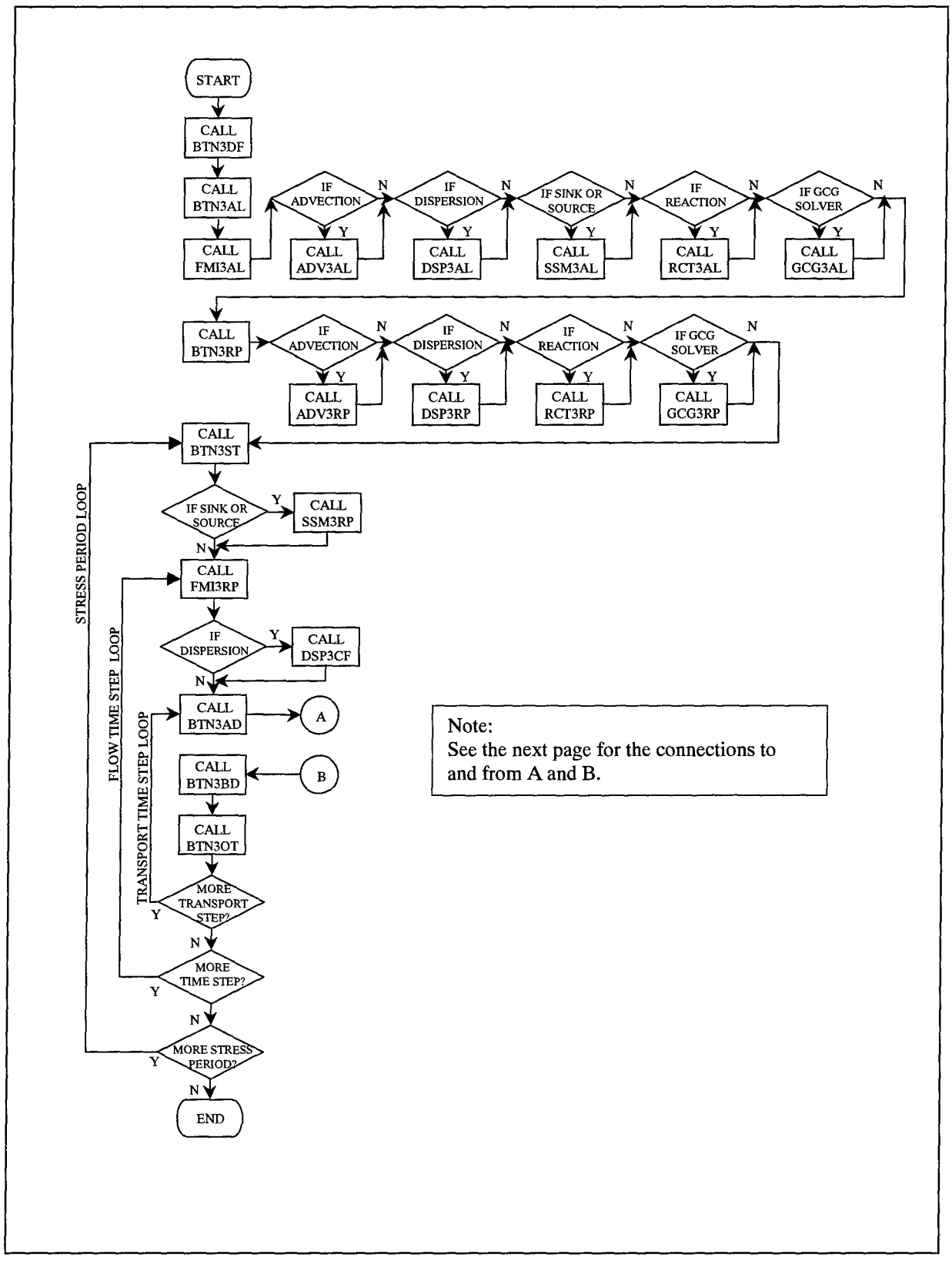


Figure 25. Flowchart for the main program of the MT3DMS code (Continued)

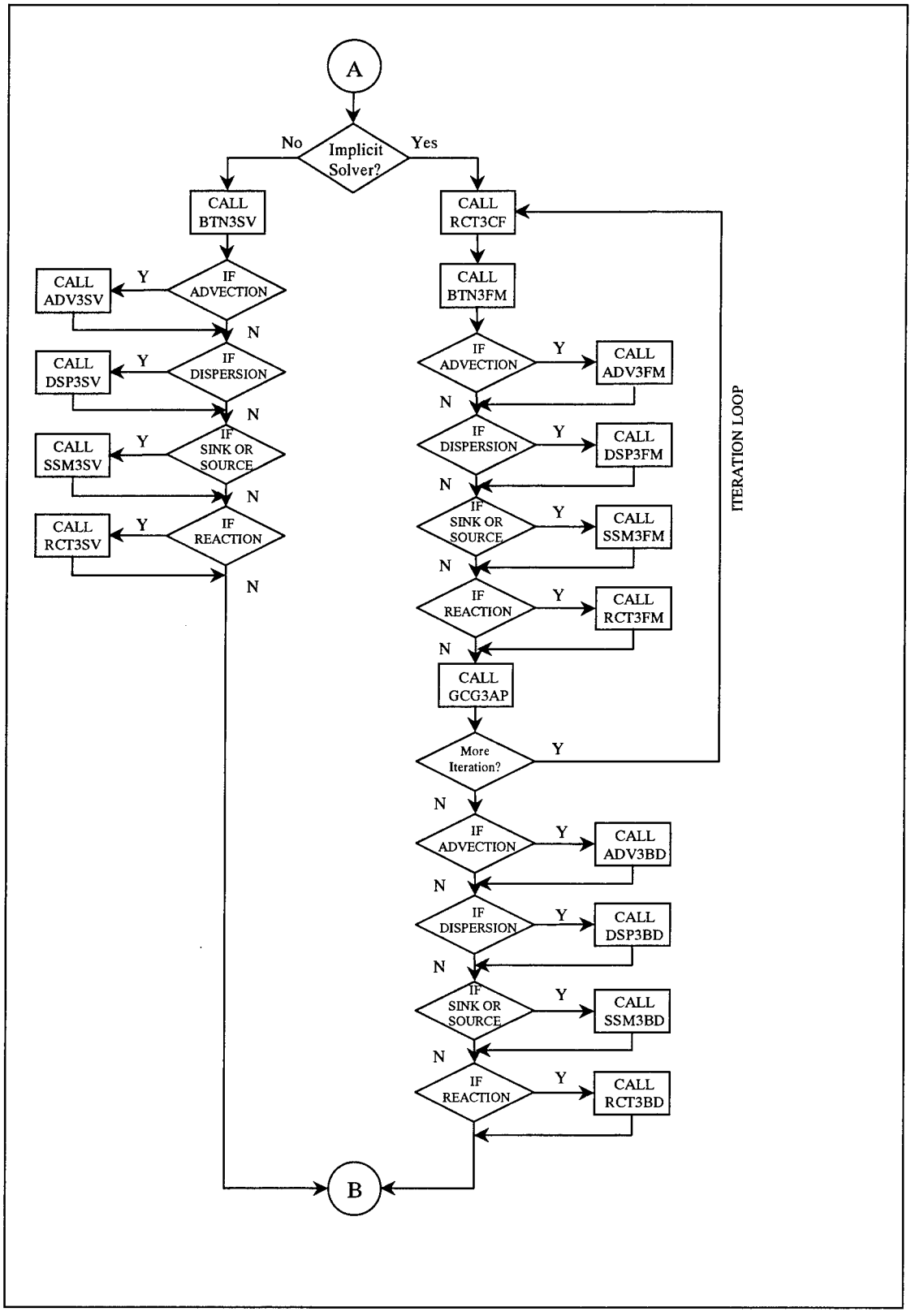


Figure 25. (Concluded)

e. Read and process input data which are constant throughout the simulation.

f. For each stress period:

(1) Obtain stress period timing information.

(2) Read and process the concentrations of sources or sinks that need to be specified.

(3) For each flow time-step:

(a) Read saturated thicknesses, flows across cell interfaces, and the locations and flow rates of sinks and/or sources.

(b) Calculate coefficients that are constant within the current time-step.

(c) For each transport time-step:

(i) Determine an appropriate step size for the current transport step.

(ii) Either solve each component of the transport equation explicitly or formulate the coefficient matrices of each transport component for the iterative solver.

(iii) Prepare mass balance information.

(iv) Print or save simulation results.

(v) If the number of transport steps exceeds the specified maximum, stop.

g. End program.

Basic transport package – *BTN350*

The BTN Package consists of nine primary modules, each of which is described below in the order of execution.

BTN3DF: Define the simulation problem by reading the number of layers, rows, columns, and stress periods as well as the total number of species and the transport options to be used.

BTN3AL: Allocate space for basic data arrays needed by the model as a whole.

BTN3RP: Read and prepare basic data arrays used by the entire model.

BTN3ST: Read stress period timing information.

BTN3AD: Determine an appropriate step size for each transport step.

BTN3FM: Reset and formulate matrix coefficients for the next step of implicit solution.

BTN3SV: Reset arrays and prepare for the next step of explicit solution.

BTN3BD: Prepare global mass balance information.

BTN3OT: Print and save simulation results according to the user-specified options.

Advection package – *ADV350*

The ADV Package consists of five primary modules and a number of secondary modules:

ADV3AL: Allocate space for data arrays needed by the Advection Package.

ADV3RP: Read and prepare input data needed for solving the advection term.

ADV3FM: Formulate matrix coefficients related to the advection term if the implicit finite-difference method is used.

ADV3SV: Solve the advection term using (a) one of the particle-tracking-based Eulerian-Lagrangian methods, (b) the explicit finite-difference method, or (c) the third-order TVD method. This primary module includes the following secondary modules, which are named according to the functions they perform:

- SADV3F: Solve the advection term using the explicit upstream finite-difference method.
- SADV3U: Solve the advection term using the third-order TVD method (ULTIMATE). A function named CFACE is called by SADV3U to evaluate interface concentrations using the third-order interpolation algorithm.
- SADV3M: Solve the advection term using the forward-tracking MOC.
- SADV3B: Solve the advection term using the backward-tracking MMOC.
- SADV3Q: Compute mass flux into or out of a finite-difference cell based on finite-difference formulation.

The following subroutines are called by the secondary modules SADV3M and SADV3B:

VPOINT: Interpolate particle velocity at an arbitrary point based on the piecewise linear scheme.

EULER: Perform particle tracking with the first-order Euler algorithm.

RK4: Perform particle tracking with the fourth-order Runge-Kutta algorithm.

PARMGR: Manage the distribution of moving particles dynamically, inserting or deleting particles as necessary.

CNGRAD: Calculate Relative Concentration Gradient between a cell and its neighboring cells.

GENPTN: Insert particles in a finite-difference cell according to fixed patterns.

GENPTR: Insert particles in a finite-difference cell randomly.

CPOINT: Calculate concentration at an arbitrary point from neighboring nodes with the first-order polynomial interpolation.

ADV3BD: Prepare budget information relevant to the advection term if the implicit scheme is used.

Dispersion package – *DSP350*

The DSP Package consists of six primary modules, each of which is described below in the order of execution:

DSP3AL: Allocate space for data arrays needed by the Dispersion Package.

DSP3RP: Read and prepare dispersion parameters.

DSP3CF: Calculate components of the hydrodynamic dispersion coefficient.

DSP3FM: Formulate matrix coefficients related to the dispersion term if the implicit scheme is used.

DSP3SV: Solve the dispersion term using the explicit finite-difference formulation.

DSP3BD: Prepare budget information relevant to the dispersion term if the implicit scheme is used.

Sink & source mixing package – *SSM350*

The SSM Package consists of five primary modules, each of which is described below in the order of execution:

SSM3AL: Allocate space for data arrays needed by the Sink & Source Mixing Package.

SSM3RP: Read and prepare concentrations of sources and/or sinks that need to be specified.

SSM3FM: Formulate matrix coefficients related to the sink/source term if the implicit scheme is used.

SSM3SV: Solve the concentration change due to sink/source mixing using the explicit finite-difference formulation.

SSM3BD: Prepare budget information relevant to the sink/source term if the implicit scheme is used.

Chemical reaction package – *RCT350*

The RCT Package consists of six primary modules and a secondary module:

RCT3AL: Allocate space for data arrays needed by the Chemical Reaction Package.

RCT3RP: Read and prepare chemical reaction parameters.

RCT3CF: Calculate nonlinear reaction coefficients that vary with each iteration if the implicit scheme is used.

RCT3FM: Formulate matrix coefficients related to the reaction term if the implicit scheme is used.

RCT3SV: Solve the concentration change due to chemical reactions using the explicit finite-difference formulation.

RCT3BD: Prepare budget information relevant to the reaction term if the implicit scheme is used.

Both RCT3RP and RCT3SV use the following secondary module:

SRCT3R: Calculate sorbed concentration and retardation factor for the specified sorption isotherm.

Generalized conjugate gradient solver package – *GCG350*

The GCG Package consists of the following primary modules:

GCG3AL: Allocate space for data arrays needed by the Generalized Conjugate Gradient Solver Package.

GCG3RP: Read and prepare matrix solution parameters.

GCG3AP: Solve the matrix equations resulting from the implicit solution of the transport equation.

Flow model interface package – *FMI350*

The FMI Package consists of three primary modules and three secondary modules:

FMI3AL: Read the header in the unformatted flow-transport link file and obtain essential information about the flow model.

FMI3RP1: Read and prepare saturated thickness and fluxes across cell interfaces in the column, row, and layer directions on a cell-by-cell basis by calling secondary module READHQ.

FMI3RP2: Read and prepare the locations and flow rates of the various sink/source terms by calling secondary modules READPS (for reading point sinks or sources) and READDS (for reading areally distributed sinks or sources).

Utility package – *UTL350*

The UTL consists of the following utility subroutines called upon by other modules to perform general computer input/output tasks:

OPENFL: Open an input or output file.

RARRAY: Read a one- or two-dimensional REAL data array using the block, zonal, list-directed, unformatted, or any user-specified format.

IARRAY: Read a one- or two-dimensional INTEGER data array with the block, zonal, list-directed, unformatted, or any user-specified format.

RPRINT: Print a one- or two-dimensional REAL data array with the wrap or strip format.

IPRINT: Print a one- or two-dimensional INTEGER data array with the wrap or strip format.

6 Input Instructions

General Information

The MT3DMS code uses a combination of formatted, list-directed, and unformatted input forms and calls two general array readers to enter data arrays. These input forms and the array readers are described in this section for easy reference in the preparation of input files.

Input forms

a. *Formatted.* If an input value is entered using the formatted form, the type of value and the space it occupies must agree with its format specifier. Four types of input variables are used in the program: integer, real, character, and logical.

   (1) In the input instructions to be followed, the format specifier for integer variables is written as `Iw` where `I` implies that the variable must have the form of an integer (it must not contain a decimal point or exponent), and `w` is the number of spaces reserved for the variable. If the input value is less than `w` spaces wide, the unoccupied spaces are treated as blanks by default. It is always good practice to enter the value right-justified.

   (2) The format specifier for real variables is written as `Fw.d` where `w` is the number of spaces reserved for the entire input value, with the fractional part taking *d* spaces. When a decimal point is present in the actual input field, it overrides the *d* specified in the format. Thus, even though the specifier for real variables is written as `F10.0` in the input instructions, input values can be entered in such forms as `12.345` or `0.12345`. Furthermore, input values in the exponential form, such as `1.2345E-5`, are also acceptable.

   (3) The format specifier for the character variable is written as `Aw` where `w` is the number of spaces reserved for the character variable. If the character string is less than `w` characters wide, unfilled spaces are treated as blanks.

(4) Finally, the format specifier for the logical variables is written as `Lw` where `w` is the number of spaces reserved for the logical variable. The logical variable must be entered either as `T` (for True) or as `F` (for False), leaving unoccupied spaces blank.

b. *List-directed (free format).* List-directed input (also referred to as free format) has several characteristics.

(1) A list-directed record is a sequence of values separated by either commas or blanks with multiple blanks allowed.

(2) A list-directed record terminates when the number of the input values equals the number of items in the input list or until a slash (/) is encountered.

(3) A list-directed record may occupy several lines of an input file, but each new record should start at a new line.

(4) List-directed input permits the use of a repeat count in the form, *n*d* where *n* is an unsigned-nonzero integer constant, and the input *n*d* causes n consecutive values of *d* to be entered.

c. *Unformatted (binary).* An unformatted file is a sequence of unformatted records in the form of binary characters. An unformatted file cannot be visually examined, but it is smaller in size and faster to process than a formatted file. The model program uses the unformatted form to enter the flow solution saved by a flow model and writes the calculated concentrations to an unformatted file for postprocessing purposes.

Array readers RARRAY and IARRAY

Most of the input data to be entered to the model by the user will consist of one- or two-dimensional real or integer arrays. Three-dimensional arrays are treated as a series of two-dimensional arrays, each of which corresponds to an individual model layer and is entered in sequence according to the layer number.

In the MT3DMS model, arrays are entered as an "array-control record", plus, optionally, a series of records containing the array elements. If all the elements of an array have the same value, the value is specified on the control record and it is not necessary to read the associated array. If the array elements vary, records containing the array values are read using the various input forms as specified on the array-control record. To perform these tasks, two utility subroutines, RARRAY for reading one- or two-dimensional real arrays, and IARRAY for reading one- or two-dimensional integer arrays, are provided in the program. These two array readers are compatible to array readers U2DREL and U2DINT provided in the MODFLOW model (McDonald and Harbaugh 1988). However, RARRAY and IARRAY also permit the input of array values by block, zonal, and list-directed (or free) formats.

For each array to be entered, RARRAY or IARRAY reads an array-control record first on the unit reserved for the major option which calls the array reader. For example, unit 1 is preset for the Basic Transport Package, thus, RARRAY or IARRAY reads the array-control record in the input file for the Basic Transport Package from unit 1. The content and form of the array control record are as follows:

FOR REAL ARRAY READER (RARRAY):

Record:	IREAD	CNSTNT	FMTIN	IPRN
Format:	I10	F10.0	A20	I10

FOR INTEGER ARRAY READER (IARRAY):

Record:	IREAD	ICONST	FMTIN	IPRN
Format:	I10	I10	A20	I10

a. IREAD determines how array values are read.

(1) If IREAD = 0, every element in array will be set equal to the value CNSTNT for RARRAY or to ICONST for IARRAY.

(2) If IREAD = 100, an array of input values follows the control record. The array values are read in the format specified in the third field of the array-control record (FMTIN) from the same unit used for reading the array-control record (Figure 26).

(3) If IREAD = 101, an array of values organized in "block" format follows the array-control record. The block format consists of a record specifying the number of blocks, NBLOCK, followed by NBLOCK records of input values (all in free format), specifying the first row (I1), last row (I2), first column (J1), and last column (J2) of each block as well as the value (ZZ/IZ) to be assigned to the cells within the block (ZZ is a real value for RARRAY and IZ is an integer value for IARRAY), as shown below:

NBLOCK		(free format)
I1, I2, J1, J2, ZZ/IZ	(block 1)	(free format)
I1, I2, J1, J2, ZZ/IZ	(block 2)	(free format)
.....		
I1, I2, J1, J2, ZZ/IZ	(block NBLOCK)	(free format)

If a subsequent block overlaps any preceding blocks, the subsequent block overrides the preceding blocks. It is always good practice to have the first block cover the entire grid and allow subsequent blocks to override portions of the grid as this averts the possibility that any portion of the grid will be left unassigned (Figure 26).

```
      IREAD          CNSTNT                                  FMTIN          IRPN
        100              0.                                 (10F5.1)          3
  2.1   2.1   2.1   2.1   2.1   2.1   2.1   2.1   2.1   2.1
  2.1   2.1   2.1   2.1   2.1   2.1   2.1   2.1   2.1   2.1
 10.2  10.2  10.2  10.2  10.2  10.2  10.2  10.2  10.2  10.2
 10.2  10.2  10.2  10.2  10.2  10.2  10.2  10.2  10.2  10.2
 23.5  23.5  23.5  23.5  23.5  23.5  23.5   0.4   0.4   0.4
 23.5  23.5  23.5  23.5  23.5  23.5  23.5   0.4   0.4   0.4
 23.5  23.5  23.5  23.5  23.5  23.5  23.5   0.4   0.4   0.4
 23.5  23.5  23.5  23.5  23.5  23.5  23.5   0.4   0.4   0.4
 23.5  23.5  23.5  23.5  23.5  23.5  23.5   0.4   0.4   0.4
    0     0     0     0     0     0     0   0.4   0.4   0.4
```

An array of 10 by 10 is read with the user-specified format (10F5.1) (IREAD=100).

```
      101              0.                  (FMTIN not used)          3
  5   <------------  NBLOCK
  1    10    1    10    0
  1     2    1    10    2.1
  3     4    1    10    10.2
  5     9    1     7    23.5
  5    10    8    10    0.4
```

The same array as shown in (a) is read using the block format (IREAD=101).

```
      102              0.                                (10F5.0)          3
  4   <---------NZONE
 2.1  10.2  23.5        0.4   <----------  ZV(NZONE)
  1    1     1     1     1     1     1     1     1     1
  1    1     1     1     1     1     1     1     1     1
  2    2     2     2     2     2     2     2     2     2
  2    2     2     2     2     2     2     2     2     2
  3    3     3     3     3     3     3     4     4     4
  3    3     3     3     3     3     3     4     4     4
  3    3     3     3     3     3     3     4     4     4
  3    3     3     3     3     3     3     4     4     4
  3    3     3     3     3     3     3     4     4     4
  0    0     0     0     0     0     0     4     4     4
```

The same array as shown in (a) is read using the zonal format (IREAD=102).
Note that format (10F5.0) is used to read the zone indicators).

```
      103              0.                  (FMTIN not used)          3
 20*2.1  20*10.2  7*23.5  3*0.4  7*23.5  3*0.4  7*23.5  3*0.4  7*23.5  3*0.4 ,7*23.5
 , 3*0.4   7*0   3*0.4
```

(d) The same array as shown in (a) is read using the list-directed format (IREAD=103).

Figure 26. Illustration of the various input forms used by RARRAY and IARRAY

(4) If IREAD = 102, an array of values organized in "zonal" format follows the array-control record. Each zone represents one value of the input variable, and each zone is identified by a zone number -- an integer which may actually be thought of as a code for the corresponding value of the input variable. The zonal format consists of a record specifying the number of zones, NZONE, followed by an array of values for the input variables, ZV(NZONE) or IZV (NZONE), which are listed in sequence according to zone numbers (1, 2, 3, ..., etc.); ZV is a real array for RARRAY and IZV is an integer array for IARRAY as explained below. Following ZV/IZV is the zone indicator array A(NCOL,NROW)/IA(NCOL,NROW) specifying the zone number assigned to each node (cell) in a given layer of the model. The elements in this array are actually always integers (i.e., the integer codes identifying the zone for each cell). However, they are stored in the array reserved for the input variable itself and thus follow the format of the input variable. Hence, the zone indicators are read using the format as specified in the third field of the array-control record (FMTIN) which must be a real format specifier for RARRAY and an integer format specifier for IARRAY. If the zone indicator for a cell is equal to zero, the value for that cell is set to zero (Figure 26).

The zonal format for RARRAY:

NZONES	(free format)
ZV(1),.ZV(2),....., ZV(NZONE)	(free format)
A(NCOL,NROW)	(using format FMTIN)

The zonal format for IARRAY:

NZONE	(free format)
IZV(1), IZV(2), ..., IZV(NZONE)	(free format)
IA(NCOL,NROW)	(using format FMTIN)

(5) If IREAD = 103, an array of values follows the array-control record. The array values are read using list-directed or free format (Figure 26).

(6) If IREAD = any value other than 0, 100, 101, 102, and 103, array values are read from a separate file.

(a) If IREAD > 0, it is the unit number on which the external file is read using the format specified in FMTIN.

(b) If IREAD < 0, the absolute value of IREAD gives the unit number on which the array values are read from an external unformatted file. The unformatted file contains one record of header, followed by an unformatted record of NCOL*NROW values.

b. CNSTNT/ICONST is a constant. Its use depends on the value of IREAD.

(1) If IREAD = 0, every element in the array is set equal to CNSTNT/ICONST.

(2) If IREAD ≠ 0, and CNSTNT/ICONST ≠ 0, elements in the array are multiplied by CNSTNT/ICONST.

c. FMTIN is the user-specified format to read array values or the zonal indicator array. The format must be enclosed in parentheses; for example, (15F5.0) for real values and (15I5) for integer input values.

d. IPRN is a flag indicating whether or not the array being read should be printed out for checking, and it also serves as a code indicating the format that should be used in printing. It is used only if IREAD is not equal to zero. IPRN is set to zero if the specified value exceeds those defined in Table 3. If IPRN is less than zero, the array will not be printed.

Table 3
Printing Formats Corresponding to the IPRN Code (after McDonald and Harbaugh 1988)

IPRN	RARRAY	IARRAY
0	10G11.4	10I11
1	11G10.3	60I1
2	9G13.6	40I2
3	15F7.1	30I3
4	15F7.2	25I4
5	15F7.3	20I5
6	15F7.4	
7	20F5.0	
8	20F5.1	
9	20F5.2	
10	20F5.3	
11	20F5.4	
12	10G11.4	

Units of Input and Output Variables

The MT3DMS code uses any consistent units for input and output variables. In the input file to the Basic Transport Package, the user decides the units for time, length, and mass. Then, any input variable or constant should be entered in units consistent with the three basic units. For example, suppose that day (d) is chosen as unit for time, feet (ft) for length, and pound (lb) for mass. Then, hydraulic heads should have the unit of ft, solute concentration the unit of lb/ft^3, dispersivity the unit of ft, distribution coefficient for linear sorption the unit of ft^3/lb, reaction rate for the first-order kinetic reactions the unit of d^{-1}, and so on.

Similarly, if SI units are preferred, and second (s), centimeter (cm), and gram (g) are selected as the units for time, length, and mass, respectively, hydraulic heads should have the unit of cm, concentration the unit of g/cm^3, dispersivity the unit of cm, distribution coefficient for linear sorption the unit of cm^3/g, reaction rate for the first-order kinetic reactions the unit of s^{-1}, and so on. It should be emphasized that the unit names entered by the user are used in MT3DMS for identification purposes only and do not affect the model simulation results in any way.

A special note is necessary on the unit for concentration. It is possible that the use of consistent units sometimes results in concentration values being extremely large or small, making the model results more vulnerable to numerical round-off errors. Furthermore, it is more convenient to work with a commonly used unit such as ppm rather than a consistent but otherwise cumbersome unit such as lb/ft^3. In the MT3DMS code, it is permissible to use any unit for concentration if no nonlinear sorption or other types of nonlinear reaction is simulated. The unit for output concentration will be identical to that of input concentration. The user should note that the mass calculated by the model has the unit of concentration times flow rate multiplied by time. Thus, if inconsistent units have been used, a conversion factor must be used to convert the calculated mass to the right unit, if the absolute values are important. It is up to the user to make the conversion. It is also possible to work with relative concentrations. In that case, all concentrations entered into the model can be scaled according to the maximum concentration (C_o) of either aquifer or fluid sources. Subsequently, the model simulates the changes in the relative concentration (C/C_o).

Interface with the Flow Model

MT3DMS is designed to be used in conjunction with a block-centered finite-difference flow model. This allows the user to construct and calibrate a flow model independently. Prior to running MT3DMS, the saturated thickness, fluxes across cell interfaces in all directions, and locations and flow rates of various sinks/sources, including transient groundwater storage, as solved by the flow model, should be saved in an unformatted "flow-transport link file". This file should be saved in such a way that it can be retrieved by MT3DMS correctly through the Flow Model Interface Package. Appendix C, Linking MT3DMS with a Flow Model, gives more detailed information on the structure and form of the unformatted flow-transport link file.

Incorporating the LinkMT3D package into the MODFLOW code

In many cases, the U. S. Geological Survey modular three-dimensional finite-difference groundwater flow model (MODFLOW) will be used for the flow solution; a package has therefore been written for use with MODFLOW to save the information needed by MT3DMS. This package, currently in the third version, is named LKMT3 and has already been incorporated into the version of the MODFLOW code distributed with MT3DMS. If it is preferred to use a version of MODFLOW other than the one distributed with MT3DMS, the LKMT3 package must be added into MODFLOW.

To add the LKMT3 package into MODFLOW, insert the statements in file LKMT3.INC into the MODFLOW main program between the following two statements:

```
CALL BAS1OC (...)
IF(CNVG .EQ. 0) STOP
```

This can be accomplished by using the `INCLUDE` statement of FORTRAN, i.e.,

```
CALL BAS1OC (...)
INCLUDE 'LKMT3.INC'
IF(CNVG .EQ. 0) STOP
```

where the syntax for the `INCLUDE` statement may vary slightly from one compiler to another. Recompile the main program after the modification. Next, compile the source file for the LKMT3 package, i.e., LKMT3.FOR or LKMT3.F90, and link it with the rest of MODFLOW files.

It should also be pointed out that the procedures described above are intended for the 1988 version of MODFLOW (referred to as MODFLOW-88) (McDonald and Harbaugh 1988). Slightly different procedures are required to add the LKMT3 package to a newer version of MODFLOW (referred to as MODFLOW-96) (Harbaugh and McDonald 1996). Refer to `MODFLOW.TXT` included with the MT3DMS distribution files for more information.

Activating the LinkMT3D package in a MODFLOW simulation

To activate the LKMT3 package added to MODFLOW, the user must enter a positive integer number in the 22nd element of the IUNIT array (i.e., IUNIT(22)) in the input file for the MODFLOW Basic Package. This instructs the MODFLOW program to save the unformatted flow-transport link file for use by MT3DMS. It also serves as the unit number on which the saturated thickness and flow terms will be saved. For this reason, the value entered must be a unique number that has not been used by other input and output files. If IUNIT(22) is specified as zero, no file will be saved by the LKMT3 package.

Input Instructions for the Basic Transport Package

Input to the BTN Package is read on unit INBTN=1, which is preset in the main program. Since the BTN package is needed for every simulation, this input file is always required. Note that underlined are new features introduced in the current version.

For each simulation:

A1 Record: HEADNG(1)
Format: A80

- HEADNG(1) is the first line of any title or heading for the simulation run. The line should not be longer than 80 characters.

A2 Record: HEADNG(2)
Format: A80

- HEADNG(2) is the second line of any title or heading for the simulation run. The line should not be longer than 80 characters.

A3 Record: NLAY, NROW, NCOL, NPER, NCOMP, MCOMP
Format: 6I10

- NLAY is the total number of layers;
- NROW is the total number of rows;
- NCOL is the total number of columns;
- NPER is the total number of stress periods;
- NCOMP is the total number of chemical species included in the current simulation. For single-species simulation, set NCOMP = 1;
- MCOMP is the total number of "mobile" species. MCOMP must be equal to or less than NCOMP. For single-species simulation, set MCOMP=1.
  Note that "mobile species" are involved in both transport and reaction while "immobile" species equal to NCOMP-MCOMP are involved in reaction only. Also, for each species included in NCOMP, MT3DMS automatically tracks a sorbed or immobile counterpart if a sorption isotherm or dual-domain mass transfer is specified through the Chemical Reaction Package. Thus, there is no need to define separate "immobile" species to simulate sorption or a dual-domain system. The ability to define separate immobile species is only intended for using MT3DMS with add-on reaction packages.

A4 Record: TUNIT, LUNIT, MUNIT
Format: 3A4

- TUNIT is the name of unit for time, such as DAY or HOUR;
- LUNIT is the name of unit for length, such as FT or M;
- MUNIT is the name of unit for mass, such as LB or KG.
  Note that these names are used for identification purposes only and do not affect the model outcome.

A5 Record: TRNOP(10)
(ADV DSP SSM RCT GCG XXX XXX XXX XXX XXX)
Format: 10L2

- TRNOP are logical flags for major transport and solution options. TRNOP(1) to (5) correspond to Advection, Dispersion, Sink & Source Mixing, Chemical Reaction, and Generalized Conjugate Gradient Solver packages, respectively. If any of these options is used, enter its corresponding TRNOP element as T, otherwise as F. TRNOP(6) through (10) are reserved for add-on packages.

A6 Record: LAYCON(NLAY)
Format: 40I2

- LAYCON is a 1-D integer array indicating the type of model layers. Each value in the array corresponds to one model layer. Enter LAYCON in as many lines as necessary if NLAY > 40.

  LAYCON = 0, the model layer is confined. The layer thickness DZ to be entered in a subsequent record will be used as the *saturated thickness* of the layer.
  LAYCON ≠ 0, the model layer is either unconfined or convertible between confined and unconfined. The saturated thickness, as calculated by the flow model and saved in the flow-transport link file, will be read and used by the transport model. (Note that this type corresponds to the LAYCON values of 1, 2, and 3 of MODFLOW; however, there is no need to distinguish between these layer types in the transport simulation.)

A7 Array: DELR(NCOL)
Reader: RARRAY

- DELR is a 1-D real array representing the cell width along rows (Δx) in the direction of increasing column indices (j). Specify one value for each column of the grid.

A8 Array: DELC(NROW)
Reader: RARRAY

- DELC is a 1-D real array representing the cell width along columns (Δy) in the direction of increasing row indices (i). Specify one value for each row of the grid.

A9 Array: HTOP(NCOL,NROW)
Reader: RARRAY

- HTOP is a 2-D array defining the top elevation of all cells in the first (top) model layer, relative to the same datum as the hydraulic heads.

If the first model layer is unconfined, HTOP can be set most conveniently to a uniform elevation above the water table (Figure 27). Note that the concentrations for the cells in the first model layer are calculated at nodal points assumed to be midway between HTOP and the bottom of the first layer. Thus, HTOP should not be set much higher than the water table. Also note that the difference between HTOP and the bottom elevation of the first layer must be equal to the layer thickness DZ to be defined in the next record.

If the first model layer is confined, HTOP is equal to the bottom elevation of the confining unit overlying the first model layer (Figure 27).

A10 Array: DZ(NCOL,NROW) (one array for each layer in the grid)
Reader: RARRAY

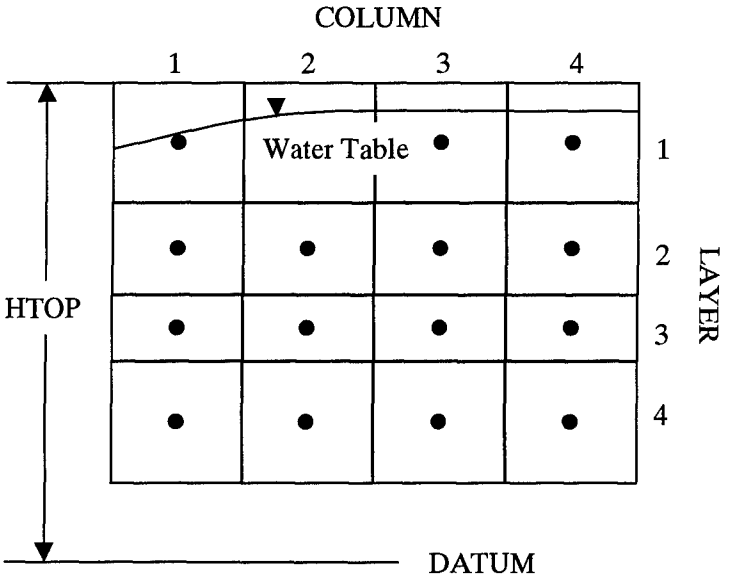
- DZ is the thickness of all cells in each model layer. DZ is a 3-D array. The input to 3-D arrays is handled as a series of 2-D arrays with one array for each layer, entered in the sequence of layer 1, 2, ..., NLAY. The thickness of the first layer must be equal to the difference between HTOP and its bottom elevation. When the grid is discretized into horizontal layers, HTOP for the first layer and DZ within each layer are uniform (Figure 28). However, if a vertically distorted grid is used, both HTOP and DZ may be variable for cells within the same layer (Figure 28).

A11 Array: PRSITY(NCOL,NROW) (one array for each layer)
Reader: RARRAY

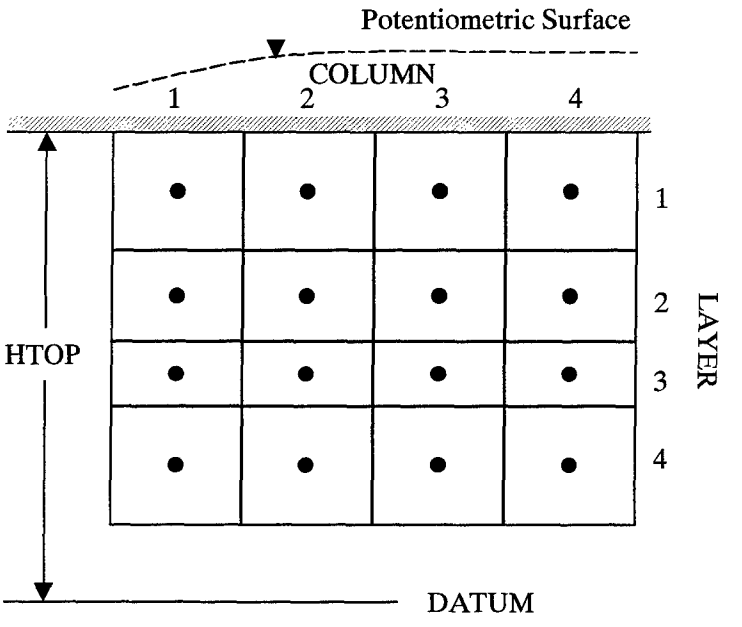
- PRSITY is the "effective" porosity of the porous medium in a single porosity system (see discussions in Chapter 2). Note that if a dual-porosity system is simulated, PRSITY should be specified as the "mobile" porosity (i.e., the ratio of interconnected pore spaces filled with mobile waters over the bulk volume of the porous medium); the "immobile" porosity is defined through the Chemical Reaction Package.

A12 Array: ICBUND(NCOL,NROW) (one array for each layer)
Reader: IARRAY

- ICBUND is an integer array specifying the boundary condition type (inactive, constant-concentration, or active) for every model cell. For multispecies simulation, ICBUND defines the boundary condition type *shared by all species.* Note that different species are allowed to have different constant-concentration conditions through an option in the Source and Sink Mixing Package.



(a) Array HTOP for an unconfined aquifer.



(b) Array HTOP for a confined aquifer.

Figure 27. Illustration of array HTOP for (a) unconfined and (b) confined aquifer layers

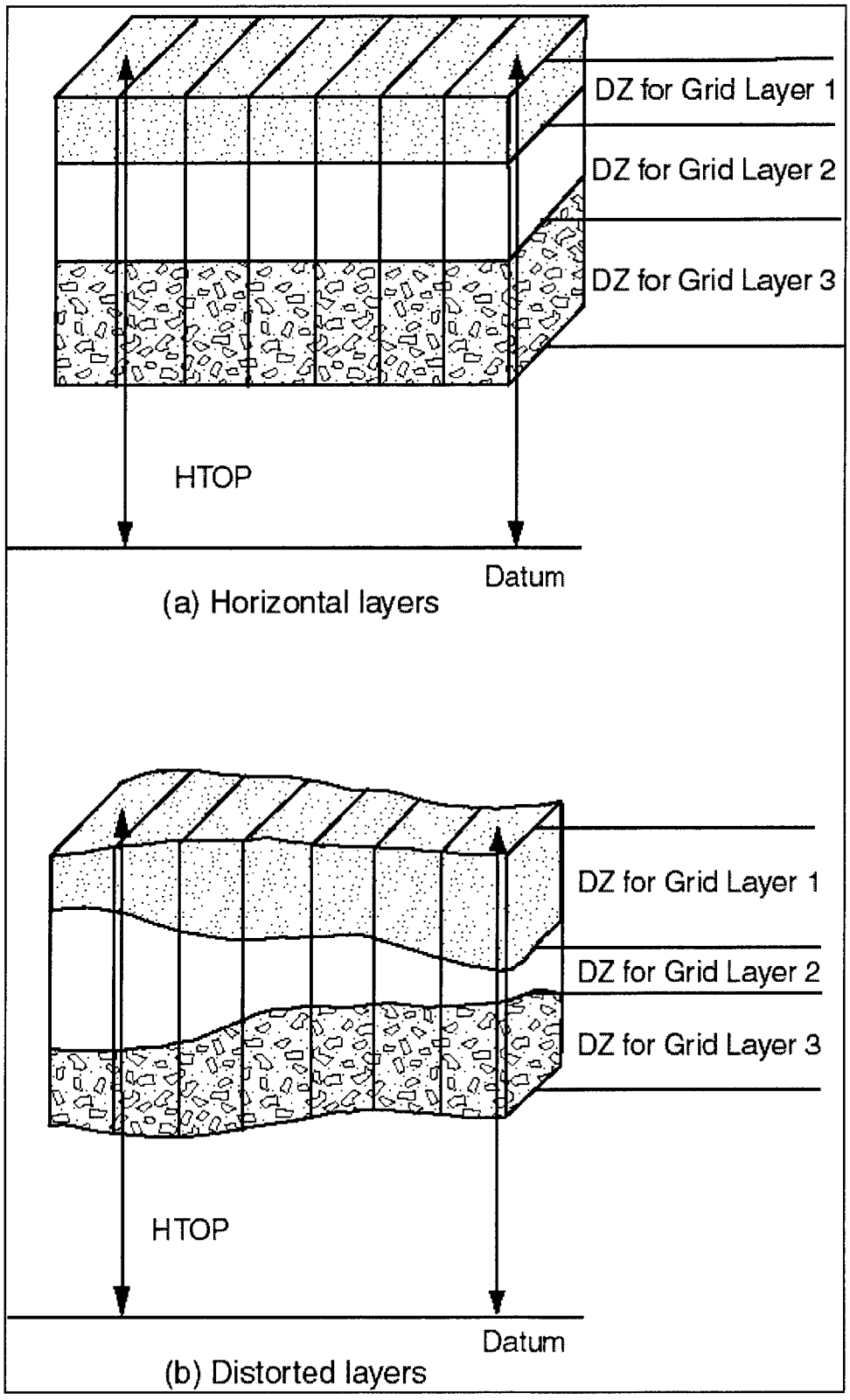


Figure 28. Arrays HTOP and DZ for different vertical discretization schemes

If ICBUND = 0, the cell is an inactive concentration cell for all species. Note that no-flow or "dry" cells are automatically converted into inactive concentration cells. Furthermore, active cells in terms of flow can be treated as inactive concentration cells to minimize the area needed for transport simulation, as long as the solute transport is insignificant near those cells.

If ICBUND < 0, the cell is a constant-concentration cell *for all species*. The starting concentration of each species remains the same at the cell throughout the simulation. (To define different constant-concentration conditions for different species at the same cell location, refer to the Sink/Source Mixing Package.) Also note that unless explicitly defined as a constant-concentration cell, a constant-head cell in the flow model is not treated as a constant-concentration cell.

If ICBUND > 0, the cell is an active (variable) concentration cell where the concentration value will be calculated.

(Enter A13 for each species)

A13 Array: SCONC(NCOL,NROW) (one array for each layer)
Reader: RARRAY

- SCONC is the starting concentration (initial condition) at the beginning of the simulation (unit, ML^{-3}). For multispecies simulation, the starting concentration must be specified for all species, one species at a time.

A14 Record: CINACT, THKMIN
Format: 2F10.0

- CINACT is the value for indicating an inactive concentration cell (ICBUND = 0). Even if inactive cells are not anticipated in the model, a value for CINACT still must be submitted.
- THKMIN is the minimum saturated thickness in a cell, expressed as the decimal fraction of the model layer thickness (DZ) below which the cell is considered inactive. The default value is 0.01 (i.e., 1 percent of the model layer thickness).

A15 Record: IFMTCN, IFMTNP, IFMTRF, IFMTDP, SAVUCN
Format: 4I10, L10

- IFMTCN is a flag indicating whether the calculated *concentration* should be printed to the standard output text file and also serves as a printing-format code if it is printed. The codes for print-formats are the same as those listed in Table 3.

- IFMTCN > 0, concentration is printed in the wrap form (Figure 29).
  < 0, concentration is printed in the strip form (Figure 29).
  = 0, concentration is not printed.
- IFMTNP is a flag indicating whether the number of particles in each cell (integers) should be printed and also serves as a printing-format code if they are printed. The convention is the same as that used for IFMTCN.
- IFMTRF is a flag indicating whether the model-calculated retardation factor should be printed and also serves as a printing-format code if it is printed. The convention is the
- same as that used for IFMTCN.
  IFMTDP is a flag indicating whether the model-calculated, distance-weighted dispersion coefficient should be printed and also serves as a printing-format code if it is printed. The convention is the same as that used for IFMTCN.
- SAVUCN is a logical flag indicating whether the concentration solution should be saved in a default unformatted (binary) file named MT3Dnnn.UCN, where nnn is the species index number, for post-processing purposes or for use as the initial condition in a continuation run.
  If SAVUCN = T, the concentration of each species will be saved in the default file MT3Dnnn.UCN. In addition, the model spatial discretization information will be saved in another default file named MT3D.CNF to be used in conjunction with MT3Dnnn.UCN for postprocessing purposes.

  If SAVUCN = F, neither MT3Dnnn.UCN nor MT3D.CNF is created.

A16 Record: NPRS
Format: I10

- NPRS is a flag indicating the frequency of the output and also indicating whether the output frequency is specified in terms of total elapsed simulation time or the transport step number. Note that what is actually printed or saved is controlled by the input values entered in the preceding record (Record A15).

  If NPRS > 0, simulation results will be printed to the standard output text file or saved to the unformatted concentration file at times as specified in record TIMPRS(NPRS) to be entered in the next record.

  If NPRS = 0, simulation results will not be printed or saved except at the end of simulation.

  If NPRS < 0, simulation results will be printed or saved whenever the number of transport steps is an even multiple of NPRS.

	1	2	3	4	5	6	7	8	9	10
	11	12	13	14	15	16	17			
1	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
2	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56			
3	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56			
4	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56			
5	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56			
6	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56			
7	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56			

(a) WRAP FORM

	1	2	3	4	5	6	7	8	9	10
1	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
2	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
3	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
4	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
5	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
6	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
7	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56

	11	12	13	14	15	16	17
1	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
2	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
3	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
4	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
5	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
6	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56
7	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56	1234.56

(b) STRIP FORM

Figure 29. Illustration of wrap and strip forms of printed output for a layer containing 7 rows and 17 columns (modified from McDonald and Harbaugh 1988)

(Enter A17 only if NPRS > 0)

A17 Record: TIMPRS(NPRS)
Format: 8F10.0

- TIMPRS is the total elapsed time at which the simulation results are printed to the standard output text file or saved in the default unformatted (binary) concentration file MT3Dnnn.UCN. Note that if NPRS > 8, enter TIMPRS in as many lines as necessary.

A18 Record: NOBS, NPROBS
Format: 2I10

- NOBS is the number of observation points at which the concentration of each species will be saved at the specified frequency in the default MT3Dnnn.OBS where nnn is the
- species index number.
  NPROBS is an integer indicating how frequently the concentration at the specified observation points should be saved in the observation file MT3Dnnn.OBS. Concentrations are saved every NPROBS step.

(Enter A19 NOBS times if NOBS > 0)

A19 Record: KOBS, IOBS, JOBS
Format: 3I10

- KOBS, IOBS, and JOBS are the cell indices (layer, row, column) in which the observation point or monitoring well is located and for which the concentration is to be printed at every transport step in file MT3Dnnn.OBS. Enter one set of KOBS, IOBS, JOBS for each observation point.

A20 Record: CHKMAS, NPRMAS
Format: L10, I10

- CHKMAS is a logical flag indicating whether a one-line summary of mass balance information should be printed, for checking and postprocessing purposes, in the default file MT3Dnnn.MAS where nnn is the species index number.

  If CHKMAS = T, the mass balance information for each transport step will be saved in file MT3Dnnn.MAS.

  If CHKMAS = F, file MT3Dnnn.MAS is not created.

- NPRMAS is an integer indicating how frequently the mass budget information should be saved in the mass balance summary file MT3Dnnn.MAS. Mass budget information is saved every NPRMAS step.

For each stress period

A21 Record: PERLEN, NSTP, TSMULT
Format: F10.0, I10, F10.0

- PERLEN is the length of the current stress period. If the flow solution is transient, PERLEN specified here must be equal to that specified for the flow model. If the flow solution is steady-state, PERLEN can be set to any desired length.
- NSTP is the number of time-steps *for the transient flow solution* in the current stress period. If the flow solution is steady-state, NSTP = 1.
- TSMULT is the multiplier for the length of successive time-steps used in the transient flow solution; it is used only if NSTP > 1.

  If TSMULT > 0, the length of each flow time-step within the current stress period is calculated using the geometric progression as in MODFLOW. Note that both NSTP and TSMULT specified here must be identical to those specified in the flow model if the flow model is transient.

  If TSMULT $\leq$ 0, the length of each flow time-step within the current stress period is read from the record TSLNGH (see A22). This option is needed in case the length of time-steps for the flow solution is not based on a geometric progression in a flow model, unlike MODFLOW.

(Enter A22 if TSMULT$\leq$0)
A22 Record: TSLNGH(NSTP)
Format: 8F10.0

- TSLNGH provides the length of time-steps for the flow solution in the current stress period. This record is needed only if the length of time-steps for the flow solution is not based on a geometric progression. Enter TSLNGH in as many lines as necessary if NSTP > 8.

A23 Record: DT0, MXSTRN, TTSMULT, TTSMAX
Format: F10.0, I10, 2F10.0

- DT0 is the user-specified transport step size within each time-step of the flow solution. DT0 is interpreted differently depending on whether the solution option chosen is explicit or implicit:

For explicit solutions (i.e., the GCG solver is not used), the program will always calculate a maximum transport step size which meets the various stability criteria. Setting DT0 to zero causes the model-calculated transport step size to be used in the simulation. However, the model-calculated DT0 may not always be optimal. In this situation, DT0 should be adjusted to find a value that leads to the best results. If DT0 is given a value greater than the model-calculated step size, the model-calculated step size, instead of the user-specified value, will be used in the simulation.

For implicit solutions (i.e., the GCG solver is used), DT0 is the initial transport step size. If it is specified as zero, the model-calculated value of DT0, based on the user-specified Courant number in the Advection Package, will be used. The subsequent transport step size may increase or remain constant depending on the user-specified transport step size multiplier TTSMULT and the solution scheme for the advection term.

- MXSTRN is the maximum number of transport steps allowed within one time step of the flow solution. If the number of transport steps within a flow time-step exceeds MXSTRN, the simulation is terminated.
- TTSMULT is the multiplier for successive transport steps within a flow time-step if the GCG solver is used *and* the solution option for the advection term is the standard finite-difference method. A value between 1.0 and 2.0 is generally adequate. If the GCG package is not used, the transport solution is solved explicitly as in the original MT3D code, and TTSMULT is always set to 1.0 regardless of the user-specified input. Note that for the particle-tracking-based solution options and the third-order TVD scheme, TTSMULT does not apply.
- TTSMAX is the maximum transport step size allowed when transport step size multiplier TTSMULT > 1.0. Setting TTSMAX=0 imposes no maximum limit.

Input Instructions for the Advection Package

Input to the Advection Package is read on unit INADV = 2, which is preset in the main program. The input file is needed only if the Advection Package is used; however, this package is needed under almost all circumstances. Note that underlined are new features introduced in the current version.

For each simulation:

B1 Record: MIXELM, PERCEL, MXPART, NADVFD
Format: I10, F10.0, 2I10

- MIXELM is an integer flag for the advection solution option.
  MIXELM = 0, the standard finite-difference method with upstream or central-in-space weighting, depending on the value of NADVFD;
  = 1, the forward-tracking method of characteristics (MOC);
  = 2, the backward-tracking modified method of characteristics (MMOC);
  = 3, the hybrid method of characteristics (HMOC) with MOC or MMOC automatically and dynamically selected;
  = -1, the third-order TVD scheme (ULTIMATE).
- PERCEL is the Courant number (i.e., the number of cells, or a fraction of a cell) advection will be allowed in any direction in one transport step.

  For implicit finite-difference or particle-tracking-based schemes, there is no limit on PERCEL, but for accuracy reasons, it is generally not set much greater than one. Note, however, that the PERCEL limit is checked over the entire model grid. Thus, even if PERCEL > 1, advection may not be more than one cell's length at most model locations.

  For the explicit finite-difference or the third-order TVD scheme, PERCEL is also a stability constraint which must not exceed one and will be automatically reset to one if a value greater than one is specified.
- MXPART is the maximum total number of moving particles allowed and is used only when MIXELM = 1 or 3.
- NADVFD is an integer flag indicating which weighting scheme should be used; it is needed only when the advection term is solved using the implicit finite-difference method.
  NADVFD = 0 or 1, upstream weighting (default);
  = 2,central-in-space weighting.

(Enter B2 if MIXELM = 1, 2, or 3)

B2 Record: ITRACK, WD
Format: I10, F10.0

- ITRACK is a flag indicating which particle-tracking algorithm is selected for the Eulerian-Lagrangian methods.

ITRACK = 1, the first-order Euler algorithm is used.
= 2, the fourth-order Runge-Kutta algorithm is used; this option is computationally demanding and may be needed only when PERCEL is set greater than one.
= 3, the hybrid first- and fourth-order algorithm is used; the Runge-Kutta algorithm is used in sink/source cells and the cells next to sinks/sources while the Euler algorithm is used elsewhere.

- WD is a concentration weighting factor between 0.5 and 1. It is used for operator splitting in the particle-tracking-based methods. The value of 0.5 is generally adequate. The value of WD may be adjusted to achieve better mass balance. Generally, it can be increased toward 1.0 as advection becomes more dominant.

(Enter B3 if MIXELM = 1 or 3)

B3 Record: DCEPS, NPLANE, NPL, NPH, NPMIN, NPMAX
Format: F10.0, 5I10

- DCEPS is a small Relative Cell Concentration Gradient below which advective transport is considered negligible. A value around 10^{-5} is generally adequate.
- NPLANE is a flag indicating whether the random or fixed pattern is selected for initial placement of moving particles.
  If NPLANE = 0, the random pattern is selected for initial placement. Particles are distributed randomly in both the horizontal and vertical directions by calling a random number generator (Figure 18b). This option is usually preferred and leads to smaller mass balance discrepancy in nonuniform or diverging/converging flow fields.

  If NPLANE > 0, the fixed pattern is selected for initial placement. The value of NPLANE serves as the number of vertical "planes" on which initial particles are placed within each cell block (Figure 18a). The fixed pattern may work better than the random pattern only in relatively uniform flow fields. For two-dimensional simulations in plan view, set NPLANE = 1. For cross sectional or three-dimensional simulations, NPLANE = 2 is normally adequate. Increase NPLANE if more resolution in the vertical direction is desired.
- NPL is the number of initial particles per cell to be placed at cells where the Relative Cell Concentration Gradient is less than or equal to DCEPS. Generally, NPL can be set to zero since advection is considered insignificant when the Relative Cell Concentration Gradient is less than or equal to DCEPS. Setting NPL equal to NPH causes a uniform number of particles to be placed in every cell over the entire grid (i.e., the uniform approach).

- NPH is the number of initial particles per cell to be placed at cells where the Relative Cell Concentration Gradient is greater than DCEPS. The selection of NPH depends on the nature of the flow field and also the computer memory limitation. Generally, a smaller number should be used in relatively uniform flow fields and a larger number should be used in relatively nonuniform flow fields. However, values exceeding 16 in two-dimensional simulation or 32 in three-dimensional simulation are rarely necessary. If the random pattern is chosen, NPH particles are randomly distributed within the cell block. If the fixed pattern is chosen, NPH is divided by NPLANE to yield the number of particles to be placed per vertical plane, which is rounded to one of the values shown in Figure 30.
- NPMIN is the minimum number of particles allowed per cell. If the number of particles in a cell at the end of a transport step is fewer than NPMIN, new particles are inserted into that cell to maintain a sufficient number of particles. NPMIN can be set to zero in relatively uniform flow fields and to a number greater than zero in diverging/converging flow fields. Generally, a value between zero and four is adequate.
- NPMAX is the maximum number of particles allowed per cell. If the number of particles in a cell exceeds NPMAX, all particles are removed from that cell and replaced by a new set of particles equal to NPH to maintain mass balance. Generally, NPMAX can be set to approximately two times of NPH.

(Enter B4 if MIXELM = 2 or 3)

B4 Record: INTERP, NLSINK, NPSINK
Format: 3I10

- INTERP is a flag indicating the concentration interpolation method for use in the MMOC scheme. Currently, only linear interpolation is implemented. Enter INTERP = 1.
- NLSINK is a flag indicating whether the random or fixed pattern is selected for initial placement of particles to approximate sink cells in the MMOC scheme. The convention is the same as that for NPLANE. It is generally adequate to set NLSINK equivalent to NPLANE.
- NPSINK is the number of particles used to approximate sink cells in the MMOC scheme. The convention is the same as that for NPH. It is generally adequate to set NPSINK equivalent to NPH.

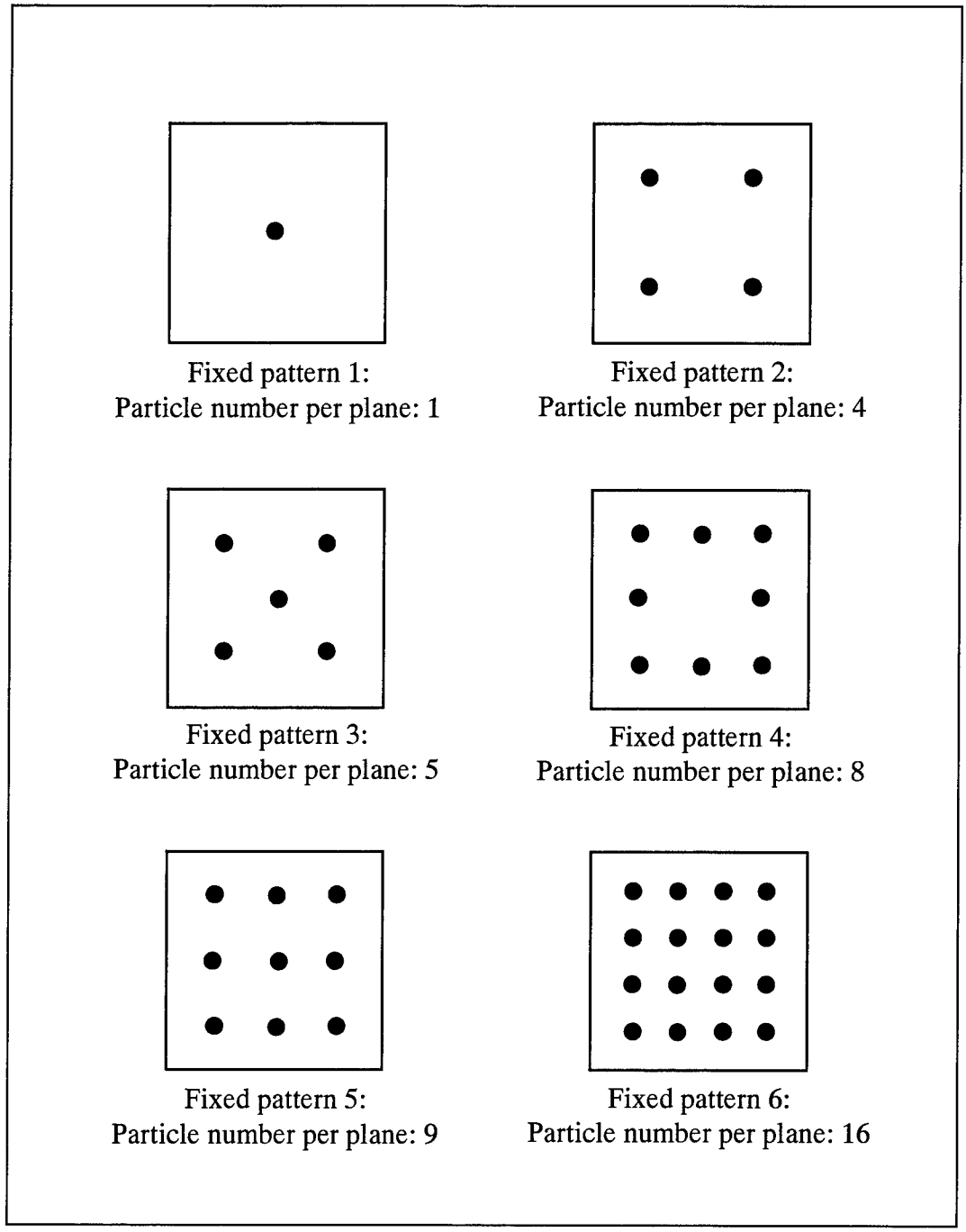


Figure 30. Distribution of initial particles using the fixed pattern (if the fixed pattern is chosen, the number of particles placed per cell (NPL or NPH) is divided by the number of vertical "planes," or NPLANE, to yield the number of particles to be placed on each vertical plane, which is then rounded to one of the values shown here)

(Enter B5 if MIXELM = 3)

B5 Record: DCHMOC
Format: F10.0

- DCHMOC is the critical Relative Concentration Gradient for controlling the selective use of either MOC or MMOC in the HMOC solution scheme.

  The MOC solution is selected at cells where the Relative Concentration Gradient is greater than DCHMOC.

  The MMOC solution is selected at cells where the Relative Concentration Gradient is less than or equal to DCHMOC.

Input Instructions for the Dispersion Package

Input to the Dispersion Package is read on unit INDSP = 3, which is preset in the main program. The input file is needed only if the Dispersion Package is used in the simulation.

FOR EACH SIMULATION:

C1 Array: AL(NCOL,NROW) (one array for each layer).
Reader: RARRAY

- AL is the longitudinal dispersivity, α_L, for every cell of the model grid (unit, L).

C2 Array: TRPT(NLAY)
Reader: RARRAY

- TRPT is a 1D real array defining the ratio of the horizontal transverse dispersivity, α_{TH}, to the longitudinal dispersivity, α_L. Each value in the array corresponds to one model layer. Some recent field studies suggest that TRPT is generally not greater than 0.1.

C3 Array: TRPV(NLAY)
Reader: RARRAY

- TRPV is the ratio of the vertical transverse dispersivity, α_{TV}, to the longitudinal dispersivity, α_L. Each value in the array corresponds to one model layer. Some recent field studies suggest that TRPT is generally not greater than 0.01.

Set TRPV equal to TRPT to use the standard isotropic dispersion model (Equation 10 in Chapter 2). Otherwise, the modified isotropic dispersion model is used (Equation 11 in Chapter 2).

C4 Array: DMCOEF(NLAY)
Reader: RARRAY

- DMCOEF is the effective molecular diffusion coefficient (unit, L^2T^{-1}). Set DMCOEF = 0 if the effect of molecular diffusion is considered unimportant. Each value in the array corresponds to one model layer.

Input Instructions for the Sink & Source Mixing Package

Input to the Sink & Source Mixing Package is read on unit INSSM = 4, which is preset in the main program. The input file is needed if any sink or source option is used in the flow model, including the constant-head or general-head-dependent boundary conditions. The classification of the sink/source types used in MT3DMS is the same as that used by MODFLOW (McDonald and Harbaugh 1988). Note that underlined are new features introduced in the current version.

For each simulation:

D1 Record: FWEL, FDRN, FRCH, FEVT, FRIV, FGHB, (FNEW(n), n=1,4)
Format: 10L2

- FWEL is a logical flag for the Well option.
- FDRN is a logical flag for the Drain option.
- FRCH is a logical flag for the Recharge option.
- FEVT is a logical flag for the Evapotranspiration
- option.
- FRIV is a logical flag for the River option.
  FGHB is a logical flag for General-Head-Dependent
- Boundary option.
  FNEW are logical flags reserved for additional sink/source options. If any of these options is used in the flow model, its respective flag must be set to T (True), otherwise, set to F (False).

  Note that when MODFLOW is used to obtain flow solutions for MT3DMS, Version 2 and later of the LKMT package for MODFLOW will store appropriate values for these flags in the unformatted flow-transport link file. If these flags are not specified correctly here, MT3DMS will issue a warning, reset the flags to correct values, and proceed with the simulation.

Also note that a commonly used add-on package to MODFLOW, the Streamflow-Routing (STR) package is supported through the River option. This is done by associating the River option in MT3DMS with the STR package instead of the RIV package in MODFLOW. For this reason, the RIV and STR packages cannot be used concurrently in the same MODFLOW simulation.

D2 Record: MXSS
Format: I10

- MXSS is the maximum number of all point sinks and sources included in the flow model. Point sinks and sources include constant-head cells, wells, drains, rivers, and general-head-dependent boundary cells. Recharge and evapotranspiration are treated as areally distributed sinks and sources; thus, they should not be counted as point sinks and sources. MXSS should be set close to the actual number of total point sinks and sources in the flow model to minimize the computer memory allocated to store sinks and sources.

For each stress period:

(Enter D3 if FRCH = T)
D3 Record: INCRCH
Format: I10

- INCRCH is a flag indicating whether an array containing the concentration of recharge flux for each species will be read for the current stress period.
  If INCRCH $\geq$ 0, an array containing the concentration of recharge flux for each species will be read.
  If INCRCH $<$ 0, the concentration of recharge flux will be reused from the last stress period. If INCRCH $<$ 0 is specified for the first stress period, then by default, the concentration of positive recharge flux (source) is set equal to zero and that of negative recharge flux (sink) is set equal to the aquifer concentration.

(Enter D4 for each species if FRCH = T and INCRCH $\geq$ 0)
D4 Array: CRCH(NCOL,NROW)
Reader: RARRAY

- CRCH is the concentration of recharge flux for a particular species. If the recharge flux is positive, it acts as a source whose concentration can be specified as desired. If the recharge flux is negative, it acts as a sink (discharge) whose concentration is always set equal to the concentration of groundwater at the cell where discharge occurs. Note that the location and flow rate of recharge/discharge are obtained from the flow model directly through the unformatted flow-transport link file.

(Enter D5 if FEVT = T)

D5 Record: INCEVT
Format: I10

- INCEVT is a flag indicating whether an array containing the concentration of evapotranspiration flux for each species will be read for the current stress period.

  If INCEVT $\geq$ 0, an array containing the concentration of evapotranspiration flux for each species will be read. If INCEVT< 0, the concentration of evapotranspiration flux for each species will be reused from the last stress period. If INCEVT < 0 is specified for the first stress period, then by default, the concentration of negative evapotranspiration flux (sink) is set to the aquifer concentration, while the concentration of positive evapotranspiration flux (source) is set to zero.

(Enter D6 for each species if FEVT = T and INCEVT$\geq$0)

D6 Array: CEVT(NCOL,NROW)
Reader: RARRAY

- CEVT is the concentration of evapotranspiration flux for a particular species. Evapotranspiration is the only type of sink whose concentration may be specified externally. Note that the concentration of a sink cannot be greater than that of the aquifer at the sink cell. Thus, if the sink concentration is specified greater than that of the aquifer, it is automatically set equal to the concentration of the aquifer. Also note that the location and flow rate of evapotranspiration are obtained from the flow model directly through the unformatted flow-transport link file.

D7 Record: NSS
Format: I10

- NSS is the number of point sources whose concentrations need to be specified. By default, unspecified point sources are assumed to have zero concentration. (The concentration of point sinks is always set equal to the concentration of groundwater at the sink location.)

Note that in MT3DMS, point sources are generalized to include not only those associated with a flow rate in the flow model, but also those independent of the flow solution. This type of "mass-loading" sources may be used to include contaminant sources which have minimal effects on the hydraulics of the flow field.

(Enter D8 NSS times if NSS > 0)

D8 Record: KSS, ISS, JSS, CSS, ITYPE, (CSSMS(n), n=1, NCOMP)

Format: 3I10, F10.0, I10, [free]

- KSS, ISS, JSS are the cell indices (layer, row, column) of the point source for which a concentration needs to be specified for each species.
- CSS is the specified source concentration or mass-loading rate, depending on the value of ITYPE, in a single-species simulation,. (For a multispecies simulation, CSS is not used, but a dummy value still needs to be entered here.)

  Note that for most types of sources, CSS is interpreted as the source concentration with the unit of mass per unit volume (ML^{-3}), which, when multiplied by its corresponding flow rate (L^3T^{-1}) from the flow model, yields the mass-loading rate (MT^{-1}) of the source.

  For a special type of sources (ITYPE = 15), CSS is taken directly as the mass-loading rate (MT^{-1}) of the source so that no flow rate is required from the flow model.

  Furthermore, if the source is specified as a constant-concentration cell (ITYPE = -1), the specified value of CSS is assigned directly as the concentration of the designated cell. If the designated cell is also associated with a sink/source term in the flow model, the flow rate is not used.
- ITYPE is an integer indicating the type of the point source as listed below:

  ITYPE = 1, constant-head cell;
  = 2, well;
  = 3, drain (note that in MODFLOW conventions, a drain is always a sink, thus, the concentration for drains cannot be specified if the flow solution is from MODFLOW);
  = 4, river (or stream);
  = 5, general-head-dependent boundary cell;
  = 15, mass-loading source;
  = -1, constant-concentration cell.

- (CSSMS(n), n=1, NCOMP) defines the concentrations of a point source for multispecies simulation with NCOMP>1. In a multispecies simulation, it is necessary to define the concentrations of *all species* associated with a point source. As an example, if a chemical of a certain species is injected into a multispecies system, the concentration of that species is assigned a value greater than zero while the concentrations of all other species are assigned zero. CSSMS(n) can be entered in free format, separated by a comma or space between values. Several important notes on assigning concentration for the constant-concentration condition (ITYPE = -1) are listed below:

  The constant-concentration condition defined in this input file takes precedence to that defined in the Basic Transport Package input file.

  In a multiple stress period simulation, a constant-concentration cell, once defined, will remain a constant-concentration cell in the duration of the simulation, but its concentration value can be specified to vary in different stress periods.

  In a multispecies simulation, if it is only necessary to define different constant-concentration conditions for selected species at the same cell location, specify the desired concentrations for those species, and assign a negative value for all other species. The negative value is a flag used by MT3DMS to skip assigning the constant-concentration condition for the designated species.

Input Instructions for the Chemical Reaction Package

Input to the Chemical Reaction Package is read on unit INRCT = 8, which is preset in the main program. The input file is needed only if chemical reactions are simulated. In addition, the option for modeling transport in a dual-domain system is specified through this file. Note that new features introduced in the current version are underlined.

For each simulation:

E1 Record: ISOTHM, IREACT, IRCTOP, IGETSC
Format: 4I10

- ISOTHM is a flag indicating which type of sorption (or dual-domain mass transfer) is simulated:
  ISOTHM = 0, no sorption is simulated;
  =1, Linear isotherm (equilibrium-controlled);
  =2, Freundlich isotherm (equilibrium-controlled);
  =3, Langmuir isotherm (equilibrium-controlled);
  =4, First-order kinetic sorption (nonequilibrium);
  =5, Dual-domain mass transfer (without sorption);
  =6, Dual-domain mass transfer (with sorption).
- IREACT is a flag indicating which type of kinetic rate reaction is simulated:
  IREACT = 0, no kinetic rate reaction is simulated;
  IREACT = 1, first-order irreversible reaction.
  Note that this reaction package is not intended for modeling chemical reactions between species. An add-on reaction package developed specifically for that purpose may be used.
- IRCTOP is an integer flag indicating how reaction variables are entered:
  IRCTOP $\geq$ 2, all reaction variables are specified as 3-D arrays on a cell-by-cell basis.
  IRCTOP < 2, all reaction variables are specified as a 1-D array with each value in the array corresponding to a single layer. This option is mainly for retaining compatibility with the previous versions of MT3D.
- IGETSC is an integer flag indicating whether the initial concentration for the nonequilibrium sorbed or immobile phase of all species should be read when nonequilibrium sorption (ISOTHM = 4) or dual-domain mass transfer (ISOTHM = 5 or 6) is simulated:
  IGETSC = 0, the initial concentration for the sorbed or immobile phase is not read. By default, the sorbed phase is assumed to be in equilibrium with the dissolved phase (ISOTHM = 4), and the immobile domain is assumed to have zero concentration (ISOTHM = 5 or 6).
  IGETSC > 0, the initial concentration for the sorbed phase or immobile liquid phase of all species will be read.

(Enter E2A if ISOTHM=1, 2, 3, 4, or 6; but not 5)

E2A Array: RHOB(NCOL,NROW) (one array for each layer)
Reader: RARRAY

- RHOB is the bulk density of the aquifer medium (unit, ML^{-3}).

(Enter E2B if ISOTHM = 5 or 6)

E2B Array: PRSITY2(NCOL,NROW) (one array for each layer)
Reader: RARRAY

- PRSITY2 is the porosity of the immobile domain, i.e., the ratio of pore spaces filled with immobile fluids over the bulk volume of the aquifer medium, when the simulation is intended to represent a dual-domain system.

(Enter E2C for each species if IGETSC > 0)

E2C Array: SRCONC(NCOL,NROW) (one array for each layer)
Reader: RARRAY

- SRCONC is the user-specified initial concentration for the sorbed phase of a particular species if ISOTHM = 4 (unit, MM^{-1}). Note that for equilibrium-controlled sorption, the initial concentration for the sorbed phase cannot be specified.
  SRCONC is the user-specified initial concentration for the immobile liquid phase if ISOTHM = 5 or 6 (unit, ML^{-3}).

(Enter E3 for each species if ISOTHM > 0)

E3 Array: SP1(NCOL,NROW) (one array for each layer)
Reader: RARRAY

- SP1 is the first sorption parameter. The use of SP1 depends on the type of sorption selected (i.e., the value of ISOTHM):

  For linear sorption (ISOTHM = 1) and nonequilibrium sorption (ISOTHM = 4), SP1 is the distribution coefficient (K_d) (unit, L^3M^{-1}).

  For Freundlich sorption (ISOTHM = 2), SP1 is the Freundlich equilibrium constant (K_f) (the unit depends on the Freundlich exponent a).

  For Langmuir sorption (ISOTHM = 3), SP1 is the Langmuir equilibrium constant (K_l) (unit, L^3M^{-1}).
  For dual-domain mass transfer without sorption (ISOTHM = 5), SP1 is not used, but still must be entered.

  For dual-domain mass transfer with sorption (ISOTHM = 6), SP1 is also the distribution coefficient (K_d) (unit, L^3M^{-1}).

(Enter E4 for each species if ISOTHM > 0)

E4 Array: SP2(NCOL,NROW) (one array for each layer)
Reader: RARRAY

- SP2 is the second sorption or dual-domain model parameter. The use of SP2 depends on the type of sorption or dual-domain model selected:

  For linear sorption (ISOTHM = 1), SP2 is read but not used.

  For Freundlich sorption (ISOTHM = 2), SP2 is the Freundlich exponent *a*.

  For Langmuir sorption (ISOTHM = 3), SP2 is the total concentration of the sorption sites available ($\overline{S}$) (unit, MM^{-1}).

  For nonequilibrium sorption (ISOTHM = 4), SP2 is the first-order mass transfer rate between the dissolved and sorbed phases (unit, T^{-1}).

  For dual-domain mass transfer (ISOTHM = 5 or 6), SP2 is the first-order mass transfer rate between the two domains (unit, T^{-1}).

(Enter E5 for each species if IREACT > 0)

E5 Array: RC1(NCOL, NROW) (one array for each layer)
Reader: RARRAY

- RC1 is the first-order reaction rate for the dissolved (liquid) phase (unit, T^{-1}). If a dual-domain system is simulated, the reaction rates for the liquid phase in the mobile and immobile domains are assumed to be equal.

(Enter E6 for each species if IREACT > 0)

E6 Array: RC2(NCOL, NROW) (one array for each layer)
Reader: RARRAY

- RC2 is the first-order reaction rate for the sorbed phase (unit, T^{-1}). If a dual-domain system is simulated, the reaction rates for the sorbed phase in the mobile and immobile domains are assumed to be equal. Generally, if the reaction is radioactive decay, RC2 should be set equal to RC1, while for biodegradation, RC2 may be different from RC1.

  Note that RC2 is read but not used, if no sorption is included in the simulation.

Input Instructions for the Generalized Conjugate Gradient Solver Package

Input to the Generalized Conjugate Gradient (GCG) Package is read on unit INGCG = 9, which is preset in the main program. The input file is needed only if the GCG solver is used for implicit solution schemes. This package is a new addition in MT3DMS.

For each simulation:

F1 Record: MXITER, ITER1, ISOLVE, NCRS
Format: Free

- MXITER is the maximum number of outer iterations; it should be set to an integer greater than one only when a nonlinear sorption isotherm is included in simulation.
- ITER1 is the maximum number of inner iterations; a value of 30-50 should be adequate for most problems.
- ISOLVE is the type of preconditioners to be used with the Lanczos/ORTHOMIN acceleration scheme:
  = 1, Jacobi
  = 2, SSOR
  = 3, Modified Incomplete Cholesky (MIC)
  (MIC usually converges faster, but it needs significantly more memory)
- NCRS is an integer flag for treatment of dispersion tensor cross terms:
  = 0, lump all dispersion cross terms to the right-hand-side (approximate but highly efficient).
  = 1, include full dispersion tensor (memory intensive).

F2 Record: ACCL, CCLOSE, IPRGCG
Format: Free

- ACCL is the relaxation factor for the SSOR option; a value of 1.0 is generally adequate.
- CCLOSE is the convergence criterion in terms of relative concentration; a real value between 10^{-4} and 10^{-6} is
- generally adequate.
  IPRGCG is the interval for printing the maximum concentration changes of each iteration. Set IPRGCG to zero as default for printing at the end of each stress period.

Start of a Simulation Run

There are two ways to start a simulation run. The first method is simply to type the name of the executable file. The program will prompt the user for the names of various input and output files. An example is given below, where "C:\>" is the command prompt and "MT3DMS" is the name of the executable file of the MT3DMS program:

```
C:\>MT3DMS


+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                                                                     +
+                                MT3DMS                               +
+             A Modular Three-Dimensional Transport Model             +
+ For Simulation of Advection, Dispersion and Chemical Reactions     +
+                of Contaminants in Groundwater Systems               +
+                                                                     +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Enter Name for Standard Output File:                 test1.m3d
Enter Name for Basic Transport Input File:           test1.btn
Enter Name for Advection Input File:                 test1.adv
Enter Name for Dispersion Input File:                test1.dsp
Enter Name for Sink & Source Input File:             test1.ssm
Enter Name for Chemical Reaction Input File:         test1.rct
Enter Name for GCG Solver Input File:                test1.gcg
Enter Name for Unformatted Flow-Transport Link File: test1.umt
Print out Heads and Flow Terms for Checking (Y/N)?   N


STRESS PERIOD NO.    1

TIME STEP NO.    1
FROM TIME =  0.00000     TO    100.00
Transport Step: 10   Step Size: 10.000
Total Elapsed Time:  100.000

Program Completed.
```

The second method is to create a response file which contains the names of input and output files in the order required by the program. The content of such a response file (RUN.FIL) for the example shown above would be as follows:

```
test1.m3d
test1.btn
test1.adv
test1.dsp
test1.ssm
test1.rct
test1.gcg
test1.umt
N
```

Then, at the command prompt, type:

```
C:\>MT3DMS < RUN.FIL
```

Continuation of a Previous Simulation Run

Sometimes it may be necessary to break a long simulation run into several shorter ones. For example, if the flow model has many stress periods and time-steps, the unformatted head and flow file generated by the flow model can be quite large. If there is not enough disk space, one has to break the simulation time into several intervals, with each interval having a separate head and flow file. Then, one needs to run the transport model once for each interval of the flow solution. The continuation of a previous simulation run in the MT3DMS transport model is similar to the continuation of a flow simulation in the MODFLOW flow model.

First, save the concentrations from the final step of the preceding run on the default unformatted file MT3Dnnn.UCN. Next, rename the file and use it as the starting concentration file for the next run. If there is more than one step of concentrations saved in the MT3Dnnn.UCN file where nnn is the species index, extract the concentrations at the last step using a program named SAVELAST included with the MT3DMS distribution files (see Appendix D). The array reader which reads the starting concentrations (RARRAY) is capable of reading a model-generated unformatted concentration file with no modification. Because mass budget terms are always set to zero at the start of a simulation run, the printed budget on a simulation run represents only that single run. Therefore, if a total budget for a series of continuation runs is desired, the totals from each run can be added externally. Similarly, the model program keeps track of simulation time only for single simulation runs; total simulation time for a series of continuation runs must be calculated externally by adding the simulation times of each run.

7 Benchmark Problems and Application Examples

This chapter describes several benchmark problems used to test the accuracy and performance of the MT3DMS code. Either analytical solutions or numerical solutions by another code are available for these benchmark problems. In addition, a number of application examples are presented to illustrate how the transport code may be applied to solving more complicated field problems. All the input files required to run these test problems are included with the MT3DMS distribution files. It is recommended that the users try these test problems first to become familiarized with the various simulation options and input/output structures of the MT3DMS code before applying it to solve their own problems.

One-Dimensional Transport in a Uniform Flow Field

A relatively complete set of one-dimensional analytical solutions for solute transport involving advection, dispersion, and some simple chemical reactions in a steady-state uniform flow field is available in Van Genuchten and Alves (1982). The problem considered in this section involves the following initial and boundary conditions:

$$\begin{aligned} &C(x,0)=0 \\ &C(0,t)=C_o \qquad t>0 \\ &\frac{\partial C}{\partial x}(\infty,t)=0 \qquad t>0 \end{aligned} \tag{135}$$

A numerical model consisting of 101 columns, 1 row, and 1 layer is used to solve the problem for comparison with the analytical solution for the same initial and boundary conditions as presented in Van Genuchten and Alves (1982). The model parameters used in the simulation are listed below:

Cell width along rows (Δx) = 10 m
Cell width along columns (Δy) = 1 m
Layer thickness (Δz) = 1 m

Groundwater seepage velocity (v) = 0.24 m/day
Porosity (θ) = 0.25
Simulation time (t) = 2,000 days

In the flow model, the first and last columns are constant-head boundaries. Arbitrary head values are used to establish the required uniform hydraulic gradient. In the transport model, the first column is a constant-concentration boundary with a relative concentration of one. The last column is set sufficiently far away from the source to approximate an infinite one-dimensional flow domain as assumed in the analytical solution. Four simulations are run using different values of dispersivity, α_L, retardation factor, R, and decay rate constant, λ, as follows:

Case 1a:	$\alpha = 0, \quad R = 0, \lambda = 0$	Advection only
Case 1b:	$\alpha = 10\,\text{m}, R = 0, \lambda = 0$	Advection and dispersion
Case 1c:	$\alpha = 10\,\text{m}, R = 5, \lambda = 0$	Advection, dispersion, and sorption
Case 1d:	$\alpha = 10\,\text{m}, R = 5, \lambda = 0.002\,\text{d}^{-1}$	Advection, dispersion, sorption, and decay

Note that in this example the linear sorption isotherm is assumed and that the dimensionless retardation factor is defined for the model by specifying a uniform bulk density of the porous medium and a uniform distribution coefficient of the linear sorption isotherm (Chapter 2).

Cases 1a and 1b are solved using the MOC option with these solution parameters: DCEPS = 10^{-5}, NPLANE = 1, NPL = 0, NPH = 4, NPMIN = 0, and NPMAX = 8. These parameters imply that no particle (i.e., NPL = 0) is placed in cells where the Relative Concentration Gradient (i.e., DCCELL) is equal to or less than 10^{-5}, considered negligible; and four particles (i.e., NPH = 4) are inserted with a fixed pattern in one vertical plane (NPLANE = 1) into cells where DCCELL > 10^{-5}. Since NPMIN = 0, new particles are added only after a cell where DCCELL > 10^{-5} becomes void of any particle. The fixed pattern and one vertical plane are chosen for initial particle placement because of the one-dimensional uniform flow field. Cases 1c and 1d are solved using the MMOC option. The solution parameters needed for the MMOC option, NLSINK and NPSINK, are set equivalent to NPLANE and NPH for the MOC option.

A close match between the analytical (solid lines) and numerical solutions (symbols) is obtained for all cases (Figure 31). Note that no numerical dispersion is introduced even for the pure advection problem (Case 1a). With a dispersivity of 10 m, the concentration distribution for Case 1b is much smoother. The concentration front moves slower in Case 1c than in Case 1b because of the sorption. The transport is the slowest in Case 1d due to the combined effects of sorption and radioactive decay.

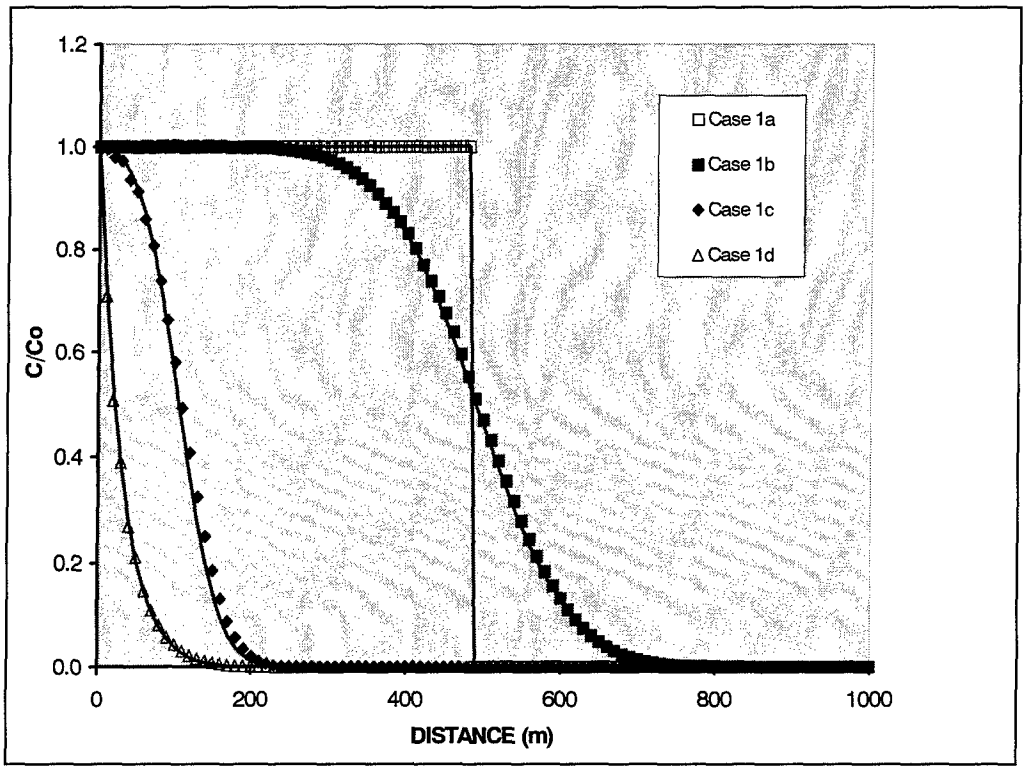
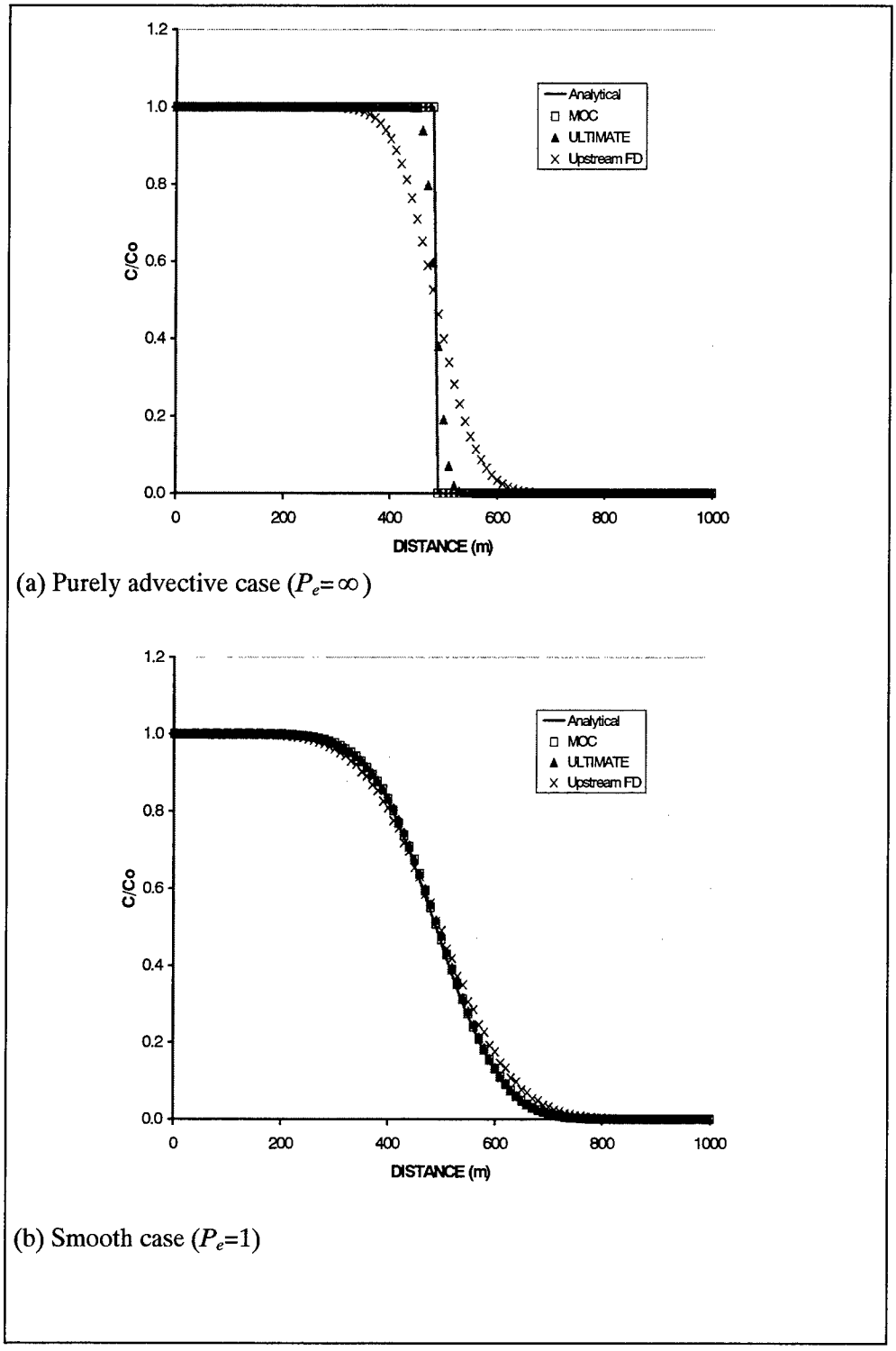


Figure 31. Comparison of the calculated concentrations with the analytical solutions (solid lines) and numerical solutions (symbols) for the one-dimensional test problem

We can examine the effect of different solution schemes on the solution accuracy. In Case 1a, the transport problem is purely advective (i.e., the grid Peclet number is infinity ($P_e = v\Delta x/D_{xx} = \Delta x/\alpha_L = \infty$)). This case represents a severe test for the various transport solution techniques. In Case 1b, the grid Peclet number is one (i.e., $P_e = v\Delta x/D_{xx} = 10/10 = 1$), which indicates that the transport problem is no longer advection dominated and can be solved by most transport techniques without any major difficulty.

Figure 32a shows the comparison of the analytical solution with numerical solutions based on the MOC, the third-order TVD scheme (ULTIMATE), and the standard explicit finite-difference method with upstream weighting for the advection term. The MOC yields a solution identical to the analytical solution. This demonstrates that the MOC scheme is most effective for the type of problems in which advection is dominant. While the ULTIMATE solution is not as close to the analytical solution as the MOC solution, it does lead to dramatic improvement over the upstream finite-difference solution.

Figure 32b shows the same comparison, but the grid Peclet number of the transport problem is equal to one. Because the transport problem is no longer dominated by advection, the standard finite-difference method is reasonably



(a) Purely advective case ($P_e=\infty$)

(b) Smooth case (P_e=1)

Figure 32. Comparison between the analytical solution and three different numerical solutions for (a) purely advective cases and (b) smooth cases

accurate, as shown in the figure. The more computationally intensive MOC and ULTIMATE schemes lead to only marginal gains in the solution accuracy. Thus, the standard finite-difference method (either explicit or implicit) can be used effectively for problems that are not dominated by advection (e.g., $P_e \leq 4$) even though it is less accurate for strongly advection-dominated problems as demonstrated in Figure 32a.

One-Dimensional Transport with Nonlinear or Nonequilibrium Sorption

Nonlinear sorption

Grove and Stollenwerk (1984) present a computer code for modeling one-dimensional advective-dispersive transport with nonlinear equilibrium-controlled sorption and ion exchange. In this section, the MT3DMS code is used to solve the same test problems as described in Grove and Stollenwerk (1984) involving the Freundlich and Langmuir isotherms. The initial and boundary conditions of the transport model are

$$
\begin{aligned}
& C(x,0) = 0 \\
& -\theta D \frac{\partial C}{\partial x} + qC\Big|_{x=0} = \begin{cases} f_o, & 0 < t < t_o \\ 0, & t > t_o \end{cases} \\
& \frac{\partial C}{\partial x}(\infty,t) = 0 \qquad\qquad t > 0
\end{aligned}
\tag{136}
$$

where the concentration of source fluid, $C(x = o, t)$, decreases to zero after a specified time period, t_o. The model parameters used in the simulation are identical to those of Grove and Stollenwerk (1984) and are listed below:

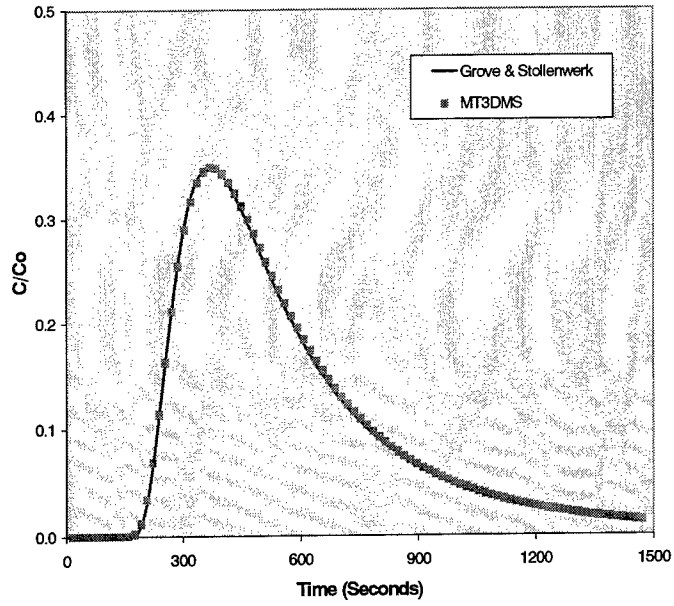
Grid spacing (Δx) = 0.16 *cm*
Dispersivity (α_L) = 1 *cm*
Groundwater seepage velocity (v) = 0.1 *cm/s*
Porosity $(\theta) = 0.37$
Bulk density $(\rho_b) = 1.587\ g/cm^3$
Freundlich equilibrium constant (K_f) = 0.3 $(\mu g/g)(\ell/mg)^a$
Freundlich sorption exponent (a) = 0.7
Langmuir equilibrium constant (K_l) = 100 ℓ/mg
Langmuir sorption capacity $(\bar{S})$ = 0.003 $\mu g/g$
Concentration of source fluid C_o = 0.05 mg/ℓ
Duration of source pulse (t_o) = 160 seconds

Since the flow field is steady state, only one stress period is required for the flow model. However, to accommodate the change in source concentration, two stress periods are used in the transport model. The boundary conditions for the flow model are constant-head at the two ends of the model domain with the values of hydraulic head set arbitrarily to establish the desired Darcy flux. The boundary conditions for the transport model are specified total mass flux (third-type) on the left and zero dispersive mass flux (second-type) on the right. The third-type boundary condition is approximated by specifying the advective mass flux (i.e., $f_o = qC_o$ where q is the rate of inflow or outflow across the boundary). This is accomplished in the test problem by setting the concentration of the inflow at the constant-head node on the left to 0.05 mg/ℓ and zero for the first and second stress periods, respectively. The second-type boundary condition is handled by setting it sufficiently far away from the source so that the plume does not reach it within the specified simulation time.

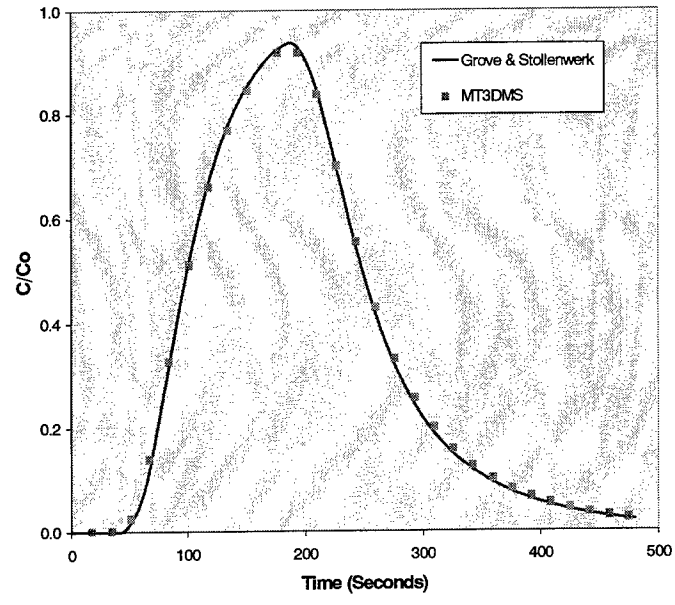
As pointed out in the input instructions, for linear sorption, inconsistent units may be used for sorption constants as long as the resulting retardation factor is dimensionless. However, for nonlinear sorption, the units of sorption constants must be consistent with the unit of concentration. For Freundlich sorption, retardation factor $R = 1 + (\rho_b / \theta) K_f a C^{a-1}$. Thus, the unit for K_f must be $[\rho_b]^{-1}[C]^{1-a}$, or $(\mu g/g)(\ell/mg)^a$ for this test problem. For Langmuir sorption, retardation factor $R = 1 + (\rho_b / \theta)\ [K_l \bar{S} / (1 + K_l C)^2]$. Thus, the unit for K_l must be $[C]^{-1}$, or l/mg in this test problem, while the unit for sorption capacity $\bar{S}$ must be $[K_l]^{-1}[\rho_b]^{-1}$, or μg/g in this test problem. Also note that for this test problem the concentration unit used is mg/ℓ while the length unit is cm. As a result, the budget terms computed by MT3DMS must be properly converted to use their absolute values.

With a small grid Peclet number of 0.16, the test problem is not advection dominated. The solution scheme used is the fully implicit finite-difference method with central-in-space weighting for the advection term. Since nonlinear reaction is involved, the maximum number of outer iterations for the Generalized Conjugate Gradient (GCG) solver is set to be greater than one. The solution convergence criterion is set at 10^{-6} and the modified incomplete Cholesky (MIC) preconditioner is used.

Figure 33 shows the comparison of the breakthrough curves as calculated by MT3DMS with those by Grove and Stollenwerk (1984) at a location 8 cm from the source. The MT3DMS solutions and those of Grove and Stollenwerk agree well for both the Freundlich and Langmuir sorption isotherms.



(a) Freundlich sorption



(b) Langmuir sorption

Figure 33. Comparison between the numerical solutions of MT3DMS and those based on Grove and Stollenwerk (1984) for one-dimensional transport involving nonlinear (a) Freundlich and (b) Langmuir sorption isotherms

Nonequilibrium sorption

A new feature of MT3DMS is the ability to simulate solute transport under either chemical or physical nonequilibrium. In this section, MT3DMS is used to solve one-dimensional transport subject to linear but nonequilibrium sorption. The test problem is identical to that described in the preceding section on nonlinear sorption except for the sorption constants, which are given as distribution coefficient (K_d), 0.933 cm^3/g, and first-order mass transfer coefficient β, ranging from zero to 20 s^{-1}.

The solution schemes used for this test problem are the explicit third-order TVD (ULTIMATE) option for advection and the implicit finite-difference method for all other terms. Since the nonequilibrium sorption is linear, the maximum number of outer iterations for the GCG solver is set to one. As in the nonlinear sorption cases, the solution convergence criterion is set at 10^{-6} and the MIC preconditioner is used.

Figure 34 shows the close agreement between the MT3DMS solutions and the analytical solutions of Weerts (1994) at an observation point 8 cm from the source for a wide range of β values. Note that the mass transfer coefficient α as defined by Weerts differs from β used in this report by a factor of $\rho_b K_d$. When β is zero, the numerical solution is identical to the analytical solution with no sorption. On the other hand, when β is equal to 20 s^{-1}, the nonequilibrium sorption process is sufficiently fast so as to approach the equilibrium condition with the retardation factor $R = 5$.

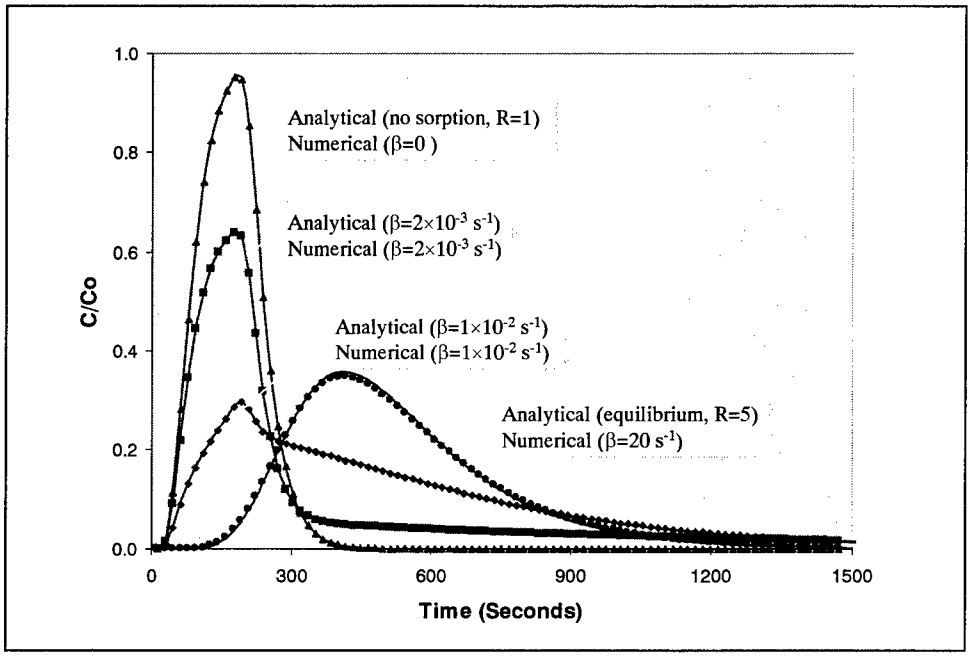


Figure 34. Comparison between the analytical solutions (solid lines) and numerical solutions (symbols) for one-dimensional transport involving nonequilibrium sorption

Two-Dimensional Transport in a Uniform Flow Field

An analytical solution for two-dimensional transport of solute injected continuously from a point source in a steady-state uniform flow field is given by Wilson and Miller (1978). The analytical solution is applicable under the assumption that: (a) the aquifer is infinite in areal extent and relatively thin in vertical extent, so that instantaneous vertical mixing can be assumed, and (b) the injection rate is insignificant compared with the ambient uniform flow.

A numerical model consisting of 46 columns, 31 rows, and 1 layer is constructed for comparison with the analytical solution of Wilson and Miller (1978). The model grid is so oriented that the flow direction is aligned with the model as shown in Figure 35. The model parameters used in the simulation are listed below:

Cell width along rows (Δx) = 10 m
Cell width along rows (Δy) = 10 m
Layer thickness (Δz) = 10 m
Groundwater seepage (v) = 1/3 m/day
Porosity (θ) = 0.3
Longitudinal dispersivity = 10 m
Ratio of transverse to longitudinal dispersivity = 0.3
Volumetric injection rate = 1 m^3/day
Concentration of the injected water = 1,000 ppm
Simulation time (t) = 365 days

The flow model is surrounded by constant-head boundaries on the east and west borders and no-flow boundaries on the north and south borders. The head values at the constant-head boundaries are arbitrarily chosen to establish the required hydraulic gradient. The simulation period is chosen so that the plume developed from the point source does not reach the boundaries.

With a grid Peclet number of one, this transport problem is not advection dominated and thus can be solved with sufficient accuracy by any of the solution schemes available in MT3DMS. The HMOC and ULTIMATE options are selected for comparison purposes. The empirical solution parameters used in the HMOC scheme are DCEPS = 10^{-5}, NPLANE=1, NPL = 0, NPH = 16, NPMIN = 2, NPMAX = 32, and DCHMOC = 10^{-3}. The ULTIMATE option does not require any empirical solution parameters.

The calculated concentrations at the end of the 365-day simulation period are shown in Figure 35. The HMOC and ULTIMATE solutions both agree well with the analytical solution. The mass balance discrepancy for the HMOC solution fluctuates around $\pm$ 5 percent, typical of the particle-tracking-based solution approach. The mass balance discrepancy for the ULTIMATE option is around 10^{-4} percent.

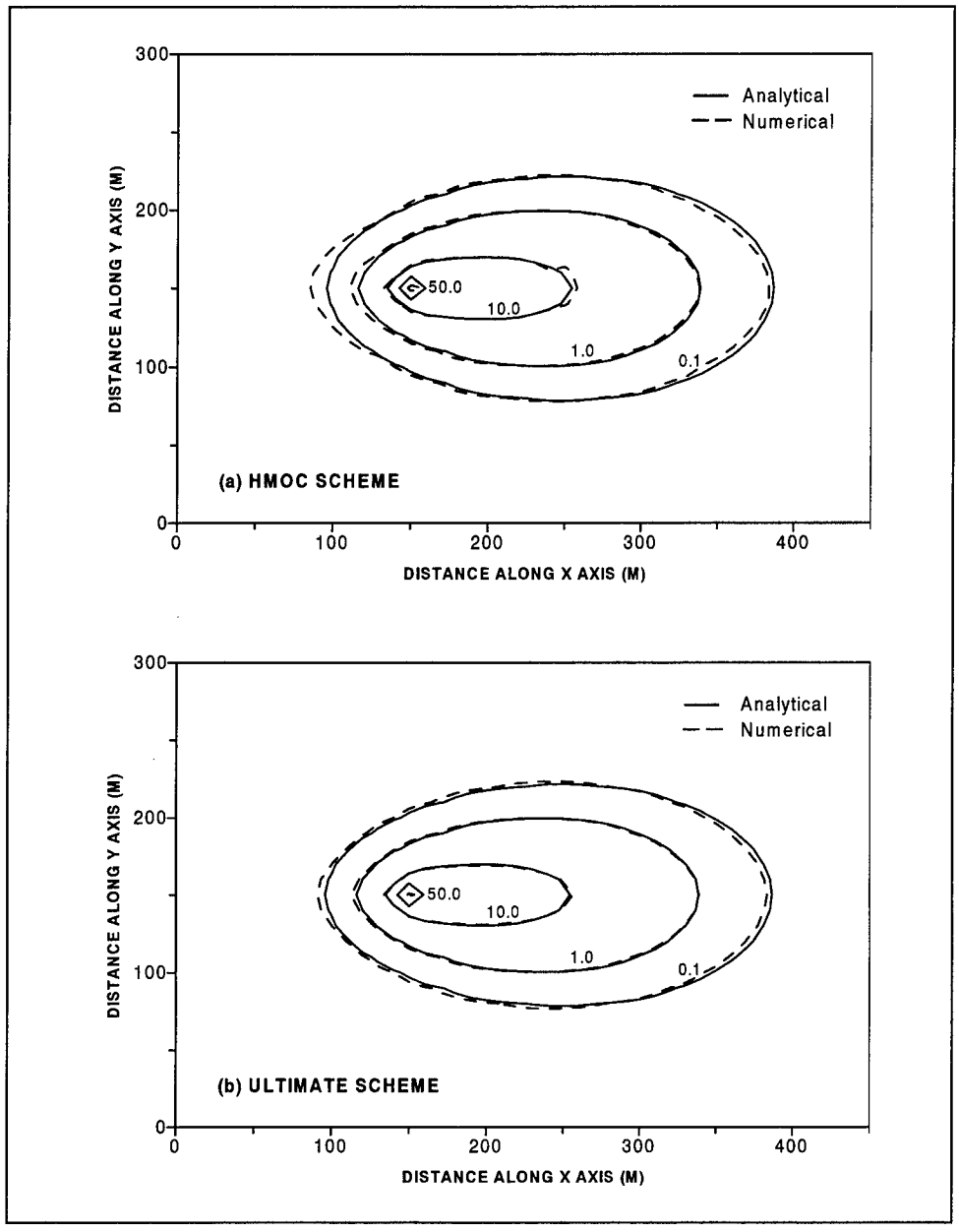


Figure 35. Comparison of the analytical and numerical solutions for two-dimensional transport from a continuous point source with the model grid aligned with the flow direction

Two-Dimensional Transport in a Diagonal Flow Field

The problem considered in this section is similar to that considered in the preceeding section except that the flow direction is oriented at a 45-deg angle to both the columns and rows of the model grid. The grid Peclet number for this

problem is 5 in the longitudinal direction and 50 in the transverse direction. The problem represents a challenging test for transport solution techniques because of the sharpness of the concentration front, compounded by the grid orientation effect.

A numerical model consisting of 100 columns, 100 rows, and 1 layer is constructed for this test problem. The configuration of the test problem is shown in Figure 36, and the model parameters used in the simulation are listed below:

Cell width along rows (Δx) = 10 m
Cell width along rows (Δy) = 10 m
Layer thickness (Δz) = 1 m
Groundwater seepage (v) = 1 m/day
Porosity (θ) = 0.14
Longitudinal dispersivity = 2 m
Ratio of transverse to longitudinal dispersivity = 0.1
Volumetric injection rate = 0.01 m^3/day
Concentration of the injected water = 1,000 ppm
Simulation time (t) = 1,000 days

Figures 36a through 36d show the analytical solution, compared with the numerical solutions based on the MOC scheme, the standard explicit finite-difference method with upstream weighting, and the third-order TVD (ULTIMATE) scheme, respectively. The solution control parameters for the MOC scheme are DCEPS = 10^{-5}, NPLANE = 0, NPL = 0, NPH = 16, NPMIN = 2, and NPMAX = 32. As seen from Figure 36b, the solution based on the MOC scheme is in close agreement with the analytical solution. The solution based on the explicit upstream finite-difference scheme, Figure 36c, yields excessive numerical dispersion. While the ULTIMATE scheme, Figure 36d, does not preserve the concentration peak as well as the MOC scheme, it greatly reduces the numerical dispersion compared with the standard upstream finite-difference scheme. Given the same transport step size, the finite difference method is nearly three times faster than the ULTIMATE scheme, which is in turn more than twice as fast as the MOC scheme. The memory requirements for the ULTIMATE scheme and the upstream finite-difference scheme are identical, while the MOC scheme requires additional memory proportional to the number of particles needed.

Two-Dimensional Transport in a Radial Flow Field

The test problem considered in this section concerns the two-dimensional transport of solute injected from a fully penetrating well. The problem is intended to test the accuracy of MT3DMS as applied to a radial flow system. The assumptions for this problem are:

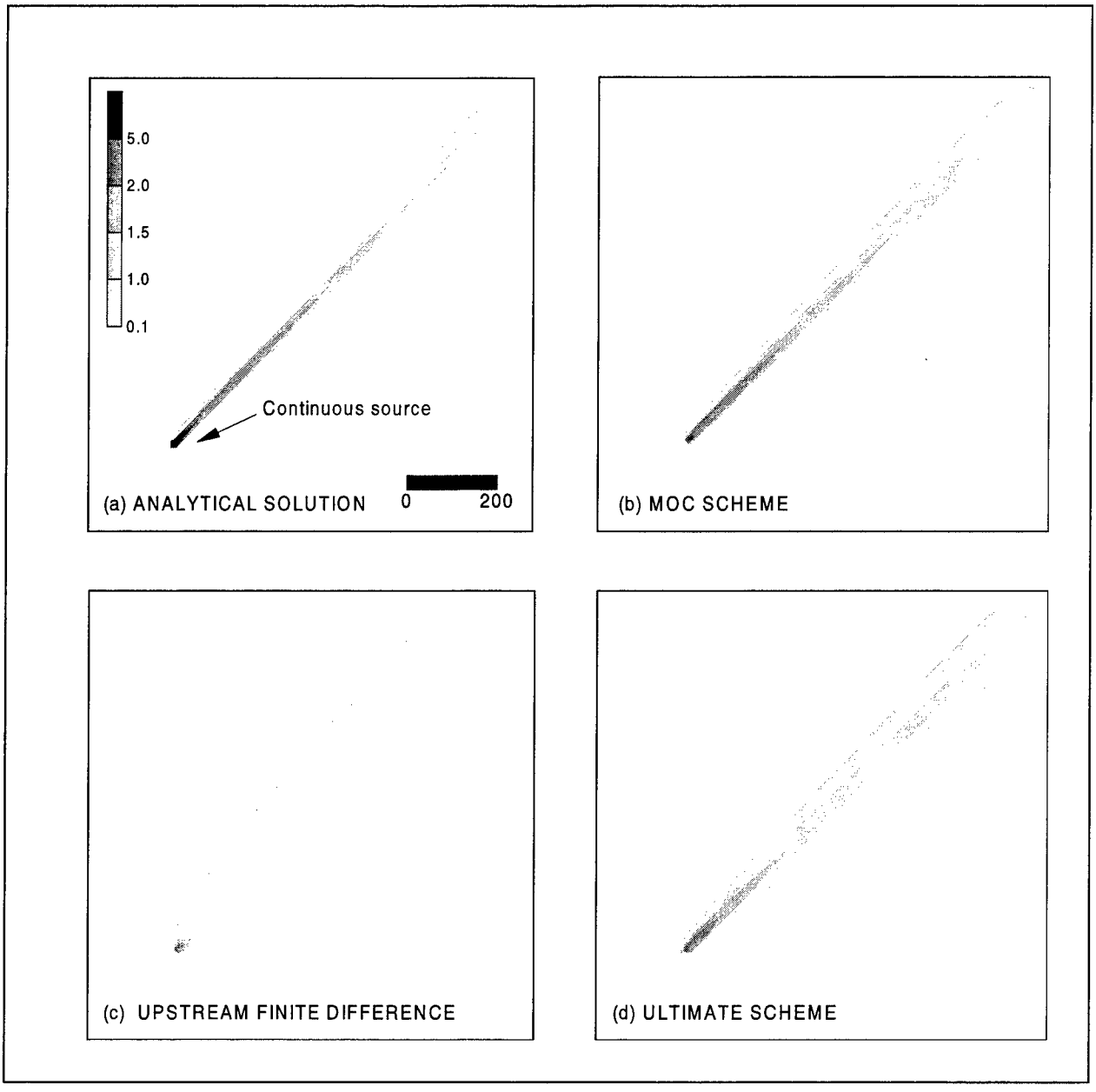


Figure 36. Comparison of the analytical and numerical solutions for two-dimensional transport from a continuous point source with the model grid at a 45-degree diagonal to the flow direction

a. The injection rate of the well is constant.

b. The ambient groundwater velocity is negligible relative to the velocity created by the injection.

c. The aquifer is homogeneous, isotropic, and infinite in areal extent.

d. The flow field is steady state.

The initial and boundary conditions for the transport problem are

$$C(r,0) = 0$$
$$C(r_w,t) = C_0 \qquad t > 0 \tag{137}$$
$$\frac{\partial C}{\partial r}\Big|_{r\to\infty} = 0 \qquad t > 0$$

where r_w is the well radius. An approximate analytical solution for this problem is given by Moench and Ogata (1981) and available in a computer program (LTIRD) provided by Javandel, Doughty, and Jsang (1984).

A numerical model consisting of 31 columns, 31 rows, and 1 layer was used to solve this problem for comparison with the analytical solution of Moench and Ogata (1981). The model parameters used in the simulation are listed below:

Cell width along rows (Δx) = 10 m
Cell width along columns (Δy) = 10 m
Layer thickness (Δz) = 1 m
Injection rate = 100 m^3/day
Porosity (θ) = 0.3
Longitudinal dispersivity = 10 m
Ratio of transverse to longitudinal dispersivity = 1.0
Simulation time (t) = 27 days

The flow model is surrounded by four constant-head boundaries. To approximate the assumption of infinite flow domain used in the analytical transport solution, an analytical flow solution is first used to calculate the steady-state head distribution. The heads from the analytical solution are then used as the specified-head values at the constant-head boundaries of the flow model. The simulation time in the transport model is adjusted to ensure the plume does not reach the constant-head boundaries.

In the transport model, the cell at column 16 and row 16 is simulated as the constant-concentration condition with a relative concentration of one. The values of the grid Peclet number along the x and y directions are spatially variable, depending on the x and y components of the velocity vector. However, a uniform grid Peclet number in the direction of the flow path may be estimated by noting that the flow direction is always 45 deg diagonal to the flow direction. Thus, the characteristic length L *which* is needed for estimating the grid Peclet number, may be set equal to $\sqrt{10^2+10^2} = 14.1$ where 10 is the grid spacing in both the x and y directions, leading to $Pe_L = vL/D_L = vL/(\alpha_L v) = 14.1/10 = 1.4$. In other words, the transport is not advection dominated, and any of the solution options available in MT3DMS can be used to obtain satisfactory results.

For comparison purposes, the ULTIMATE scheme and the implicit upstream finite-difference method are used to solve the advection term, while the dispersion and sink/source terms are solved with the implicit finite-difference method. The MIC preconditioner is used for the GCG solver with the convergence error criterion set to 10^{-4}.

The calculated concentration breakthrough curves at the injection well are shown in Figure 37. It can be seen that the numerical solution based on the ULTIMATE scheme is in close agreement with the analytical solution. The fully implicit finite-difference solution with upstream weighting agrees reasonably well with the analytical solution in spite of some numerical dispersion. However, as the transport step size used in the fully implicit finite-difference solution is increased by using larger transport step size multipliers, the numerical solution becomes less accurate.

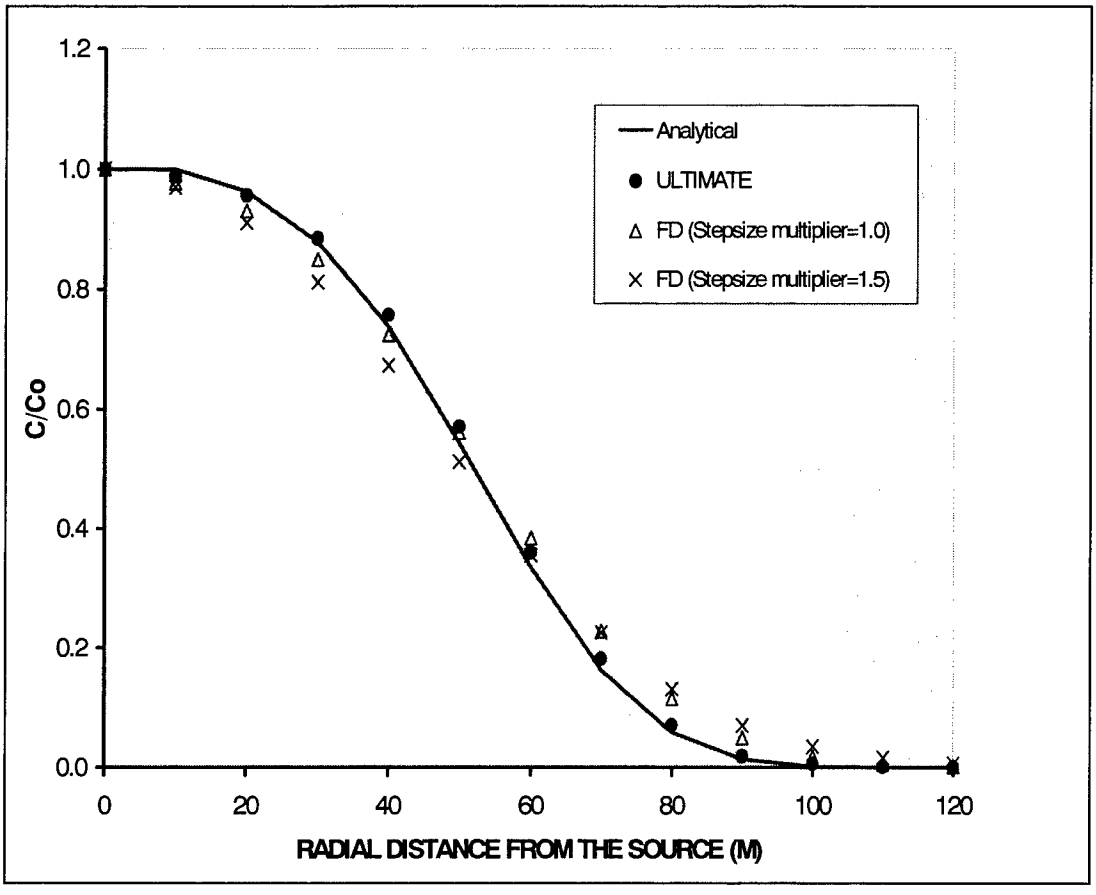


Figure 37. Comparison of the analytical and numerical solutions for two-dimensional transport from a point source in a radial flow field

Concentration at an Injection/Extraction Well

The test problem considered in this section involves the concentration change during an injection/pumping cycle at a fully penetrating well in a confined aquifer. The problem was first proposed by El-Kadi (1988) and then used by Zheng (1993) as a difficult test for the MOC. Water of a constant concentration, C_o, is injected into the well. After a certain period of time, t_1, the flow is reversed, and the contaminated water is pumped out. Assuming that the flow field has an infinite extent and reaches steady state instantaneously after the injection and pumping cycles start, an approximate analytical solution for this problem is given by Gelhar and Collins (1971).

A numerical model consisting of 31 columns, 31 rows, and 1 layer is used to simulate the concentration change at the injection/extraction well for comparison with the approximate analytical solution of Gelhar and Collins (1971). The model setup and input parameters used in the simulation are the same as those used by El-Kadi (1988) and Zheng (1993) and are listed below:

Cell width along rows (Δx) = 900 ft
Cell width along rows (Δy) = 900 ft
Layer thickness (Δz) = 20 ft
Hydraulic conductivity of the aquifer = 0.005 ft/s
Porosity (θ) = 0.3
Longitudinal dispersivity = 100 ft
Ratio of transverse to longitudinal dispersivity = 1.0
Volumetric injection rate = 1 ft^3/s
Relative concentration of the injected water = 100 percent
Length of the injection period = 2.5 years
Length of the extraction period = 7.5 years

The grid Peclet number along the flow direction is approximately 13 for this problem, indicating that it is dominated by advection. The MOC and ULTIMATE schemes are used to solve the advective component of this problem; sinks/sources and dispersion are solved implicitly by the GCG package with the convergence criterion set equal to 10^{-4}. For comparison purposes, the fully implicit finite-difference method with upstream weighting is also used to solve this problem. The solution parameters for the MOC scheme are NPL = 16, NPH = 16, NPMIN = 4, and NPMAX = 32. Because NPL is set equal to NPH, a uniform distribution of 16 particles per cell is initialized over the entire grid. Initial particles are placed randomly within each cell block. Because of the coarse grid and the strongly diverging and converging nature of the flow field, the fourth-order Runge-Kutta particle-tracking algorithm is used to increase the accuracy of the particle-tracking solution.

The concentration breakthrough curves at the injection/extraction well are plotted in Figure 38. Given the same transport step size of 56 days, the MOC scheme achieves the best match with the analytical solution, while the match between the fully implicit upstream finite-difference solution and the analytical solution is the poorest. This is not unexpected considering the relatively

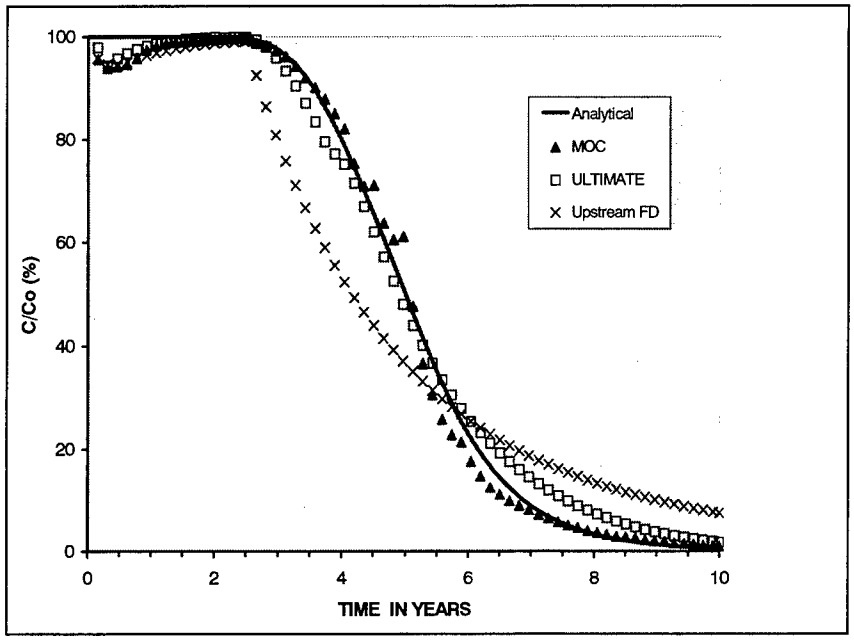


Figure 38. Calculated concentrations at an injection/pumping well as compared with the analytical solution of Gelhar and Collins (1971)

large-grid Peclet number for this problem. While the MOC scheme performs well in terms of matching the concentration at the injection/pumping well, it can lead to significant mass balance discrepancy if the empirical solution parameters are not properly adjusted. On the other hand, the ULTIMATE and the finite-difference solutions are inherently mass conservative.

Three-Dimensional Transport in a Uniform Flow Field

In the ealier section, Two-Dimensional Transport in a Uniform Field, two-dimensional transport from a point source in a uniform flow field was considered under the assumption of a relatively thin aquifer and instantaneous vertical mixing. However, if the aquifer is thick and instantaneous vertical mixing cannot be assumed, the transport of solute away from the point source should be considered three dimensional. An analytical solution for three-dimensional transport with the same set of initial and boundary conditions as discussed earlier is given by Hunt (1978).

A numerical model consisting of 21 columns, 15 rows, and 8 layers is used to solve the three-dimensional transport problem for comparison with the analytical solution of Hunt (1978). The point source is simulated at column 3, row 8, and layer 7. The model parameters used in the simulation are listed below:

Cell width along rows (Δx) = 10 m
Cell width along rows (Δy) = 10 m
Layer thickness (Δz) = 10 m
Hydraulic conductivity of the aquifer = 0.5 m/day
Porosity (θ) = 0.2
Longitudinal dispersivity = 10 m
Ratio of horizontal transverse to longitudinal dispersivity = 0.3
Ratio of vertical transverse to longitudinal dispersivity = 0.3
Volumetric rates of injection = $0.5 m^3$/day
Relative concentration of the injected water = 100 percent
Simulation time = 100 days

Because of the small grid Peclet number for this test problem ($Pe_x = v_x \Delta x / D_{xx} = v_x \Delta x / (\alpha_L v_x) = 10/10 = 1$; $Pe_y = Pe_z = 0$), any of the solution techniques available in MT3DMS should yield satisfactory results. Shown in Figure 39 are the calculated concentrations based on the ULTIMATE scheme for layers 5, 6, and 7 at the end of the 100-day simulation period, which agree well with the analytical solution. It should be noted that the analytical solution assumes the aquifer domain has an infinite extent. Because the point source is located close to the bottom layer which is a no-mass-flux boundary, an image source has been used to account for the boundary effect on the analytical solution.

Two-Dimensional, Vertical Transport in a Heterogeneous Aquifer

This problem was developed by Sudicky (1989) to demonstrate the application of the Laplace Transform Galerkin method for a hypothetical field-scale example. The problem considers flow and solute transport in a heterogeneous cross section with a highly irregular flow field, dispersion parameters that are small compared with the spatial discretization, and a large contrast between longitudinal and transverse dispersivities. Segol (1994) refers to this problem as "The Waterloo Problem with Discontinuities" in a comprehensive compilation of benchmark simulations. Van der Heijde (1995) presents this problem as an example of "Level 2" testing, in which the objectives are to test potentially problematic parameter combinations and to demonstrate a code's applicability to typical real-world problems.

Conceptual model

The domain considered by Sudicky (1989) is the deformed quadrilateral shown in Figure 40a, with a length of 250 m and a depth ranging from about 6.5 m along the left boundary and 5.375 m along the right boundary. The flow system is assumed to be at steady state. The boundary conditions for flow are shown in Figure 40a. The left and bottom boundaries are impermeable, and a uniform head of 5.375 m is specified along the right boundary. The water table along the top boundary of the system is represented as a free surface across

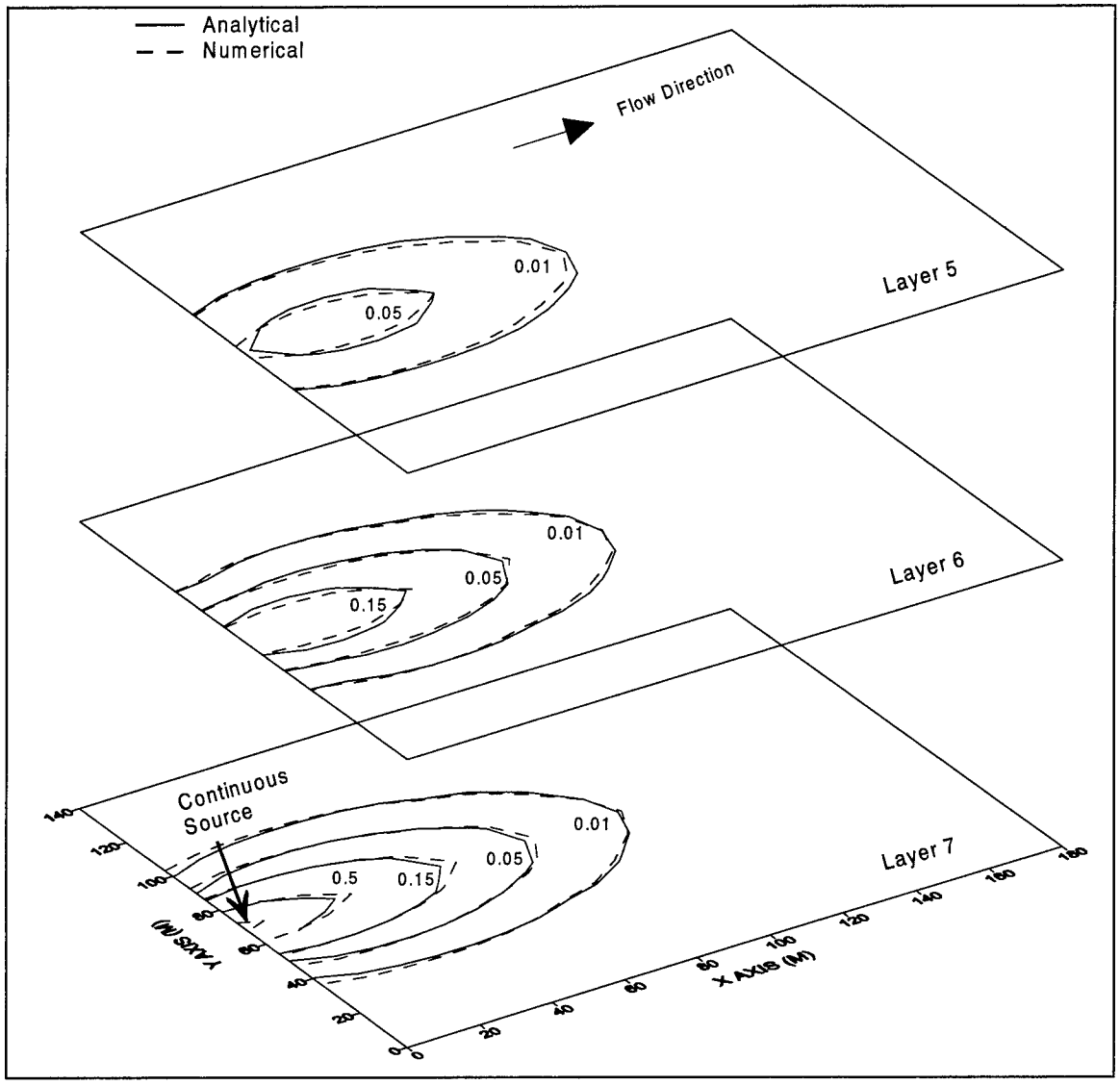


Figure 39. Comparison of the analytical and numerical solutions for three-dimensional transport from a continuous point source in a uniform flow field

which a uniform recharge of 10 cm/yr is applied. The aquifer consists of a fine-grained silty sand ($K_H = 5\times10^{-4}$ cm/sec) within which are located two lenses of medium-grained sand ($K_H = 10^{-2}$ cm/sec). The hydraulic conductivity is assumed to be isotropic.

The boundary conditions for solute transport are shown in Figure 40b. The concentration is specified along the top of the system. A relative concentration of 1.0 is assigned for the patch extending from x = 40 m to 80 m and is 0.0 elsewhere. After 5 years the source is removed and the concentration along the top reverts to a uniform value of zero. The domain is initially devoid of contaminants. A uniform porosity of 0.35 is assigned, the longitudinal (α_L) and transverse vertical dispersivity (α_{TV}) are 0.5 m and 0.005 m, respectively, and the effective diffusion coefficient is 1.34×10^{-5} cm^2/sec.

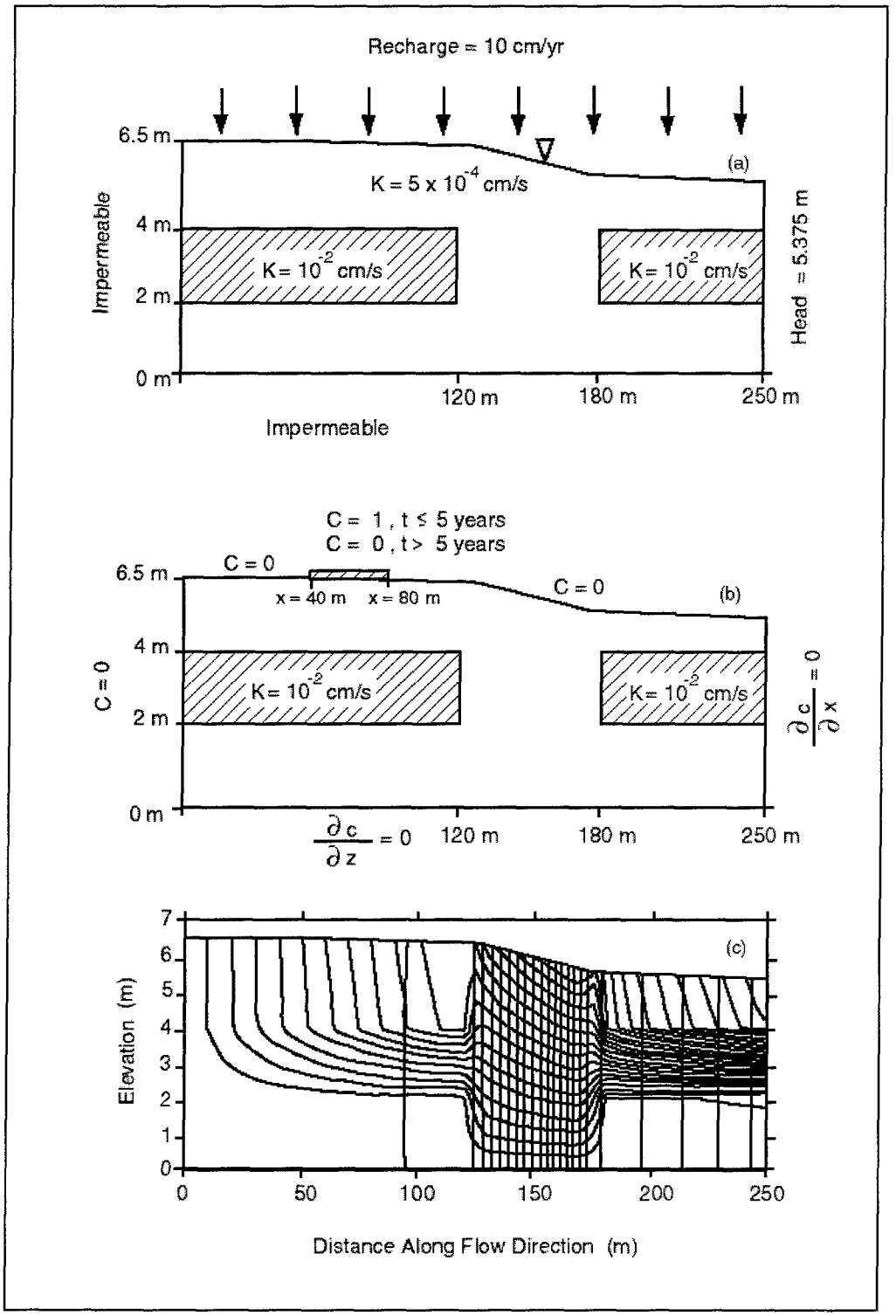


Figure 40. Physical system and flow boundary conditions (a), transport boundary conditions (b); and hydraulic head and stream function solutions (c) (contour intervals are 0.05 m for the hydraulic head and 1 m^2/day for the stream function (Sudicky 1989))

Flow solution

Sudicky (1989) solved the flow problem using the dual-formulation finite-element technique of Frind and Matanga (1985). The system was discretized into triangular elements using 51 equally spaced nodes in the horizontal direction (Δx = 5 m) and 25 nodes in the vertical direction. The location of the water table was determined by iterative adjustment of the vertical positions of nodes located above the medium-grained sand lenses. Figure 40c shows the hydraulic head and stream function solutions.

For the block-centered flow model used by MODFLOW and MT3DMS, the cross section is represented by a single row and fifty 5-m-long columns. Since MODFLOW determines the elevation of the water table by an iterative solution within a fixed grid, the vertical discretization of the cross section is intended to approximate rather than replicate the finite element model. The elevation of the water table at the left boundary is estimated from Figure 6 of Sudicky (1989) to be about 6.5 m, so that the average vertical spacing between nodes is about 0.25 m. The MODFLOW model is divided into 27 layers of uniform thickness 0.25 m, with the water table allowed to fall within the 6 uppermost layers.

Transport solution

Sudicky (1989) solved the transport problem using the Laplace Transform Galerkin (LTG) finite-element method with the same grid as that used to solve the flow problem. The LTG method is well suited for handing problems with relatively small dispersivities.

For the MT3DMS simulation, the transient top boundary condition is modeled using time-varying constant-concentration cells available as an option in the Sink/Source Mixing package. The MOC solution option is used for the advection term, with the solution control parameters listed below:

PERCEL	= 1.0
ITRACK	= 3 (mixed Euler and fourth-order Runge-Kutta)
WD	= 0.5
DCEPS	= 10^{-5}
NPLANE	= 0 (random initial particle placement)
NPL	= 0
NPH	= 10
NPMIN	= 2
NPMAX	= 20

The transport solutions obtained by the LTG solution and MT3DMS with the MOC option are shown on Figures 41 and 42, respectively. The agreement between the two solutions is reasonable, in light of the fundamental differences between the transport solution techniques and between the vertical discretization approaches in the finite-element and finite-difference models. Sudicky (1989) concluded that the LTG solution was relatively free of numerical dispersion

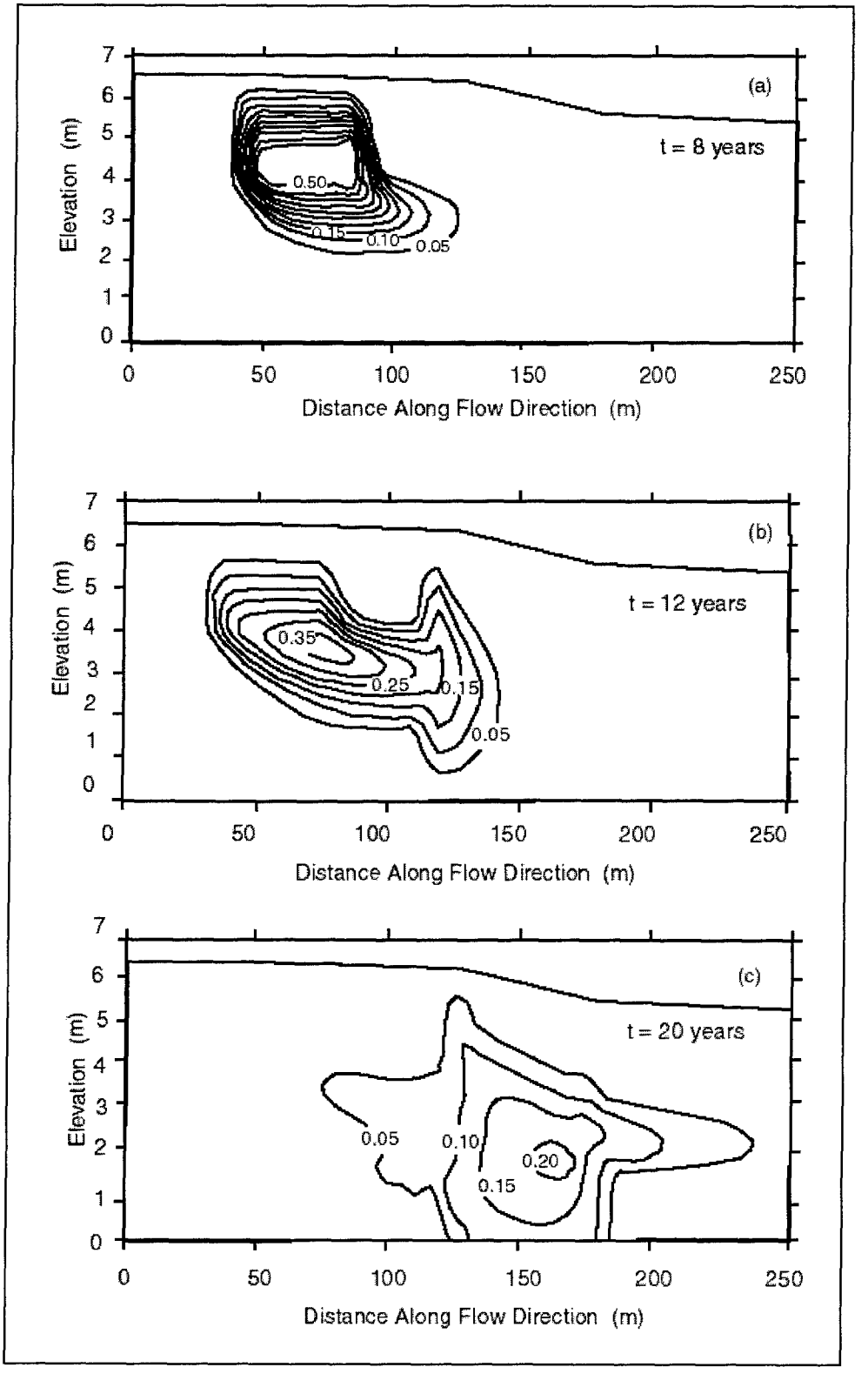


Figure 41. Plume configuration at (a) t = 8 years, (b) t = 12 years, and (c) t = 20 years based on the LTG method from Sudicky (1989)

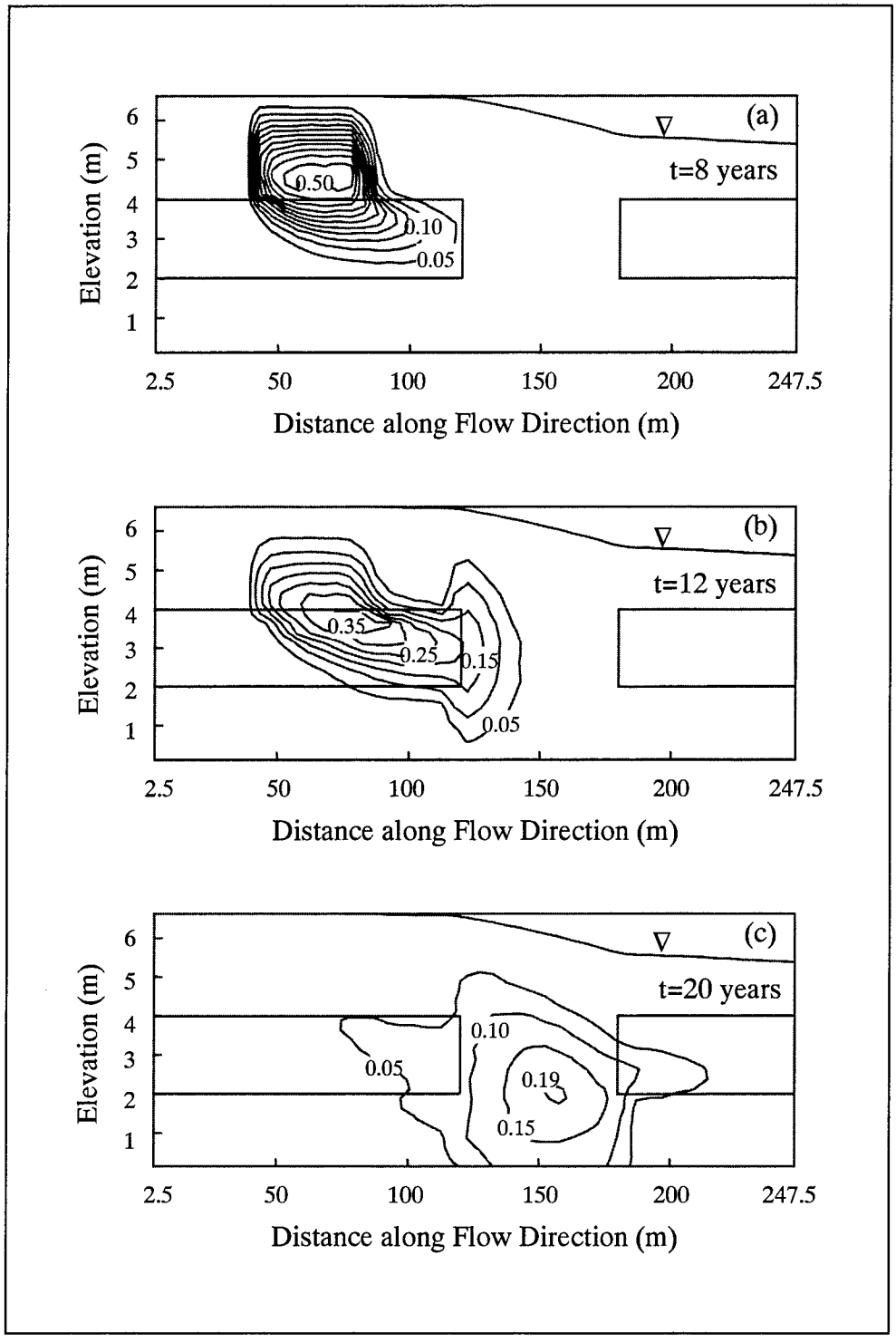


Figure 42. Plume configuration at (a) t = 8 years, (b) t = 12 years, and (c) t = 20 years as calculated by MT3DMS with the MOC scheme

because the C = 0.05 contour follows the pattern of the flow lines closely. The good agreement between the LTG and MT3DMS solutions suggests that the MT3DMS code is capable of simulating flow and solute transport in a heterogeneous cross section with a highly irregular flow field and relatively sharp fronts characteristic of field-scale applications. Furthermore, the mass balance discrepancy for the MT3DMS simulation is less than 1 percent at the end of the 20-year simulation period.

A Two-Dimensional Application Example

This section describes the application of the MT3DMS code to a hypothetical problem involving transport of contaminants in a two-dimensional heterogeneous aquifer in plan view. This problem is intended to compare the performance of various solution options available in MT3DMS for modeling transport in heterogeneous aquifers for which analytical solutions do not exist.

The configuration of the test problem is shown in Figure 43. The flow domain is discretized into 14 columns, 18 rows, and 1 layer with a uniform spacing of 100 m. The model grid is bounded by no-flow boundaries on the east and west sides. The north side is a constant-head boundary with a uniform head of 250 m. The south side is a specified-head boundary with the hydraulic head equal to 20 m at the center of column 1 (marked by +) and increasing at a gradient of 2.5/100 to 52.5 m in column 14. Water of a specific concentration is injected into the aquifer through a fully penetrating well, while a pumping well located downstream removes solute mass from the aquifer. Between the injection and pumping wells there is a zone of a low hydraulic conductivity as shown in Figure 43 in which the hydraulic conductivity is three orders of magnitude smaller than that elsewhere in the model region. The aquifer parameters used in the simulation are listed below and in Figure 43:

Cell width along rows (Δx) = 100 m
Cell width along rows (Δy) = 100 m
Layer thickness (Δz) = 10 m
Porosity (θ) = 0.3
Longitudinal dispersivity = 20 m
Ratio of horizontal transverse to longitudinal dispersivity = 0.2
Simulation time = 2 years

The boundary conditions for the transport model are no-mass flux boundaries on the east, west, and north borders of the model. The south border is a specified advective mass flux boundary, acting as a line of point sinks taking mass out of the aquifer. The mass removed is equal to the flow entering the constant-head cells multiplied by the concentration at the cells.

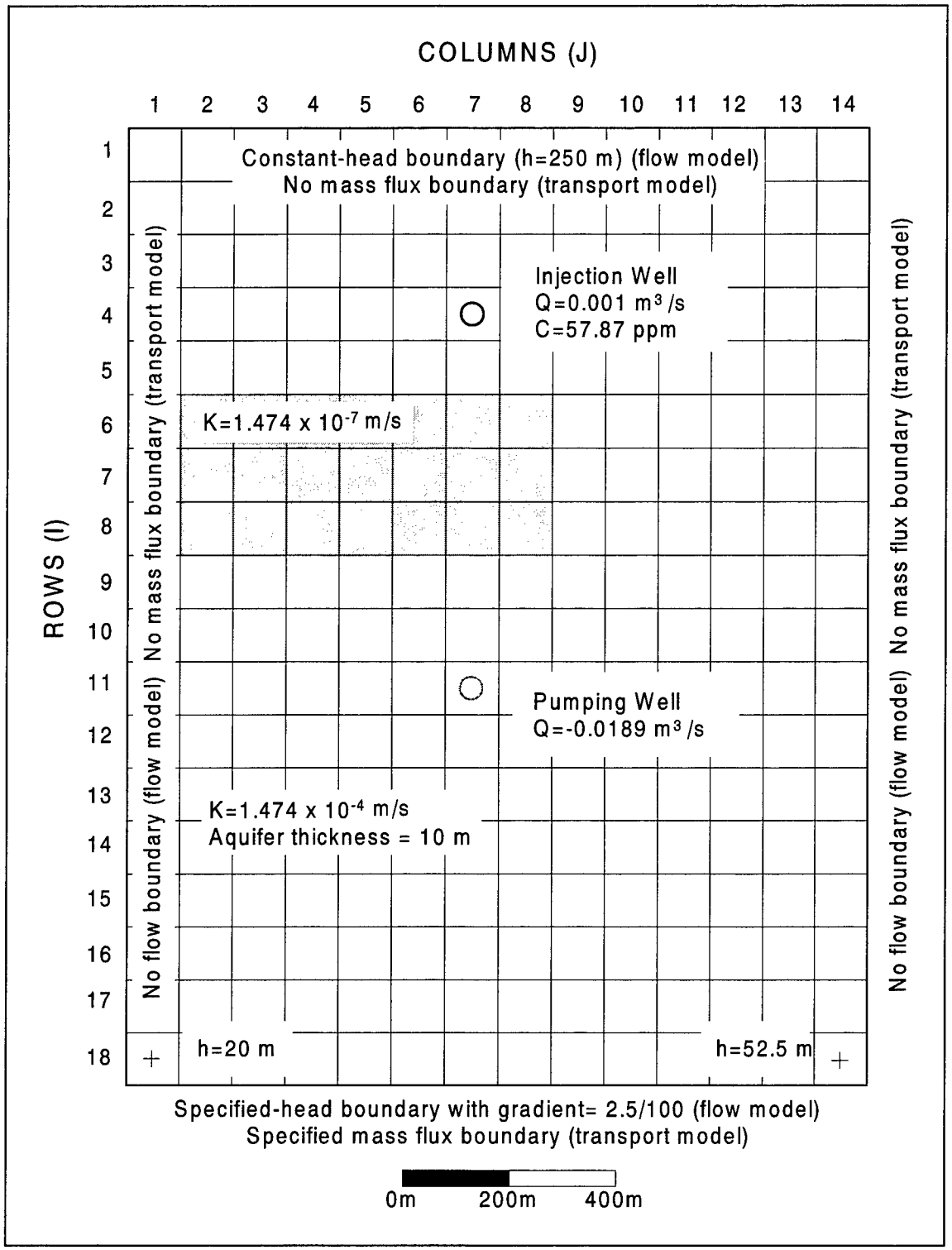


Figure 43. Configuration of the test problem involving transport in a heterogeneous aquifer with a strong regional gradient

The aquifer is assumed to be confined and the flow field is steady state with constant injection and pumping rates at the two wells as shown in Figure 43. The concentration of the injected water is 57.78 ppm in the first year but becomes zero in the second year. Because the flow rates at the injection and pumping wells are constant, only one stress period is necessary for the flow model. However, two stress periods are needed for the transport model. The HMOC and the third-order TVD (ULTIMATE) schemes are used to solve the advection term, while the dispersion and sink/source terms are solved using the implicit finite-difference method with the convergence criterion set equal to 10^{-4} in the GCG solver. The solution control parameters for the HMOC scheme are listed below:

PERCEL	= 1.0
ITRACK	= 2 (fourth-order Runge-Kutta)
DCEPS	= 10^{-5}
NPLANE	= 0 (random initial particle placement)
NPL	= 0
NPH	= 16
NPMIN	= 0
NPMAX	= 32

The calculated concentrations at the end of the first year are plotted in Figure 44. While the overall shapes of the two plumes based on the HMOC and ULTIMATE schemes are similar, the HMOC solution has a more "rough" appearance than the ULTIMATE solution, a characteristic of the particle-tracking-based solution approach. The mass balance discrepancy for the HMOC solution varies with time but can be reduced to less than 1 percent at the end of the simulation by experimenting with the various empirical solution parameters. The ULTIMATE solution, on the other hand, has a mass balance discrepancy error of less than 10^{-4} percent for the entire simulation.

Figure 45 shows the calculated concentrations based on different solution schemes at the pumping well. Again, the discrete nature of the particle-tracking-based approach is apparent in the HMOC solution, exaggerated by the artificially coarse grid used in this test problem. The ULTIMATE solution, on the other hand, agrees well with the HMOC solution without the undesirable roughness. It is also clear that the finite-difference solution with upstream weighting introduces a significant amount of numerical dispersion while the finite-difference solution with central-in-space weighting results in some oscillation.

A Three-Dimensional Field Case Study

This section describes the application of the MT3DMS code to an actual field problem involving the evaluation of the effectiveness of proposed groundwater remediation schemes. The discussion is intended to demonstrate the performance of the MT3DMS code in a representation of real-world problems.

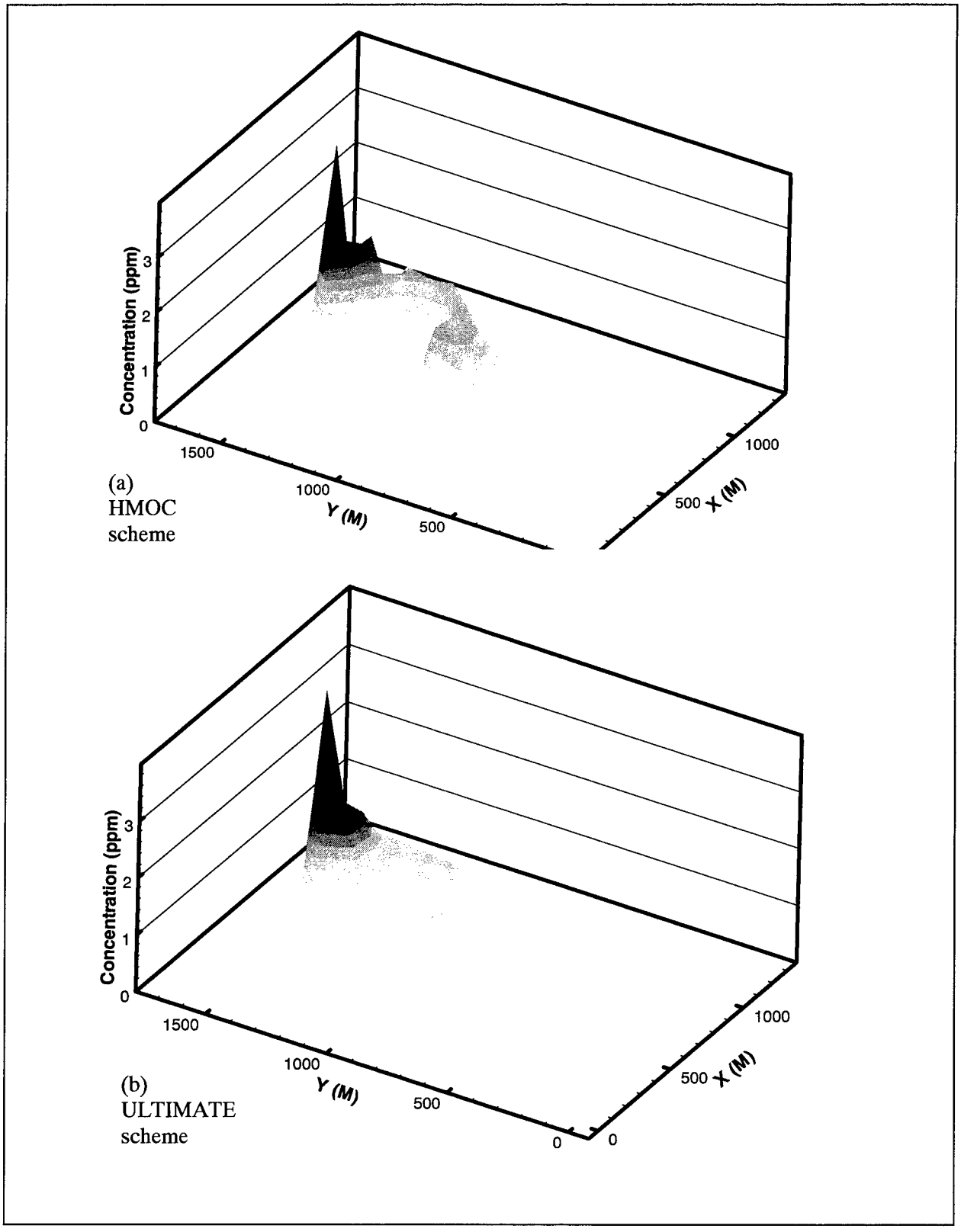


Figure 44. Comparison of the calculated concentrations at the end of the 1-year simulation period based on the HMOC and the third-order TVD (ULTIMATE) schemes

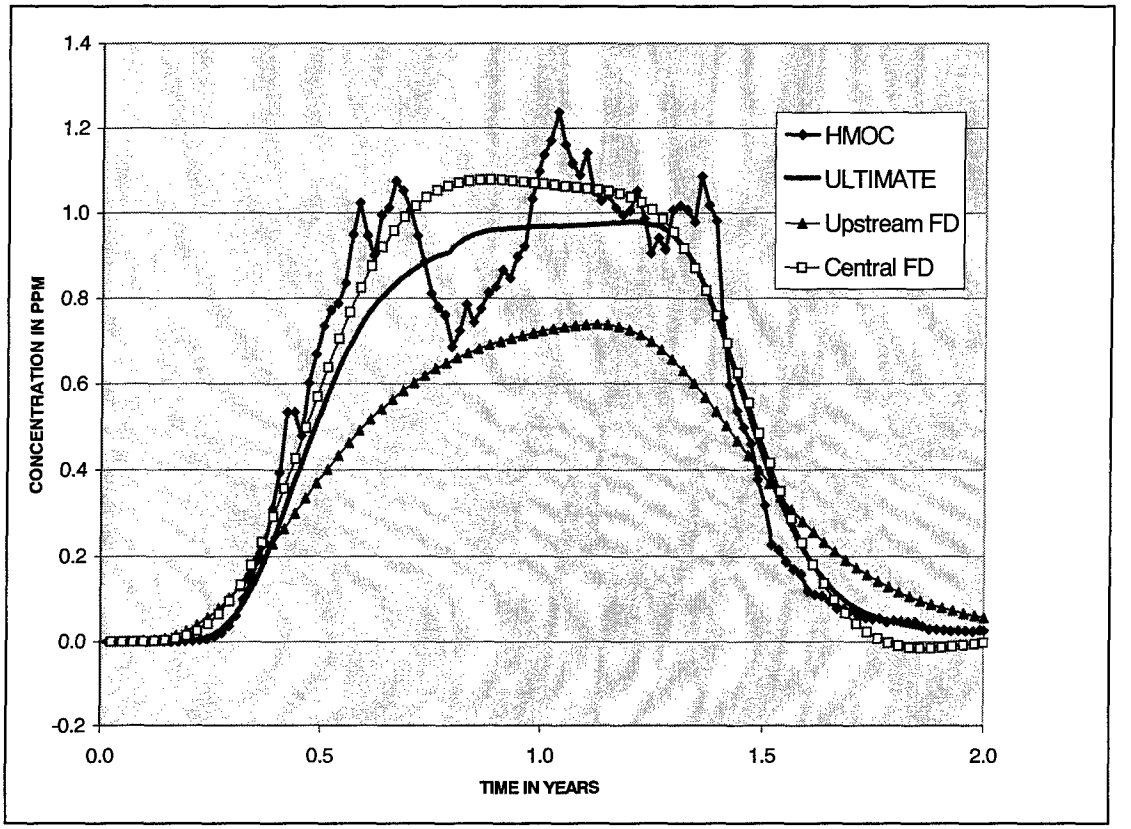


Figure 45. Comparison of the calculated concentration breakthrough curves based on different solution schemes at the pumping well

Site description

The geologic setting of the study site is illustrated in Figure 46. The unconfined aquifer beneath the site consists of an upper zone of fine- and medium-grain sands, with occasional discontinuous lenses of silty sands, and a lower zone of medium to coarse sands with some gravel. The hydraulic conductivity for the upper and lower zones are approximately 60 and 520 ft/day, respectively. The average groundwater recharge rate estimated for the site is around 12.7 cm (5 in.) per year. The porosity value is approximately 30 percent. Other relevant aquifer parameters are listed in Table 4. Several organic contaminants were detected in groundwater beneath the site. Among them is 1,2-dichloroethane (1,2-DCA), which was found over an area of more than 670 $\times$ 396 m (2200 $\times$ 1300 ft, with the maximum concentration exceeding 200 ppb (Figure 47).

Table 4
Summary of Aquifer Parameters at the Field Study Site

Parameter	Value
Hydraulic conductivity (medium to fine sand)	18 m/day (60 ft/day)
Hydraulic conductivity (coarse sand)	159 m/day (520 ft/day)
Ratio of vertical to horizontal hydraulic conductivity	0.1
Recharge rate	12.7 cm/yr (5 in./yr)
Saturated thickness	30.5 m (100 ft)
Longitudinal dispersivity	3 m (10.0 ft)
Transverse dispersivity	0.6 m (2.0 ft)
Porosity	0.3
Aquifer bulk density	1.7 g/cm^3
Distribution coefficient (K_d)	0.176 cm^3/g

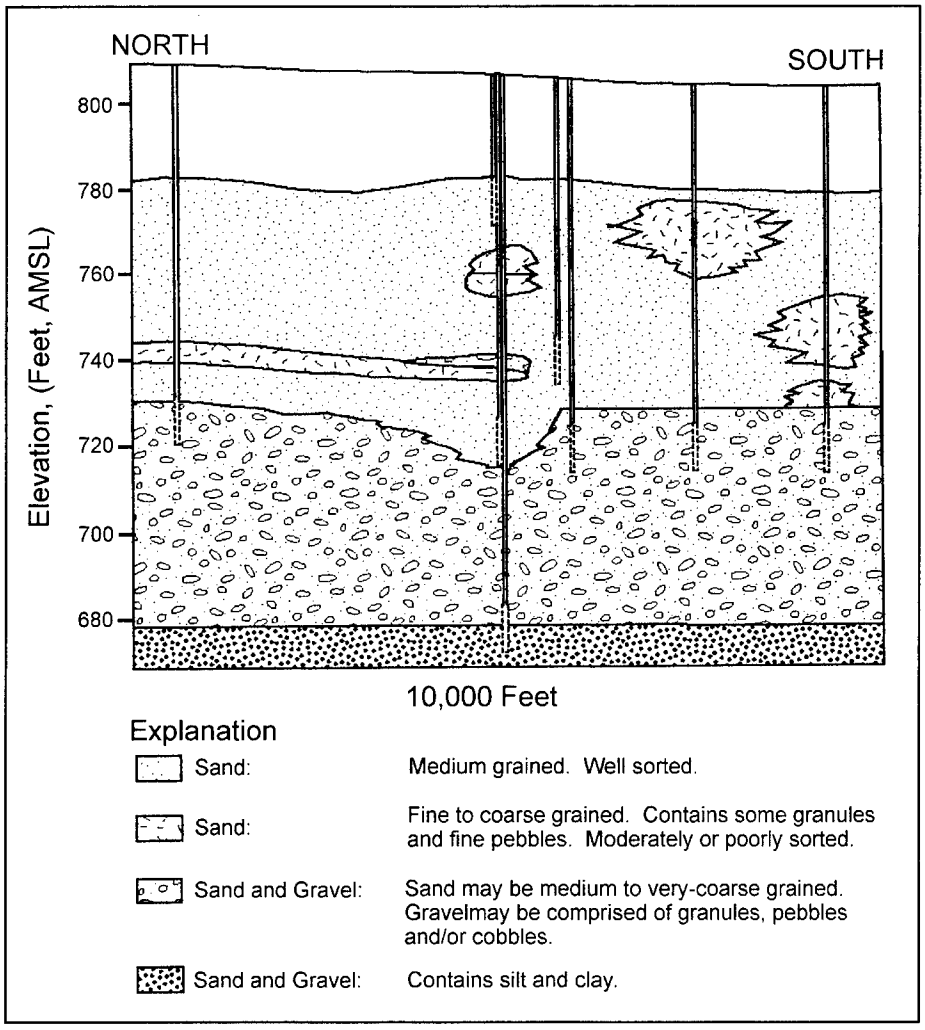


Figure 46. Geological setting at the study site of the three-dimensional field example

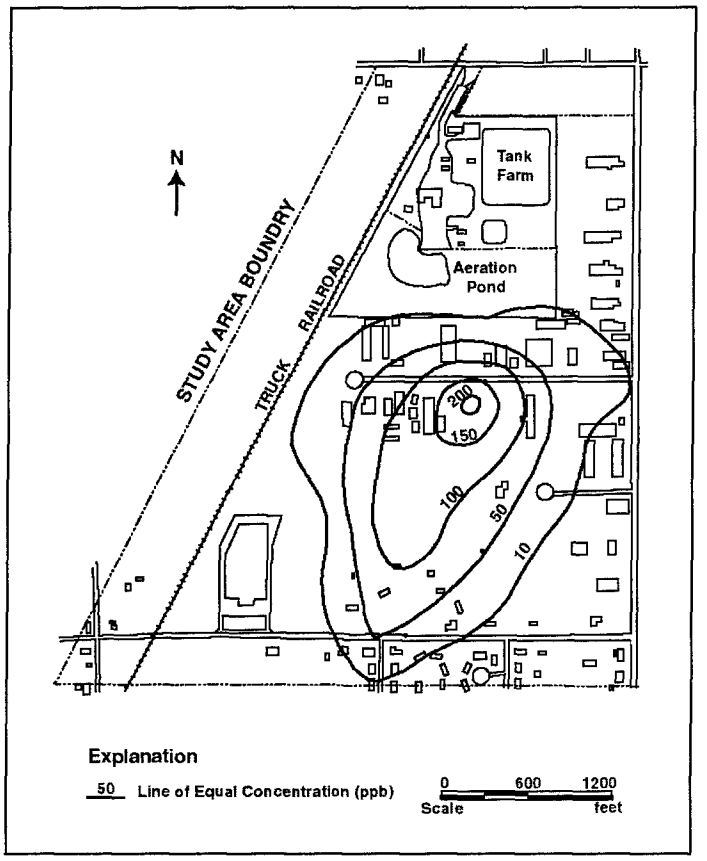


Figure 47. Initial distribution of 1,2-DCA in ppb at approximately 30 m (100 ft) below the land surface at the study site for the field example problem

Flow and transport models

The objective of numerical simulation at this site is to investigate the effectiveness and performance of various remedial scenarios designed to contain the 1,2-dichloroethane plume and eventually clean up the aquifer. The numerical model consists of four layers in the vertical direction as shown in Figure 48. Each layer has a uniform thickness of 7.62 m (25 ft). In plan view, each layer is discretized into 61 rows and 40 columns. The mesh spacing is 15.24 × 15.24 m (50 × 50 ft) in the detailed study area and progressively increases toward the model boundaries. The boundary conditions for the flow model are specified-head on the four sides, no-flow at the bottom, and specified-flux from recharge at the water table. The heads at the side boundaries are set to establish a regional gradient of 5×10^{-4} from east to west and 1×10^{-3} from north to south. The boundary conditions for the transport model are no-mass-flux at the bottom and specified, advective mass flux elsewhere. The advective mass flux is determined internally in the model by the rate of inflow or outflow across each boundary node and the concentration of inflow or outflow (inflow concentration is zero by default). The boundaries are sufficiently far away from the detailed study site so that their effects on the flow and transport in the immediate vicinity of the site are minimized. Most of the cells are concentrated in the central part of the mesh,

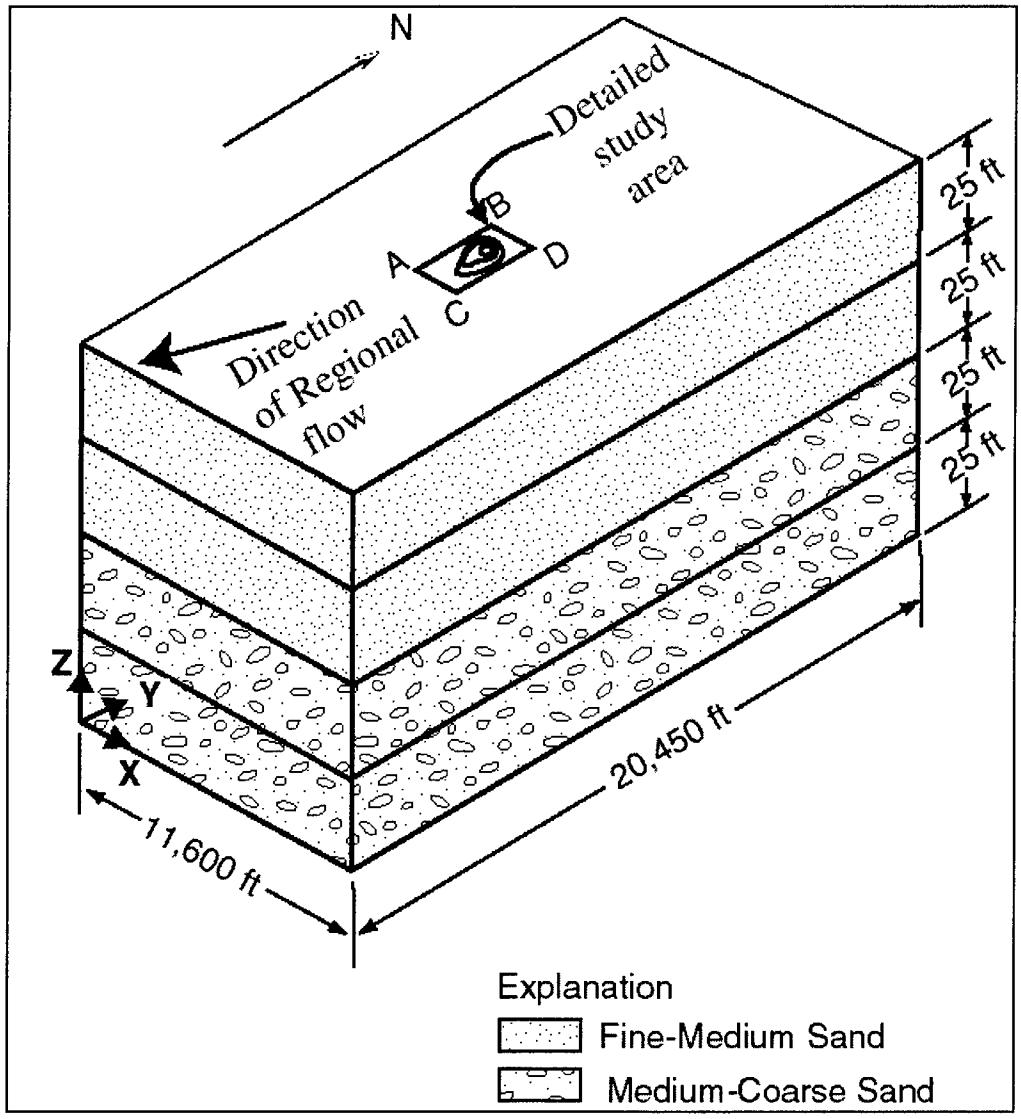


Figure 48. Schematic diagram showing the structure of the flow and transport models developed for the field example (11,600 ft = 3.535.68 m; 20,450 ft = 6,233 m; 25 ft = 7.62 m)

denoted by ABCD in Figure 48, in the area of the contaminant plume. Only steady-state conditions are represented in the flow model.

The measured 1,2-dichloroethane concentrations are used as the initial condition for the transport model prior to any remediation effort. The initial concentration distribution within the detailed study area for model layer three is shown in Figure 49a. The initial concentration in layer two is assigned 20 percent of that in layer three while the top and bottom layers are initially clean. One of the scenarios for cleanup is to use eight extraction wells as shown in Figure 49 to pump the contaminated groundwater out of the aquifer for treatment. The proposed total extraction rate for the eight wells is about 4.25×10^3 m^3/day, all from layer three.

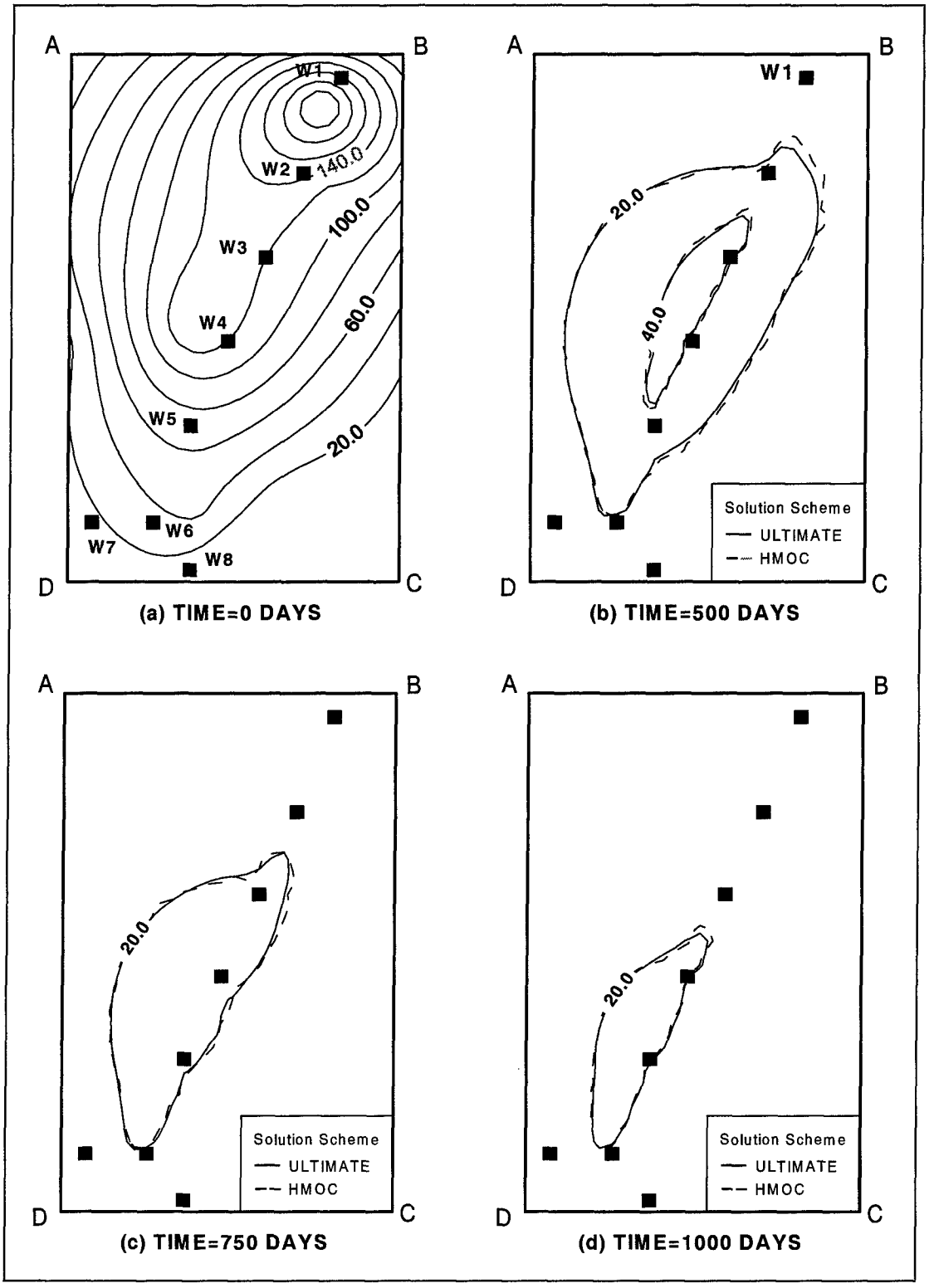


Figure 49. Initial and calculated concentration distributions in model layer three at 0, 500, 750, and 1,000 days (ABCD indicates the detailed study area as depicted in Figure 48)

The transport simulation incorporates an equilibrium-controlled linear sorption isotherm. A constant retardation factor of 2.0 is calculated from the bulk density and distribution coefficient given. With the retardation factor of two, the initial total mass in the aquifer is twice the total dissolved mass.

Comparison of solution schemes

The transport model is solved using the following combinations of solution options: (a) the explicit ULTIMATE scheme for advection and the implicit finite-difference method for all other terms, (b) the particle-tracking-based HMOC scheme for advection and the implicit finite-difference method for all other terms, and (c) the fully implicit finite-difference method with upstream weighting for the advection term. The solution control parameters used in the HMOC scheme are DCEPS = 10^{-5}, NPLANE = 0, NPL = 0, NPH = 16, NPMIN = 2, NPMAX = 32, and DCHMOC = 0.01. Initial particles are distributed using the random pattern (i.e., NPLANE = 0). Particle tracking is performed using the mixed first-order Euler and fourth-order Runge-Kutta algorithms. The Courant number is fixed at 1.0 for all three solutions.

The calculated concentration distributions in model layer three (3) within the detailed study area, at times equal to 500, 750, and 1,000 days after the proposed pump-and-treat system is started, are shown in Figures 49b, c, and d. The concentrations calculated by the ULTIMATE and HMOC schemes agree well with each other, and also with those calculated by the fully implicit finite-difference method with upstream weighting (Figure 50). The mass balance discrepancy for the HMOC solution is less than 1 percent throughout the simulation. The concentration breakthrough curves at pumping well W4, which is located near the middle of the initial plume, are shown in Figure 50. Again, all three solutions agree well with one another, particularly between the ULTIMATE and HMOC solutions.

Consideration of efficiency versus accuracy

For this application example, the accuracy of the fully implicit finite-difference solution is comparable to that of the ULTIMATE and HMOC solutions. Thus, it is advantageous to use the fully implicit finite-difference option because of its computational efficiency. With the fully implicit finite-difference scheme, there is no restriction on the transport time-step size that may be used, and the user can run the simulation in as few transport steps as desired. However, as the time-step size increases, the accuracy of the transport solution usually deteriorates as the Courant number becomes much greater than one. This accuracy requirement on the transport step size should be taken into consideration when specifying a transport step size multiplier for the fully implicit finite-difference scheme.

To illustrate the point above, the fully implicit finite-difference scheme is used to solve the same transport problem but with different transport step size multipliers. The explicit finite-difference scheme, which has a maximum

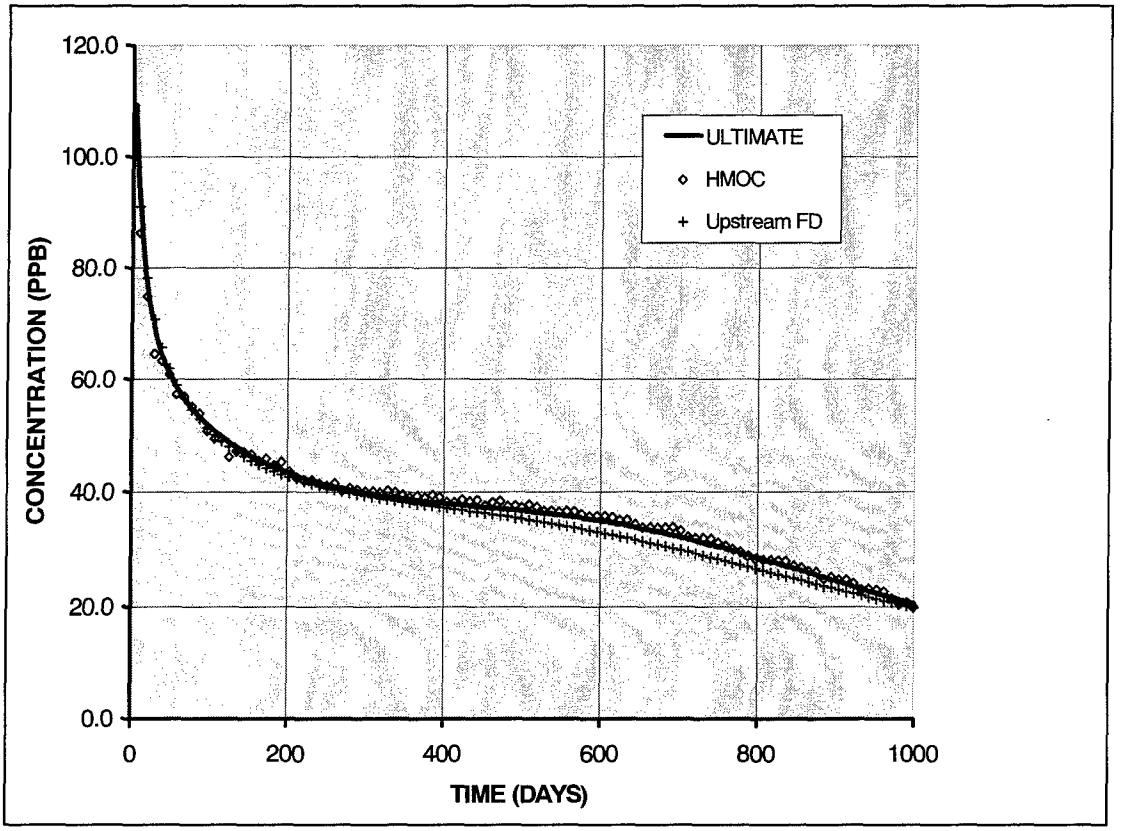


Figure 50. Concentration breakthrough curves as calculated using different solution schemes at the pumping well W4, located near the middle of the initial plume

transport step size of 2.25 days to meet the various stability constraints, is used as the base case. The computation times cited in Table 5 are computer clock times on a Pentium Pro 200 MHz PC. The GCG solver uses the MIC preconditioner with the convergence error criterion set to 10^{-4}. The time inside the parentheses is that obtained with the option of lumped dispersion cross terms turned on. Although the simulation can be speeded up dramatically by using greater step size multipliers, the solutions become less accurate, as measured approximately by the total masses removed from the aquifer relative to that of the base case.

Table 5
Comparison of Computation Times Using Explicit and Implicit Schemes

Explicit or Implicit Schemes	Initial Transport Step Size	Transport Step Multiplier	Total Number of Steps Used	Total Computation Time, sec[1]	Relative Mass Removed
Explicit	2.25	N/a	889	128	1.00
Implicit Run 1	2.25	1.2	29	64 (24)	0.98
Implicit Run 2	2.25	1.5	16	38 (16)	0.95
Implicit Run 3	2.25	2.0	10	26 (11)	0.93
Implicit Run 4	2000	N/a	1	8 (3)	0.79

[1] Time inside parentheses was obtained with the option of lumped dispersion cross terms turned on.

Effect of nonequilibrium sorption

In the simulations already discussed, the sorption is modeled using the equilibrium-controlled linear isotherm. To assess the effect of the local equilibrium assumption on the effectiveness of the proposed remedial system, new simulations are made using the kinetic (nonequilibrium) sorption isotherm available in MT3DMS. Only one additional parameter, the first-order kinetic rate β, is needed. In the simulations performed in this study, the first-order kinetic rates of 0.1, 1.5×10^{-4}, and 10^{-6} day^{-1} are used to represent the fast, moderate, and slow sorption processes, respectively.

As shown in Figure 51, when the sorption rate constant is equal to 0.1 day^{-1}, the kinetic sorption approaches the equilibrium sorption. In other words, the local equilibrium assumption used in earlier simulations is valid. However, when the rate constant is in the range of 1.5×10^{-4} day^{-1}, the local equilibrium assumption would result in considerable overestimation of mass removed. As the rate constant decreases further (i.e., the sorption process is sufficiently slow compared to the transport process), the kinetic sorption approaches the case without sorption. Note that less mass is removed without sorption because by assuming no sorption, there is initially less mass in the aquifer.

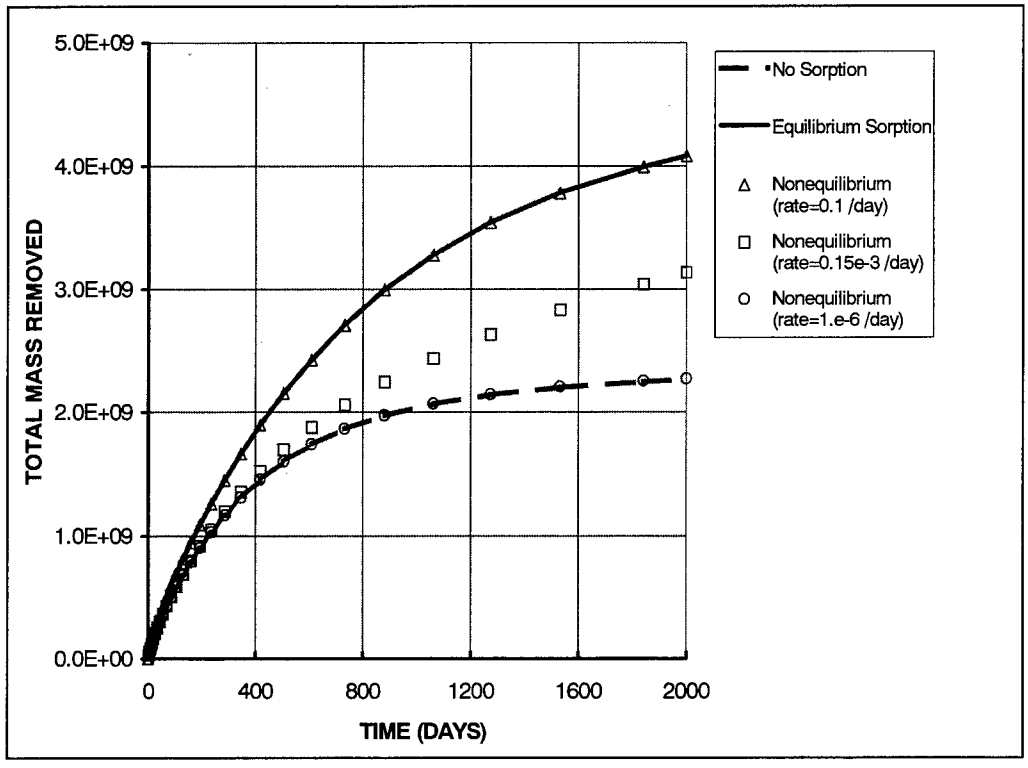


Figure 51. The effect of kinetic sorption rates on the mass removal for the field example

References

Anderson, M. P. (1979). "Using models to simulate the movement of contaminants through groundwater flow systems," *CRC Critical Rev. Environ. Control* 9(2), 97-156.

_____. (1984). "Movement of contaminants in groundwater: groundwater transport -- advection and dispersion." *Groundwater Contamination.* National Academy Press, Washington, DC, 37-45.

Bear, J. (1972). *Dynamics of Fluids in Porous Media.* American Elsevier Publishing Company, New York.

_____. (1979). *Hydraulics of Groundwater.* McGraw-Hill, New York.

Burnett, R. D., and Frind, E. O. (1987). "An alternating direction Galerkin technique for simulation of groundwater contaminant transport in three dimensions, 2, dimensionality effects," *Water Resour. Res.* 23(4), 695-705.

Cheng, R. T., Casulli, V., and Milford, S. N. (1984). "Eulerian-Lagrangian solution of the convection-dispersion equation in natural coordinates," *Water Resour. Res.* 20(7), 944-52.

Clement, T. P. (1997). RT3D, A Modular Computer Code for Simulating Reactive Multispecies Transport in 3-Dimensional Groundwater Aquifers, Pacific Northwest National Laboratory, Richland, Washington.

Cox, R. A., and Nishikawa, T. (1991). "A new total variation diminishing scheme for the solution of advective-dominant solute transport," *Water Resour. Res.* 27(10), 2645-54.

El-Kadi, A. I. (1988). "Applying the USGS mass-transport model (MOC) to remedial actions by recovery wells," *Ground Water* 26(3), 281-88.

Farmer, C. L. (1987). "Moving particle techniques." *Advances in transport phenomena in porous media.* J. Bear and M.Y. Corapcioglu, eds., *NATO ASI Series #128, Nijhoff, Boston.*

Frind, E. O., and Matanga, G. B. (1985). "The dual formulation of flow for contaminant transport modeling, 1. Review of theory and accuracy aspects," *Water Resour. Res.* 21(2), 159-69.

Garder, A. O., Jr., Peaceman, D. W., and Pozzi, A. L., Jr. (1964). "Numerical calculation of multidimensional miscible displacement by the method of characteristics," *Soc. Pet. Eng. J.* 6(2), 175-82.

Gelhar, L. W., and Collins, M. A. (1971). "General analysis of longitudinal dispersion in nonuniform flow," *Water Resour. Res.* 7(6), 1511-21.

Golden Software, Inc. (1996). Surfer® Version 6.0 Reference Manual. Golden, CO.

Goode, D. J. (1990). "Particle velocity interpolation in block-centered finite difference groundwater flow models," *Water Resour. Res.* 26(5), 925-40.

Goode, D. J., and Konikow, L. F. (1989). "Modification of a method-of-characteristics solute transport model to incorporate decay and equilibrium-controlled sorption and ion exchanges," *U.S. Geological Survey Water-Resources Investigations Report 89-4030.*

Grove, D. B., and Stollenwerk, K. G. (1984). "Computer model of one-dimensional equilibrium-controlled sorption processes," *U.S. Geological Survey Water-Resources Investigations Report 84-4059.*

Gustafsson, I. (1978). "A class of first order factorization methods," *BIT* 18, 142-56.

Hagman, L. A., and Young, D. M. (1981). *Applied Iterative Methods.* Academic Press, New York.

Harbaugh, A. W., and McDonald, M. D. (1996). "User's documents 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.*

Harten, A. (1983). "High resolution schemes for hyperbolic conservation laws," *J. Comp. Phys.* 49, 357-93.

_____. (1984). "High resolution schemes for hyperbolic conservation laws," *J Comp. Phys.* 21, 955.

Healy, R. W., and Russell, T. F. (1993). "A finite-volume Eulerian-Lagrangian localized adjoint method for solution of the advection-dispersion equation," *Water Resour. Res.* 29(7), 2399-2413.

Healy, R. W., and Russell, T. F. (1989). "Efficient implementation of the modified method of characteristics in finite-difference models of solute transport." *Proc. 4th int. conf. on the use of models to analyze and find working solutions to groundwater problems*, National Water Well Association.

Hestenes, M. R., and Stiefel, E. L. (1952). "Methods of conjugate gradients for solving linear systems," *J. Res. Nat. Bur. Standard* 49, 409-36.

Hill, M. C. (1992). "A computer program (MODFLOWP) for estimating parameters of a transient, three-dimensional, ground-water flow model using nonlinear regression," *U.S. Geological Survey Open-File Report 91-484.*

Hunt, B. W. (1978). "Dispersive sources in uniform groundwater flow," *ASCE Journal of the Hydraulics Division* 104(HY1), 75-85.

Hunt, B. (1983). *Mathematical analysis of groundwater resources.* Butterworths, Cambridge, MA.

Javandel, I., Doughty, C., and Tsang, C. F. (1984). "Groundwater transport: Handbook of mathematical models," *American Geophysical Union Water Resources Monogram 10.*

Jea, K. C. (1982). "Generalized conjugate gradient acceleration of iterative methods," Ph.D. diss., also report CNA-176, Center for Numerical Analysis, The University of Texas at Austin, Texas.

Jea, K. C., and Young, D. M. (1983). "On the simplification of generalized conjugate gradient methods for nonsymmetrizable linear systems," *Linear Algebra and Its Application* 52, 399-417.

Konikow, L. F., and Bredehoeft, J. D. (1978). "Computer model of two-dimensional solute transport and dispersion in ground water." *U.S. Geological Survey. Water Resour. Invest. Book 7.*

Konikow, L. F., Goode, D. J., and Hornberger, G. Z. (1996). "A three-dimensional method-of-characteristics solute-transport model (MOC3D)," *U.S. Geological Survey Water-Resources Investigations Report 96-4267.*

LaBolle, E. M., Fogg, G. E, and Tompson, A. F. B. (1996). "Random-walk simulation of transport in heterogeneous porous media: Local mass-conservation problem and implementation methods," *Water Resour. Res.* 32(3), 583-93.

Lanczos, C. (1950). "Solution of systems of linear equations by minimized iterations," *J. Res. Nat. Bur. Standards* 49, 33-53.

Leonard, B. P. (1979). "A stable and accurate convective modeling procedure based on quadratic upstream interpolation." *Computer Methods Appl. Mech. Engng.* 19, 59.

Leonard, B. P. (1988). "Universal Limiter for transient interpolation modeling of the advective transport equations: The ULTIMATE conservative difference scheme," NASA Technical Memorandum 100916 ICOMP-88-11, Washington, DC.

Leonard, B. P., and Niknafs, H. S. (1990). "Cost-effective accurate coarse-grid method for highly convective multidimensional unsteady flows," NASA Conference Publication 3078: Computational Fluid Dynamics Symposium on Aeropropulsion, Washington, DC.

_____. (1991). "Sharp monotonic resolution of discontinuities without clipping of narrow extrema," *Computer & Fluids* 19(1), 141-154.

McDonald M. D., and Harbaugh, A. W. (1988). "A modular three-dimensional finite-difference flow model." *Techniques of Water Resources Investigations of the U.S. Geological Survey, Book 6.*

Manteuffel, T. A. (1977). "The Tchebyshev iteration for non-symmetric linear systems," *Num. Math.* 28, 307-27.

Mitchell, A. R. (1984). "Recent developments in the finite element method." *Computational techniques and applications: CTAC-83.* J. Noye and C.A.J. Fletcher, eds., Elsevier, North-Holland.

Moench, A. F., and Ogata, A. (1981). "A numerical inversion of the Laplace transform solution to radial dispersion in a porous medium," *Water Resour. Res.* 17(1), 250-53.

Neuman, S. P. (1981). "A Eulerian-Lagrangian numerical scheme for the dispersion-conversion equation using conjugate space-time grids," *J. of Computational Physics* 41, 270-94.

_____. (1984). "Adaptive Eulerian-Lagrangian finite element method for advection-dispersion," *Int. J. Numerical Methods in Engineering* 20, 321-37.

Osher, S., and Chakravarthy, S. (1984). "High resolution schemes and the entropy condition," *SIAM J. Numer. Anal.* 21, 955.

_____. (1986). "Very high order accurate TVD schemes." *IMA Volumes in Mathematics and its Applications,* Vol 2, Springer-Verlag, 229-274, New York.

Pollock, D. W. (1988). "Semianalytical computation of path lines for finite-difference models," *Ground Water* 26(6), 743-750.

Press, W. H. (1986). *Numerical recipes*: *The art of scientific computing*. University Cambridge Press, MA.

Roache, P. J. (1992). "A flux-based modified method of characteristics," *Int. J. Numerical Methods in Fluids* 15, 1259-75.

Russell, T. F., and Wheeler, M. F. (1983). "Finite element and finite difference methods for continuous flows in porous media." *SIAM the mathematics of reservoir simulation.* R.E. Ewing, ed., Chapter II., 35-106.

Saad, Y. (1982). "The Lanczos biorthogonalization algorithm and other oblique projection methods for solving large unsymmetric systems, *SIAM J. Num. Analy.* 19, 485-506.

Segol, G. (1994). *Classic Groundwater Simulations.* PTR Prentice Hall, Englewood Cliffs NJ, 176-180.

Sudicky, E. A. (1989). "The Laplace transform Galerkin technique: Application to mass transport in groundwater," *Water Resour. Res.* 25(8), 1833-1846.

Van der Heijde, P. K. M. (1995). "Model testing: A functionality analysis, performance evaluation, and applicability assessment protocol." *Groundwater Models for Resources Analysis and Management.* A.I. El-Kadi, ed., CRC Press, Lewis Publishers, Boca Raton, FL.

Van Genuchten, M. Th., and Alves, W. J. (1982). "Analytical solutions of the one-dimensional convective-dispersive solute transport equation," *U.S. Department of Agriculture Technical Bulletin No. 1661.*

Weerts, A. H. (1994). "Analytical models for chemical transport in the subsurface environment," Ph.D. diss., Wageningen Agricultural University, The Netherlands.

Widdowson, M. A., and Waddill, D. W. (1997). "SEAM3D, a numerical model for three-dimensional solute transport and sequential electron acceptor-based bioremediation in groundwater," Draft Report to the U.S. Army Corps of Engineers Waterways Experiment Station, Vicksburg, MS.

Wilson, J. L., and Miller, P. J. (1978). "Two-dimensional plume in uniform ground-water flow," *J. Hyd. Div.*, ASCE 4, 503-514.

Yee, H. C. (1987). "Construction of explicit and implicit symmetric TVD schemes and their applications," *J. Comp. Phys.* 68, 151-179.

Yeh, G. T. (1990). A Lagrangian-Eulerian method with zoomable hidden fine-mesh approach to solving advection-dispersion equations, *Water Resour. Res.* 26(6), 1133-1144.

Yeh, G. T., Chang, J. R., and Short, T. E. (1992). "An exact peak capturing and oscillation-free scheme to solve advection-dispersion transport equations," *Water Resour. Res.* 28(11), 2937-2951.

Yeh, T. C. J., Srivastava, R., Guzman, A., and Harter, T. (1993). "A numerical model for water flow and chemical transport in variably saturated porous media," *Ground Water* 31(4), 634-644.

Young, D. M., and Jea, K. C. (1980). "Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods," *Linear Algebra and Its Application* 34, 159-194.

Young, D. M., and Mai, T. Z. (1988). "Iterative algorithms and software for solving large sparse linear systems," *Comm. in Appl. Num. Meth.* 4, 435-456.

Zheng, C. (1988). "New solution and model for evaluation of groundwater pollution control," Ph.D. diss., Univ. of Wisconsin-Madison.

_____. (1989). "PATH3D -- a ground-water path and travel-time simulator." *Version 2.0 user's manual.* S.S. Papadopulos & Associates, Inc.

_____. (1990). "MT3D, A modular three-dimensional transport model for simulation of advection, dispersion and chemical reactions of contaminants in groundwater systems," Report to the U.S. Environmental Protection Agency, Robert S. Kerr Environmental Research Laboratory, Ada, OK.

_____. (1993). "Extension of the method of characteristics for simulation of solute transport in three dimensions," *Ground Water* 31(3), 456-465.

Zheng, C., and Bennett, G. D. (1995). *Applied contaminant transport modeling: Theory and practice*, John Wiley and Sons, New York.

Appendix A
Description of the Generalized Conjugate Gradient Solver[1]

This section describes iterative algorithms in the Generalized Conjugate Gradient Solver (GCG) Package for solving the linear system

$$\boldsymbol{Au} = \boldsymbol{b} \tag{A1}$$

where the coefficient matrix $\boldsymbol{A}$ is a given real nonsingular matrix of order N, $\boldsymbol{b}$ is a given column vector, and vector $\boldsymbol{u}$ is to be determined. The matrix $\boldsymbol{A}$ is assumed to be large and sparse.

The iterative algorithms which we consider include the following components (Young and Mai 1988):[2]

a. A basic iterative method.

b. An acceleration procedure

c. A stopping procedure.

Basic Iterative Method

A basic iterative method for solving $\boldsymbol{Au} = \boldsymbol{b}$ is a method of the form

$$\boldsymbol{u}^{(n+1)} = \boldsymbol{G}\boldsymbol{u}^{(n)} + \boldsymbol{k} \tag{A2}$$

where $\boldsymbol{u}^{(0)}$ is arbitrary, $\boldsymbol{G}$ is the iteration matrix defined later, and $\boldsymbol{k} = \boldsymbol{Q}^{-1}\boldsymbol{b}$. It can be shown that a basic iterative method can be defined in terms of a nonsingular matrix $\boldsymbol{Q}$, referred to as a "splitting" or "preconditioning" matrix. Thus, if $\boldsymbol{A}$ is represented in the form

$$\boldsymbol{A} = \boldsymbol{Q} - (\boldsymbol{Q} - \boldsymbol{A})$$

[1] This appendix is prepared by T. –Z. Mai, Department of Mathematics, University of Alabama.

[2] References cited in this appendix are listed in the References at the end of the main text.

the system $Au = b$ can be written in the form

$$Qu = (Q - A)u + b. \tag{A3}$$

If the iterative method is defined by

$$Qu^{(n+1)} = (Q - A)u^{(n)} + b \tag{A4}$$

then the basic form of the iterative method with $G = I - Q^{-1}A$, $k = Q^{-1}b$ is obtained.

An ideal choice of the splitting matrix would be $Q = A$. In this case, the basic method would converge in one iteration. However, the work involved in carrying out the single iteration would be the same as that required to solve the system $Au = b$ itself by a direct method. In practice, Q is chosen so that the solution of the system $Qx = y$ for x, given any y, can be easily carried out. For many of the standard iterative methods, Q is chosen to be a matrix of the form $Q = L\Sigma U$ where L and U are sparse matrices such that L is a lower triangular, Σ is a diagonal, and U is an upper triangular matrix. In this case, the solution of the problem $Qx = y$ can be obtained by the following steps:

a. Solve $Lz = y$ for z by forward substitution.

b. Solve $\Sigma w = z$ for w.

c. Solve $Ux = w$ for x by backward substitution.

The splitting matrices for some standard basic iterative methods, which are involved in the GCG package of MT3DMS, are listed below. Here, A is represented in the form

$$A = D - C_L - C_U \tag{A5}$$

where D is a diagonal matrix, and C_L and C_U are strictly lower and strictly upper triangular matrices, respectively.

Method	Splitting Matrix Q
Jacobi	D
Symmetric Successive-Over-Relaxation (SSOR)	$\left(\omega^{-1}D - C_L\right)\left[\left(\frac{2-\omega}{\omega}\right)D\right]^{-1}\left(\omega^{-1}D - C_U\right)$
Modified Incomplete Cholesky (MIC) (Gustafsson 1978)	$L\Sigma U$

The real number ω in the SSOR splitting matrix is an iteration parameter (relaxation factor) which is chosen to make the convergence of the basic iterative method as fast as possible. However, the optimum value of the iteration parameter is seldom known in advance; thus, an estimate is needed so that the rate of convergence is reasonable. For the case where the coefficient matrix is

symmetric and positive definite (SPD), a very good approximation of the optimum value of ω is given by

$$\omega = \frac{2}{1+\sqrt{2(1-\rho(\boldsymbol{B}))}} \tag{A6}$$

where $\rho(\boldsymbol{B})$ is the spectral radius of the matrix $\boldsymbol{B} = \boldsymbol{D}^{-1}\boldsymbol{A}$ (Young 1971). It should be noted that there is a complicated algorithm for the estimation of the optimal value of ω adaptively described in Young (1971). It should also be noted that $\rho(\boldsymbol{B}) = \cos(\pi \boldsymbol{h})$ for the Laplace equation with central finite-difference discretization with mesh size $\boldsymbol{h}$. The model considered in this package is more complicated than the Laplace equation; however, the value of ω may be estimated by this formula when the SSOR method is chosen. Fortunately, the SSOR method is not sensitive to the value of ω; thus the rate of convergence may not be slowed down by the choice of ω.

The matrices $\boldsymbol{L}$, Σ, and $\boldsymbol{U}$ are obtained from the MIC factorization so that $\boldsymbol{L}$ and $\boldsymbol{U}$ are lower and upper triangular matrices, respectively, such that $\boldsymbol{A} = \boldsymbol{L}\Sigma\boldsymbol{U} + \boldsymbol{R}$, where $\boldsymbol{R}$ is an error matrix. The procedure for the incomplete Cholesky factorization is based on the Gaussian elimination procedure; however, the sparse pattern of $\boldsymbol{A}$ must be preserved. The elements of zero value within $\boldsymbol{A}$ must not be modified. If there is a fill-in, this fill-in must be added to the diagonal element of $\boldsymbol{A}$ in the same row before the next elimination process can be continued. The algorithm is given below:

for $\boldsymbol{i}$ = 1,2,..., N, **do**

$$\boldsymbol{d}_{ii} = \boldsymbol{a}_{ii} - \sum_{\boldsymbol{k}=1}^{\boldsymbol{i}-1} \boldsymbol{l}_{jk}\boldsymbol{u}_{ki}\boldsymbol{d}_{kk}$$

for j = 1,2,...,N, do

$$d_{\mathrm{ii}} l_{\mathrm{ji}} = \begin{cases} a_{\mathrm{ji}} - \sum_{k=1}^{i-1} l_{\mathrm{jk}} u_{\mathrm{ki}} d_{\mathrm{kk}} & (j,i) \notin P \text{ and } j \geq i \\ 0 & (j,i) \in P \text{ and } j < i \end{cases}$$

$$d_{\mathrm{ii}} u_{ij} = \begin{cases} a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} d_{\mathrm{kk}} & (j,i) \notin P \text{ and } j \geq i \\ 0 & (j,i) \in P \text{ and } j < i \end{cases}$$

$$\boldsymbol{a}_{jj} = \boldsymbol{a}_{jj} - \sum_{k=1}^{i-1} \boldsymbol{l}_{jk}\boldsymbol{u}_{ki}\boldsymbol{d}_{kk} - \sum_{k=1}^{i-1} \boldsymbol{l}_{ik}\boldsymbol{u}_{kj}\boldsymbol{d}_{kk} (j,i) \in P$$

where $P = \{(i,j) \mid \boldsymbol{a}_{ij} = 0, \forall i,j = 1,2,...,N\}$ is the sparsity set of $\boldsymbol{A}$. An example is given:

Suppose that

$$A=\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}$$

The first step of Gaussian elimination is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{4} & 1 & 0 & 0 \\ \frac{1}{4} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}=\begin{bmatrix} 4 & -1 & -1 & 0 \\ 0 & \frac{15}{4} & -\frac{1}{4} & -1 \\ 0 & -\frac{1}{4} & \frac{15}{4} & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}$$

Since the zero pattern should be kept the same as A, the fill-in entries $\frac{1}{4}$ (which were 0 originally) should be discarded before proceeding to the next elimination step. However, for the MIC factorization, the fill-ins are added to the corresponding diagonal elements. Thus the matrix

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ 0 & \frac{7}{4} & 0 & -1 \\ 0 & 0 & \frac{7}{4} & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}$$

is to be carried on. Continuing with the procedure yields

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{4} & 1 & 0 & 0 \\ -\frac{1}{4} & 0 & 1 & 0 \\ 0 & -\frac{4}{7} & -\frac{4}{7} & 1 \end{bmatrix}\begin{bmatrix} 4 & -1 & -1 & 0 \\ 0 & \frac{7}{4} & 0 & -1 \\ 0 & 0 & \frac{7}{4} & -1 \\ 0 & 0 & 0 & \frac{20}{7} \end{bmatrix}=LU=A+R. \qquad \text{(A7)}$$

Acceleration Procedures

In many cases a basic iterative method can be speeded up using an acceleration procedure. There are two classes of acceleration procedures. One class is for the linear system whose coefficient matrix is SPD, such as the conjugate gradient (CG) method proposed by Hestenes and Stiefel (1952). The other class is for nonsymmetric cases. There are many algorithms in this class; however, they can all be treated as generalized conjugate gradient (GCG) methods. The GCG methods are equivalent to the CG method when the system is symmetric.

Symmetric matrix

First the CG acceleration for the symmetric case, for example the flow case, will be discussed. In the absence of round-off error, the CG method converges to the exact solution in no more than N iterations, where N is the order of the system. The iterates $u^{(n)}$ generated by the CG acceleration satisfy the following condition:

$$u^{(n)} - u^{(0)} \in K_n(\delta^{(0)}, I-G),\ n = 0,1,2,... \tag{A8}$$

where $\delta^{(n)} = Q^{-1}(b-Au^{(n)})$ is called pseudo-residual vector and $K_n(\delta^{(0)}, I-G)$ is the Krylov space of $\delta^{(0)}$ of degree n with respect to the matrix $I-G$, and is defined by

$$K_n(\delta^{(0)}, I-G) = span\{\delta^{(0)}, (I-G)\delta^{(0)}, ..., (I-G)^{n-1}\delta^{(0)}\} \tag{A9}$$

The error $\varepsilon^{(n)} = u^{(n)} - \bar{u}$ satisfies the inequality

$$\left\|\varepsilon^{(n)}\right\|_{(Z(I-G))^{1/2}} \le \left\|w-\bar{u}\right\|_{(Z(I-G))^{1/2}},\ n = 0,1,2,... \tag{A10}$$

where the matrix Z is chosen so that $Z(I-G)$ is SPD.

Let (u,v) be the inner product of the vectors u and v, the formula for the CG acceleration is given below:

Formulas for CG acceleration

$u^{(0)}$ *is arbitrary*

$\delta^{(0)} = Q^{-1}(b - Au^{(0)})$

$u^{(n+1)} = u^{(n)} + \lambda_n p^{(n)},\ n = 0,1,2,...$

$\delta^{(n+1)} = \delta^{(n)} - \lambda_n Q^{-1}Ap^{(n)},\ n = 0,1,2,...$

$\lambda_n = (\delta^{(n)}, p^{(n)})/(Q^{-1}Ap^{(n)}, Ap^{(n)}),\ n=0,1,2,...$

$p^{(n)} = \delta^{(n)} + \alpha_n p^{(n-1)},\ n \ge 1,\ \ p^{(0)} = \delta^{(0)}$

$\alpha_n = (\delta^{(n)}, Ap^{(n-1)})/(p^{(n-1)}, Ap^{(n-1)}),\ n = 0,1,2,...$ (A11)

where $p^{(n)}$ is the A-conjugate direction vectors

Nonsymmetric matrix

The case where A is neither symmetric nor positive definite will now be discussed. If A is similar to an SPD matrix, by the use of an auxiliary matrix, a GCG method can be developed. The GCG method is equivalent to the CG method if the matrix A is an SPD matrix. But, in this package of subroutines, the coefficient matrix of the linear system is derived from an advection and dispersion/diffusion partial differential equation which is in general a nonsymmetric matrix and not similar to an SPD matrix. In such a case there are several consequences: some of the eigenvalues of the iteration matrix G may be complex and/or may have real parts greater than unity; the Jordan canonical form of I-G may not be diagonal and hence there may exist principal vectors of grade greater than one which may slow down the convergence (e.g., Manteuffel 1977). It should be noted that if the eigenvalues of G are distributed over a circle in the complex plane then little or no acceleration of the convergence is possible (Hagman and Young 1981).

Young and Jea (1980) considered the idealized generalized CG acceleration procedure (IGCG acceleration) for speeding up the convergence of a nonsymmetric basic iterative method. The IGCG acceleration is defined by the following two conditions:

$$u^{(n)} - u^{(0)} \in K_n(\delta^{(0)}, I\text{-}G),\ n=1,2,3,... \tag{A12}$$

and

$$(Z\delta^{(n)}, v) = 0,\ for\ all\ v \in K_n(\delta^{(0)}, I\text{-}G). \tag{A13}$$

The first condition simply implies that there is a polynomial acceleration procedure. The second condition is called the Galerkin condition. We should point out that if $Z(I\text{-}G)$ is SPD, then the Galerkin condition is equivalent to the minimization condition:

$$||u^{(n)} - \bar{u}||_W \leq ||w - \bar{u}||_W \tag{A14}$$

for all w such that $w\text{-}u^{(0)} \in K_n(\delta^{(0)}, I\text{-}G)$, where $W = (Z(I\text{-}G))^{1/2}$.

Young and Jea (1980) considered three forms of the IGCG method including ORTHODIR, ORTHOMIN, and ORTHORES. In the general case, the computation of $u^{(n+1)}$ for all three forms of the IGCG method involves the use of information from all previous iterations. Thus the procedures often require excessive amounts of computational work and computer storage. Truncated forms of the IGCG method, ORTHODIR(s), ORTHOMIN(s), and ORTHORES(s), are often used. The truncated forms are the same as for the IGCG method except that for a given integer s only the information from the last s iteration is required. However, we need to point out that the convergence properties may not hold. The methods may actually break down.

Jea and Young (1983) simplified the IGCG method by introducing an expanded system involving the use of A^T which leads to the Lanczos method.

With a suitable choice of Z, the three forms of the IGCG acceleration (i.e., ORTHODIR, ORTHOMIN, and ORTHORES) applied to the expanded system reduce to ORTHODIR(2), ORTHOMIN(1), and ORTHORES(1), respectively. Thus these simplified procedures are referred to as Lanczos/ORTHODIR, Lanczos/ORTHOMIN, and Lanczos/ORTHORES methods. Details of these methods can be found in Jea and Young (1983). In this package, the Lanczos/ORTHOMIN acceleration procedure is used because its formulas are simpler than the others.

Formulas for Lanczos/ORTHOMIN

$u^{(0)}$ *is arbitrary*

$$\delta^{(0)} = Q^{-1}(b - Au^{(0)})$$

$$u^{(n+1)} = u^{(n)} + \lambda_n p^{(n)},\ n = 0,1,2,...$$

$$\delta^{(n+1)} = \delta^{(n)} - \lambda_n Q^{-1} A p^{(n)},\ n = 0,1,2,...$$

$$\eta^{(n+1)} = \eta^{(n)} - \lambda_n (Q^{-1}A)^T p^{(n)},\ n = 0,1,2,...$$

$$\lambda_n = (\delta^{(n)}, \eta^{(n)})/(Q^{-1}Ap^{(n)}, q^{(n)}),\ n=0,1,2,...$$

$$p^{(n)} = \delta^{(n)} + \alpha_n p^{(n-1)},\ n \geq 1,\quad p^{(0)} = \delta^{(0)}$$

$$q^{(n)} = \eta^{(n)} + \alpha_n Q^{(n-1)},\ n \geq 1,\quad q^{(0)} = \delta^{(0)}$$

$$\alpha_n = (\delta^{(n)}, \eta^{(n)})/(\delta^{(n-1)}, \eta^{(n-1)}),\ n = 0,1,2,... \qquad \text{(A15)}$$

In addition to the simple formulas, fast convergence and less computational work have been reported in some applications (Jea 1982 and Saad 1982).

Stopping Procedures

The problem of stopping the iteration procedure involves deciding whether or not the current iterate $u^{(n)}$ is a sufficiently accurate approximation to the true solution $\bar{u}$ of the linear system. If the true solution were known, then it would seem reasonable to accept the approximate solution $u^{(n)}$ if

$$\frac{\left\| u^{(n)} - \bar{u} \right\|_\alpha}{\left\| \bar{u} \right\|_\alpha} \leq \varsigma \qquad \text{(A16)}$$

Here α is a prescribed norm and ζ is a prescribed tolerance, perhaps 10^{-5}. The above stopping test is called an exact stopping test.

Since the true solution is usually not known in advance, alternative stopping tests must often be used. In a stopping test based on the use of the pseudo-residual vector $\delta^{(n)} = Q^{-1}\varepsilon^{(n)}$, where $\varepsilon^{(n)} = b - Au^{(n)}$, it follows that

$$\varepsilon^{(n)} = (G-I)^{-1}\delta^{(n)} \tag{A17}$$

For any vector norm α and any compatible matrix norm β,

$$\left\|\varepsilon^{(n)}\right\|_\alpha \le \left\|(G-I)^{-1}\right\|_\beta \left\|\delta^{(n)}\right\|_\alpha \tag{A18}$$

Therefore the exact stopping test can be replaced by

$$\frac{\left\|\varepsilon^{(n)}\right\|_\alpha}{\left\|\bar{u}\right\|_\alpha} \le \left\|(G-I)^{-1}\right\|_\beta \frac{\left\|\delta^{(n)}\right\|_\alpha}{\left\|\bar{u}\right\|_\alpha} \tag{A19}$$

If $\bar{u}$ is replaced by $u^{(n)}$, the stopping test obtained is

$$\left\|(G-I)^{-1}\right\|_\beta \frac{\left\|\delta^{(n)}\right\|_\alpha}{\left\|u^{(n)}\right\|_\alpha} \le \varsigma \tag{A20}$$

For the symmetric case, $\left\|(G-I)^{-1}\right\|_\beta$ may be replaced by $\dfrac{1}{1-M(G)}$ where $M(G)$ is the largest eigenvalue of the iteration matrix G. The 2-norm or the ∞-norm is chosen for the vector norm; thus the stopping test becomes

$$\frac{1}{1-M(G)} \frac{\left\|\delta^{(n)}\right\|_2}{\left\|u^{(n)}\right\|_2} \le \varsigma \tag{A21}$$

This stopping test is adequate for an iterative algorithm. Since $M(G)$ is usually not known, an estimate for $M(G)$ may be used, or it may be set to zero. For the nonsymmetric case, there is no rigorous stopping test. Nevertheless, the test for the nonsymmetric case is used. Since the eigenvalue of G may be complex, $M(G)$ is set to be zero.

It should be noted that the tests presented above are preferable to some other tests such as $\left\|\delta^{(n)}\right\|_2 \le \varsigma$ or $\left\|b - Au^{(n)}\right\|_2 \le \varsigma$ which are sometimes used. In this package the maximum change of the concentration field at each grid point is required. Thus it is suitable to use the infinity norm of two successive iterations:

$$\left\|u^{(n+1)} - u^{(n)}\right\|_\infty \le \varsigma \tag{A22}$$

This stopping test can fulfill the maximum change requirement without additional effort.

Description of Subroutines

The iterative solver consists of one driver routine, GCG3AP, and five supporting utility routines, MVPRD, ATVPRD, QSOLVE, QTSLVE, and MIC.

The subroutine GCG3AP performs the iterative algorithms discussed above. There is a flag that allows the selection of CG acceleration for the symmetric system or Lanczos/ORTHOMIN acceleration for the nonsymmetric system. The algorithm flowchart is given in Figure A1.

The utility subroutines only communicate with GCG3AP. They do not interact with other routines in the GCG package. The subroutine MVPRD performs the matrix-vector product. It takes a matrix A and a vector v and returns the result of Av. The subroutine ATVPRD performs the product of the transpose of an input matrix A^T and a vector v, and returns the result of A^Tv.

The subroutine QSOLVE performs $Q^{-1}y$ where $\boldsymbol{Q}$ is the preconditioning matrix in one of the forms discussed above. Since Q^{-1} is not known explicitly, the system $Qx = y$ is solved for x. This subroutine takes the preconditioning matrix $\boldsymbol{Q}$ and a vector y as input and returns the solution x. Similarly, the subroutine QTSLVE solves the system $Q^Tx = y$ for x. Finally, the subroutine MIC performs the modified incomplete Cholesky factorization to obtain the preconditioning matrix $\boldsymbol{Q}$ with the same sparsity as A.

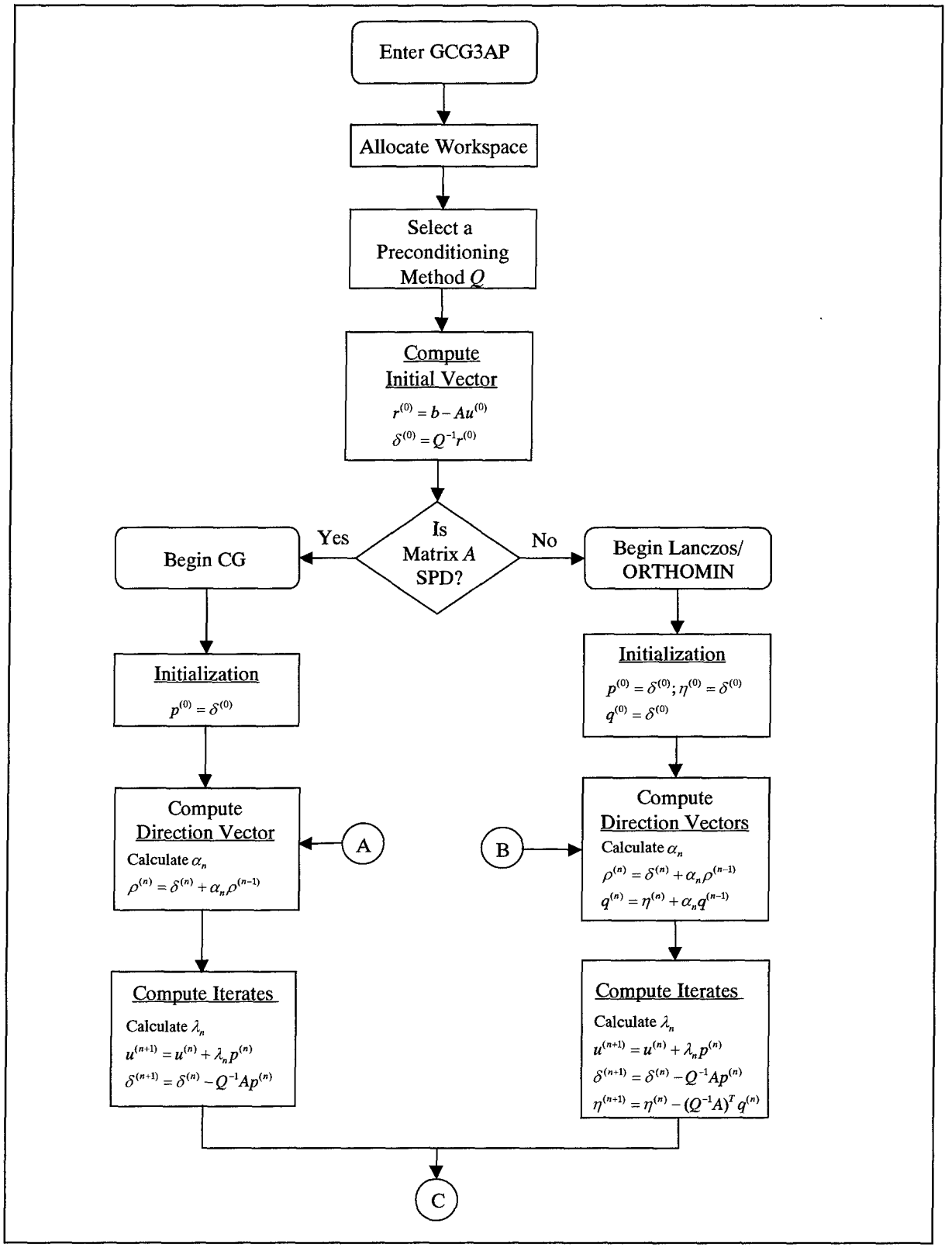


Figure A1. Flowchart for the GCG3AP module of the GCG package (Continued)

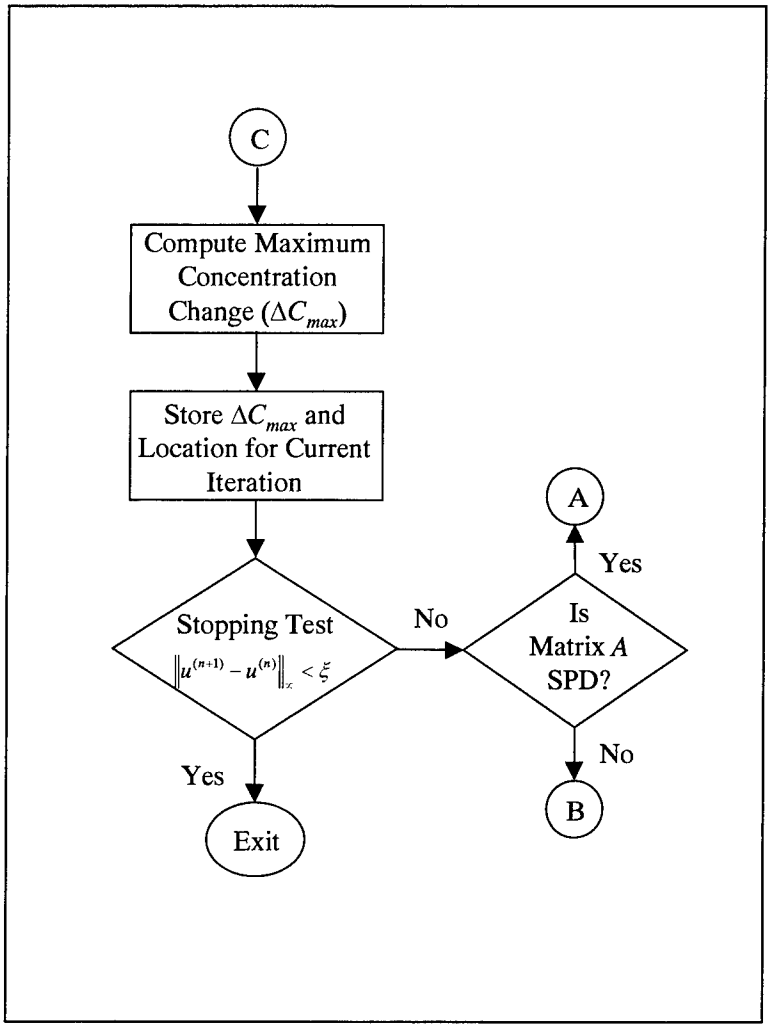


Figure A1. (Concluded)

Appendix B Computer Memory Requirements

The elements of the X and IX arrays required by each package for a given problem are always printed by the model program every time it is run. They can also be calculated according to the formulas that follow. Variables used in the formulas are defined as follows:

NCOL = Number of columns.

NROW = Number of rows.

NLAY = Number of layers.

NCOMP = Number of total species.

MCOMP = Number of mobile species.

NODES = NCOL*NROW*NLAY, the number of all nodes in the model.

NCR = NCOL*NROW, the number of nodes per model layer.

MXSS = Maximum number all point sinks and sources present in the flow model, including constant-head and general-head-dependent boundary cells, but excluding recharge and evapotranspiration cells.

MXPART = Maximum total number of moving particles allowed.

ND = Dimensions of the given problem (ND=1 for a one-dimensional problem; 2 for a two-dimensional problem and 3 for a three-dimensional problem).

MXITER = Maximum number of outer iterations for the GCG solver.

ITER1 = Maximum number of inner iterations for the GCG solver.

Basic Transport Package (BTN)

a. X Array

NODES*(5*NCOMP+10)+2*NROW+2*NCOL+NCR

b. IX Array

NODES*NCOMP+NLAY

Advection Package (ADV)

a. X Array

If one of the particle-tracking-based Eulerian-Lagrangian methods is used

(MIXELM > 0):

(ND+2)*MXPART*MCOMP

If the TVD or finite-difference method is used (MIXELM≤0):

None

b. IX Array

If one of the particle-tracking-based Eulerian-Lagrangian methods is used

(MIXELM>0):

ND*NODES*MCOMP

If the TVD or finite-difference method is used (MIXELM≤0):

None

Dispersion Package (DSP)

a. X Array

10*NODES+3*NLAY

b. IX Array

None

Sink & Source Mixing Package (SSM)

a. X Array

6*MXSS*NCOMP

If recharge is simulated, add NCR*(NCOMP+1)

If evapotranspiration is simulated, NCR*(NCOMP+1)

b. IX Array

If recharge is simulated, NCR

If evapotranspiration is simulated, add NCR

Chemical Reaction Package (RCT)

a. X Array

If sorption is simulated, NODES*(2*NCOMP+1)

If first-order rate reaction is simulated, add 2*NODES*NCOMP

b. IX Array

None

Generalized Conjugate-Gradient Solver Package (GCG)

a. X Array

8*NODES+MXITER*ITER1 plus one of the following:

(1) If the MIC preconditioner is selected (ISOLVE = 3) and the full dispersion tensor is included in the coefficient matrix (NCRS = 1): add 38*NODES

(2) If the MIC preconditioner is selected (ISOLVE = 3) and the dispersion cross terms are lumped to the right-hand side (NCRS = 0): add 14*NODES

(3) If the Jacobi or SSOR preconditioner is selected (ISOLVE = 1 or 2) and the full dispersion tensor is included in the coefficient matrix (NCRS = 1):
add 19*NODES

(4) If the Jacobi or SSOR preconditioner is selected (ISOLVE = 1 or 2) and the dispersion cross terms are lumped to the right-hand side (NCRS = 0):
add 7*NODES

b. IX Array

3*MXITER*ITER1

Appendix C
Linking MT3DMS with a Flow Model

The Flow Model Interface Package of MT3DMS reads the saturated thickness, fluxes across cell interfaces in all directions, and locations and flow rates of the various sources and sinks from an unformatted flow-transport link file saved by a flow model used in conjunction with the MT3DMS transport model. If the U.S. Geological Survey modular flow model (MODFLOW) is used for flow simulation, a package named LKMT3 (where 3 denotes the version number) for addition to MODFLOW is included with the MT3DMS distribution files. The structure and contents of the flow-transport link file, as saved by the LKMT3 package, are listed below. (Note that MT3DMS can also read the unformatted flow-transport link file saved by previous versions of the LKMT package for backward compatibility.) The MODFLOW program distributed with MT3DMS already contains the LKMT3 package. If a different version of MODFLOW is desired, the LKMT3 package must be added to the MODFLOW code as explained in Chapter 6. If it is necessary to link MT3DMS with a flow model other than MODFLOW, the flow model should be modified to save the same information with the same structure as listed below. Unless specified otherwise, each integer or real variable is assigned four bytes in the unformatted flow-transport link file.

FOR EACH SIMULATION:

F0. Record: VERISON, MTWEL, MTDRN, MTRCH, MTEVT, MTRIV, MTGHB, MTCHD, MTISS, MTNPER

HEADER (character*11)—a character string used by MT3D to identify the unformatted flow-transport link. VERSION='MT3DXXXXXXX' where XXXXXXX can be any characters.

MTWEL—an integer flag indicating whether wells (the WEL package) are included in the MODFLOW simulation:

MTWEL>0, wells are included in the MODFLOW simulation;

MTWEL=0, wells are not included in the MODFLOW simulation.

MTDRN—an integer flag indicating whether drains (the DRN package) are included in the MODFLOW simulation. The convention is the same as that for MTWEL.

MTRCH—an integer flag indicating whether recharge (the RCH package) is included in the MODFLOW simulation. The convention is the same as that for MTWEL.

MTEVT—an integer flag indicating whether evapotranspiration (the EVT package) is included in the MODFLOW simulation. The convention is the same as that for MTWEL.

MTRIV—an integer flag indicating whether rivers (the RIV package) or streams (the STR package) are included in the MODFLOW simulation. The convention is the same as that for MTWEL.

MTGHB—an integer flag indicating whether general-head-dependent boundaries (the GHB package) are included in the MODFLOW simulation. The convention is the same as that for MTWEL.

MTCHD—an integer flag indicating whether any constant-head boundary cells are included in the MODFLOW simulation. The convention is the same as that for MTWEL.

MTISS—an integer flag indicating whether the flow simulation is steady state or transient:

MTISS>0, the flow simulation is steady state;

MTISS=0, the flow simulation is transient.

MTNPER—the number of stress periods used in the flow simulation.

FOR EACH TIME-STEP OF THE FLOW SOLUTION:

F1. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL

F2. Record: THKSAT(NCOL,NROW,NLAY)

KPER—the stress period at which the cell saturated thicknesses are saved.

KSTP—the time-step at which the cell saturated thicknesses are saved.

NCOL, NROW, NLAY—numbers of columns, rows, and layers, respectively.

LABEL(character*16)—a character string equal to 'THKSAT', the identifier for the saturated thickness array.

THKSAT—Saturated thickness of unconfined cells. For inactive cells, the value must be set equal to 1.E30. For confined cells, the value must be set to –111.

(If NCOL=1, skip F3 and F4)

F3. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL

F4. Record: QX(NCOL,NROW,NLAY)

LABEL(character*16)—a character string equal to 'QXX', the identifier for the QX array.

QX—volumetric flow rates (L^3T^{-1}) between cells at cell interfaces along rows (or the x-axis). Positive in the direction of increasing J index.

(If NROW=1, skip F5 and F6)

F5. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL

F6. Record: QY(NCOL,NROW,NLAY)

LABEL(character*16)—a character string equal to 'QYY', the identifier for the QY array.

QY—volumetric flow rates (L^3T^{-1}) between cells at cell interfaces along columns (or the y-axis). Positive in the direction of increasing I index.

(If NLAY=1, skip F7 and F8)

F7. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL

F8. Record: QZ(NCOL,NROW,NLAY)

LABEL(character*16)—a character string equal to 'QZZ', the identifier for the QZ array.

QZ—volumetric flow rates (L^3T^{-1}) between cells at cell interfaces along layers (or the z-axis). Positive in the direction of increasing K index.

(If MTISS>0, skip F9 and F10)

F9. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL

F10. Record: QSTO(NCOL,NROW,NLAY)

LABEL(character*16)—a character string equal to 'STO', the identifier for the QSTO array.

QSTO—volumetric flow rates (L^3T^{-1}) released from or accumulated in transient groundwater storage. Positive for release and negative for accumulation.

F11. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL, NCNH

LABEL(character*16)—a character string equal to 'CNH', the identifier for constant-head boundaries.

NCNH—total number of constant-head boundary cells.

(If NCHN>0, there must be NCNH records of F12)

F12. Record: KCNH, ICNH, JCNH, QCNH

KCNH, ICNH, JCNH—cell indices of each constant-head boundary cell.

QCNH—volumetric net flow rate (L^3T^{-1}) out of or into each constant-head cell, including the exchange between constant-heads (this is different from the net flow rate calculated at a constant-head cell by MODFLOW). Positive if the flow is out of the constant-head cell, negative otherwise.

(If wells are not present in the flow model, skip F13 and F14)

F13. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL, NWEL

LABEL(character*16)—a character string equal to 'WEL', the identifier for wells.

NWEL—total number of wells.

(If NWEL>0, there must be NWEL records of F14)

F14. Record: KWEL, IWEL, JWEL, QWEL

KWEL, IWEL, JWEL—cell indices of each well.

QWEL—volumetric flow rate (L^3T^{-1}) of each well. Positive if the flow is into the cell, negative otherwise. (The same convention is followed by the rest of sink/source terms).

(If drains are not present in the flow model, skip F15 and F16)

F15. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL, NDRN

LABEL(character*16)—a character string equal to 'DRN', the identifier for drains.

NDRN—total number of drains.

(If NDRN>0, there must be NDRN records of F16)

F16. Record: KDRN, IDRN, JDRN, QDRN
KDRN, IDRN, JDRN—cell indices of each drain.

QDRN—volumetric flow rate (L^3T^{-1}) of each drain.

(If recharge is not present in the flow model, skip F17, F18, and F19)

F17. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL

LABEL(character*16)—a character string equal to 'RCH', the identifier for recharge.

F18. Record: IRCH(NCOL,NROW)

IRCH—layer indices of the recharge flux.

F19. Record: RECH(NCOL,NROW)

RECH—volumetric recharge rate (L^3T^{-1}).

(If evapotranspiration is not present in the flow model, skip F20, F21, and F22)

F20. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL

LABEL(character*16)—a character string equal to 'EVT', the identifier for evapotranspiration.

F21. Record: IEVT(NCOL,NROW)

IEVT—layer indices of the evapotranspiration flux.

F22. Record: EVTR(NCOL,NROW)

EVTR—volumetric evapotranspiration rate (L^3T^{-1}).

(If rivers or streams are not present in the flow model, skip F23 and F24)

F23. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL, NRIV

LABEL(character*16)—a character string equal to 'RIV', the identifier for rivers (or streams).

NRIV—total number of rivers or streams.

(If NRIV>0, there must be NRIV records of F24)

F24. Record: KRIV, IRIV, JRIV, QRIV

KRIV, IRIV, JRIVN—cell indices of each river (or stream) cell.

QRIV—volumetric flow rate (L^3T^{-1}) of each river (or stream) cell.
(If general-head-dependent boundaries are not present in the flow model, skip

F25 and F26)

F25. Record: KPER, KSTP, NCOL, NROW, NLAY, LABEL, NGHB

LABEL(character*16)—a character string equal to 'GHB', the identifier for general-head-dependent boundaries.

NGHB—total number of GHB cells.

(If NGHB>0, there must be NGHB records of F26)

F26. Record: KGHB, IGHB, JGHB, QGHB

KGHB, IGHB, JGHB—cell indices of each general-head-dependent boundary cell.

QGHB—volumetric flow rate (L^3T^{-1}) of each general-head-dependent boundary cell.

Appendix D
Postprocessing Programs

Two postprocessing programs are included with the MT3DMS distributions files: (a) PostMT3D|MODFLOW for generating graphical data files from an unformatted concentration file saved by MT3DMS (or an unformatted head/drawdown file saved by MODFLOW) and (b) SAVELAST for extracting the concentrations at the final step from an unformatted concentration file and saving them in a separate unformatted file as the starting concentrations for a continuation run. These two programs are documented in this appendix.

PostMT3D|MODFLOW (PM)

PM can be used to extract the calculated concentrations from the unformatted concentration files saved by MT3DMS and other versions of MT3D within a user-specified window along a model layer or cross section (two-dimensional (2-D)) or within a user-specified volume (three-dimensional (3-D)) at any desired time interval. The concentrations within the specified window or volume are saved in such formats that they can be used by any commercially available graphical package to generate 2-D or 3-D contour maps and other types of graphics.

To use PM, two input files are required. The first is the unformatted file saved by MT3DMS or MODFLOW after appropriate output control options have been set. The second is a text file which contains information on the spatial configuration of the model grid, referred to as the model configuration file. For output, PM generates data files either in the POINT format, where the spatial coordinates of a nodal point are saved along with the data value at the nodal point, or in the ARRAY format that is directly readable by Golden Software's (1996) 2-D contouring package Surfer©. In addition, the data files saved in the POINT format can include an optional header which is compatible with Amtec Engineering's 2-D and 3-D visualization package Tecplot©.

Unformatted concentration file

In the previous versions of MT3D, the unformatted concentration file is named MT3D.UCN by default. In MT3DMS, one unformatted concentration file is saved for each species, with the default name MT3Dnnn.UCN where nnn is

the species index number as in MT3D001.UCN, MT3D002.UCN, and so on. The unformatted concentration files are saved by setting the flag SAVUCN to T (for True) in the BTN input file. The structure and contents of the unformatted concentration files are shown below:

MT3Dnnn.UCN (unformatted):

For each transport step saved:

For each layer of the 3-D concentration matrices:

Record 1: NTRANS,KSTP,KPER,TIME2,TEXT,NCOL,NROW,ILAY

Record 2: ((CNEW(J,I,ILAY),J=1,NCOL),I=1,NROW)

where

NTRANS	is the transport step at which the concentration is saved;
KSTP	is the time-step at which the concentration is saved;
KPER	is the stress period at which the concentration is saved;
TIME2	is the total elapsed time at which the concentration is saved;
TEXT	is a character string (character*16) set equal to "CONCENTRATION";
NCOL	is the total number of columns;
NROW	is the total number of rows;
ILAY	is the layer at which the concentration is saved; and
CNEW	is the calculated concentration.

Model configuration file

The model configuration file, named MT3D.CNF by default, is saved automatically by MT3DMS along with MT3Dnnn.UCN after setting the output flag SAVUCN to T. If PM is used to process the unformatted head or drawdown file saved by MODFLOW before MT3DMS is used, the user needs to create the model configuration file manually using a text editor. The structure and contents of the model configuration file follow.

MT3D.CNF (free format):

Record 1:	NLAY, NROW, NCOL
Record 2:	(DELR(J), J=1,NCOL)
Record 3:	(DELC(I),I=1,NROW)
Record 4:	((HTOP(J,I), J=1,NCOL),I=1,NROW)
Record 5:	(((DZ(J,I,K), J=1,NCOL),I=1,NROW),K=1,NLAY)
Record 6:	CINACT

where

NLAY	is the total number of layers;
DELR	is the width of columns (along the rows or x-axis);

DELC	is the width of rows (along the columns or y-axis);
HTOP	is a 2-D array defining the top elevation of the first model layer;
DZ	is a 3-D array defining the thickness of each model cell;
CINACT	is the value used in the model for indicating inactive cells.

The values in the model configuration file are arranged in list-directed (or free) format. Therefore, each record should begin at a new line and may occupy as many lines as needed. Either blank spaces or commas can be used to separate values within a record. In addition, input by free format permits the use of a repeat count in the form *n*d* where *n* is an unsigned-nonzero integer constant, and the input *n*d* causes n consecutive values of *d* to be entered. HTOP is a 2-D array, and its values should be arranged in the order of column first, sweeping from column 1 to column NCOL along the first row; then continuing on to row 2, row 3, ..., until row NROW. DZ is a 3-D array, and its values for each layer should be arranged similarly to those for HTOP, starting from the first layer, then continuing on to layer 2, layer 3, ..., until layer NLAY. Note that if one is only interested in creating data files for certain layers in plan view, then HTOP and DZ are never used and thus may be entered as some dummy numbers with the use of repeat counts.

Running PM

PM can be run in either interactive or batch mode. To run it interactively, the user simply types the name of the executable file at the DOS prompt:

C:\> **PM**

The program will prompt the user for the various input items, and the user responds to the input requests directly from the keyboard. To run PM in batch mode, the user must write all responses in the order required by PM to a text file and then redirect PM to get responses from the response file instead of keyboard by issuing a command as follows:

C:\> **PM < response.file**

where `response.file` is the name of the text file containing all responses to PM which the user would otherwise type in from the keyboard.

The user can select the concentrations, heads, or drawdowns at a desired time by specifying either a) the numbers of transport step, time-step, and stress period, or (b) the total elapsed time, whichever is more convenient. (Note that transport step is used for MT3DMS only and is not used for MODFLOW.) The value of -1 may be entered to obtain the results at the final step stored in the unformatted file. The user can also define a 2-D window or 3-D volume within which the graphical data files are desired by specifying the **starting** and **ending** column (J), row (I), and layer (K) indices of the window.

For example, to generate a data file for a cross-sectional contour map along the 5th column, from row 20 to row 40 and from layer 1 to layer 10, the user

would enter the starting (J, I, K) indices as 5, 20, 1 and the ending (J, I, K) indices as 5, 40, 10. Similarly, to generate a data file for a cross-sectional contour map along the 5th row, from column 20 to column 40 and from layer 1 to layer 10, the user would enter the starting (J, I, K) indices as 20, 5, 1 and the ending indices as 40, 5, 10. Moreover, to generate a data file for a contour map on the 5th layer, from column 20 to column 40 and from row 1 to row 10, the user would enter the starting (J, I, K) indices as 20, 1, 5 and at the lower right corner as 40, 10, 5. Finally, to generate a 3-D data file within a volume defined from column 1 to column 40, from row 1 to row 20, and from layer 1 to layer 5, the user would enter the starting (J, I, K) indices as 1, 1, 1 and the ending indices as 40, 20, 5. It is also possible to generate a data file for a contour map on the water table, that is the cells in the uppermost active layers instead of a specific layer, for example from column 20 to column 40 and from row 1 to row 10, the user would enter the starting (J, I, K) indices as 20, 1, 0 and the ending indices as 40, 10, 0.

It should be pointed out that in MT3DMS, the origin of the internal coordinate system (O_m) is set at the upper, top, left corner of the cell in the first column, first row, and first layer (i.e., cell (1, 1, 1)) and the positive x, y and z coordinates are in the directions of increasing column, row, and layer indices, respectively (Figure D1). However, in the output files generated by PM, the origin (O) is transformed to the lower, bottom, left corner of cell in the first column, last row, and last layer (i.e., cell (1, NROW, NLAY) (Figure D1)) as is customary in most graphical packages. As a result, the y- and z-axes used in MT3DMS are reversed by PM whereas the x-axis remains the same.

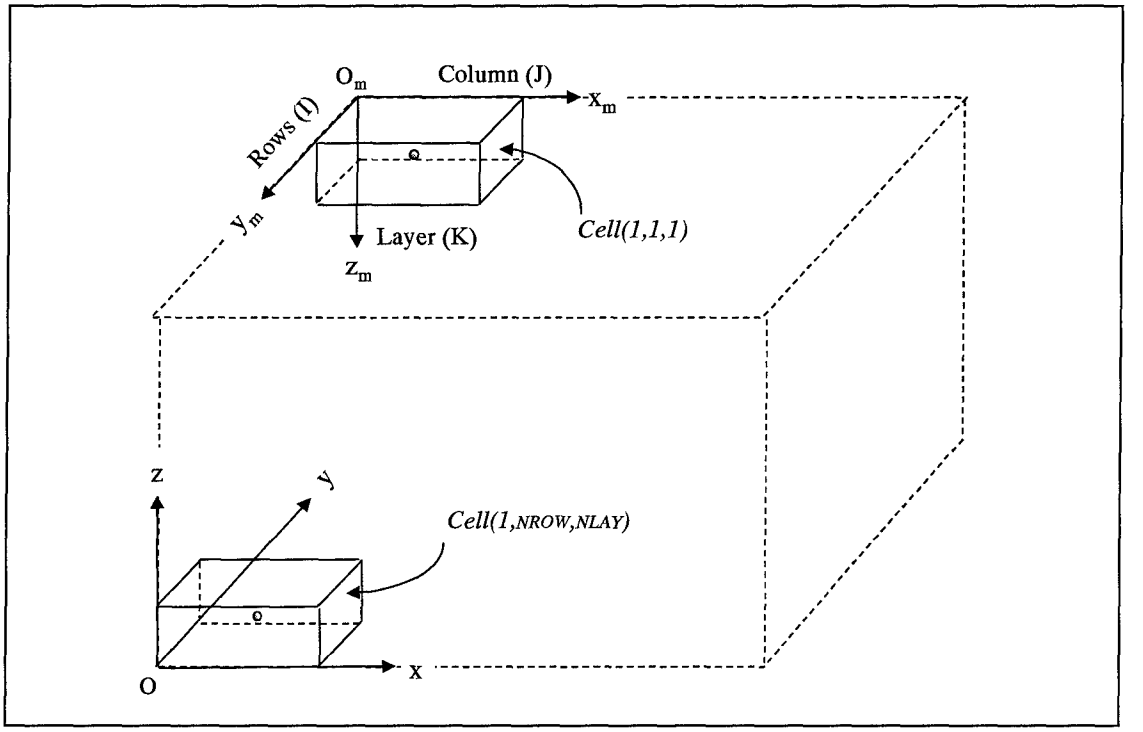


Figure D1. Transformation from the model internal coordinate system to the coordinate used by the postprocessor for plotting purposes

Therefore, if the contour map is on a layer or water table (i.e., the x-y plane), the horizontal axis of the map is along the direction of increasing column (J) indices, and the vertical axis is along the direction of decreasing row (I) indices. If the contour map is on a cross section along a row (i.e., the x-z plane), the horizontal axis of the map is along the direction of increasing column (J) indices, and the vertical axis is along the direction of decreasing layer (K) indices. If the contour map is on a cross section along a column (i.e., the y-z plane), the horizontal axis of the map is along the direction of decreasing (I) indices, and the vertical axis is along the direction of decreasing layer (K) indices. All the necessary transformations are done by PM automatically. As an option, PM also allows the user to add an offset in the x and y directions to the map's origin for the convenience of posting data points on the map.

Output files

For output, PM writes data files in one of the two formats, referred to as the ARRAY format (with the file extension .GRD) and the POINT format (with the file extension .DAT). The ARRAY format follows the convention used by Golden Software's Surfer© (1996) graphical contouring package. It saves the concentrations, heads, or drawdowns within a user-defined window of *regular* model mesh spacing to the output file, directly usable for generating contour maps by Surfer©. Note that if concentrations, heads, or drawdowns in an *irregular* portion of the model mesh are written to a ".GRD" file, no interpolation is performed, and the contour map is thus deformed. The POINT format saves concentration, head, or drawdown at each nodal point along with the nodal coordinates within the user-defined 2-D window or 3-D volume to the output file. This format is useful for generating data files of *irregular* model mesh spacing to be used by interpolation routines included in any standard contouring packages. It is also useful for generating plots of concentrations, heads, or drawdowns versus distances along a column, row, or layer at a selected time. (Note that the plots of concentrations versus times at a selected node can be generated from the observation files, MT3Dnnn.OBS, saved by MT3DMS). An optional header can be added to a ".DAT" file so that the file can be used directly by Amtec Engineering's 2-D and 3-D visualization package Tecplot©. The ARRAY and POINT formats are presented here for reference:

a. ARRAY(Surfer© GRD) file format (free format).

DSAA
NX, NY, XMIN, XMAX, YMIN, YMAX, CMIN, CMAX
CWIN(NX,NY)

where

DSAA is the character keyword required by Surfer©.

NX is the number of nodal points in the horizontal direction of the window;

NY is the number of nodal points in the vertical direction of the window;

XMIN is the minimum nodal coordinate in the horizontal direction of the window;

XMAX is the maximum nodal coordinate in the horizontal direction of the window;

YMIN is the minimum nodal coordinate in the vertical direction of the window;

YMAX is the maximum nodal coordinate in the vertical direction of the window;

CMIN is the minimum concentration value within the window;

CMAX is the maximum concentration value within the window; and

CWIN is a 2-D array containing all the calculated concentrations within the window.

b. POINT (DAT) file format without header (free format).

For each active cell inside the specified 2-D window or 3-D volume:

X, Y, Z, CXYZ

where

X is the nodal coordinate in the x-axis (along the rows);

Y is the nodal coordinate in the y-axis (along the columns);

Z is the nodal coordinate in the z-axis (along the layers); and

CXYZ is the calculated concentration at the nodal point defined by (X,Y,Z).

Note that for data files defined in a 2-D window, one of the coordinates will be constant. In using an interpolation routine for gridding purposes, the user should be sure to specify appropriate columns. For example, to create a contour map in a x-z cross section, the y column should either be deleted or skipped.

c. POINT (DAT) file format with header (free format).

This file format is identical to type 2 except for the addition of a Tecplot©-compatible header consisting of the following information:

VARIABLES="X" "Y" "Z" "DATA"

ZONE I=NX **J**=NY **K**=NZ **F=POINT**

where

VARIABLES, ZONE, I, J, K, F, and **POINT** are keywords used by Tecplot©;

NX is the number of columns within the user-specified 2-D window or 3-D volume;

NY is the number of rows within the user-specified 2-D window or 3-D volume;

NZ is the number of layers within the user-specified 2-D window or 3-D volume;

X, Y, Z, and DATA are character labels for the four data columns saved in the file.

SAVELAST

If a continuation run as described in Chapter 6 is desired, the concentrations from the final step of the preceding run can be used as the starting concentrations for the continuation run. The concentrations for species #nnn are saved in the default unformatted concentration file, MT3Dnnn.UCN, which is directly readable by the array reader RARRAY. If there is more than one step of concentration saved in MT3Dnnn.UCN, then SAVELAST can be used to extract the concentrations of the final step and save them in a separate unformatted file. To run it, the user simply types SAVELAST at the DOS command prompt and enter the names of input and output files from the monitor screen.

Appendix E
Abbreviated Input Instructions

Tables E1 through E6 present abbreviated input instructions for the Basic Transport Package, the Advection Package, the Dispersion Package, the Sink and Source Mixing Package, the Chemical Reaction Package, and the Generalized Conjugate Gradient Solver Package, respectively.

Table E1
Basic Transport Package

ID	Variable Name	Format	Explanation
A1	HEADING(1)	A80	First line of title for simulation run
A2	HEADING(2)	A80	Second line of the title
A3	NLAY, NROW, NCOL, NPER, **NCOMP, MCOMP**	6I10	Number of layers, rows, columns, stress periods, **total, and mobile species**
A4	TUNIT, LUNIT, MUNIT	3A4	Unit names for time, length, and mass
A5	TRNOP(10)	10L2	Transport and solver options (ADV, DSP, SSM, RCT, GCG)
A6	LAYCON (NLAY)	40I2	Model layer type code (=0 confined; >0 unconfined or convertible)
A7	DELR (NCOL)	RARRAY	Cell width along rows (or x-axis)
A8	DELC (NROW)	RARRAY	Cell width along columns (or y-axis)
A9	HTOP (NCOL, NROW)	RARRAY	Top elevation of the first model layer
A10	DZ (NCOL, NROW) (One array for each layer)	RARRAY	Cell thickness
A11	PRSITY (NCOL, NROW) (One array for each layer)	RARRAY	Porosity **(or "mobile" porosity if the dual-domain model is used)**
A12	ICBUND (NCOL, NROW) (One array for each layer)	IARRAY	Concentration boundary indicator array shared by all species
(Enter A13 for each species)			
A13	SCONC (NCOL, NROW) (One array for each layer)	RARRAY	Starting concentration
A14	CINACT, **THKMIN [THKMIN = 0.01 if not specified]**	2F10.0	Value indicating inactive cells; **minimum saturated thickness relative to total cell thickness below which the cell is considered inactive**
A15	IFMTCN, IFMTNP, IFMTRF, IFMTDP, SAVUCN	4I10, L10	Output control options
A16	NPRS	I10	Interval for printing or saving simulation results
(Enter A17 if NPRS > 0)			
A17	TIMPRS (NPRS)	8F10.0	Total elapsed times at which simulation results should be printed or saved
A18	NOBS, NPROBS	2I10	Number of observation points and output frequency
(Enter A19 NOBS times if NOBS > 0)			
A19	KOBS, IOBS, JOBS	3I10	Location of observation points (layer, row, column)
A20	CHKMAS, NPRMAS	L10, I10	Flag for saving mass budget summary file and output frequency
(Repeat A21 through A23 for each stress period)			
A21	PERLEN, NSTP, TSMULT	F10.0, I10, F10.0	Length of stress period, number of flow time-steps and time-step multiplier
(Enter A22 if TSMULT < or = 0)			
A22	TSLNGH (NSTP)	8F10.0	Length of time-steps if geometric progression of time-steps is not used in flow model
A23	DT0, MXSTRN, **TTSMULT, TTSMAX**	F10.0, I10, 2F10.0	Transport step size; maximum transport steps allowed in one flow time-step; **transport step multiplier and maximum transport step size (the last two only applicable to the fully implicit finite-difference method)**

Note: The new features introduced in MT3DMS are shown in bold type.

Table E2
Advection Package

ID	Variable Name	Format	Explanation
B1	MIXELM, PERCEL, MXPART, **NADVFD**	I10, F10.0, 2I10	Advection solution option; number of cells that advection is allowed to move in one transport step (Courant number); maximum number of moving particles allowed; **weighting scheme for the finite-difference method** MIXELM = 0, finite-difference = 1, MOC = 2, MMOC = 3, HMOC **= -1, 3rd-order TVD (ULTIMATE)**
(Enter B2 if MIXELM = 1, 2, or 3)			
B2	ITRACK, WD	I10, F10	Particle tracking option, concentration weighting factor
(Enter B3 if MIXELM=1 or 3)			
B3	DCEPS, NPLANE, NPL, NPH, NPMIN, NPMAX	F10.0, 5I10	Particle distribution control parameters for the MOC scheme
(Enter B4 if MIXELM=2 or 3)			
B4	INTERP, NLSINK, NPSINK	3I10	Solution flags for the MMOC scheme
(Enter B5 if MIXELM=3)			
B5	DCHMOC	F10.0	Critical cell concentration gradient for controlling the selective use of the MOC or MMOC scheme: MOC is used if DCCELL > DCHMOC MMOC is used if DCCELL < or = DCHMOC
Note: The new features introduced in MT3DMS are shown in bold type.			

Table E3
Dispersion Package

ID	Variable Name	Format	Explanation
C1	AL (NCOL, NROW) (one array for each layer)	RARRAY	Longitudinal dispersivity
C2	TRPT (NLAY) (one array for all layers)	RARRAY	Ratio of horizontal transverse dispersivity to longitudinal dispersivity
C3	TRPV (NLAY) (one array for all layers)	RARRAY	Ratio of vertical transverse dispersivity to longitudinal dispersivity
C4	DMCOEF (NLAY) (one array for all layers)	RARRAY	Effective molecular diffusion coefficient

Table E4
Sink and Source Mixing Package

ID	Variable Name	Format	Explanation
D1	FWEL, FDRN, FRCH, FEVT, FRIV, FGHB, X, X, X, X	10L2	Flags for well, drain, recharge, evapotranspiration, river (or stream), general-head-dependent boundary, respectively. (X indicates the flags reserved for additional sink/source options)
D2	MXSS	I10	Maximum number of all point sources and sinks in the flow model
(Repeat D3 through D8 for each stress period)			
(Enter D3 if FRCH = T)			
D3	INCRCH	I10	Flag indicating whether concentration of recharge flux should be read
(Enter D4 **for each species** if FRCH = T and INCRCH > or = 0)			
D4	CRCH (NCOL, NROW)	RARRAY	Concentration of recharge flux
(Enter D5 if FEVT = T)			
D5	INCEVT	I10	Flag indicating whether concentration of evapotranspiration fluxes should be read
(Enter D6 **for each species** if FEVT = T and INCEVT > or = 0)			
D6	CEVT (NCOL, NROW)	RARRAY	Concentration of evapotranspiration flux
D7	NSS	I10	Number of point sources of specified concentrations
(Enter D8 NSS times if NSS > 0)			
D8	KSS, ISS, JSS, **CSS, ITYPE, (CSSMS(n), n = 1,NCOMP)**	3I10, F10.0, I10, (free)	Layer, row, column, concentration, and type of point sources which are of specific concentrations, and **concentrations of all species** ITYPE = 1, constant-head cell = 2, well = 3, drain = 4, river (or stream) = 5, general-head-dependent boundary cell **= 15, mass-loading** **= -1, time-varying constant-concentration cell**
Note: The new features introduced in MT3DMS are shown in bold type.			

Table E5
Chemical Reaction Package

ID	Variable Name	Format	Explanation
E1	ISOTHM, IREACT, **IRCTOP, IGETSC**	4I10	Flags indicating type of sorption and first-order rate reactions; **input option for reaction package data arrays; flag for initial condition of nonequilibrium sorbed/immobile phase** ISOTHM = 1, Linear = 2, Freundlich = 3, Langmuir **= 4, Nonequilibrium** **= 5, dual-domain mass transfer (without sorption)** **= 6, dual-domain mass transfer (with sorption)** = 0, None IREACT = 1, Decay or biodegradation = 0, None Note: set IRCTOP = 2 to assign all reaction coefficients cell-by-cell, i.e., one array for each layer; set IRCTOP = 0 or 1 to assign all reaction coefficients layer-by-layer, i.e., one array for all layers (this is compatible with all previous versions of MT3D except MT3D'96).
(Enter E2A if ISOTHM = 1, 2, 3, 4, or 6; but not 5)			
E2A	RHOB (NCOL, NROW) (one array for each layer)	RARRAY	Bulk density of the porous medium
(Enter E2B is ISOTHM = 5 or 6)			
E2B	**PRSITY2 (NCOL, NROW) (one array for each layer)**	**RARRAY**	**Porosity of the immobile domain if a dual-domain system is simulated)**
(Enter E2C for each species if IGETSC>0)			
E2C	**SRCONC (NCOL, NROW) (one array for each layer)**	**RARRAY**	**Initial concentration for the sorbed or immobile liquid phase (the sorbed phase is assumed to be in equilibrium with the dissolved phase and the concentration is zero in the immobile domain if SRCONC is not specified)**
(Enter E3 and E4 **for each species**)			
E3	SP1 (NCOL, NROW) (one array for each layer)	RARRAY	First sorption constant (equilibrium constant)
E4	SP2 (NCOL, NROW) (one array for each layer)	RARRAY	Second sorption constant **or dual-domain mass transfer rate**
(Enter E5 and E6 **for each species** if IREACT > 0)			
E5	RC1(NCOL, NROW) (one array for each layer)	RARRAY	First-order rate reaction constant for the dissolved phase
E6	RC2(NCOL, NROW) (one array for each layer)	RARRAY	First-order rate reaction constant for the sorbed phase
Note: The new features introduced in MT3DMS are shown in bold type.			

Table E6
Generalized Conjugate Gradient Solver Package (NEW)

ID	Variable Name	Format	Explanation
F1	MXITER, ITER1, ISOLVE, NCRS	Free	Maximum number of outer iterations; maximum of inner iterations; type of preconditioners used with Lanczos/ORTHOMIN acceleration, flag for treatment of dispersion cross terms ISOLVE = 1, JACOBI = 2, SSOR = 3, MIC NCRS = 0, dispersion cross terms lumped to the right-hand side = 1, full dispersion tensor
F2	ACCL, CCLOSE, IPRGCG	Free	Relaxation factor for the SSOR preconditioner; convergence criterion; interval for printing maximum concentration changes to the output file

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1999	Final report

4. TITLE AND SUBTITLE
MT3DMS: A Modular Three-Dimensional Multispecies Transport Model for Simulation of Advection, Dispersion, and Chemical Reactions of Contaminants in Groundwater Systems; Documentation and User's Guide

5. FUNDING NUMBERS
Work Unit No. CU-1062

6. AUTHOR(S)
Chunmiao Zheng, P. Patrick Wang

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Geological Sciences
University of Alabama
Tuscaloosa, AL 35487

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Corps of Engineers, Washington, DC 20314-1000;
U.S. Army Engineer Research and Development Center, Environmental Laboratory
3909 Halls Ferry Road, Vicksbug, MS 39180-6199

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
Contract Report SERDP-99-1

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

This manual describes the next generation of the modular three-dimensional transport model, MT3D, with significantly expanded capabilities, including the addition of (a) a third-order total-variation-diminishing (TVD) scheme for solving the advection term that is mass conservative but does not introduce excessive numerical dispersion and artificial oscillation, (b) an efficient iterative solver based on generalized conjugate gradient methods and the Lanczos/ORTHOMIN acceleration scheme to remove stability constraints on the transport time-step size, (c) options for accommodating nonequilibrium sorption and dual-domain advection-diffusion mass transport, and (d) a multicomponent program structure that can accommodate add-on reaction packages for modeling general biological and geochemical reactions.

The new modular multispecies transport model, referred to as MT3DMS in this manual, is unique in that it includes the three major classes of transport solution techniques (i.e., the standard finite-difference method, the particle-tracking-based Eulerian-Lagrangian methods, and the higher-order finite-volume TVD method) in a single code. Since no single numerical technique has been effective for all transport conditions, the combination of these solution techniques, each having its own strengths and limitations, is believed to offer the best approach for solving the most wide-ranging transport problems with desired efficiency and accuracy.

(Continued)

14. SUBJECT TERMS
Groundwater MT3DMS
Model Transport

15. NUMBER OF PAGES
221

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED		

13. (Concluded).

MT3DMS can be used to simulate changes in concentrations of miscible contaminants in groundwater considering advection, dispersion, diffusion, and some basic chemical reactions, with various types of boundary conditions and external sources or sinks. The basic chemical reactions included in the model are equilibrium-controlled or rate-limited linear or nonlinear sorption and first-order irreversible or reversible kinetic reactions. More sophisticated multispecies chemical reactions can be simulated by add-on reaction packages. MT3DMS can accommodate very general spatial discretization schemes and transport boundary conditions, including: (a) confined, unconfined, or variably confined/unconfined aquifer layers, (b) inclined model layers and variable cell thickness within the same layer, (c) specified concentration or mass flux boundaries, and (d) the solute transport effects of external hydraulic sources and sinks such as wells, drains, rivers, areal recharge, and evapotranspiration. MT3DMS is designed for use with any block-centered finite-difference flow model, such as the U.S. Geological Survey modular finite-difference groundwater flow model, MODFLOW, under the assumption of constant fluid density and full saturation. However, MT3DMS can also be coupled with a variably saturated or density-dependent flow model for simulation of transport under such conditions.