USGS
science for a changing world

Volume-Weighted Particle-Tracking Method for Solute-Transport Modeling: Implementation in MODFLOW–GWT

Chapter 58 of
Section A, Groundwater
Book 6, Modeling Techniques

Techniques and Methods 6–A58

U.S. Department of the Interior
U.S. Geological Survey

Volume-Weighted Particle-Tracking Method for Solute-Transport Modeling: Implementation in MODFLOW–GWT

By Richard B. Winston, Leonard F. Konikow, and George Z. Hornberger

Chapter 58 of
Section A, Groundwater
Book 6, Modeling Techniques

Techniques and Methods 6–A58

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
RYAN K. ZINKE, Secretary

U.S. Geological Survey
William H. Werkheiser, Deputy Director
exercising the authority of the Director

U.S. Geological Survey, Reston, Virginia: 2018

Suggested citation:
Winston, R.B., Konikow, L.F., and Hornberger, G.Z., 2018, Volume-weighted particle-tracking method for solute-transport modeling; Implementation in MODFLOW–GWT: U.S. Geological Survey Techniques and Methods, book 6, chap. A58, 44 p., https://doi.org/10.3133/tm6A58.

ISSN 2328-7055 (online)

Contents

Figures

Tables

Conversion Factors

Multiply	By	To obtain
	Length	
foot (ft)	0.3048	meter (m)
	Area	
square foot (ft^2)	0.09290	square meter (m^2)
	Volume	
cubic foot (ft^3)	0.02832	cubic meter (m^3)
	Flow rate	
foot per day (ft/d)	0.3048	meter per day (m/d)
cubic foot per second (ft^3/s)	0.02832	cubic meter per second (m^3/s)
cubic foot per day (ft^3/d)	0.02832	cubic meter per day (m^3/d)
	Hydraulic conductivity	
foot per day (ft/d)	0.3048	meter per day (m/d)

Volume-Weighted Particle-Tracking Method for Solute-Transport Modeling: Implementation in MODFLOW–GWT

By Richard B. Winston, Leonard F. Konikow, and George Z. Hornberger

Abstract

In the traditional method of characteristics for groundwater solute-transport models, advective transport is represented by moving particles that track concentration. This approach can lead to global mass-balance problems because in models of aquifers having complex boundary conditions and heterogeneous properties, particles can originate in cells having different pore volumes and (or) be introduced (or removed) at cells representing fluid sources (or sinks) of varying strengths. Use of volume-weighted particles means that each particle tracks solute mass. In source or sink cells, the changes in particle weights will match the volume of water added or removed through external fluxes. This enables the new method to conserve mass in source or sink cells as well as globally. This approach also leads to potential efficiencies by allowing the number of particles per cell to vary spatially—using more particles where concentration gradients are high and fewer where gradients are low. The approach also eliminates the need for the model user to have to distinguish between "weak" and "strong" fluid source (or sink) cells. The new model determines whether solute mass added by fluid sources in a cell should be represented by (1) new particles having weights representing appropriate fractions of the volume of water added by the source, or (2) distributing the solute mass added over all particles already in the source cell. The first option is more appropriate for the condition of a strong source; the latter option is more appropriate for a weak source. At sinks, decisions whether or not to remove a particle are replaced by a reduction in particle weight in proportion to the volume of water removed. A number of test cases demonstrate that the new method works well and conserves mass. The method is incorporated into a new version of the U.S. Geological Survey's MODFLOW–GWT solute-transport model.

Introduction

The modular finite-difference groundwater flow model (MODFLOW) developed by the U.S. Geological Survey (USGS) is a widely used and flexible computer program for simulating flow in three-dimensional groundwater systems (McDonald and Harbaugh, 1988; Harbaugh and McDonald, 1996a; Harbaugh and others, 2000). MODFLOW–GWT is a solute-transport program that is integrated with MODFLOW and has the capability to calculate changes in concentration of a single solute subject to the processes of advection, hydrodynamic dispersion (which includes diffusion), fluid sources, decay, and retardation (Konikow and others, 1996; Kipp and others, 1998; Goode, 1999; Heberton and others, 2000). MODFLOW–GWT solves the solute-transport equation in three dimensions. One of its primary solution options uses the method of characteristics (MOC), which evolved from an earlier two-dimensional model by Konikow and Bredehoeft (1978). The MOC uses forward particle tracking to represent advection, coupled with either an explicit or implicit finite-difference method to calculate dispersive flux. This approach is optimal for advection-dominated systems, which are typical of many field problems involving groundwater contamination, as it minimizes numerical dispersion by representing advection by moving particles rather than by using a finite-difference approximation (see Konikow and Bredehoeft, 1978; Bredehoeft, 1971; Pinder and Cooper, 1970). The model assumes that fluid properties are homogeneous and independent of concentration. The MOC solution technique as implemented, however, does not guarantee a global mass balance, and this has been a major criticism of the approach.

The standard MOC approach has been modified to use weighted particles, wherein the particle weights represent fluid volumes. Thus, each particle actually tracks a solute mass as the product of its concentration and its weight (volume). The use of volume-weighted particles was applied in the MOCDENSE model (Sanford and Konikow, 1985), but only applied in source or sink cells to more accurately account for variable inflows and (or) outflows. Similarly, the MT3DMS model (Zheng and Wang, 1999) included a partial implementation of volume-weighted particles in their MOC procedure. Neither of these models, however, accounted for (or varied the weights in accordance with) spatially varying pore volumes between cells, which arise from cell-to-cell differences in saturated thickness or effective porosity. This report documents a full and comprehensive implementation of the volume-weighting approach to the method of characteristics. By assuring that the initial fluid volumes and solute mass in storage in each cell is matched exactly by corresponding weights and masses on

particles, and that all inflows and outflows of fluid volumes and mass over time are matched exactly by corresponding changes in weights and mass on particles, a global mass balance for the solution will be assured at all times. This can substantially improve the overall accuracy of the method relative to the original MOC solution. The use of volume-weighted particles also facilitates improved flexibility in adjusting the spatial density of particles in that a larger number of particles per cell can be initialized in more critical areas where greater numerical accuracy and precision are desired, and few particles used in areas of lesser concern or interest. This allows the model user to control and optimize the tradeoff between computational time and numerical accuracy for the model.

Purpose and Scope

The purpose of this report is to describe the improved conceptual model that uses volume-weighted particles in the MOC and provide guidance on its implementation and application. The report describes its numerical implementation, documents the testing and evaluation of the new method, and provides detailed instructions for its use. Both the strengths and weaknesses of the new method are illuminated.

Governing Equations

Groundwater Flow and Velocity

The equations governing groundwater flow and interstitial velocity are those used in MODFLOW–GWT (Konikow and others, 1996) and are not repeated here. Details related specifically to the solution of the flow equation are described by McDonald and Harbaugh (1988), Harbaugh and McDonald (1996a), Harbaugh and others (2000), and Harbaugh (2005). Solution to the flow equation provides the hydraulic gradients used to compute the interstitial velocity field, which couples the solute-transport equation to the groundwater flow equation.

Solute Transport

The basic form of the solute-transport equation solved in MODFLOW–GWT is that presented by Konikow and others (1996; eq. 16), as shown below:

$$\frac{\partial C}{\partial t}+\frac{V_i}{R_f}\frac{\partial C}{\partial x_i}-\frac{1}{\varepsilon R_f}\frac{\partial}{\partial x_i}\left(\varepsilon D_{ij}\frac{\partial C}{\partial x_j}\right)-\frac{\sum\left[W\left(C'-C\right)\right]}{\varepsilon R_f}+\lambda C=0 \quad (1)$$

where

C is volumetric concentration (mass of solute per unit volume of fluid, ML^{-3});
ε is the effective porosity (dimensionless);
V is a vector of interstitial fluid velocity components (LT^{-1});
D is a second-rank tensor of dispersion coefficients (L^2T^{-1});
W is a volumetric fluid sink ($W<0$) or fluid source ($W>0$) rate per unit volume of aquifer (T^{-1});
C' is the volumetric concentration in the sink/source fluid (ML^{-3});
λ is the decay rate (T^{-1});
t is time (T);
R_f is the retardation coefficient (dimensionless); and
x_i are the Cartesian coordinates (L).

As explained in more detail by Konikow and others (1996), the retardation factor represents a simplified conceptualization of the sorption process for the case of reversible, instantaneous equilibrium sorption governed by a linear isotherm. It can be related to sorption parameters by

$$R_f=1+\frac{\rho_b K_d}{\varepsilon} \quad (2)$$

where

ρ_b is the bulk density of the aquifer material (mass of solids per unit volume of aquifer, ML^{-3}); and
K_d is the sorption coefficient, or distribution coefficient, which is constant in time (L^3M^{-1}).

The third term in equation 1 represents the change in concentration due to hydrodynamic dispersion. This Fickian model assumes that the driving force is the concentration gradient and that the dispersive flux occurs in a direction from higher concentrations towards lower concentrations. The dispersion coefficient is related to the velocity of groundwater flow and to the nature of the aquifer using Scheidegger's (1961) equation, as shown below:

$$D_{ij}=\alpha_{ijmn}\frac{V_m V_n}{|V|} \quad (3)$$

where

α_{ijmn} is a component of the dispersivity tensor (L), a fourth rank tensor (see Scheidegger, 1961), and i,j,m,n=1,2,3;
V_m and V_n are components of the velocity vector in the m and n directions, respectively (LT^{-1}); and
$|V|$ is the magnitude of velocity, where

$$|V|=\sqrt{V_x^2+V_y^2+V_z^2}\ . \quad (4)$$

Scheidegger (1961) further shows that for an isotropic aquifer, the dispersivity tensor can be defined in terms of the longitudinal and transverse dispersivities, α_L and α_T. These parameters are specified by the user in the model input datasets, and the model will calculate the dispersion coefficients accordingly.

The solute-transport model can also consider more complex processes such as double-porosity media (matrix diffusion); direct age simulation; and more flexible retardation, decay, and zero-order growth reactions, as described by Goode (1999). The new volume-weighted particle method maintains compatibility with the processes and effects considered by Goode (1999), who provides detailed instructions for implementing them.

Numerical Methods

The basic numerical approach used in this new volume-weighted particle version of MOC (abbreviated as "MOCWT") is similar to that used in previous implementations of MOC. This report describes the major changes and differences developed and applied in the MOCWT approach. The reader is referred to Konikow and others (1996), Kipp and others (1998), Goode (1999), and Konikow and Hornberger (2003) for more details on the basic MOC numerical approach. The notation and conventions used in this report and in the computer code to describe the grid and to number the nodes are described in detail by Konikow and others (1996) and Konikow and Hornberger (2003). The indexing notation is consistent with that used in the MODFLOW–2000 documentation (Harbaugh and others, 2000).

Solute transport is simulated in the MOC algorithms of GWT by tracking a large number of representative particles (or water parcels) moving through a groundwater flow system at a speed and direction determined by the velocity at the location of each particle. The velocity is determined from the heads (and head gradients) calculated by MODFLOW. Each particle has a concentration associated with it, and numerically moving particles thereby simulate the advection process. Also, particle concentrations are adjusted during each time increment for any effects of dispersion, dilution, and reactions using either an explicit finite-difference approach (Konikow and others, 1996) or an implicit finite-difference approach (Kipp and others, 1998). In the standard MOC algorithms, the concentration for each node (or cell) of the model grid is then computed as the average of the concentrations of all particles in that cell at the end of each time increment, as shown in the following equation:

$$C_{j,i,k} = \frac{\sum_{n=1}^{N} C_n}{N} \tag{5}$$

where

$C_{j,I,k}$ is the average concentration in cell j,i,k;

C_n is the concentration of the nth particle in the cell; and

N is the total number of particles in the cell.

If a sufficiently large number of representative particles are tracked, then the method yields an accurate estimate of the change in concentration in each cell during each time increment.

In this new volume-weighted particle version of MOC (abbreviated as "MOCWT"), both a concentration and weight are assigned to each particle. The weight represents the volume of fluid associated with the particle; hence, each particle effectively tracks solute mass. The method will maintain an accurate global mass balance for the solute if (1) all solute mass initially present in the aquifer is accurately represented on particles, (2) all mass entering the system at fluid sources and by reactions is accurately added to particles, and (3) all mass leaving the system at fluid sinks or by reactions is accurately removed from particles. Dispersion calculations by finite-difference methods are mass conservative; therefore, they should not create any mass imbalance. In MOCWT algorithms, the average concentration in a cell at the end of a time increment is computed as

$$C_{j,i,k} = \frac{\sum_{n=1}^{N} (C_n \omega_n)}{\sum_{n=1}^{N} \omega_n} \tag{6}$$

where ω_n is the weight associated with particle n. Because ω_n corresponds to an appropriate fluid volume, the cell concentration is being calculated as solute mass divided by fluid volume.

Initial Conditions

The weights assigned to particles at the start of the simulation depend on the initial pore volume of water in a given cell and the initial number of particles assigned to that cell, such that for every cell

$$\sum_{n=1}^{N} \omega_n = \overline{V}_{j,i,k} \tag{7}$$

where

$\overline{V}_{j,i,k}$ is the pore volume of cell j,i,k; and

$$\overline{V}_{j,i,k} = (\varepsilon b \Delta x \Delta y)_{j,i,k} \tag{8}$$

where Δx and Δy are the grid dimensions in the x- and y- directions, respectively, for cell j,i,k, and b is the saturated thickness of the cell. The initial point density (or number of particles placed in each cell) can vary from cell to cell, as specified by the user. Thus, the weight of every particle initially placed in a cell would be set equal to

$$\omega_n = \overline{V}_{j,i,k} / N \ . \tag{9}$$

The need and value for using volume-weighted particles in setting up initial conditions is illustrated with a simple example in figure 1. An aquifer having uniform properties and cell pore volumes is represented in the left half (1*A* and 1*B*) of the figure. There is steady flow from left to right. Cell *j* and cell *j*+1 both have pore volumes equal to 1.0 (in arbitrary units). Cell *j* has an initial concentration of 1.0, and elsewhere the initial concentration equals 0 (fig. 1*A*). Thus, the initial mass in cell *j* equals 1.0, and each of four initial particles (shown in red) would have a concentration of 1.0. In cell *j*+1, the four initial particles (shown in black) would have a concentration of 0.0. For simplicity, consider a case of advective transport only, with no dispersion. After one time increment, particles have advected to the right one-half cell distance (fig. 1*B*). Then in both cells *j* and *j*+1, the average cell concentration is 0.5 and the total solute mass in both cells equals 0.5. However, if we consider a case in which pore volume varies spatially (figs. 1*C* and 1*D*), it is evident that the use of nonweighted particles can lead to a mass balance error. The same initial conditions are represented in figure 1*C*, noting that the cell pore volume increases to the right. (The pore volume might increase in this manner if the porosity increased from 0.10 to 0.11, or if the saturated thickness increased from 10.0 to 11.0, or some other combination of porosity and thickness changes yielding a 10-percent increase in pore volume.) After one time increment, two particles initially in cell *j* have advected into cell *j*+1. Thus, with nonweighted particles, the average cell concentration in cell *j*+1 would be computed as *C*=0.5; however, because the cell volume is 1.1, the equivalent solute mass is 0.55. With solute mass present only in cells *j* and *j*+1, the total mass in the system is now computed to be 1.05, rather than the actual 1.0, creating a total mass-balance error of 5 percent.

The relative advantage of using volume-weighted particles is illustrated in figure 2, which also shows the case of spatially varying pore volumes of cells. The cell pore volume increases from 0.9 to 1.1 moving from left to right (from cell *j*-1 to *j*+1). Initially, there is a mass of 1.0 in the system, all contained in cell *j* (fig. 2*A*). After one time increment, all particles have been advected approximately one-half cell distance to the right (fig. 2*B*). Then after using the weighted averaging method to compute cell concentrations, the resulting mass balance is accurate.

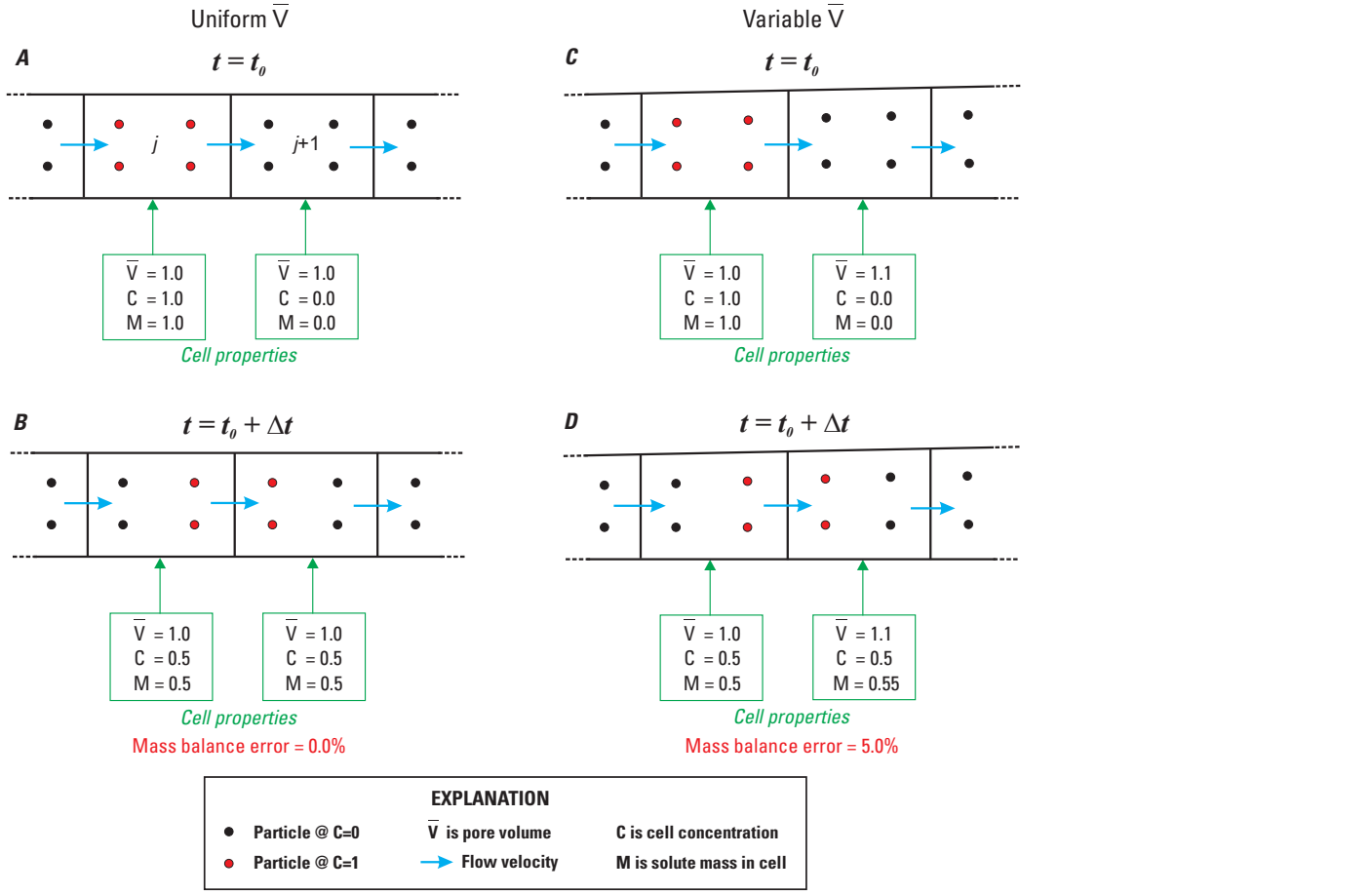

Figure 1. Simplified example illustrating the use of the standard method of characteristics with nonweighted particles to compute concentrations for case of uniform pore volumes (*A* and *B*) and for case of varying pore volumes (*C* and *D*). *A* and *C* represent initial conditions and *B* and *D* represent particle positions after one time increment. Particles are colored according to their concentration. All quantities are in arbitrary units.

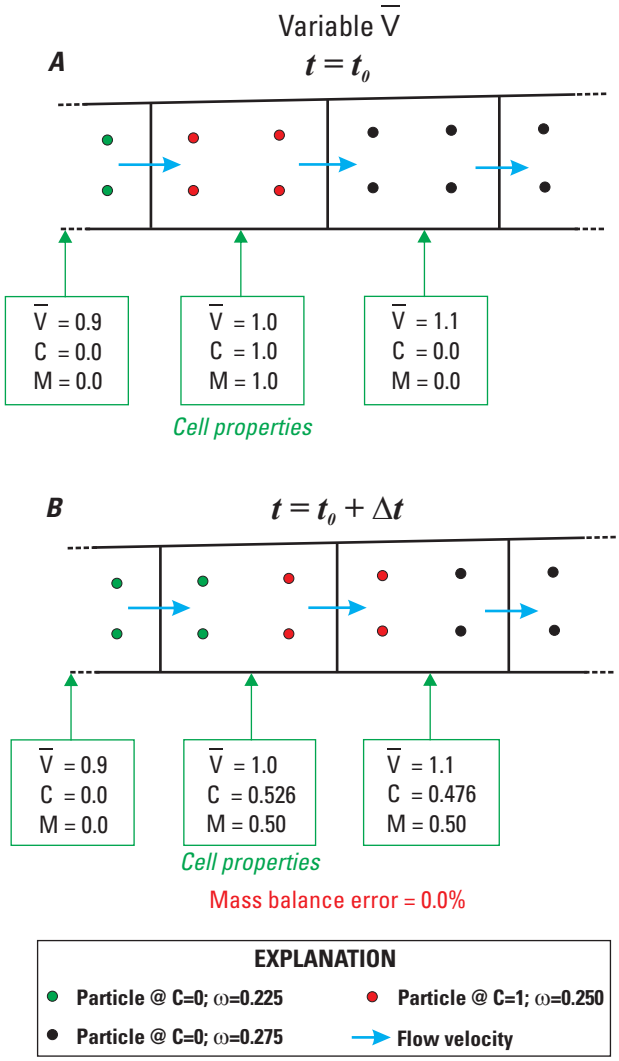

Figure 2. Same example as figure 1 illustrating the use of the MOCWT with volume-weighted particles to compute concentrations, in the case of spatially varying pore volumes. *A* represents initial conditions and *B* represents particle positions after one time increment. $\overline{V}$ is the pore volume, ω is the particle weight, *C* is the cell concentration, and *M* is the solute mass in the cell. Particles are colored according to their weights. All quantities are in arbitrary units.

Boundary Conditions

A major difficulty in particle-tracking methods is dealing with fluid sources and sinks, especially for the situation where the specified flux is "weak" or "intermediate" in strength relative to the grid size and the regional flow field (for example, see Konikow and Bredehoeft, 1978; Konikow and others, 1996; and Pollock, 2012). This is one cause of mass-balance errors in the original MOC formulation. A "weak" source or sink exhibits a negligible effect on the local flow field. An "intermediate" source or sink has a small effect on the flow field, and there can be inward flow (for a source) or outward flow (for a sink) across at least one cell face. "Intermediate" sources and sinks are lumped with "weak" sources and sinks for purposes of numerical treatment. A "strong" source or sink is characterized by outward flow (for a source) or inward flow (for a sink) across all active cell faces.

Special considerations are required for cells representing boundary conditions where fluid enters or leaves the model domain. Where a boundary condition represents a fluid source (that is, fluid enters the model domain), that fluid must have a concentration associated with it, which in turn will affect the concentration in the aquifer. The basic requirement of maintaining a mass balance stipulates that the volume of fluid added at a source cell during a time increment must be balanced by a corresponding increase in weights added to particles in that cell, and the total solute mass entering the cell with the fluid source must equal the sum of the solute mass added to particles in that cell. Similarly, where a boundary condition represents a fluid sink (that is, water leaves the model domain), the volume of fluid leaving the sink cell during a time increment must be balanced by a corresponding decrease in weights on particles in that cell.

Inflow Boundaries

Boundary conditions representing a fluid source may be characterized as being either a "strong" source or a "weak" source, depending on the strength of the inflow from the boundary flux relative to the cell-to-cell outflow to adjacent cells. That is, if most or all of the outflow from a cell is derived from the boundary condition's inflow, then the source term can be considered to represent a "strong" source. If a boundary flux represents a strong fluid source, then the added volume of fluid will be represented as weights on newly created particles added to the cell. If a boundary flux represents a weak fluid source, then the added volume of fluid will be represented as weights added to existing particles in the cell. The model user sets a numerical parameter (GENCRIT, a particle GENeration CRITerion, appendix 1, dataset 7.2) that controls whether sources and sinks would be considered to be "strong" or "weak" for numerical treatment in the code, and whether or not to create new particles to represent fluid added to the system.

A cell having a strong fluid source would have little to no inflow from adjacent cells and can induce a divergence in the flow field if it is a point source (fig. 3*A*). A new particle will be created in such a cell to replace any particle that originated in that cell and had been advected out of the cell during that time increment. The weight added to each newly created particle is, therefore,

$$\omega_{p(NEW)} = W\Delta t / N^*_{p(NEW)} \qquad (10)$$

where $N^*_{p(NEW)}$ is the total number of new particles added to that cell during the time increment. The concentration assigned to each new particle equals C' (the concentration of the fluid

source). The sum of the weights of all new particles equals the volume of water entering the cell from the source term during the time increment ($W\Delta t$). Note that if no particles leave a strong source cell during a time increment, and, hence, no new particles are created, then the rules for a weak source cell will be followed, as described next.

A cell having a weak fluid source may also have substantial inflow from adjacent cells (fig. 3*B*). In such a case, the volume of water added from the external source is balanced by increasing the weights on particles still in the cell after particles are moved by advection during that time increment, which can be expressed as

$$W\Delta t = \Delta \sum_p \omega_p \quad . \tag{11}$$

The total weight added in the cell is distributed proportionately to each particle as

$$\Delta\omega_p = (\omega_p / \sum \omega_{j,i,k}) W\Delta t \quad . \tag{12}$$

The updated weight for each particle is therefore $\omega_p + \Delta\omega_p$ and the updated concentration is

$$C_p^{new} = \frac{(C\omega)_p^{old} + C'\Delta\omega_p}{\omega_p^{old} + \Delta\omega_p} \quad . \tag{13}$$

The change in solute mass in the cell attributable to the source fluid can be expressed as

$$\Delta M_{j,i,k} = \sum \left(\Delta\omega_p C' \right) \quad . \tag{14}$$

Where there is a relatively large flux into the system across a model boundary (such as a subgrid boundary or across a boundary cell face, as defined with the BFLX [Boundary Flux] Package; see Konikow and Hornberger, 2003), special care must be taken to assure that the fluid flux is adequately represented on particles. As particles leave a "strong" source cell, they are normally replaced with a new particle placed in the original initial position in the cell of the particle that left. This placement scheme works well when the fluid source represents a distributed internal source or a point source, such as would be associated with an injection well. But if the fluid source is derived from a flux across a cell face that lies on a boundary, then a uniform initial placement of particles throughout the cell causes particles farther away from the influx boundary to get replaced too often. This results in a reduced average residence time in the cell for new particles. If this method is used with volume-weighted particles, then the fluid volume represented on the new particles will exit the source cell faster than fluid enters across the boundary representing the source. After several transport time increments, the sum of the weights of the particles in the source cell stabilizes at a level below the initial volume of the cell. Because the solute mass crossing the boundary into the cell is fixed, the resulting concentration in the cell will be too high if the volume (sum of particle weights) is too low.

When using volume-weighted particles, an alternative scheme for positioning new replacement particles must be used to approach a volume balance in the cell. Thus, in cells with boundary faces that represent a strong fluid source, instead of placing new particles throughout the entire volume of the cell, particles are replaced only within a reduced area closer to the boundary face across which flow enters the

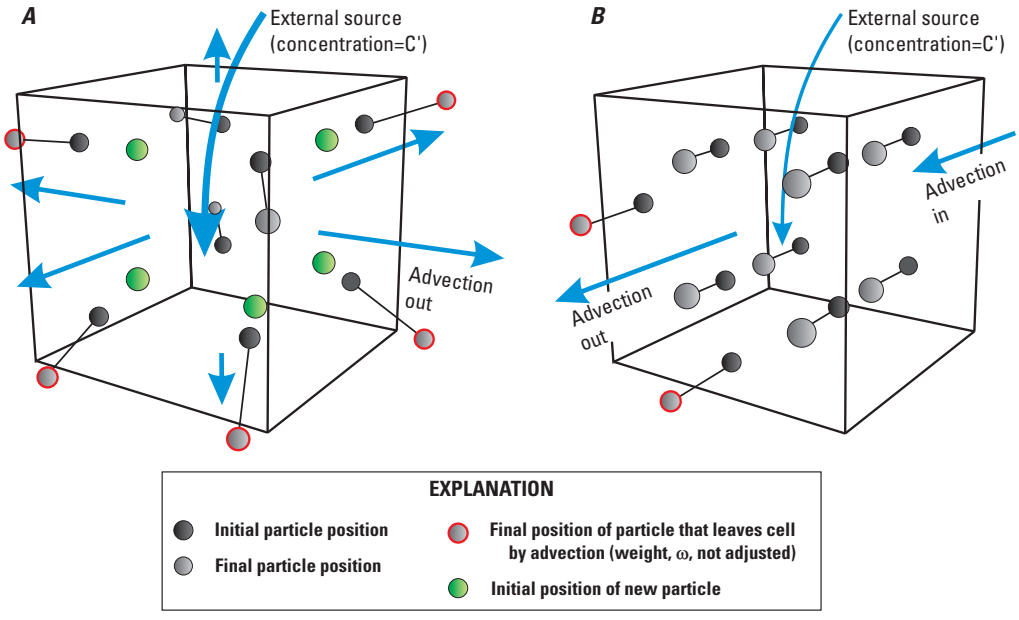

Figure 3. Schematic diagram illustrating inflow boundary conditions in a representative cell of the model grid. Particle sizes are schematically proportional to weights (ω). *A*, A "strong" fluid source cell in which particles leave the cell by advection and are replaced by new particles representing the inflow from the fluid source. *B*, A "weak" fluid source cell in which the volume of fluid added from the external source is balanced by adding weights to particles in the cell; for simplicity, no particles are depicted moving laterally into the cell from the upgradient cell.

transport grid. By appropriately placing the locations of new (replacement) particles closer to the boundary flux face, the average residence (or transit) time of particles through the cell is increased, and, therefore, the cell will maintain a total volume (sum of weights) on all particles in the cell that closely matches the initial pore volume of the cell. The appropriate limit on the area (or volume) in which to replace particles is based on the maximum distance that a parcel of water crossing the boundary face of interest (where the velocity is known) can travel during the length of the transport time increment. The end result will be a more uniform marching of particles through the grid from source areas to discharge areas and a more accurate estimate of concentrations in the boundary source cells, and, hence, in downgradient cells.

The approach is illustrated by using a simple one-dimensional flow example with a cell on a subgrid boundary representing a strong fluid source (fig. 4). There is an influx across the subgrid boundary on the left side of the cell with a certain velocity V_x. The displacement distance associated with this flux can be determined precisely by multiplying the velocity on the face by the transport time increment. In figure 1, this is represented by the length L, and L corresponds to the furthest distance a particle entering this cell could possibly move in one time increment. New particles placed in this cell to replace particles that leave are then restricted to the area defined by this distance. To implement this in the computer code, the particle that started at x_{init} and moved to x_{final} is replaced by a new particle that is placed in its original position. Then, the x-coordinate of the particle is transformed to a new position (x_{new}) coinciding with the same relative location in the partial cell area between the bounding face (in this case, the left face) and the displacement distance. The relative location can be expressed as

$$\frac{x_{init}}{\Delta x} = \frac{x_{new}}{L} \tag{15}$$

where x_{init} is the original location of the particle at the start of the transport time increment. Solving for x_{new}, the position of the new particle at the start of the next transport time increment, yields

$$x_{new} = \frac{x_{init} L}{\Delta x} \ . \tag{16}$$

In the example shown in figure 4, the velocity at the original location of particle x_{init} is high enough so that it moves out of the cell to location x_{final} during one time increment. To compensate for the weight (and associated fluid volume) that has left the source cell, a new particle will be placed in the source cell to represent the source fluid entering across the left boundary face. This fluid and its dissolved constituents can only migrate in the x-direction a distance less than or equal to L, so the x-coordinate of the new particle is placed at the same relative position of the grid cell with respect to L as it initially was, relative to Δx.

If a cell has more than one face on a boundary that has an influx, then the flux across any one of those inflow faces represents a known fraction of total boundary flux into the cell. In this case, each of the boundary flux faces is assigned a probability function based on the relative strengths of the fluxes across each face. This probability function is used to determine which face a replacement particle will be assumed to have entered across, and, hence, which coordinate (x-, y-, or z-direction) for the new particle will be constrained. For example, if there are two boundary faces with inflow velocities equal to V_x and V_y, respectively (see fig. 5), then the probability of the x-coordinate being restricted by L_1 is

$$\frac{V_x}{V_x + V_y} \tag{17}$$

and the probability of the y-coordinate being restricted by L_2 is

$$\frac{V_y}{V_x + V_y} \ . \tag{18}$$

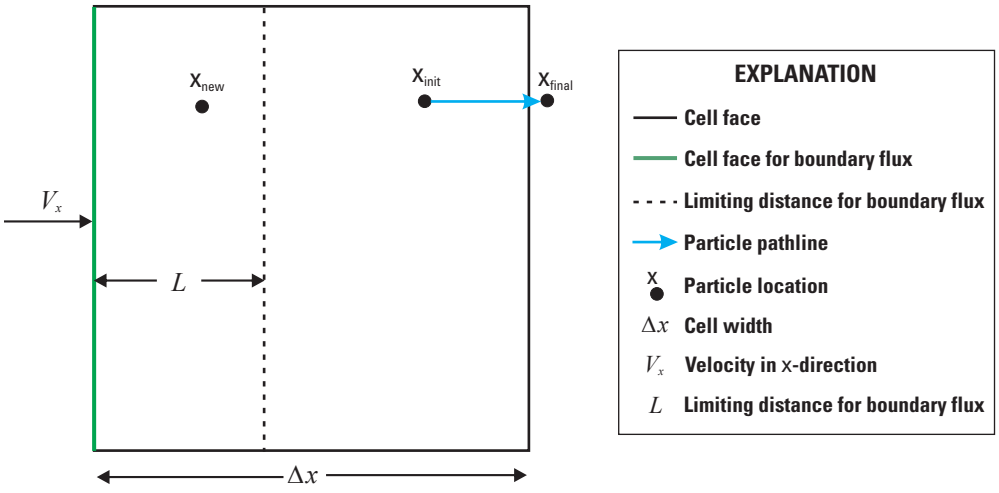

Figure 4. Example for one-dimensional flow (in x-direction) illustrating method to place new particles in area of boundary cell close to face across which flux enters cell across a boundary of the transport grid at a velocity V_x.

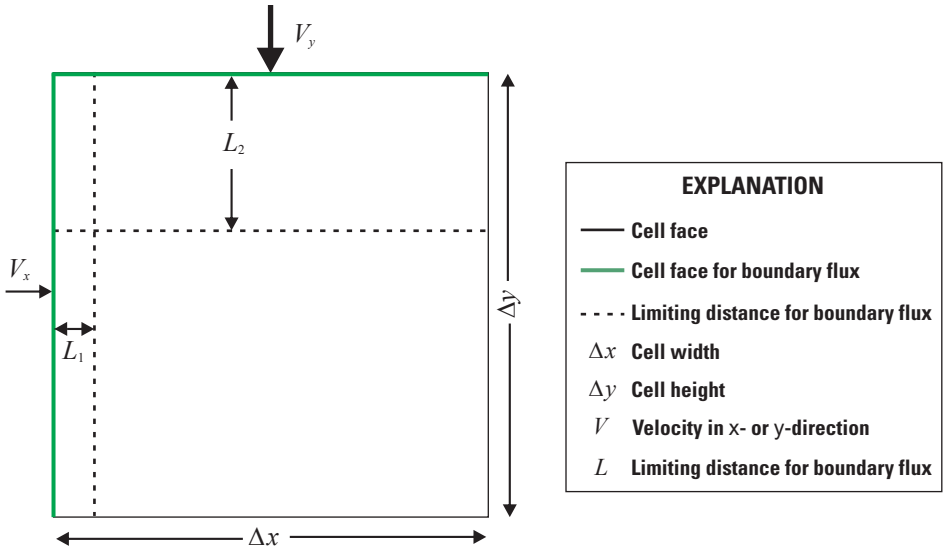

Figure 5. Example showing constrained areas for placement of new particles in a boundary cell in which boundary flux enters cell across two boundary faces at different rates (with the magnitude of V_y exceeding that of V_x).

Because the sum of the two probabilities equals 1.0, each time a new particle is to be placed in the cell to replace one that left, we can select a random number between 0 and 1 to decide which boundary face (and which constrained area) to represent for that particular particle. Given a sufficient periodic refreshment of particles in the cell, both boundary-face inflows will be accounted for in proportion to their respective fluxes. For the case illustrated in figure 5, if V_y is four times greater than V_x, then approximately 20 percent of the new particles generated to represent source fluid entering that cell across its boundaries will be placed in the area defined by L_1 and the left face, and approximately 80 percent of the new particles will be placed in the area defined by L_2 and the top face.

Outflow Boundaries

Boundary conditions representing a fluid sink can be characterized as being either a "strong" sink or a "weak" sink, depending on the strength of the outflow from the boundary flux relative to the cell-to-cell influx from adjacent cells. However, unlike for fluid sources, the relative strength of the sink term does not affect the way in which particles are treated or weights are adjusted. Instead, the weights of particles in sink cells are always and consistently adjusted (reduced) to account for the fluid volume removed from the cell according to equation 12 (fig. 6). The concentrations of the particles are not changed because reducing the particle weight removes solute mass from the system and has no direct effect on the concentration in the fluid remaining in the cell.

If the cell is a strong sink, then particles will never leave the cell, but might accumulate indefinitely. To prevent this from occurring, particles will be removed when their weights become smaller than some critical value, at which point they are tracking a trivially small mass. This criterion is set by the user (REMCRIT, appendix 1, dataset 7.2). REMCRIT represents a fraction of the volume and mass in a cell; if both the weight and the mass of a particle are less than this criterion,

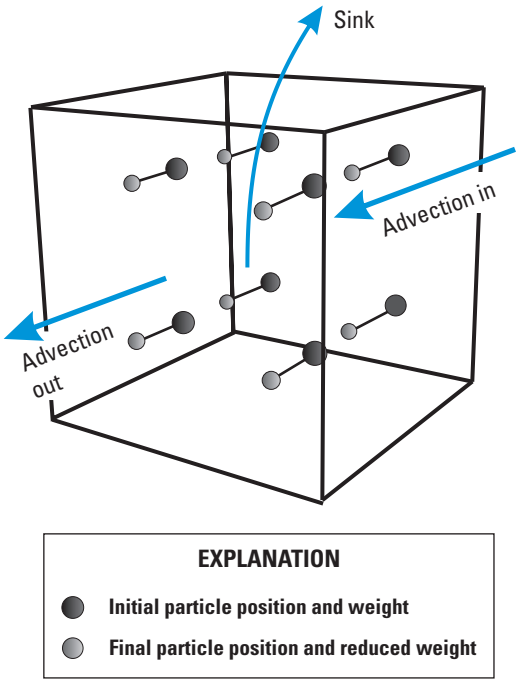

Figure 6. Schematic diagram illustrating a simple example of a cell with a weak sink (outflow boundary) in which no particles enter or leave the cell during the time increment. The volume of fluid removed from the cell by the sink term is balanced by removing weights from particles in the cell. Particle sizes shown are schematically proportional to weights.

then the particle is removed from active tracking and the small remaining weight and mass of the removed particle are distributed to the remaining particles in the cell. If the volume to be removed in the sink cell is greater than the sum of the particle weights in the cells, all the particles are removed and any leftover solute mass is added to particles entering the cell in a subsequent transport step.

Local Volume Balance in Source Cells

The volume-weighted particle method strives to maintain a global mass balance over the entire grid by assuring that the sum of the weights within the grid matches the total water volume within the grid. Because MOCWT uses a finite number of discrete particles that each track (or represent) a fixed volume of water (the weight), there will generally not be an exact match between the sum of the weights on particles that leave a finite-difference cell during a time increment and the fluid volume that leaves as determined by the solution to the groundwater flow equation. The same is true for particles and fluid entering a cell during a time increment. Therefore, even though a global volume balance is maintained, a discrepancy between the sum of weights of all particles in a particular cell and the pore volume of water contained in that cell can occur. This discrepancy, in turn, will cause the estimated concentration in the cell to be too high or too low, even though the correct mass is added to the cell. This produces local oscillations in the calculated average cell concentration field, although it would not lead to global mass-balance errors and locally would tend to smooth out over time. However, the problem may be exacerbated in cells that represent strong fluid or solute sources. Any local oscillations on particles originating in the source cell will be propagated downstream through the flow field.

In a source cell, the solute mass and fluid volume entering the cell from external sources are known. The new average concentration in such a cell is computed by a simple mixing formula as

$$\bar{C}^{t+adv}_{j,i,k} = \frac{M_{in} + M_{jik}}{\bar{V}_{in} + \bar{V}_{jik}} = \frac{\sum_n Q_{bdy_n} \Delta t \; C'_n + \sum_p \left(\omega_p C_p \right)}{\sum_n Q_{bdy_n} \Delta t \; + \sum_p \left(\omega_p \right)} \quad (19)$$

where

$\bar{C}^{t+adv}_{j,i,k}$ is the average concentration in the source cell at the intermediate time level (t+adv) after advective transport has been simulated by moving particles (see Konikow and others, 1996, p. 13);

M_{in} is the net solute mass entering the cell with specified flows across boundaries;

M_{jik} is the total mass on particles in the cell;

$\bar{V}_{in}$ is the net fluid volume entering the cell from all boundary condition fluxes;

$\bar{V}_{jik}$ is the total fluid volume associated with particles remaining in the cell after advection;

n is an index for the total number of fluid sources and sinks in the cell (including cell-to-cell flow to or from adjacent strong source cells or entering across a subgrid boundary);

Q_{bdy_n} is an individual fluid flux associated with a particular type of boundary condition;

C'_n is the solute concentration in the n^{th} fluid flux term; and

ω_p is the weight associated with a particle in the cell at the intermediate time level.

If the denominator in the last term on the right side of equation 19 (the net fluid volume entering from specified boundary sources plus the sum of the weights of all particles in a cell) matches the pore volume of the cell ($\varepsilon b \Delta x \Delta y$), the average concentration in the cell will be correctly calculated and mass on the particles will be locally conserved. If the denominator does not match the pore volume of the cell, this approach will conserve mass on the particles within the cell but produce an incorrect average concentration in the cell. (If the sum of weights plus the source volume is greater than the pore volume, the calculated average concentration will be too low, and vice versa.) If the equation were modified to calculate the average concentration on the basis of the pore volume instead of the sum of weights of particles in the cell after advection, then the proper concentration would be calculated; however, the sum of the weights times the concentrations of all particles would not yield the correct solute mass, producing and perpetuating a global mass-balance error.

An approach to eliminate this source of error is based on maintaining a local volume balance, as well as a local solute mass balance, in fluid and solute source cells. If the denominator in the last term of equation 19 is less than the pore volume, then there is a deficit in the sum of particle weights in the cell because the first part of that term ($Q_{bdy_n} \Delta t$) is known and fixed. Therefore, proportionately too much volume has left the cell during the time increment on particles that were advected out of the cell and (or) too little has entered the cell on particles that were advected into the cell (recalling that even though a cell represents a net fluid source, depending on local head gradients, there can still be some amount of cell-to-cell flow into such a source cell). If the denominator in the last term of equation 19 is greater than the pore volume, then there must be an excess in the sum of particle weights remaining in the cell. Therefore, not enough volume (weight) has left the cell during the time increment on particles that were advected out of the cell and (or) too much has entered the cell on particles that were advected into the cell.

The approach taken to achieve a local volume balance in fluid- and solute-source cells is to compare the fluid volume entering the cell from adjacent cells with the sum of particle weights on particles that advect into the cell and assure that they match. Similarly, if there is fluid flow out of a source cell into adjacent cells, a match is also required between it and the sum of particle weights leaving the cell during that time increment. Because there is always a match between the volume of fluid sources and added particle weights, and between the volume of fluid sinks and removed particle weights, then meeting the balance conditions for all cell-to-cell flows with changes in particle weights will assure that the sum of particle weights in a source cell will continue to match the cell's pore volume.

In considering fluxes and advection across the six faces of source cells, flow between adjacent source cells is considered separately and handled differently than flow between a source cell and an adjacent active cell that is not a fluid source. As explained below, for the latter case, after all particles have been moved during a time increment, the weights and masses of particles moving into or out of a source cell may have to be adjusted to assure the local volume balance. But if both cells include sources, the particle corrections in one cell would lead to readjustments in the other, as well as a need for complex iterations in achieving a local balance. A simpler alternative is to treat flow and advective-solute flux between adjacent source cells designated for a local volume balance as a known solute-mass boundary flux based on the fluid flux computed by the solution to the flow equation and the average cell concentration at the start of the transport time increment.

This explicit approach is computationally simple and consistent with the concept of full volumetric mixing in a strong source cell, although it also has time-step size constraints. To be consistent with explicitly and exactly counting the cell-to-cell flux in this manner, any particles that move into a volume-balanced source cell from an adjacent volume-balanced source cell are removed from the calculation process and their weight and mass are returned to (or kept in) the originating strong source cell.

There are four possibilities to consider with respect to the remaining fluxes across cell faces between a designated source cell and adjacent active cells that are not strong fluid sources:

1. Where the total volume advected out of the designated source cell into adjacent active cells that are not designated for volume balance is overestimated (too much weight left the cell on particles), the excess in weight advected out of the cell is recovered by removing weights proportionately from all of the particles that have left the designated source cell during this time increment. The proportionate removal of weight implies that the weight removed from one particle is proportional to the weight of the particle, and that the sum of all weights removed equals the excess. The solute mass retrieved in this manner will be determined by the weight removed from each particle times the concentration of each particle summed over all particles. The retrieved volume and mass are then placed in the designated source cell by distributing the weights and mass equally onto all particles remaining in the source cell. If there are no particles left in the cell, new particles are created by using the same number and distribution as initially used and the weight and mass are distributed on these newly created particles.

2. Where the total volume advected out of the designated cell into adjacent active cells that are not strong fluid sources is underestimated (too little weight left on particles exiting the cell), the deficit in weight advected out of the cell is recovered by removing weights proportionately from all of the particles remaining in the source cell and placed in a cell receiving outflow. The deficit is placed in a forward (downstream) cell on a single newly created particle. The cell in which the particle is placed is selected randomly on the basis of the relative flux across all faces where flow leaves the designated source cell and enters an adjacent active undesignated cell. For example, if there are two such outflow faces and the flux across one of the faces is twice that across the other, then over many time increments, the newly created particle will have been placed in the cell receiving the higher flow twice as often as in the cell receiving the lesser flow. The new particle is placed a distance from the cell face equal to one-half of the length of the time increment times the velocity at the face; this places it at a point consistent with the center of mass of a continuous release of mass across the face during the time increment. The position of the particle in the plane located $0.5\Delta tV$ away from the face is selected randomly in both directions within the plane (that is, the coordinates are selected randomly from a uniform distribution within the range from –0.5 to +0.5 of the relative cell distances of the two dimensions in which the plane lies). This procedure introduces a small stochastic component into the otherwise deterministic modeling solution.

3. Although the flux into a designated source cell, by definition, must be less than the flux out of it, the flux into a designated source cell can still be nonzero and must be considered in striving for a local volume balance. Where the volume advected into the source cell on particles from adjacent active undesignated cells is overestimated (too much weight entered on advected particles), the excess is retrieved proportionately from all such particles that entered the strong source cell and replaced in a contributing adjacent cell. If more than one adjacent cell has contributed, the cell to receive the replacement of the excess weight and mass is selected randomly on the basis of the relative flows across the cell faces into the strong source cell. As above for case 2, this will assure that over many time increments, all adjacent contributing cells will receive the proper proportional share of the excess weights. However, it does not assure that contributing cells receive the proper share of the solute mass, and there can be some numerical dispersion in the contributing cells. Within the cell selected to receive the excess weight and mass, the excess is redistributed equally onto all particles in that cell (for example, if there are three particles in the cell, then one-third of the excess weight and mass is added to each particle and an updated concentration for each particle is recomputed accordingly). If there are no particles remaining in the cell, new particles are created by using the same number and distribution as initially used and the excess weight and mass are distributed on these newly created particles.

4. Where the volume advected into the source cell on particles from adjacent active nonstrong-source cells is underestimated (not enough weight entered, often because no particles have advected into the strong source cell in response to a typically small flow into a strong source cell), the deficit is retrieved by removing the weights (and related mass) from particles in all such contributing cells in proportion to the flux contributed from those cells. Within each of the contributing cells, the volume (weight) and related mass are removed from all particles in proportion to their relative weight within the cell. However, if the sum of the weights in a cell is insufficient to meet the demand generated by the deficit, the process will remove all the weight present and the remainder will remain as an error in the local volume balance within the designated source cell. The weights removed to match the deficit are then applied as equal additions of mass and volume to all particles in the designated source cell. If there are no particles in the cell, the deficit is added to a newly created particle at the center of the cell.

Further adjustments of particle weights to account for specified (external) fluid sources and sinks are made independently of these adjustments. If all of the above conditions are met, it will guarantee that the sum of the weights on particles in the strong source cell after advection plus the known net-fluid inflow volume will equal the cell's pore volume. Maintenance of the local volume balance should also yield a local solute mass balance, as well as the calculation of an accurate and smooth rate of change of concentration in the source cell for a steady mass influx associated with a specified boundary condition, thereby minimizing or eliminating oscillations originating in the fluid-source cells. This process is particularly sensitive to accuracy of the flow solution. A small error in calculating the flow across a cell face may cause the deficit to be calculated incorrectly, resulting in a small change in total volume in the cell at each transport time increment.

Case 2 (above) of the volume-balancing option balances the weight in the source cell by creating a new particle in an adjacent cell. When this situation occurs frequently, then over many time increments, a cumulatively large number of particles may be created in the downstream cell. To maintain computational efficiency, the model includes criteria to automatically "lump" low-weight particles together in downgradient cells immediately adjacent to a designated volume-balance cell. That is, if a particle created due to balancing the volume in a source cell was underestimated by particles advected out of the cell and numerous particles so generated have trivial weights, then the computational cost of individually tracking all trivial-weight particles may not be worth the effort. Removing at least some of these low-weight particles and redistributing their weight and mass among the other remaining particles in the downgradient cell will reduce computational efforts without much loss of numerical accuracy or precision.

The implementation of the local volume balancing is optional. In dataset 7.2 (see Input Data Instructions in appendix 1), the `ISRCFIX` flag controls whether this option is implemented in a simulation. If `ISRCFIX=0`, the corrections to maintain a local volume balance are never done. If `ISRCFIX=1`, the local volume balance is activated and implemented at cells, depending on the value of `GENCRIT` in dataset 7.2 and any additional specific cells designated in the optional VBAL Package. If `GENCRIT` flags a source cell as being a strong fluid source and `ISRCFIX=1`, then local volume balancing will be implemented in that cell. Additionally, a user may want to implement local volume balancing at specific cells that may represent weak fluid sources but strong solute sources (for example, at a cell having a low recharge rate with a very high source concentration). This can be accomplished by listing the grid coordinates of those specific cells in the optional VBAL Package.

If local volume balancing is active, then "lumping" of newly generated particles having trivial weights that have migrated into cells adjacent to the designated volume-balanced cell will be implemented. In such cells, lumping will only be implemented if the total number of particles in that cell has increased to more than the initial number of particles placed in that cell plus eight. If that is the case, then the model will search for all particles in that cell that have a trivial weight, in which the condition of "trivial" is defined as having a weight less than the weight of particles initially generated in that cell times the value of `REMCRIT` (defined in dataset 7.2). Then it will remove the weight and solute mass associated with every other particle so flagged and lump it onto one newly created particle that is placed at the average location of all of the removed particles. (Removing just every other particle assures that a reasonable number of particles will always remain after the lumping is implemented.)

Applying Dispersive Flux to Particles

After calculating the change in concentration in a cell caused by dispersion using either the explicit or implicit finite-difference approach, that change must be applied to all particles within the cell. An overriding consideration is that this transfer must conserve mass, and, hence, the sum of the changes in mass on the particles must equal the change in mass computed for the cell. A concern is that because the particle concentrations will usually include a range of concentration values about the mean of the cell, the adjustment of particle concentrations for the change due to dispersion might cause the concentration of individual particles to decrease to a negative value or to increase to a value higher than the highest source concentration. Such undershoot and overshoot could propagate spatially as particles move with the flow, and should be minimized or avoided.

Recognizing that particle concentrations deviate from the mean for the cell, and that according to the governing equation the dispersive flux is driven by the concentration gradient,

we can deduce that the appropriate change in concentration on a particle should be related to both the magnitude of its concentration and the mean concentration in the cell. That is, if the computed change in concentration due to dispersion in a cell is negative, then particles within that cell that have concentrations high relative to the mean should show a greater reduction in concentration than particles having a relatively low concentration because the concentration difference driving the dispersive flux would be greatest for a particle having the highest concentration. Conversely, if the computed change in concentration due to dispersion in a cell is positive (an increase in concentration), then particles within that cell that have a relatively low concentration should show a greater increase in concentration than particles having a relatively high concentration because the concentration difference driving the dispersive flux would be greatest for a particle having the lowest concentration. In MOCWT, a modified approach of that used in MOC for adjusting particle concentrations for dispersion is implemented, as described in more detail by Konikow (2010).

Decreasing Concentration

If dispersion indicates that the concentration of a cell should decrease over a time increment because the dispersive flux out of the cell is greater than the dispersive flux into the cell, the method for applying the change in concentration for the cell to particles in the cell is to reduce the concentrations of all particles in the cell by the percentage reduction calculated for the cell, as described by Konikow and Bredehoeft (1978) and Konikow and others (1996). Reducing the concentration of particles by the percentage of the reduction at the cell is equivalent to using the value of zero as a base or limit to the amount of decrease that can occur on a particle. Applying the same percentage reduction to all of the particles in the cell results in a greater absolute decrease in concentration on the particles having a higher concentration and avoids the particle with the lowest concentration being reduced to a value below zero.

However, when dispersion induces a decrease in concentration in a cell, the base value for the percentage reduction might more properly be the minimum concentration occurring in any adjacent cell, which drives the concentration gradient. Then, the percentage reduction can be scaled to this minimum concentration as a base, and would be directly related to the ratio $\Delta C \Big/ \left(C_{jik}^{n} - C_{\min} \right)$, where $C_{\min}$ is the minimum concentration in any cell adjacent to cell j,i,k that is used in the calculation of dispersive flux. For a three-dimensional grid accounting for cross-product terms, this could include as many as 26 adjacent cells.

The concentration at any particle or any node can thereby be computed directly from

$$C_{p}^{n+1} = \left(C_{p}^{n} - C_{\min} \right) \left(1 + \left[\Delta C \Big/ \left(C_{jik}^{n} - C_{\min} \right) \right] \right) + C_{\min} \quad . \quad (20)$$

The use of equation 20 leads to the greatest reduction in the concentration of the particle with the highest initial concentration, as well as the smallest reduction in the concentration of the particle with the lowest initial concentration. This smaller range in concentration change serves to minimize any possible numerical dispersion arising from the transfer of dispersive changes from nodes to particles. Furthermore, the final concentration of the particle with the lowest concentration remains greater than the minimum concentration of the adjacent cells, which is a desired end result of using this new algorithm.

In the event that a particle had a concentration less than the base level ($C_{\min}$) at the beginning of the time increment, this mass-conservative algorithm has the advantage that it will compute an increase in concentration for such a particle, even though the cell concentration is decreasing. This self-correcting aspect minimizes the potential for the propagation of oscillations.

Although the use of equation 20 requires slightly more computational effort than the original simpler approach, it produces results that are more consistent with conceptual expectations for the dispersion process when concentrations are decreasing due to dispersion; hence, it is incorporated into the new code.

Although the method described above would be applied in most circumstances, there are several circumstances in which other methods are used. If the percent change in concentration on a particle is computed to be less than –100 percent (for example, –110 percent), then a base of zero is used and the percent change in concentration is recalculated by using a base of zero. If the percent change in concentration is still less than –100 percent, or if the percent change is greater than 0 or if the concentration in the cell is less than 0, then the change in the concentration of the cell is applied to the particle arithmetically. This can result in particles with negative concentrations. Dispersion during following time increments will tend to eliminate the negative concentrations.

Increasing Concentration

If dispersion indicates that the concentration of a cell should increase over a time increment because the dispersive flux out of the cell is less than the dispersive flux into the cell, then the new concentration for the cell at the end of the time increment, represented by the nodal value, would be $C_{jik}^{n+1} = C_{jik}^{n} + \Delta C_{jik}$. At the start of the calculation of dispersive fluxes for the time increment, the cell contains a number of particles of varying individual concentrations, whose mean equals C_{jik}^{n}. To apply the calculated change in concentration in the cell to the particles in that cell, the simplest approach would be to add the same change in concentration calculated for the cell to the concentration of each particle. This would assure that the mean of the new concentrations of all the particles $\overline{C}_{p}$ would equal the new cell concentration (C_{jik}^{n+1}) and that the process would necessarily conserve mass. This was the method used in both the original MOC3D code (Konikow

and others, 1996) and its predecessor two-dimensional MOC code (Konikow and Bredehoeft, 1978). However, this computationally simple approach can lead to individual particle concentrations that are higher than the value of the highest concentration used to compute the dispersive flux from an adjacent cell into cell *j,i,k*. But just as C_{min} represents a lower limit to updated particle concentrations when cell concentrations are decreasing due to dispersion, as described in the preceding section, so should the highest concentration in an adjacent cell represent an upper limit to increased particle concentrations.

Thus, a new method for inclusion in MOCWT was developed for increasing individual particle concentrations in a cell where dispersion is causing a net flux of mass into the cell (see Konikow, 2010). Analogous to the method used when concentrations are decreasing, the new method increases concentrations on particles on a percentage basis over a scaled range where the maximum concentration in any adjacent cell (C_{max}) represents the theoretical upper limit for increased particle concentrations. This method has the following characteristics:

- Particles with a concentration greater than the mean concentration at the cell node (C^n_{jik}) should increase less than the nodal increase, with the particle having the highest concentration increasing the least.
- Particles with a concentration less than the mean concentration at the cell node should increase more than the nodal increase, with the particle having the lowest concentration increasing the most.
- Particles with a concentration equal to C_{max} should not change in value, even though the mean cell concentration is increasing.
- Particles with a concentration equal to C^n_{jik} should increase by the same amount as at the node (ΔC_{jik}).
- Particles with a concentration already above C_{max} should decrease in concentration.
- The mean of the particle concentrations at the end of the time increment should equal the new concentration at the node (C^{n+1}_{jik}) (that is, the application of the dispersive change to particles should be mass conservative).

These objectives (or constraints) are met by developing an algorithm that first increases the particle concentration by some arbitrary constant amount (C_x) more than the computed nodal increase (ΔC), and then subsequently reducing the overestimated particle concentration on a percentage basis, where the fractional reduction is related to the ratio of C_x to $\left(C^n_{jik} + \Delta C + C_x\right)$.

The fourth constraint above can thereby be expressed mathematically as

$$\left(C^n_{jik} + \Delta C + C_x\right)\left(1 - \frac{C_x}{C^n_{jik} + \Delta C + C_x}\right) = C^n_{jik} + \Delta C \quad . (21)$$

Next, we must determine the value of C_x. Based on the third constraint above, we can express equation 21 for a particle having a value of C_{max}. For this case, equation 21 can be rewritten as

$$\left(C_{max} + \Delta C + C_x\right)\left(1 - \frac{C_x}{C^n_{jik} + \Delta C + C_x}\right) = C_{max} \quad . \quad (22)$$

We can expand equation 22 and combine terms to yield

$$\Delta C - \frac{C_x \Delta C}{C^n_{jik} + \Delta C + C_x} + C_x - \frac{\left(C_x\right)^2}{C^n_{jik} + \Delta C + C_x} - \frac{C_{max} C_x}{C^n_{jik} + \Delta C + C_x} = 0 \quad . (23)$$

Solving equation 23 for the unknown quantity, C_x, results in

$$C_x = \frac{-C^n_{jik}\Delta C - \left(\Delta C\right)^2}{C^n_{jik} + \Delta C - C_{max}} \quad . \quad (24)$$

Once C_x is defined for a particular cell and time increment on the basis of the local C_{max}, then the new concentration for any given particle in the cell can be calculated from the following generalized version of equation 21:

$$C^{n+1}_P = \left(C^n_P + \Delta C + C_x\right)\left(1 - \frac{C_x}{C^n_{jik} + \Delta C + C_x}\right) \quad . \quad (25)$$

As was true for equation 20, the use of equations 24 and 25 also requires slightly more computational effort than a simpler additive approach (as used in earlier models) when the cell concentration is increasing due to dispersion. However, it produces results that are more consistent with conceptual expectations for the dispersion process and cannot create particle concentrations that overshoot physically based bounds; hence, it is incorporated into the new code.

If C_x is calculated to be less than or equal to 0, the change in the concentration of the cell is applied to the particle arithmetically.

Water-Table Cells

Change in Saturated Thickness

Under many hydrologic conditions, the water table can rise or fall over time. The MODFLOW–2000 program simulates this type of transient change, and at the limit, even allows a "convertible" cell to go dry if the water-table elevation (or head) falls below the elevation of the bottom of the cell. To assure consistency between the flow model and the transport model, MODFLOW–GWT always adjusts and updates the saturated thickness of cells after each flow time step to assure the use of accurate volumetric parameters in accounting for changes in solute mass and concentration.

Drying Cells

If a cell goes completely dry, the volume of water in that cell goes to 0; therefore, if this occurs, there will be no solute mass left in the cell. However, if the rate of decline in the water table is fast relative to the size of the time increments used to solve the transport equation, it is possible (though unlikely) that one or more particles can be "trapped" in a cell that has gone dry. If this happens, and if the cell below the newly dry cell is active, then the total mass (weight and concentration) of the trapped particles will be translated onto a newly created particle placed at the center of the underlying active cell. If particles are trapped in a dry cell and the cell below is inactive, then the total mass of the trapped particles will be transferred to a new particle placed at the center of whichever active cell of the eight cells laterally surrounding the newly dry cell has the lowest head. This is based on the assumption that the active cell, with the lowest head after the cell went dry, would indicate the direction of the steepest hydraulic gradient, and that trapped particles would have moved laterally in the direction of the highest velocity out of the drying cell. These adjustments also globally conserve mass.

Rewetting Cells

The MODFLOW program has an option to allow cells that have previously gone dry to rewet at a later time if changes in hydrologic conditions cause the simulated water table to rise above the bottom elevation of a dry cell. If this occurs, the solute-transport model must recognize this condition and develop a consistent method for reinitializing solute mass in the rewetted cell.

A conceptual problem is that prior to rewetting, the solution to the flow equation necessarily indicates no flow (and zero velocity) into the still-dry cell, so the solution to the transport equation for times under this head distribution will not allow particles (and solute) to move into the cell. Next, after a new time step is completed in solving the flow equation, this formerly dry cell may now become partly saturated. Because the flow and transport equations are solved sequentially, the progressing transient solution to the transport equation has an incomplete starting condition for the next time step immediately after a cell has rewetted. That is, the cell now has a volume of water in it based on the most recent solution to the flow equation, but no solute mass or concentration is yet associated with that volume of water.

To solve this problem, after the flow equation is solved for a new time step, the model checks to determine if any dry cells have rewet. If any cells are so identified, a new initial distribution of particles is placed in that cell prior to commencing the solution of the transport equation for the time interval over which the flow equation has been solved. The number of new particles placed in the cell is determined by the value of `NPTPND` or by the IPDA or IPDL Packages, as selected by the user. The weight assigned to each particle is equal to the updated volume of water contained in the cell divided by the number of new particles placed in the cell. The concentration in the rewetted cell (and the concentration assigned to each particle) should be related to the source of the water that has flowed into the cell with its rewetting. That water may have been derived from some combination of upward flow from an active cell below the rewetted cell and (or) lateral inflow from active adjacent cells. The MODFLOW program includes a parameter (`WETDRY`) that constrains the source of water for rewetting. If `WETDRY`<0, only the cell below a dry cell can cause the cell to become wet; if `WETDRY`>0, the cell below the dry cell and the four horizontally adjacent cells can cause a cell to become wet. The details of setting the concentration value in the rewet cell is, therefore, based on the value of `WETDRY`.

If a cell (*j,i,k*) has rewet only by upward flow from below (`WETDRY`<0), then the concentration for the rewetted cell is assumed to equal the mean concentration in the underlying source cell at the end of the previous time increment ($C^{t-1}_{j,i,k+1}$). This is an approximation that ignores any concentration gradients present within the underlying cell, and inherently assumes that the fluid that moved from this cell to the rewetted cell moved in at the average concentration of the source cell. To maintain a balance of solute mass, however, the mass transferred to new particles in the rewetted cell ($M_{j,i,k}$, where $M_{j,i,k} = \sum\left(\omega_p \times C_p\right)$ for all particles in the rewetted cell) must be subtracted from the mass stored on particles in the underlying source cell. This approach conserves mass, but may cause some temporary spatial oscillations in the concentration field. In addition, because of the discrete nature of particles and the variability in their spatial distribution, it is possible that the sum of the weights of particles present in a cell might be smaller than the saturated pore volume of that cell. In such a case, there might be insufficient solute mass present on the particles in the source cell to match the required solute mass transferred to the rewetted cell based on the criteria described above. Thus, the concentration assigned to the rewetted cell (and, hence, the mass transferred between cells) is made subject to the constraint that no more than half the mass in the underlying cell is allowed to be transferred to the rewetted cell. If this constraint is activated, the mass and concentration in the rewetted cell may be lower than initially assumed, but the constraining condition will assure a global mass balance and provide a locally smooth concentration distribution.

Evapotranspiration

When the Evapotranspiration Package (EVT) is in use, the flux calculated due to evapotranspiration (ET) is included in the sink term. However, under the default assumption for solutes that evapotranspiration removes water but excludes solute, this particular type of sink should not affect the mass of solute remaining in the cell. Therefore, when the appropriate volume of water is removed from a cell subject to an ET flux, the associated solute is not removed. This is accomplished in the weighted particle method by decoupling the ET flux from

the other sink terms, and by recalculating particle concentrations in cells with ET losses in accordance with a reduced volume (particle weight) and constant solute mass. Thus, ET will cause particle concentrations to increase. Note, however, that if groundwater age is being simulated with the AGE Package (Goode, 1999), groundwater discharge by ET is not allowed to exclude "age" or increase the age concentration, and "age" mass is removed with the fluid as it is with a normal sink.

The volume removed from a weighted particle during a time increment due to evapotranspiration is related to the fraction of the sum of weights in a cell represented by the individual particle weight as

$$\Delta\omega_p^{ET} = \frac{\omega_p}{\sum \omega_{jik}} \left| ET_{jik} \right| \Delta t \qquad (26)$$

where

$\sum \omega_{jik}$ is the sum of the weights (volumes) of all of the particles in the cell; and

ET_{jik} is the volumetric ET flux for the cell (L^3/T).

If ET is the only sink in the cell, the new particle concentration will be

$$C_p = \frac{M_p}{\omega_p - \Delta\omega_p^{ET}} \qquad (27)$$

where

M_p is mass of the particle.

If there are multiple sink terms associated with a cell where there is ET, the sequence in which the effects of the sinks are calculated will affect the calculated particle concentration and solute mass at the end of the time increment. In other words, because ET leads to an increase in particle concentration, the final value of C_p will differ, depending on whether ω_p and M_p in eq. 27 have or have not been already reduced to account for the non-ET sinks. To eliminate sensitivity to the order of calculations to account for fluid sinks, a stepwise approach is used in which half of the ET is accounted for first by solving eq. 27 before adjusting for other non-ET sinks. Then particle weights and mass are adjusted to account for fluid removed from all non-ET sinks on the basis of the intermediate particle concentration and mass. Finally, eq. 27 is again applied to determine the particle concentration after removing the remaining volume (weight) due to ET.

Treatment of Empty Cells

Particle movement can cause cells occasionally to become void of particles (herein called an "empty cell"). In source cells, new particles are generated at the original particle locations to represent the added mass. Empty cells that are not source cells must be treated differently. When the number of such cells exceeds a user-specified critical value (`FZERO`, dataset 7), an attempt will be made to generate a new particle in empty cells at the face of the cell that has the greatest fluid flow into the cell. However, this only is done if the outflowing cell on the opposite side of the face contains at least one particle, so some cells may remain void of particles. The position of the new particle is determined by the size and positions of the two cells having the greatest fluid inflow into the cell. If there is only one cell from which fluid flows into the empty cell, the particle will be placed in the center of the face of the cell with the inflow. The same will be done if the cell with the second largest amount of flow into the cell is directly opposite the cell with the largest flow into the cell. In other cases, the position of the particle will be displaced from the face center toward the cell with the second largest inflow in proportion to the relative sizes of the two inflows, according to the following equation:

$$F = \frac{I_s}{I_l + I_s} \qquad (28)$$

where

F is the particle displacement as a fraction of the length of the cell side;

I_s is the fluid inflow on the cell face with the smaller inflow; and

I_l is the fluid inflow on the cell face with the larger inflow.

The fluid volume (weight) assigned to the new particle will be set equal to the volumetric flow rate into the cell from the cell with the highest flow into the cells times the length of the transport step. However, the volume is restricted to be no more than either half the pore volume of the cell or half the total volume assigned to particles in the upstream cell. The fluid volume assigned to the new particle is not allowed to be less than or equal to zero. The fluid volume assigned to the new particle is removed from all the particles in the upstream cell in proportion to previous particle weight as a fraction of the total particle weight in the cell.

The concentration assigned to the new particle will be the sum of solute mass removed from particles in the upstream cell divided by the sum of the weights removed from the particles in the upstream cell.

This process of generating a new particle in a cell that becomes void of particles will greatly reduce the chances that the number of cells void of particles will exceed `FZERO`. However, if after making these adjustments to add new particles on the inflow face of a cell void of particles and the number of cells void of particles remains greater than the value of `FZERO`, then the simulation will be terminated and relevant information will be written to the output file to inform the user about the locations of the void cells. This will enable the user to increase the initial number of particles along critical pathlines to reduce the chance of this occurring again. One possible approach to this problem is to use MODPATH to track particles from these locations backwards to their source cells, and then increase the number of particles per node at those cells and rerun the simulation.

Constant-Concentration Boundary Condition

An optional package, Constant-Concentration Boundary Condition (CCBD) Package, has been added to the model to allow users to specify cells of the transport domain as constant-concentration boundary cells. At such cells, the concentration will remain constant at a user-defined value for each stress period. Concentration values must be specified for each stress period. This boundary condition is not associated with a fluid source or sink; therefore, its conceptual implementation is limited and should be used with the knowledge that mass will be created or removed from the boundary cell as a function of the change in concentration in the cell due to advection, dispersion, sources or sinks, decay, and other processes. This package is only available when using one of the volume-weighted particles options (MOCWT and MOCWTI, method of characteristics with volume-weighted particles using explicit and implicit finite-difference solutions, respectively, to calculate the dispersive flux).

At constant-concentration boundary cells, each particle in the cell will maintain the specified constant-concentration value. This ensures that even if the total volume of particles in the cell changes, the concentration of the cell will remain constant. If the concentration of a particle in the cell changes (due to dispersion, a source or sink term, or by other means), it is "corrected" and the mass either added or removed by this process is tracked. Similarly, if a particle enters a constant concentration boundary cell due to advection, its concentration is changed to the fixed concentration and any change in mass that occurs is also tracked. This mass that is either added or removed to each constant-concentration boundary cell is reported in the mass-balance section of the output as mass either entering or leaving the system at constant-concentration boundaries.

If the initial concentration in the aquifer is not equal to the concentration value specified in the constant-concentration boundary cell, the CCBD Package will set the initial concentration to that value and print a warning message. There is a similar consistency check for the initial concentration specified in the Double-Porosity (DP) Package. It is not compatible with direct age simulations (the AGE Package) or with the zero-order growth reaction in the DP or Simple Reactions (DK) Packages. If a cell in the DP or DK Package is defined with a nonzero zero-order growth, that value is reset to 0.0 by the CCBD Package and a warning is printed.

The CCBD Package is activated by including Ftype CCBD in the GWT name file. See "GWT Input Instructions" (appendix 1) for details on the format of the input data.

Noninitial Steady-State Stress Periods

A transition to a new steady-state stress period can involve a discontinuous and abrupt change in the volume of water contained in finite-difference cells. When using the volume-weighted particles option, this sudden change in volume does not have a concentration associated with it consistent with the solution of the solute-transport equation, and, hence, can induce a global mass-balance error if not treated correctly. Therefore, if a second or later stress period represents steady-state flow, whether following a stress period representing either transient or steady-state flow, the model checks to see if there is a change in volume in any cell at the transition to the new steady-state flow solution. If the new steady-state stress period resulted in changes in volume for water-table cells, the model will conserve solute mass in those cells while adjusting the weights on all particles in those cells to account for the change in volume in the cell. At the same time, the concentration associated with those particles will be adjusted proportionately in the opposite direction to assure that the solute mass represented by each particle will remain the same. This will assure a continuity of solute mass over the transition to a new steady-state flow stress period.

Model Testing and Evaluation

The capability of the MOCWT method to accurately solve the governing equations under a range of boundary conditions is demonstrated by using the same suite of test problems described by Konikow and others (1996) and summarized below. Details about the properties and boundary conditions for each of these tests are described by Konikow and others (1996). In addition, to help assess the utility of MOCWT relative to other numerical algorithms, a test case was developed for benchmarking purposes. This test problem represents a highly simplified approximation of a documented field problem that has more realistic and complex properties and boundaries. Numerical solutions using MOCWT were evaluated and compared to solutions for the same problem using alternatively available codes and solvers.

One-Dimensional Steady Flow

This relatively simple test case compares the numerical solution using MOCWT with the analytical solution of Wexler (1992). Properties and boundary conditions for this test are described in detail by Konikow (1996; table 11); the MOCWT solutions used four initial particles per cell. Results (fig. 7) are in excellent agreement with the analytical solutions for two different values of longitudinal dispersivity. Furthermore, there are no oscillations or loss of precision at nodes close to the source, as was evident in the original MOC solutions (Konikow, 1996; figs. 18–20). The mass balance was highly accurate, with a discrepancy of approximately 0.0001 percent.

Three-Dimensional Steady Flow

To further evaluate and test MOCWT for three-dimensional cases, we compare the numerical solutions with those of the analytical solution developed by Wexler (1992) for the case of three-dimensional solute transport from a continuous

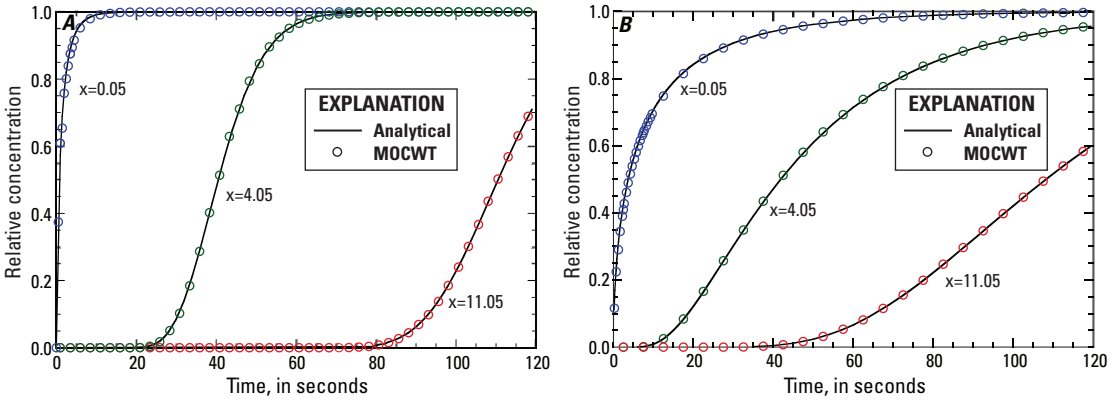

Figure 7. Numerical (MOCWT) and analytical solutions at three different locations for solute transport in a one-dimensional, steady flow field. *A*, α_L=0.1 cm (same as grid spacing); numerical solution data shown for every fifth point for clarity, except for x=0.05, where every point is shown for t<5 seconds (s). *B*, α_L=1.0 cm (10 times the grid spacing); numerical solution data shown for every 100th point for clarity, except for x=0.05, where every 10th point is shown for t<10 s.

point source in a steady, uniform flow field in a homogeneous aquifer of infinite extent. Properties and boundary conditions for this test are described in detail by Konikow (1996; table 12); the MOCWT solutions used 27 initial particles per cell. The results (fig. 8) show that the MOCWT solution very closely matches the analytical solution in the horizontal plane; similar close agreements exist in the vertical planes (not shown). The minor differences that exist are primarily attributable to differences in the nature of the source in the two solutions. That is, in the analytical solution, the source area represents a true point in space, whereas in the numerical solution the source term is inherently assumed to be distributed throughout the volume of the source cell, which has a surface area of 1.5 square meters (m^2) and a thickness of 0.05 meter (m). The mass balance was highly accurate, with a discrepancy of only about 2×10^{-6} percent. The slight waviness in the contours is an artifact of the contouring routine, which uses only the same nodal locations in both figures 8*A* and 8*B*.

Point Initial Condition in Uniform Flow

Another test problem to evaluate MOCWT includes three-dimensional solute transport from an instantaneous point source, or Dirac initial condition, in a uniform, steady, 3D flow field, which is described in more detail by Konikow and others (1996) and Kipp and others (1998). An analytical solution for this problem is given by Wexler (1992, p. 42) for the case of three-dimensional solute transport from a continuous point source; this analytical solution was modified for application to a case of an instantaneous point source (see Konikow and others, 1996). Properties and boundary conditions for this test case are described in detail by Konikow (1996; table 9); the MOCWT solutions used 25 initial particles per cell. The

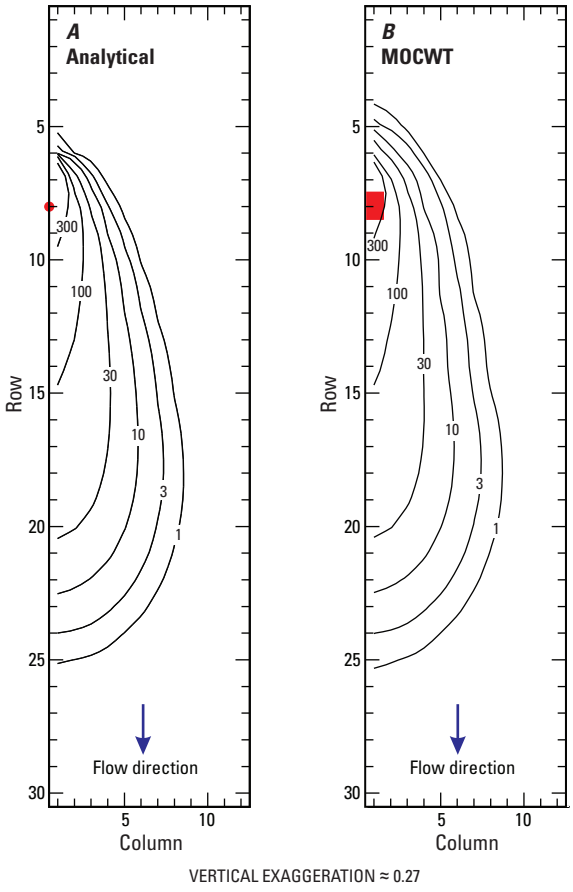

Figure 8. Concentration contours for *A*, analytical and *B*, numerical (MOCWT) solutions in the horizontal plane containing a continuous solute source (layer 1) for three-dimensional solute transport in a uniform steady flow field after 400 days. Red areas indicate the location of the continuous point source.

initial particle locations were specified by using the IPDL Package, and because flow is parallel to the *x*-direction, we used a 25×1×1 pattern of points in the *x*-, *y*-, and *z*-directions, respectively. The value of CELDIS was set to 0.25, where CELDIS is the maximum fraction of a cell dimension that a particle may move during one time increment, and is input in dataset 7 (appendix 1). The results (fig. 9) show that the MOCWT solution closely matches the analytical solution in the horizontal plane when flow is parallel to the *x*-axis; similar close agreements were achieved for a case in which the boundary conditions for the flow model were modified to generate uniform, steady, horizontal flow at 45 degrees to the grid (not shown).

The minor differences that exist are primarily attributable to differences in the nature of the initial condition source in the two solutions. That is, in the analytical solution, the initial mass represents a true point in space, whereas in the numerical solution the initial mass is inherently assumed to be distributed throughout the volume of the source cell, which has a surface area of about 11.1 m^2 and a thickness of 10.0 m. A small amount of numerical dispersion may also be contributing to the spreading in the numerical solution. The mass balance was highly accurate, with a discrepancy of only about 2×10^{-6} percent.

Benchmarking Test Case

A numerical experiment illustrates the possible effects of the numerical solution algorithm on the accuracy of the calculated concentrations and the efficiency of the solution. As described in more detail by Konikow (2011), the example implements a variety of solution algorithms used in two widely available public-domain solute-transport models (MT3DMS [Zheng and Wang, 1999] and MODFLOW–GWT [Konikow and others, 1996]) and compares selected results obtained after applying them to a hypothetical contamination problem for a nonreactive solute species.

As described by Konikow (2011), the test problem represents an analog based on, and greatly simplified from, the groundwater contamination problem at the Rocky Mountain Arsenal, Colorado (see Konikow, 1977). The aquifer is a thin, gently sloping, alluvial system with moderate hydraulic conductivity. The source of contamination (C'=1,000 milligrams per liter [mg/L]) is an unlined disposal pond, represented in the model as two adjacent injection wells. A freshwater reservoir (lake) is located on the north boundary of the model and a river is located along the south boundary of the model. It is assumed that the aquifer receives no recharge from precipitation and has a uniform hydraulic conductivity (K=1×10^{-4} meters per second [m/s]), an effective porosity that varies spatially in an uncorrelated random manner about a mean value of 0.20, and a steady-state two-dimensional flow field.

The boundary conditions produce groundwater flow that is generally from north to south, influenced by irregular lateral no-flow boundaries and two internal impermeable zones. The flow directions are also influenced by the river acting as a sloping constant-head boundary on the southern edge of the model domain (fig.10). Grid cells are 100 m on a side. No analytical (or "true") solution is available for this problem. It is not the goal of this numerical experiment to assess which model is better or best in any sense, in part because the relative strength of one method over another can change greatly, depending on the characteristics of the test problem. Instead, the goals are to demonstrate possible variability in answers

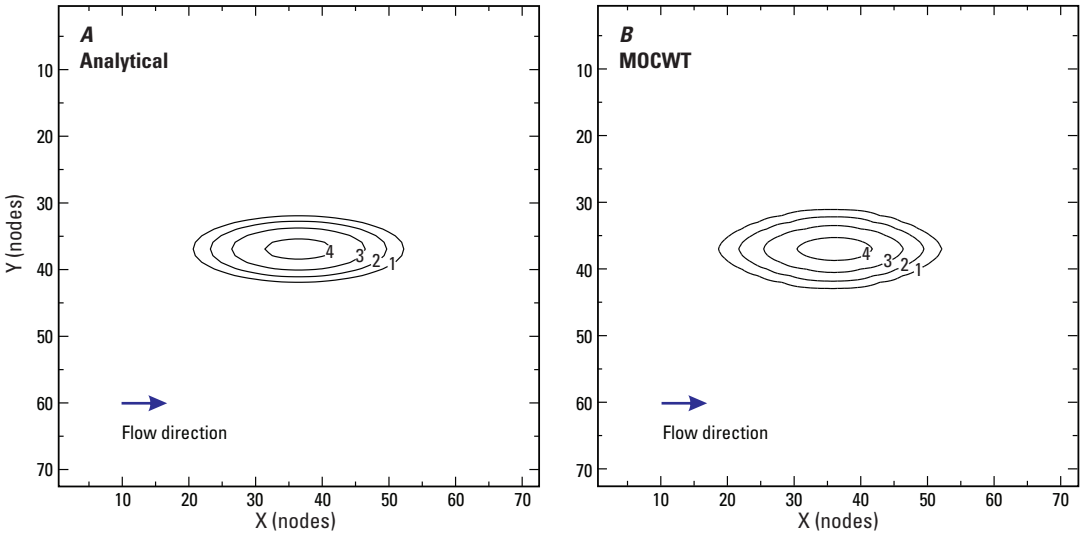

Figure 9. Concentration contours for *A*, analytical and *B*, numerical (MOCWT) solutions for transport of a point initial condition in a uniform flow field in the x-direction after 90 days. The z-component of flow is 0, but there is dispersion in all three directions. Contour values are the log of the concentrations.

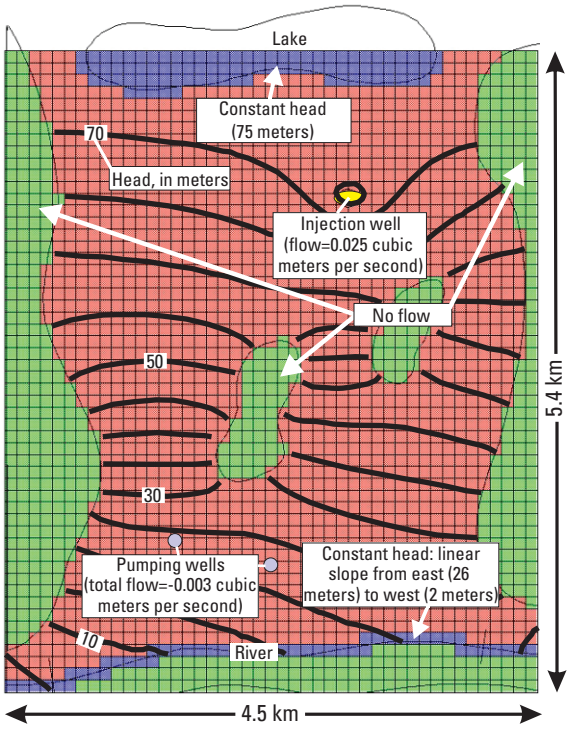

Figure 10. Diagram showing boundary conditions, finite-difference grid, and calculated steady-state heads for solute-transport test problem (from Konikow, 2011).

as affected by the choice of generic model and numerical algorithm while using a typical grid spacing for the scale of the problem, and to show how the new MOCWT algorithm might compare to other solution methods for one representative problem, further noting that even for a given test case, the results with any particular method can vary substantially, depending on the selection of numerical parameters required to implement that method.

This test problem was simulated for 20 years using various solution algorithms available in both MODFLOW–GWT and MT3DMS, including two finite-difference algorithms (FD and TVD) and eight different Eulerian-Lagrangian methods (ELLAM and seven varieties of MOC). Table 1 compares the results for these simulations for several measures of accuracy and efficiency. These results have been updated from Konikow (2011) using the most recent available versions of the models, so these results vary slightly from the previous results. Note that the indicated run times for the MT3DMS simulations do not include the small time required to solve the groundwater-flow equation using MODFLOW–2000 (approximately 0.3 s) because MT3DMS is run as a separate postprocessor that only solves the transport equation; however, MODFLOW–GWT simulations include solutions to both the flow and solute-transport equations, so the time required to solve the flow equation is included in the reported run times for the MODFLOW–GWT simulations. Also, the latest version of MT3DMS uses a new method to numerically solve the matrix equations (the General Conjugate-Gradient [GCG] solver), which improves run times substantially relative to those shown in Konikow (2011). Several methods produced noticeable undershoot and overshoot, especially the ELLAM results. The MOC methods (except for the weighted particle method) had notable mass-balance errors. The computational times varied by about a factor of 7. Note that the relative characteristics of each algorithm listed in table 1 are representative of the properties and characteristics of this particular test problem only, and the numerical solution parameters for each method were not optimized. Additional considerations are discussed by Konikow (2011).

The plumes calculated by all of the solutions indicate a plume that has spread to the south (towards the river) from the disposal pond, as would be expected based on the flow directions. But some differences exist in the details of the calculated solute concentration distributions among the various solutions (fig. 11). For example, a comparison of the MOCWT solution (11*A*) with the MT3DMS–MOC solution (11*B*) shows that the former indicates somewhat less lateral spreading of the lower concentrations and further downgradient migration of the higher concentrations. These differences can have substantial impacts on receptor and toxicity predictions. A comparison of the MOCWT solution (11*A*) with the MT3DMS–TVD solution (11*C*) shows that the calculated plumes are very similar. However, table 1 shows that there are differences among the solutions in terms of undershoot (negative concentrations, which are less than the initial background concentration of 0) and overshoot (values greater than 1,000 mg/L, which is the source concentration and represents an upper limit in this system).

The undershoot and overshoot are visualized by using threshold values of 0 and 1,000 mg/L when colorizing the calculated concentration values at 20 years (fig. 12). The results show that the MOCWT (12*A*) and MT3DMS–MOC (12*B*) solutions include small areas of slightly negative concentrations (between 0 and –0.4 mg/L), whereas the MT3DMS–TVD solution (11*C*) has somewhat greater negative concentrations (up to –9.1 mg/L) pervasive throughout the background field in an oscillatory pattern. Neither the MOCWT nor the MT3DMS–MOC solutions exhibit any overshoot, but the MT3DMS–TVD solution includes a relatively large area in the core of the plume emanating from the source that is calculated to exceed 1,000 mg/L (with a maximum of 1,045 mg/L). Undershoot and overshoot are common problems in the solution to the classical governing advection-dispersion equation.

The breakthrough curves for the various simulations at key observation points show a surprisingly large variability considering that all methods are solving the exact same set of boundary conditions in identical flow fields (fig. 13). The arrival times and concentration values at selected times differ by as much as 20 percent. As noted by Konikow (2011), these differences appear to be large enough to be of concern, and they arise solely from the choice of the numerical algorithm and the selected values for numerical parameters for that particular method.

Table 1. Comparison of several measures of accuracy and efficiency for simulating the test problem for 20 years using various solution algorithms available in public-domain solute-transport models.

[mg/L, milligram per liter; %, percent; and solution algorithms: MOC=method of characteristics; MOCIMP=method of characteristics with implicit finite-difference solution for dispersive flux; ELLAM=Eulerian-Lagrangian localized adjoint method; MOCWT=method of characteristics with volume-weighted particles; MOCWTI=method of characteristics with volume-weighted particles and implicit solution for dispersive flux; MMOC=modified method of characteristics; HMOC=hybrid method of characteristics; FD=finite-difference solution; TVD=total-variation-diminishing finite-difference method]

Variable	MODFLOW–GWT					MT3DMS (version 5.3)				
	MOC[1]	MOCIMP[1]	ELLAM[2]	MOCWT[3]	MOCWTI[3]	MOC[1]	MMOC	HMOC	FD	TVD
Transport time steps	190	99	99	190	99	84	84	84	84	173
Run time (seconds)[4]	3.6	2.2	4.1	3.0	1.9	1.2	0.72	0.905	0.69	2.1
Maximum concentration (mg/L)	1,010	1,004	1,351	1,000	999	992	968	976	1,076	1,045
Minimum concentration (mg/L)	–3.6	–7.5	–82	–0.4	–5.2	–0.3	0	0	–219	–9.1
Mass-balance error (%)	2.3	2.7	$7.2x10^{-5}$	$1.2x10^{-5}$	$7.0x10^{-6}$	–1.87	9.5	6.94	$–1.3x10^{-5}$	$–1.6x10^{-4}$

[1]16 particles per cell.

[2]ELLAM parameters: NSCEXP=NSREXP=2; NSLEXP=1; NTEXP=2.

[3] Initial number of particles per cell varies from 9 to 25.

[4]MT3DMS run times are for transport only and do not include time to solve flow equation using MODFLOW.

Notes: All simulations run on a Dell PC with Intel Xeon CPU E5502 1.87 GHz, Windows 7, 64-bit. CELDIS (in GWT)=PERCEL (in MT3DMS)=1.0.

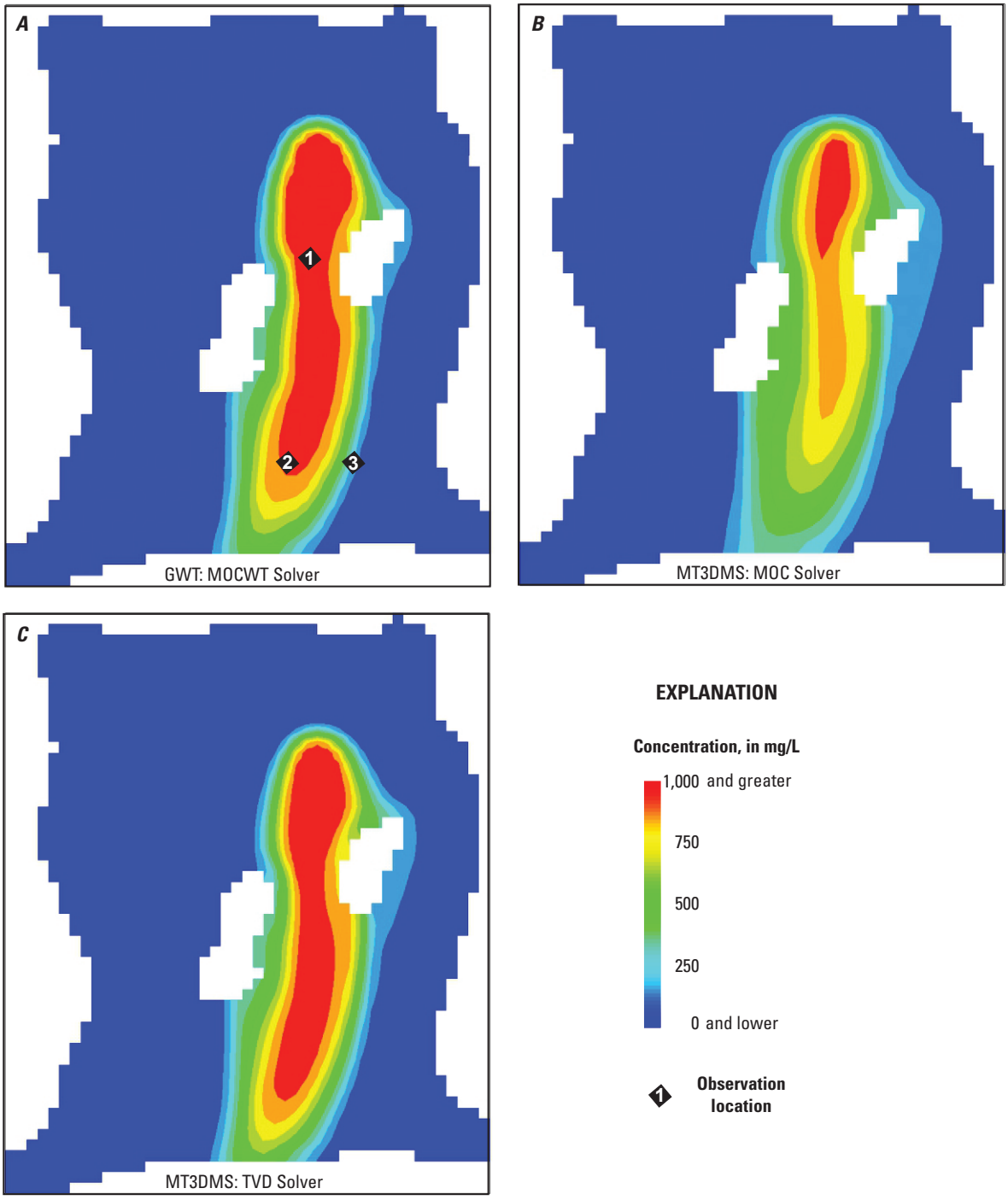

Figure 11. Illustration showing calculated concentrations after simulating 20 years of solute transport with three of the available solution algorithms for test problem with complex boundaries: *A*, MOCWT; *B*, MT3DMS–MOC; and *C*, MT3DMS–TVD. Colorized maps were derived by using Model Viewer software (Hsieh and Winston, 2002). White areas represent nonaquifer (impermeable) areas. Red colors include some areas with concentrations greater than 1,000 milligrams per liter (mg/L); blue colors include some areas less than 0 mg/L.

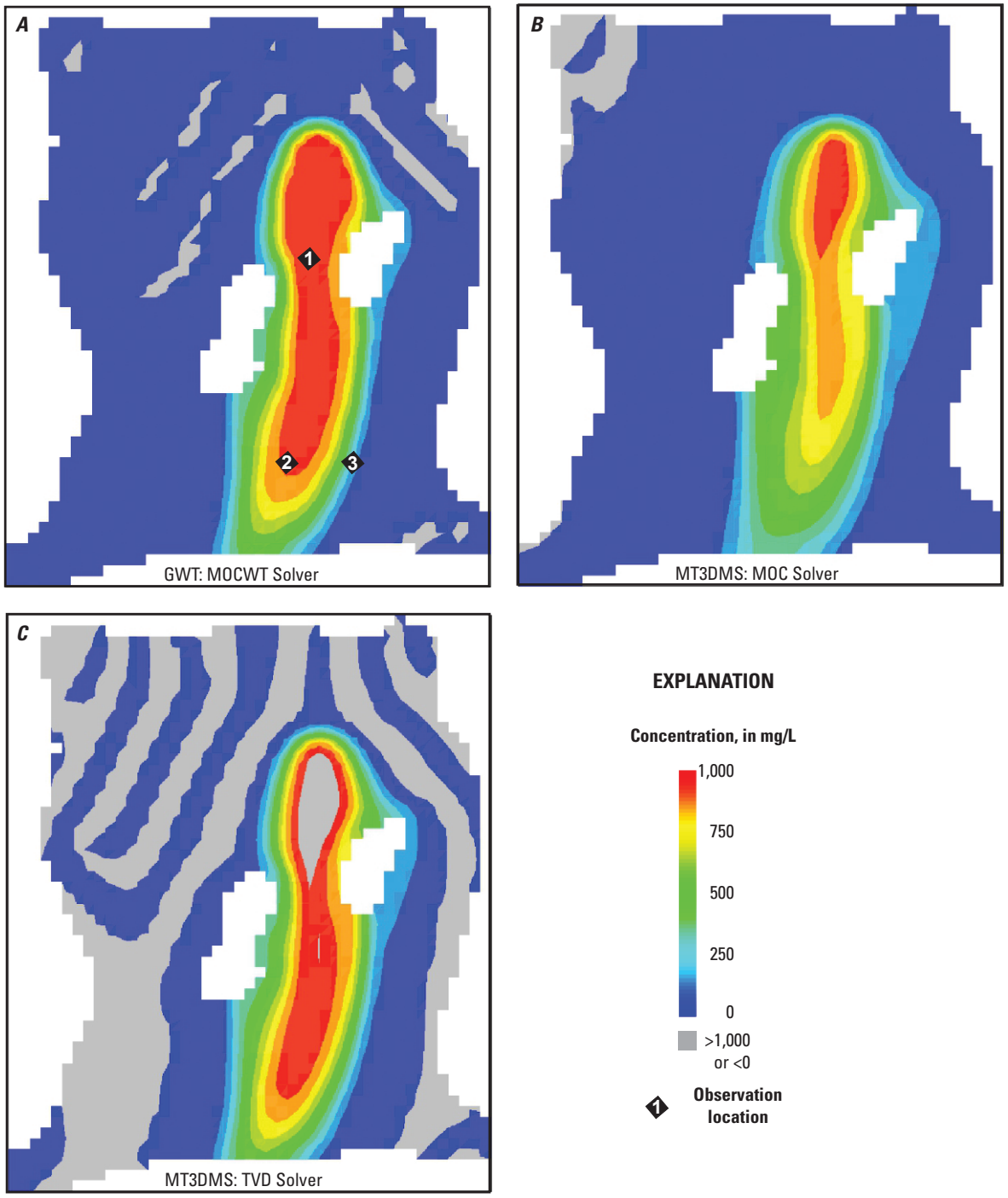

Figure 12. Illustration showing calculated concentrations after simulating 20 years of solute transport, as in figure 11, but showing areas of numerical undershoot (lower than background concentration) and overshoot (concentrations greater than the source concentration) for three of the available solution algorithms: *A*, MOCWT; *B*, MT3DMS–MOC; and *C*, MT3DMS–TVD. White areas represent nonaquifer (impermeable) areas. Gray areas within red colors have concentrations above 1,000 milligrams per liter; gray areas adjacent to blue colors have small negative concentrations.

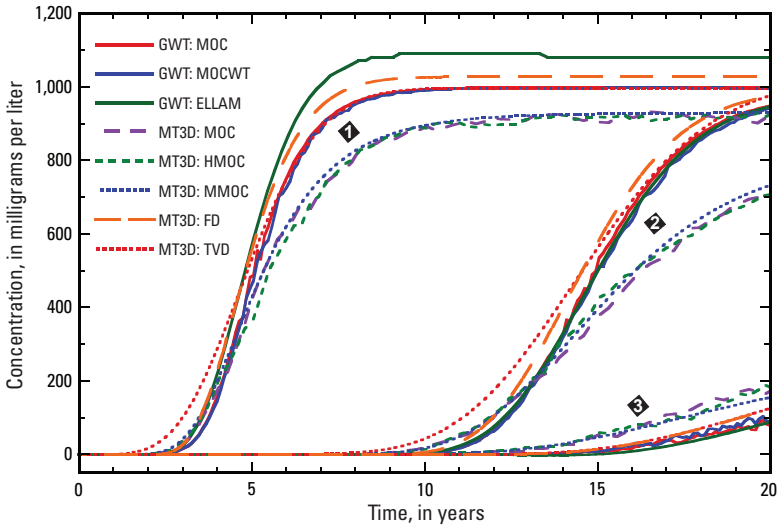

Figure 13. Comparison of simulated breakthrough curves at three observation points (locations shown in fig. 11*A*). Source concentration (*C′*) is 1,000 milligrams per liter (mg/L). Curves for the MOCIMP and MOCWTI solutions for GWT are not shown because they overlap closely with the MOC and MOCWT solutions.

Summary and Conclusions

In the traditional method of characteristics for groundwater solute-transport models, advective transport is represented by moving particles that track concentration. This approach can lead to global mass-balance problems (even if the grid spacing is uniform) because in aquifers with complex boundary conditions, particles can originate in cells having different pore volumes and (or) be introduced (or removed) at cells representing fluid sources (or sinks) of varying strengths. Use of volume-weighted particles means that each particle tracks solute mass. In source or sink cells, the changes in particle weights will match the volume of water added or removed through external fluxes. This enables the new method to conserve mass in source or sink cells as well as globally.

This approach also leads to potential efficiencies by allowing the number of particles per cell to vary spatially—using more particles where concentration gradients are high or where pathlines diverge, and fewer particles where gradients are low or pathlines are parallel. The approach also eliminates the need for the model user to have to distinguish between "weak" and "strong" fluid source (or sink) cells. The new model determines whether solute mass added by fluid sources in a cell should be represented by (1) new particles having weights representing appropriate fractions of the volume of water added by the source, or (2) distributing the solute mass added over all particles already in the source cell. The first option is more appropriate for the condition of a strong source. The latter option is more appropriate for a weak source. At sinks, decisions whether or not to remove a particle are replaced by a reduction in particle weight in proportion to the volume of water removed. Particles are removed if their weights approach zero.

The new weighted-particle algorithm was implemented as a solver option in the MODFLOW–GWT model. Two variants are available for calculating dispersive flux. The MOCWT option uses an explicit finite-difference method to calculate concentration changes caused by hydrodynamic dispersion, and the MOCWTI option uses an implicit finite-difference method to calculate concentration changes caused by hydrodynamic dispersion.

A new algorithm was applied in the MOCWT and MOCWTI solver options of MODFLOW–GWT for transferring concentration changes, between nodal (cell) values and advecting particles present within the cell, more precisely and realistically compared to currently used methods. The new method scales the changes and adjustments of particle concentrations relative to limiting bounds of concentration values determined from adjacent nodal values. The method preserves realistic concentration variations and gradients within a cell and precludes unrealistic undershoot or overshoot for concentrations of individual particles. If dispersion is causing concentrations in a cell to decrease during a time step, those particles in the cell having the highest concentration will decrease the most, and those having the lowest concentration will decrease the least, as would be expected conceptually. The converse is true if dispersion is causing cell concentrations to increase. Furthermore, if the initial concentration on a particle is outside the range of the bounding cell values, it will automatically be adjusted in the direction of the acceptable range of values.

A number of test cases demonstrate that the new MOCWT method works well and conserves mass. In the benchmarking test case, a comparison of the MOCWT results with those of the MODFLOW–GWT MOC results shows that overshoot is reduced (actually eliminated in this case), undershoot is reduced (negative concentrations are smaller),

and the mass-balance error is much less (essentially zero) in the MOCWT solution. The MOCWT results also compare favorably with the MT3DMS solutions in terms of accuracy, although the latter show faster computational times. In general, MOCWT will work best for advection-dominated transport problems and will work better than standard MOC approaches for cases involving strongly diverging or converging flow fields, such as are generated by one- and two-well tracer tests, and for heterogeneous aquifers in which the pore volume varies substantially among grid cells. The new algorithm (MOCWT) is implemented as a solver option in the U.S. Geological Survey's MODFLOW–GWT solute-transport model.

Acknowledgments

The authors appreciate the helpful review comments provided by USGS colleagues P.E. Misut and Eric Morway.

Software

The report describes the Volume-Weighted Particle-Tracking Method for Solute-Transport Modeling (MODFLOW–GWT) for the U.S. Geological Survey (USGS) modular hydrologic simulation program called MODFLOW–2000. The program can be downloaded from the USGS for free. The USGS software release is documented by Winston and others (2017). The performance of MODFLOW–GWT has been tested in a variety of applications. Future applications, however, might reveal errors that were not detected in the test simulations. Users are requested to send notification of any errors found in this model documentation report or in the model program to the MODFLOW contact listed on the web page. Updates might be made to both the report and to the model program. Users can check for updates on the MODFLOW–GWT web page (https://water.usgs.gov/nrp/gwsoftware/mf2k_gwt/mf2k_gwt.html, https://doi.org/10.5066/F78050RV).

References Cited

Bredehoeft, J.D., 1971, Comment on "Numerical solution to the convective diffusion equation," by C.A. Oster, J.C. Sonnichseann, and R.T. Jaske: Water Resources Research, v. 7, no. 3, p. 755–756.

Goode, D.J., 1999, Age, double porosity, and simple reaction modifications for the MOC3D ground-water transport model: U.S. Geological Survey Water-Resources Investigations Report 99–4041, 34 p.

Halford, K.J., and Hanson, R.T., 2002, MODFLOW–2000, User guide for the drawdown-limited, Multi-Node Well (MNW) Package for the U.S. Geological Survey's modular three-dimensional ground-water flow model, versions MODFLOW–96 and MODFLOW–2000: U.S. Geological Survey Open-File Report 02–293, 33 p.

Harbaugh, A.W., 2005, MODFLOW–2005, The U.S. Geological Survey modular ground-water model—The ground-water flow process: U.S. Geological Survey Techniques and Methods, book 6, chap. A16, variously paged.

Harbaugh, A.W., Banta, E.R., Hill, M.C., and McDonald, M.G., 2000, MODFLOW–2000, The U.S. Geological Survey modular ground-water model—User guide to modularization concepts and the ground-water flow process: U.S. Geological Survey Open-File Report 00–92, 121 p.

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

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

Heberton, C.I., Russell, T.F., Konikow, L.F., and Hornberger, G.Z., 2000, A three-dimensional finite-volume Eulerian-Lagrangian Localized Adjoint Method (ELLAM) for solute-transport modeling: U.S. Geological Survey Water-Resources Investigations Report 00–4087, 63 p.

Hornberger, G.Z., Konikow, L.F., and Harte, P.T., 2002, Simulating horizontal-flow barriers using the MODFLOW Ground-Water Transport Process: U.S. Geological Survey Open-File Report 02–52, 28 p.

Hsieh, P.A., and Winston, R.B., 2002, User's guide to Model Viewer, A program for three-dimensional visualization of ground-water model results: U.S. Geological Survey Open-File Report 02–106, 18 p.

Kipp, K.L., Konikow, L.F., and Hornberger, G.Z., 1998, An implicit dispersive transport algorithm for the U.S. Geological Survey MOC3D solute-transport model: U.S. Geological Survey Water-Resources Investigations Report 98–4234, 54 p.

Konikow, L.F., 1977, Modeling chloride movement in the alluvial aquifer at the Rocky Mountain Arsenal, Colorado: U.S. Geological Survey Water-Supply Paper 2044, 43 p.

Konikow, L.F., 2010, Applying dispersive changes to Lagrangian particles in groundwater transport models: Transport in Porous Media, v. 85, no. 2, p. 437–449.

Konikow, L.F., 2011, The secret to successful solute-transport modeling: Ground Water, v. 49, no. 2, p. 144–159.

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

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, 87 p.

Konikow, L.F., and Hornberger, G.Z., 2003, Use of boundary fluxes when simulating solute transport with the MODFLOW ground-water transport process: U.S. Geological Survey Open-File Report 03–303, 17 p.

Konikow, L.F., Hornberger, G.Z., Halford, K.J., and Hanson, R.T., 2009, Revised multi-node well (MNW2) package for MODFLOW ground-water flow model: U.S. Geological Survey Techniques and Methods, book 6, chap. A30, 67 p.

McDonald, M.G., and Harbaugh, A.W., 1988, A modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Techniques of Water-Resources Investigations, book 6, chap. A1, 576 p.

Merritt, M.L., and Konikow, L.F., 2000, Documentation of a computer program to simulate lake-aquifer interaction using the MODFLOW ground-water flow model and the MOC3D solute-transport model: U.S. Geological Survey Water-Resources Investigations Report 00–4167, 146 p.

Niswonger, R.G., and Prudic, D.E., 2005, Documentation of the Streamflow-Routing (SFR2) Package to include unsaturated flow beneath streams—A modification to SFR1: U.S. Geological Survey Techniques and Methods, book 6, chap. A13, 48 p.

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

Pollock, D.W., 1994, User's guide for MODPATH/MODPATH–PLOT, version 3: A particle tracking post-processing package for MODFLOW, the U.S. Geological Survey finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 94–464, 249 p.

Pollock, D.W., 2012, User Guide for MODPATH Version 6—A particle-tracking model for MODFLOW: U.S. Geological Survey Techniques and Methods, book 6, chap. A41, 58 p.

Sanford, W.E., and Konikow, L.F., 1985, A two-constituent solute-transport model for ground water having variable density: U.S. Geological Survey Water-Resources Investigations Report 85–4279, 88 p.

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

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

Winston, R.B., Konikow, L.F., and Hornberger, G.Z., 2017, Software Release: Volume-Weighted Particle-Tracking Method for Solute-Transport Modeling: Implementation in MF2K-GWT: U.S. Geological Survey software release, accessed October 5, 2017, at https://doi.org/10.5066/F78050RV.

Zheng, Chunmiao, and Wang, P.P., 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: U.S. Army Corps of Engineers Contract Report SERDP–99–1.

Appendix 1.

Appendix 1. Revised Data Input Instructions for Groundwater Transport Process (GWT) with MOCWT Option

The Groundwater Transport Process (GWT) is a solute-transport simulation package that is integrated with MODFLOW–2000. It is derived directly from the MOC3D model (Konikow and others, 1996), which had been integrated with MODFLOW–96 (Harbaugh and McDonald, 1996). Following is a set of instructions for preparing an input dataset for the **GWT** process. For more comprehensive descriptions of input parameters, options, and underlying assumptions, the user should also refer to Konikow and others (1996), Kipp and others (1998), Goode (1999), Heberton and others (2000), and Konikow and Hornberger (2003). One major change that has been implemented since the release of MOC3D Version 3.5 is the elimination of former item 18a, which was used to define the thickness of layers. This is no longer used because vertical discretization (and hence thickness) is now defined in the input data for MODFLOW–2000 (see Harbaugh and others, 2000). Another change is in the file type used in the MODFLOW name file. This version of the code is also compatible with the Lake and Gage Packages, the Constant-Head Boundary Package, the Streamflow-Routing Package, the Multi-Node Well Package (MNW2), and the Drain-Return Package. The Multi-Node Well Package Version 1 is no longer supported for solute transport.

MODFLOW Name File

Transport simulation is activated by including a record in the MODFLOW name file using the file type (Ftype) "GWT" to link to the transport name file. The transport name file specifies the files to be used when simulating solute transport in conjunction with a simulation of groundwater flow using MODFLOW. The transport name file works in the same way as the MODFLOW name file. (The Ftype "CONC" is not valid with MODFLOW–2000.)

MODFLOW Source and Sink Packages

Except for recharge and lakes, concentrations associated with fluid sources (C') are read as auxiliary parameters in the MODFLOW source package. The source concentration is read from a new column appended to the end of each line of the data file describing a fluid sink/source (see documentation for revised MODFLOW model; Harbaugh and McDonald, 1996a and 1996b). For example, concentrations associated with well nodes should be appended to the line in the WEL Package where the well's location and pumping rate are defined. These concentrations will be read if the auxiliary parameter "CONCENTRATION" (or "CONC") appears on the first line of the well input data file. The concentration in recharge is defined separately, as described in a following section, "Source Concentration in Recharge File." The preparation of input data files for using the Lake Package when solute-transport is also simulated is described in detail in the Lake Package documentation (Merritt and Konikow, 2000). The preparation of input data files for using the Streamflow-Routing Package (SFR2) when solute transport is also simulated is described in detail in the SFR2 documentation (Niswonger and Prudic, 2005). The preparation of input data files for using the Multi-Node Well Package (MNW2) when solute transport is also simulated is described in detail by Konikow and others (2009).

In MODFLOW, fluid sources and sinks are treated numerically as being effectively distributed throughout the volume of the cell. Some fluxes, however, actually represent flow across the aquifer boundary (for example, recharge typically represents flux into the aquifer across the top boundary of the aquifer—usually the top face of a cell in the uppermost active layer in the model). Representing such a recharge flux as a distributed source term would be consistent with having a vertical velocity of zero on the top face. In the presence of recharge, however, the vertical velocity at the water table boundary is actually nonzero. MODPATH recognizes this inconsistency and allows the user to specify that certain fluxes represent boundary fluxes, and it then assigns that flux to the appropriate or assigned face. For example, recharge would be assigned to the top face of a cell, and the velocity on that face would be computed as the recharge flux divided by the effective porosity of the cell. The difference between these two conceptualizations would affect interpolation results for estimating the velocity and pathline of a particle in a boundary cell. Thus, it would also affect concentrations calculated with GWT. Therefore, the GWT model has implemented an option to assign fluxes on boundaries in the same manner as in MODPATH. This is implemented by using the boundary flux input file (Ftype "BFLX") and (or) using the auxiliary parameter IFACE in list directed sink/source packages for MODFLOW.

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

GWT Input Data Files

All input variables are read using free formats, except as specifically indicated. In free format, variables are separated by one or more spaces or by a comma and optionally one or more spaces. Blank spaces are not read as zeros. Variables that are optional are enclosed in brackets, as in {option}.

Groundwater Transport Name File (GWT)

FOR EACH SIMULATION:

1. Data: FTYPE NUNIT FNAME

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

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

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

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

MOC, MOCIMP, ELLAM, MOCWT, or MOCWTI Main GWT input data file [required]. Specifying ***MOC*** indicates dispersion calculations will be explicit (as described by Konikow and others, 1996); specifying ***MOCIMP*** indicates dispersion calculations will be implicit (as described by Kipp and others, 1998); and specifying ***ELLAM*** indicates that the solute-transport equation will be solved with the ***ELLAM*** method (as described by Heberton and others, 2000). ***MOCWT*** and ***MOCWTI*** indicate particle tracking will be done using volume-weighted particles and explicit or implicit dispersion calculation, respectively.

IPDL or IPDA Input information to describe spatially varying initial locations of particles in either list-based format (***IPDL***) or array format (***IPDA***). Only use if weighted particle method is used and MOCWT or MOCWTI are specified for the main GWT input file *[optional]*.

CRCH Concentrations in recharge *[optional]*.

CNCA Separate output file containing concentration data in ASCII (text-only) format *[optional]*. Frequency and format of printing controlled by NPNTCL and ICONFM. If concentrations are written to a separate output file, they will not be written to the main output file.

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

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

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

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

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

PRTP Presence of this file type is a flag to indicate that the *z*-location of printed particle locations for PRTA and (or) PRTB options will be adjusted for changes in water-table elevation relative to cell dimensions. This will allow plotting packages, such as Model Viewer (Hsieh and Winston, 2002), to accurately plot relative positions of particles within cells in which the saturated thickness has changed. Only specify if PRTA or PRTB is active. A data file, filename, and unit number needs to be associated with this file type; however, the file does not need to include any data. *[optional]*.

PTOB Input file to indicate grid cells for which detailed information about particles are to be printed *[optional]*.

MBRP Separate output file with solute mass-balance components printed in a space-delimited spreadsheet. These data are printed after every transport time increment. *[optional]*.

MBIT Separate output file with solute mass-balance components as well as detailed itemization of mass transfer associated with flow packages (printed in a space-delimited spreadsheet). These data are printed after every transport time increment. *[optional]*.

OBS Observation wells input file *[optional]*.

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

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

AGE Groundwater age simulation input file *[optional]*.

DP Double porosity input file *[optional]*. (Not compatible with *ELLAM* option.)

DK Simple reactions (decay, zero-order growth, retardation) input file *[optional]*. (Not compatible with *ELLAM* option.)

CHFB Transport properties for Horizontal Flow Barriers and alternate calculation of dispersive flux near HFB cells *[optional]*. This option only should be used if the HFB Package is active.

BFLX Input file to convert certain distributed source or sink fluxes (recharge, evapotranspiration, and constant-head cells) to boundary fluxes *[optional]*.

CBDY Input file to specify spatially varying source concentrations in horizontal or vertical flow across boundaries of a transport subgrid *[optional]*.

SSTR Input file to specify that transient transport calculations will begin after the first stress period of the flow simulation *[optional]*.

CCBD Constant-concentration boundary input file *[optional]*. (Not compatible with *MOC*, *MOCIMP*, or *ELLAM* option.)

VBAL Input file to specify list of source cells where volume balancing is to be applied *[optional]*. (Not compatible with *MOC*, *MOCIMP*, or *ELLAM* option.)

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

FNAME The name of the file.

Notes:

AGE, DP, and DK file types are described by Goode (1999). The CHFB file type is described by Hornberger and others (2002). The CCBD file type is described in this report.

Files of type DATA and DATA(BINARY) can be designated as either input or output files. One of the options (either "OLD" for an input file or "REPLACE" for an output file) may be placed after the file name on the line listing the file type, unit number, and file name. If "OLD" is specified, the file must exist when the program is started. If "REPLACE" is specified and the file exists when the program is started, the existing file is deleted and then opened. The status of each file ("OLD," "REPLACE," or "UNKNOWN") is now shown in the output file. Note that the "OLD" or "REPLACE" option is not required. If neither is listed, the file status is shown as "UNKNOWN" and program execution continues normally. When output to a BINARY file from an earlier model run exceeds the amount of output generated by the current model run, specifying "REPLACE" may be required to ensure the file does not include output from the previous run after the output generated by the current run. The options may be entered in any combination of uppercase and lowercase letters.

Main GWT Package Input (*MOC, MOCIMP, MOCWT, MOCWTI,* or *ELLAM*)

Input for the solute-transport package is read from the unit specified in the transport name file. The input consists of up to 19 separate items, as described in detail below (note that item numbers do not necessarily correspond with line numbers in the file). These data are used to specify information about the transport subgrid, physical and chemical transport parameters, numerical solution variables, and output formats. Output file controls for the GWT package are specified in the transport name file, described previously.

FOR EACH SIMULATION:

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

2. Data: HEDMOC (continued)

3. Data: ISLAY1 ISLAY2 ISROW1 ISROW2 ISCOL1 ISCOL2

ISLAY1 — Number of first (uppermost) layer for transport.
ISLAY2 — Last layer for transport.
ISROW1 — First row for transport.
ISROW2 — Last row for transport.
ISCOL1 — First column for transport.
ISCOL2 — Last column for transport.

Notes:

Transport may be simulated within a subgrid, which is a "window" within the primary MODFLOW grid used to simulate flow. Row and column numbers specified here are those in the MODFLOW grid. Within the subgrid, the row and column spacing must be uniform if FTYPE *MOC*, *MOCIMP, MOCWT,* or *MOCWTI* are specified in the transport name file, but subgrid spacing can vary as in MODFLOW if *ELLAM* is specified. The thickness can vary from cell to cell and layer to layer. However, the range in thickness values (or product of thickness and porosity) should be as small as possible, although this restriction is relaxed for *MOCWT* and *MOCWTI*.

4. Data: NODISP DECAY DIFFUS

NODISP — Flag for no dispersion (set NODISP=1 if no dispersion in problem; this will reduce storage allocation; in most cases, NODISP=0).
DECAY — First-order decay rate [1/T] (DECAY=0.0 indicates no decay occurs).
DIFFUS — Effective molecular diffusion coefficient [L^2/T].

Notes:

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

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

IF Ftype *MOC*, *MOCIMP, MOCWT*, OR *MOCWTI* IS ACTIVE, AND *IPDL* AND *IPDA* ARE INACTIVE:

5a. Data: NPMAX NPTPND

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

NPMAX=2×NPTPND×NSROW×NSCOL×NSLAY.

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

IF Ftype *ELLAM* IS ACTIVE:

5b. Data: NSCEXP NSREXP NSLEXP NTEXP

NSCEXP Exponent used to calculate the number of subcells in the column direction (NSC, where NSC=2**NSCEXP).

NSREXP Exponent used to calculate the number of subcells in the row direction (NSR).

NSLEXP Exponent used to calculate the number of subcells in the layer direction (NSL).

NTEXP Exponent used to calculate the number of sub-time steps per transport time increment (NT).

Notes:

In general, numerical accuracy will be increased by increasing the value of these parameters. This will also, however, increase computational costs. For each of the four parameters above, the value represents the exponent y in the expression 2^y.

Entering a 0 or negative value for any of the above variables will cause the code to use default values. Default values for NSCEXP, NSREXP, and NSLEXP are 2 in active dimensions and 1 in inactive dimensions (for example, if a simulation represented a two-dimensional areal problem in which the number of rows and columns were greater than one and the number of layers equals one, then default settings would be NSCEXP=2, NSREXP=2, and NSLEXP=1, and the number of subcells in each direction would be 4, 4, and 2, respectively). The default value of NTEXP is 2.

IF 5a IS READ AND NPTPND IS NEGATIVE IN SIGN:

6. Data: PNEWL PNEWR PNEWC

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

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

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

Notes:

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

FOR EACH SIMULATION:

7. Data: CELDIS {FZERO} {INTRPL}

CELDIS Maximum fraction of cell dimension that particle may move in one step (typically, 0.5≤CELDIS≤1.0). For *ELLAM*, CELDIS can be greater than 1.0, and specifying CELDIS=0.0 will result in one transport time step being used (which is not generally recommended).

FZERO If the fraction of active cells having no particles exceeds FZERO, then if *MOC* or *MOCIMP* is active, program will automatically regenerate an initial particle distribution before continuing the simulation. If the *MOCWT* or *MOCWTI* options are active and this criteria is exceeded, an attempt will be made to create a new particle on an upstream face of the cell that is void of particles with mass transferred from the upstream neighboring cell; if FZERO is still exceeded after this process, the simulation is terminated and a list of the cells with zero particles is printed to the end of the main output file. The format for this list is compatible with the input for starting locations in *MODPATH* (Pollock, 2012). Typically, 0.01≤FZERO≤0.10. However, to ignore this criteria for *MOCWT* or *MOCWTI* simulations (or for "debugging" model runs in which completion is more important than accuracy), specify FZERO=1.0. Do not specify if *ELLAM* is active.

INTRPL Flag for interpolation scheme used to estimate velocity of particles. The default (INTRPL=1) will use a linear interpolation routine; if INTRPL=2, a scheme will be implemented that uses bilinear interpolation in the row and column (*j* and *i*) directions only (linear interpolation will still be applied in the *k*, or layer, direction). Do not specify if *ELLAM* is active, in which case the code will automatically set INTRPL=1.

IF *MOCIMP* OR *MOCWTI* IS ACTIVE:

7.1 Data: FDTMTH NCXIT IDIREC EPSSLV MAXIT

FDTMTH Weighting factor for temporal differencing of dispersion equation (0.0≤FDTMTH≤1.0). We suggest using either a value of FDTMTH=0.5, a centered-in-time (or Crank-Nicolson) approximation, or FDTMTH=1.0, a backward-in-time (or fully implicit) approximation. [Default value=1]

NCXIT Number of iterations for the explicitly lagged cross-dispersive flux terms (NCXIT≥1). We suggest that the user initially specify a value of 2, but if the solution exhibits significant areas of negative concentrations, then the value of NCXIT should be increased to require more iterations, which typically will reduce the extent and magnitude of negative concentrations (at the cost of increased computational time). [Default value=2]

IDIREC Direction index for permutation of the red-black node renumbering scheme. The order is as follows: 1: x,y,z; 2: x,z,y; 3: y,x,z; 4: y,z,x; 5: z,x,y; and 6: z,y,x. The first direction index is advanced most rapidly and the last direction index is advanced least rapidly. In some cases, there can be a significant variation in the number of iterations needed to achieve convergence, depending on the order of the directions for the red-black renumbering. We suggest that the user initially specify IDIREC=1. If this leads to a relatively large number of iterations (more than 10), then the user should experiment with alternate choices to determine the one requiring the fewest number of iterations for their particular problem. [Default value=1]

EPSSLV Tolerance on the relative residual for the conjugate-gradient solution of the matrix of the difference equations. We suggest that the user initially specify EPSSLV≤10^{-5}. An adequately small value of EPSSLV has the property that a smaller value does not change the numerical solution within the number of significant digits desired by the user. In the single-precision code implemented here, EPSSLV should not be less than 10^{-7}. [Default value=10^{-5}]

MAXIT Maximum number of iterations allowed for the iterative solution to the difference equations for dispersive transport. In most cases, MAXIT=100 is satisfactory. [Default value=100]

Notes:

Entering a zero or out-of-range value for any of these five variables will cause the code to use the indicated default value.

IF *MOCWT* OR *MOCWTI* IS ACTIVE:

7.2 Data: REMCRIT GENCRIT IRAND ISRCFIX

REMCRIT REMove CRITerion for calculating minimum particle weight in a cell below which low-weight particles in (a) cells that are net fluid sinks or (b) cells adjacent to strong sources when the ISRCFIX option is used are considered to be "trivial" and are removed from the particle tracking calculations. REMCRIT represents a fraction of the volume and mass in a cell; if both the weight and mass of a particle are less than this criterion, then it is removed. Particles will not be removed if the number of particles in the cell is less than the number of particles specified in the initial distribution of particles for the cell plus 8 (that is if NPTPND(t)>(NPTPND(0)+8)). The weight and mass of the removed particle are distributed to the remaining particles in the cell. [Default value=0.01]

GENCRIT GENerate CRITerion for determining whether a cell containing a fluid source should be considered "strong" or "weak." GENCRIT represents the fraction of the fluid flowing out of a cell that was derived from an external source; if the ratio of the net external source flux to the total flux out of the cell is greater than this criterion, the cell is flagged as "strong." If a cell contains a strong source, new particles will be generated (created) to represent and track the source fluid. If a cell contains a weak source, the source fluid will be represented and tracked by adjusting the weights and concentrations on existing particles in the cell. If ISRCFIX=1, volume balancing will be implemented in cells where GENCRIT indicates the presence of a strong source. [Default value=0.50]

IRAND Flag for determining method of initial positioning for newly generated particles in strong source cells. When a particle leaves a strong source, it is replaced by a new particle. If IRAND=0, the new particle is placed at the originating position of the particle that left. If IRAND>0, the particle is placed at a random location in the source cell. If IRAND=1, the "seed" value for the random number generation is itself selected randomly. If IRAND>1, the specified value of IRAND is used as the seed value.

ISRCFIX Flag for implementing volume balancing in source cells designated either by the value of GENCRIT or by listing in the VBAL Package. If ISRCFIX=0, the sum of particle weights in all source cells is allowed to change. If ISRCFIX=1, the sum of particle weights in designated source cells remains constant at the value of the fluid volume of the cell.

Notes:

Entering a 0 or out-of-range value for either of the first two variables will cause the code to use the indicated default value. Specifying the seed value by setting IRAND>1 allows the user to lock in the same sequence of "random" numbers, which may be desired when testing sensitivity of results to parameter variations.

FOR EACH SIMULATION:

8. Data: NPNTCL ICONFM NPNTVL IVELFM NPNTDL IDSPFM {NPNTPL}

NPNTCL Flag for frequency of printing concentration data. If NPNTCL=–2, concentration data will be printed at the end of every stress period; if NPNTCL=–1, data will be printed at the end of every flow time step; if NPNTCL=0, data will be printed at the end of the simulation; if NPNTCL=N>0, data will be printed every Nth particle moves, and at the end of the simulation. Initial concentrations are always printed. Solute budget and mass balance information are only printed every time concentration data are saved.

ICONFM Flag for output format control for printing concentration data. If concentration data are written to main output file (file type CNCA is not used), ICONFM represents a code indicating the format style (table 1–1, also see Harbaugh and McDonald, 1996a, p. 19). If concentration data are written to a separate output file (file type CNCA exists), specifying ICONFM≥0 will indicate that concentration data are to be written as a matrix of values for each layer of the subgrid, whereas specifying ICONFM<0 will indicate that concentration data are to be written as a table of values having one row for each node in the subgrid and four columns (x, y, z, and concentration), where x, y, and z are the actual nodal coordinates in the length units of the model simulation. In both cases for an external file, values are written using a format of (1P10E12.4). Note that we follow the *MODFLOW* convention in that y increases from top to bottom row, and z increases from top layer to bottom layer. Also note that the x and y values are given with respect to the entire *MODFLOW* grid, but the z location is calculated only for vertical distances within the layers of the transport subgrid. If data are written in matrix style, one header line precedes and identifies the data for each layer. If data are written as a table of values, one header line is written each time that concentration data are saved.

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

IVELFM Specification for format of velocity data, if being printed in main output file (see table 1–1).

NPNTDL Flag for printing dispersion equation coefficients that include cell dimension factors (see Konikow and others, 1996, p. 39–40). If NPNTDL=–2, coefficients will be printed at the end of every stress period; if NPNTDL=–1, coefficients will be printed at the end of the simulation; if NPNTDL=0, coefficients will not be printed; if NPNTDL>0, coefficients will be printed every Nth flow time step.

IDSPFM Specification for format of dispersion equation coefficients (see table 1–1).

NPNTPL Flag for printing particle locations in a separate output file (only read if file types "PRTA" or "PRTB" appear in the *GWT* name file). If either "PRTA" or "PRTB" is entered in the name file, initial particle locations will be printed to the separate file first, followed by particle data at intervals determined by the value of NPNTPL. If NPNTPL=–2, particle data will be printed at the end of every stress period; if NPNTPL=–1, data will be printed at the end of every flow time step; if NPNTPL=0, data will be printed at the end of the simulation; if NPNTPL>0, data will be printed every Nth particle moves, and at the end of the simulation. Only specify if *MOC, MOCIMP, MOCWT,* or *MOCWTI* is active.

Table 1–1. Fortran formats associated with print flags.

[Positive values for wrap format; negative values for strip format. Also see Harbaugh and McDonald, 1996a, p. 19[1]]

Print flag	Format
Print flag numbers 0–6	
0	10G11.4
1	11G10.3
2	9G13.6
3	15F7.1
4	15F7.2
5	15F7.3
6	15F7.4

Print flag	Format
Print flag numbers 7–13—Continued	
7	20F5.0
8	20F5.1
9	20F5.2
10	20F5.3
11	20F5.4
12	10G11.4
13	10F6.0

Print flag	Format
Print flag numbers 14–18—Continued	
14	10F6.1
15	10F6.2
16	10F6.3
17	10F6.4
18	10F6.5

[1]The Fortran formats specified here take the form rFw.d or rGw.d. For the "F" formats, the "r" represents the number of values printed per line, the "w" represents the number of characters used to represent the number, and the "d" represents the number of digits in the fractional part. The "G" formats are similar, except that the number of characters in the decimal part may be increased so that the number can be printed with greater precision.

FOR EACH SIMULATION:

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

FOR EACH LAYER OF THE TRANSPORT SUBGRID:

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

Module: U2DREL*

FOR EACH SIMULATION, ONLY IF TRANSPORT SUBGRID DIMENSIONS ARE SMALLER THAN FLOW GRID DIMENSIONS, AND ONLY IF *CBDY* PACKAGE IS INACTIVE:

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

Module: U1DREL*

Notes:

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

FOR EACH SIMULATION:

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

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

IF *NZONES*>0:

Data: IZONE ZONCON

IZONE Value identifying a particular zone.

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

Notes:

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

If heads or source concentrations associated with fixed-head cells vary with time, the CHD or FHB packages should be used to simulate those cells and to specify the associated source concentrations using auxiliary variables. If the source concentration value for a given fixed-head cell is specified both here (in dataset 12) and in CHD or FHB, the latter values will override the former values.

FOR EACH LAYER OF THE TRANSPORT SUBGRID IF *MOC* OR *MOCIMP* IS ACTIVE:

13. Data: IGENPT(NSCOL,NSROW) Flag to treat fluid sources and sinks as either "strong" or "weak."

Module: U2DINT*

Notes:

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

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

14. Data: ALONG(NSLAY) Longitudinal dispersivity.

Module: U1DREL*

15. Data: ATRANH(NSLAY) Horizontal transverse dispersivity.

Module: U1DREL*

16. Data: ATRANV(NSLAY) Vertical transverse dispersivity.

Module: U1DREL*

Notes:

Items 14–16 should include one value for each layer in subgrid.

FOR EACH SIMULATION:

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

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

Module: U1DREL*

Notes:

If RF=0.0 in input, the code automatically resets it as RF=1.0 to indicate no retardation. Spatially varying values of RF can be implemented as described by Goode (1999).

FOR EACH LAYER OF TRANSPORT SUBGRID:

18. Data: POR(NSCOL,NSROW) Cell porosity.

Module: U2DREL*

Notes:

The porosity is input as a separate array for each layer of the transport subgrid. The product of thickness and porosity should not be allowed to vary greatly among cells in the transport subgrid.

Source Concentration in Recharge File (CRCH)

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

FOR EACH STRESS PERIOD, IF RECHARGE PACKAGE USED:

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

Notes:

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

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

Module: U2DREL*

Initial Particle Density File—List-Based Input Format (IPDL)

Input information to describe spatially varying initial locations of particles for weighted-particle method. When using weighted particles, this input approach is most applicable when the same initial particle density is desired at most cells of the grid, and either higher or lower initial particle densities are desired at just a small number of cells.

1. Data: NPTLAY NPTROW NPTCOL NPTLIST NPMAX

NPTLAY The number of uniformly spaced particles to be placed initially in the layer direction in each cell (except for those cells defined in part 2 of this file).

NPTROW The number of uniformly spaced particles to be placed initially in the row direction in each cell (except for those cells defined in part 2 of this file).

NPTCOL The number of uniformly spaced particles to be placed initially in the column direction in each cell (except for those cells defined in part 2 of this file).

NPTLIST The number of cells in the grid for which the density of initially placed particles will differ from the uniform default values indicated by the above three parameters. The second part of this data file must include NPTLIST lines of data.

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

NPMAX Maximum number of particles available for particle tracking of advective transport. If set to 0, the model will automatically calculate NPMAX according to the following equation:

$$NPMAX=2\times NPGRID,$$

where NPGRID is the total number of particles in all cells.

IF *NPTLIST*>0:

2. Data: LAYER ROW COLUMN NPTLAY {NPTROW NPTCOL}

LAYER Layer in which cell is located.

ROW Row in which cell is located.

COLUMN Column in which cell is located.

NPTLAY If NPTLAY is a positive number, then for this particular cell it is defined as above for item 1. Specifying NPTLAY as a negative number indicates that the initial particles in this cell will be distributed in a radial pattern. The absolute value of NPTLAY defines how many particles are placed in the pattern and must be greater than 3. By default, the radius of the radial pattern is assumed to be 0.33 of the cell distances in the *x*- and *y*-directions. The relative *z*-coordinate is set to 0.0.

NPTROW As defined above for item 1. Not read if NPTLAY<0.

NPTCOL As defined above for item 1. Not read if NPTLAY<0.

Notes:

If the transport grid is only one cell wide in any direction and the number of particles specified to be placed in that direction is greater than one, then the code will automatically reduce the particle density in that direction (that is, NPTCOL, NPTROW, or NPTLAY) to a value of one.

There must be NPTLIST repetitions of item 2. Each record will be used to specify the initial density of particles generated in the layer, row, and column directions, respectively, for that one particular cell located at the indicated (LAYER,ROW,COLUMN) coordinates of the primary MODFLOW grid.

Initially distributing particles in a radial pattern may be useful for a cell in which a strong fluid source is located. In this case, the absolute value of NPTLAY must be greater than 3, although higher values are recommended. If the cell dimensions are equal, the pattern will be a circle. If cell dimensions are not equal, the pattern will be distorted in an elliptical shape relative to the magnitude of the dimensions.

Initial Particle Density File—Array-Based Input Format (IPDA)

Input information to describe spatially varying initial locations of particles for weighted-particle method. When using weighted particles, this input approach is most applicable when the initial particle density will vary among most cells of the grid.

1. Data: NPMAX

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

$$NPMAX=2\times NPGRID,$$

where NPGRID is the total number of particles in all cells.

FOR EACH LAYER OF THE TRANSPORT SUBGRID:

2. Data: NPTLAYA(NSCOL,NSROW) Initial number of weighted particles generated in layer direction.

Module: U2DINT*

NPTLAYA Two-dimensional array defining the number of uniformly spaced particles to be placed initially in the layer direction in each cell.

3. Data: NPTROWA(NSCOL,NSROW) Initial number of weighted particles generated in row direction.

Module: U2DINT*

NPTROWA Two-dimensional array defining the number of uniformly spaced particles to be placed initially in the row direction in each cell.

4. Data: NPTCOLA(NSCOL,NSROW) Initial number of weighted particles generated in column direction.

Module: U2DINT*

NPTCOLA Two-dimensional array defining the number of uniformly spaced particles to be placed initially in the column direction in each cell.

Notes:

If the transport grid is only one cell wide in any direction and the number of particles specified to be placed in that direction is greater than one, then the code will automatically reduce the particle density in that direction (that is, NPTCOLA, NPTROWA, or NPTLAYA) to a value of one. If any of the three parameters are to have a uniform value in a particular layer, the U2DREL array reading utility allows the specification of just a constant value for the entire array, thereby simplifying the input process (for a complete description of MODFLOW Array Reading Utilities, see Harbaugh and others, 2000, p. 86–88).

Volume Balancing File—List-Based Input Format (VBAL)

Input information to describe to which cells volume balancing should be applied. In general, listed cells should be fluid or solute sources. Particle tracking in these cells will be the same as if the cell was a strong fluid source.

1. Data: NBAL

NBAL The number of cells in the transport subgrid in which volume balancing should be applied. The second part of this data file must include NBAL lines of data.

2. Data: LAYER ROW COLUMN

LAYER Layer in which cell is located.
ROW Row in which cell is located.
COLUMN Column in which cell is located.

Notes:

This file is optional for use when ISRCFIX>0. The file will not be read if ISRCFIX=0.

There must be NBAL repetitions of item 2. Each record will be used to specify a particular cell where volume balancing will be implemented; such cells are located at the indicated (Layer,Row,Column) coordinates of the primary MODFLOW grid.

The volume balance will be applied in the cell for each stress period in which there is a fluid source to the cell.

Transport Boundary Flux File (BFLX)

Options to specify recharge, evapotranspiration, and specific constant-head fluxes as representing a boundary flux rather than a distributed flux are read from a separate unit specified in the *GWT* name file for ftype "BFLX."

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

In the same manner as in *MODPATH*, the user can choose to have all recharge applied (1) as a distributed source or sink term (which yields a zero flux and zero velocity condition on the top face of recharge cells), or (2) as a boundary flux (which, as in *MODPATH*, assigns the recharge flux as inflow across the top face of recharge cells). Similarly, the user can choose to have all evapotranspiration flux applied (1) as a distributed sink term, or (2) as a boundary flux (which, as in *MODPATH*, defines the evapotranspiration flux as an outflow across the top face of evapotranspiration cells, resulting in an upward velocity component on the top face of the cell). For constant-head cells that are adjacent to an aquifer boundary (or inactive cell), the user can assign the model-calculated constant-head flux to boundary faces.

For other stress packages, BLFX uses the auxiliary variable IFACE; see "MODFLOW Source and Sink Packages" above.

FOR EACH SIMULATION, IF RECHARGE PACKAGE USED:

1. Data: IRCHTP

IRCHTP Flag to indicate how recharge is applied within cell:

0 indicates distributed source or sink term

not 0 indicates recharge is applied as a boundary flux on the top face

Note:

If this package is not used to define IRCHTP, the model will assume IRCHTP=0.

FOR EACH SIMULATION, IF EVT OR ETS PACKAGE USED:

2. Data: IEVTTP

IEVTTP Flag to indicate how evapotranspiration is applied within cell:

0 indicates distributed sink term

not 0 indicates evapotranspiration is applied as a boundary flux on the top face

Notes:

If this package is not used to define IEVTTP, the model will assume IEVTTP=0.

FOR EACH SIMULATION:

3. Data: NCHNDS

NCHNDS Number of constant-head cells (defined in Basic Package by IBOUND<0, in CHD Package, or in FHB Package as constant-head cells) for which flux is to be assigned to a boundary face, as defined in the following list in dataset 4.

Notes:

For simplified input preparation in certain situations, the use of negative values will apply the boundary flux assignment to all constant-head cells within the transport subgrid, according to the following convention. If NCHNDS=–1, the source or sink flow term is distributed uniformly (per unit area) across any of the faces 1 through 4 that form boundaries with inactive cells (IBOUND=0) or are adjacent to the external boundary of the aquifer. If NCHNDS=–2, the source or sink flow term is distributed uniformly across any of the six faces that form boundaries with inactive cells (IBOUND=0) or are adjacent to an external boundary of the aquifer. If NCHNDS=–3, the source or sink flow term is assigned to the top face (6) of all constant-head cells located in the uppermost active layer in each row and column of the grid (that is, the uppermost active cell may lie in a different layer at various points within the grid). If NCHNDS=–4, the source or sink flow term is assigned to the bottom face (5) of all constant-head cells located in the lowermost active layer in each row and column of the grid (that is, the lowermost active cell may lie in a different layer at various points within the grid).

FOR EACH CONSTANT-HEAD CELL, IF NCHNDS>0:

4. Data: LAYER ROW COLUMN IFACE

LAYER	Layer of constant-head node.
ROW	Row of constant-head node.
COLUMN	Column of constant-head node.
IFACE	Cell face on which to assign constant-head flux.

Notes:

There must be NCHNDS lines or records in dataset 4. If NCHNDS is less than or equal to 0, dataset 4 is skipped. Following the convention for MODPATH (Pollock, 1994), if IFACE equals 0 or is greater than 6, the flow term is treated as an internal source or sink. If IFACE equals a number from 1 through 6, the flow term is assigned to the cell face corresponding to that value (using MODPATH conventions). (If the specified face is not a boundary face, the assignment will be ignored and the flux will remain internal to the cell.) If IFACE is less than 0, the source or sink flow term is distributed uniformly across any of the faces 1 through 4 that form boundaries with inactive cells (IBOUND=0) or are adjacent to an external boundary of the aquifer.

Concentration on Subgrid Boundary File (CBDY)

If a subgrid boundary is used in a transport simulation, the user must specify the concentration in fluid that enters the transport domain with flow across the boundary of the transport subgrid from those parts of the flow model domain that are excluded from the transport subgrid. This is normally defined in dataset 11 of the main *GWT* input file, which allows a separate but single value to be specified for each model layer. It may be desirable, however, to allow the concentration in the fluid flux across the subgrid boundary to vary spatially. This package allows the specification of spatially varying source concentrations to be associated with (1) vertical flow across a subgrid boundary from the layer above and (or) below the transport subgrid, if the number of layers in the transport subgrid is less than the number of layers in the *MODFLOW* simulation, and (2) lateral inflow across a subgrid boundary within a *MODFLOW* layer. The data are read from a separate unit specified in the *GWT* name file for ftype "CBDY."

FOR EACH SIMULATION, IF **CBDY** PACKAGE USED:

1. Data: {CINFLA(NSCOL,NSROW)} — *C′* to be associated with vertical fluid inflow across the upper boundary of the subgrid. Only read if ISLAY1>1.

Module: U2DREL*

2. Data: {CINXY(NSCOL,NSROW)} — *C′* to be associated with horizontal fluid inflow across the lateral boundaries of the subgrid. Data are read for all cells; however, only data for the cells on the boundary are used. At corner cells the user is responsible for entering an appropriate value (there can be two lateral faces at a corner; the code uses only one value for both faces). Only read if the lateral subgrid dimensions are smaller than the lateral flow grid dimensions.

Module: U2DREL*

3. Data: {CINFLB(NSCOL,NSROW)} — *C′* to be associated with vertical fluid inflow across the lower boundary of the subgrid. Only read if ISLAY2<NLAY, the number of layers in the *MODFLOW* grid.

Module: U2DREL*

Notes:

If this package is used, then the program will skip Data Set 11 in the main *GWT* input file. If a transport subgrid is not used, the data in this input file will not be read.

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

If the transport subgrid has the same row and column dimensions as the flow grid, CINXY (Record 2) should not be included in the CBDY input file.

Values are required for records 1 and (or) 3 only if there is a flow layer above the subgrid and (or) below the subgrid, respectively. For example, record 1 is skipped if layer 1 of the flow grid coincides with the uppermost layer of the transport subgrid (that is, ISLAY1=1).

Starting Stress Period File (SSTR)

MODFLOW–2000 includes the capability to simulate multiple stress periods in which some represent steady-state flow and some represent transient flow. This is commonly implemented to simulate predevelopment or natural conditions using an initial steady-state stress period, and to simulate transient conditions under modern developed (or stressed) conditions. Among other advantages, this yields an internally consistent set of heads for the start of the transient flow simulation.

Under such a simulation scenario, it may not be necessary or desired to simulate transient transport during the initial steady-state stress period for the flow simulation. When *MODFLOW* is set up to simulate multiple stress periods, this optional GWT Package provides the user the option to start the transport simulation in any subsequent stress period following the first one. When used, it might be common to start transport with the beginning of the second stress period. However, any stress period following the first can be specified for the initialization and start of the transport simulation.

If this package is activated, the model will read the SSTR input file to define the stress period number in which to start the transport simulation. The data are read from a separate unit specified in the *GWT* name file for Ftype "SSTR."

Because some of the input data for the transport simulation are read as auxiliary variables in the *MODFLOW* input files, and because the SSTR Package is implemented without making any changes to the *MODFLOW* input structure, the user must continue to define the solute auxiliary variables for all stress periods when auxiliary variables are read each stress period. However, the solute-related auxiliary variables defined for pretransport stress periods will not be used or applied in the transport simulation. For example, the Well Package allows the use of "AUXILIARY CONC" to let the user define the source fluid concentration associated with an injection well. If the user indicates that an auxiliary variable is to be used, *MODFLOW* input formats require that the auxiliary variable must be used for all stress periods.

FOR EACH SIMULATION:

1. Data: IPERGWT

IPERGWT Sequential number of the MODFLOW stress period in which the transport simulation starts.

Note:
IPERGWT should be ≥2.

Observation Well File (OBS)

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

FOR EACH SIMULATION, IF *OBS* PACKAGE USED:

1. Data: NUMOBS IOBSFL

NUMOBS Number of observation wells.

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

FOR EACH OBSERVATION WELL:

2. Data: LAYER ROW COLUMN UNIT

LAYER	Layer of observation well node.
ROW	Row of observation well node.
COLUMN	Column of observation well node.
UNIT	Unit number for output file.

Notes:

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

Layer, row, and column numbers are specified for the *MODFLOW* grid (and not for the optional transport subgrid).

Particle Observation File (PTOB)

For some studies, it is of interest to know not only the mean concentration in a cell, but also the statistical characteristics (such as range or variance) of the distribution of concentrations of all particles within the cell from which the mean is calculated. Nodes of the transport subgrid, or groups of nodes, can be designated as "particle observation" locations. At each such node, the cell coordinates (column, row, and layer in the main *MODFLOW* grid), total simulation time, particle concentration, and volumetric discharge rate into external sinks (two columns are reported: one with rate from standard sinks [QSINK], and one with rate from Multi-Node Wells that include nodes with both inflow and outflow [QMNWSINK]) after each move increment will be written to a separate output file to facilitate statistical postprocessing of the distribution of particle concentrations in the cell. There are also two blank columns that are reserved for use in a future release of *GWT*. The input file for specifying particle observations is read if the file type (Ftype) "PTOB" is included in the GWT name file. The model distinguishes between cells comprising a Multi-Node Well (MNW) and other cells or groups of cells.

FOR EACH SIMULATION, IF *PTOB* PACKAGE USED:

1. Data: NUMPTOB NUMPTOB_MNW

NUMPTOB	Number of non-MNW particle observation locations.
NUMPTOB_MNW	Number of MNW particle observation locations.

FOR EACH PARTICLE OBSERVATION LOCATION DEFINED BY NUMPTOB:

2. Data: LAYER ROW COLUMN UNIT

LAYER	Layer of particle observation node.
ROW	Row of particle observation node.
COLUMN	Column of particle observation node.
UNIT	Unit number for output file.

FOR EACH PARTICLE OBSERVATION LOCATION DEFINED BY NUMPTOB_MNW:

3. Data: WELLID UNIT

WELLID	Name of multi-node well, as used in input dataset 2a for MNW2 Package (see Konikow and others, 2009, p. 46).
UNIT	Unit number for output file.

Notes:

Layer, row, and column numbers are specified for the MODFLOW grid (and not for the optional transport subgrid).

Unit numbers must be matched to a DATA file type and file name in the *GWT* name file. The volumetric discharge, volume removed from the particle, and mass removed from the particle written to the output are a summation of, volume removed from the particle, and mass removed from the particle account for all sinks that remove water from the cell.

If NUMPTOB>1, the unit numbers for each particle observation location do not need to be unique; observations with the same unit number will be written to the same file and, in this way, particles associated with a group of nodes can be tracked.

If NUMPTOB_MNW>1, the user must supply the name of a multi-node well, and each of the cells in that well will be used to record data. This file prints particle records only for cells where there is flow into the MNW (that is, the MNW is a sink in the cell). If more than one MNW is defined for a cell, the volumetric discharge will include the summed flows into each of the wells. For stress periods when the MNW is inactive, particle records will not be printed.

Constant-Concentration Boundary (CCBD)

This package can be used to simulate a solute-transport boundary condition in which the concentration remains constant during a stress period. For each designated constant-concentration node, the concentration value of the boundary condition must be specified. The input file for specifying CCBD cells is read if the file type (Ftype) "CCBD" is included in the *GWT* name file. The CCBD Package is only compatible with a volume-weighted particle option (*MOCWT* or *MOCWTI*). It is not compatible with direct-age simulations (AGE Package).

FOR EACH STRESS PERIOD, IF *CCBD* PACKAGE USED:

1. Data: ITMP

ITMP — If ITMP≥0, the number of constant-concentration boundary cells this stress period.

If ITMP<0, constant-concentration boundary data from the previous stress period will be used (this option not valid for first stress period).

FOR EACH CONSTANT-CONCENTRATION BOUNDARY CELL:

2. Data: LAYER ROW COLUMN CCCONC

LAYER — Layer of constant-concentration cell.

ROW — Row of constant-concentration cell.

COLUMN — Column of constant-concentration cell.

CCCONC — Constant-concentration value that will be fixed in cell (CCCONC must be ≥0.0).

Notes:

If the initial concentration (defined in dataset 10) of a CCBD cell does not match CCCONC for that cell, the CCBD value overrides and replaces the initial concentration value specified in dataset 10. Layer, row, and column numbers are specified for the MODFLOW grid (and not for the optional transport subgrid).

If the SSTR package is used, values should only be specified for stress periods in which the solute-transport equation is being solved.

ISSN 2328-7055 (online)
https://doi.org/10.3133/tm6A58