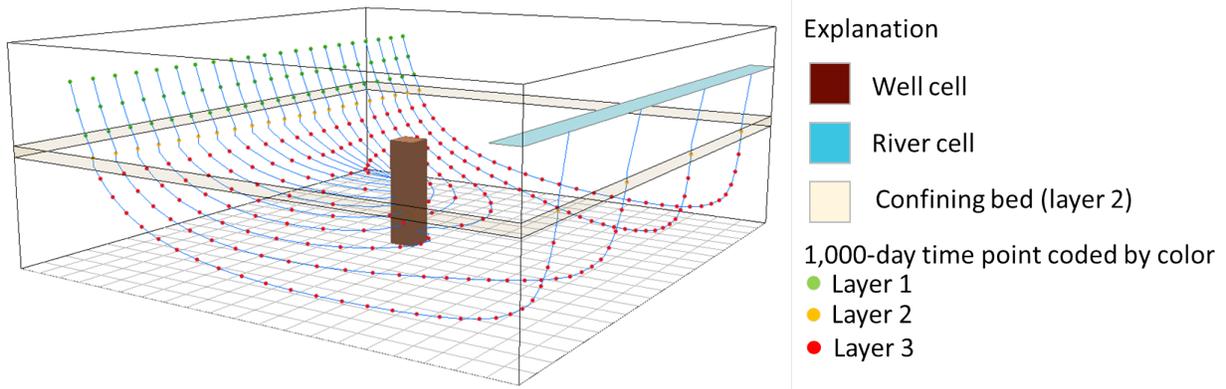




User Guide for MODPATH Version 7—A Particle-Tracking Model for MODFLOW

By David W. Pollock



Open-File Report 2016–1086

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

U.S. Department of the Interior
SALLY JEWELL, Secretary

U.S. Geological Survey
Suzette M. Kimball, Director

U.S. Geological Survey, Reston, Virginia: 2016

For more information on the USGS—the Federal source for science about the Earth, its natural and living resources, natural hazards, and the environment—visit <http://www.usgs.gov> or call 1-888-ASK-USGS (1-888-275-8747).

For an overview of USGS information products, including maps, imagery, and publications, visit <http://store.usgs.gov>.

Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Although this information product, for the most part, is in the public domain, it also may contain copyrighted materials as noted in the text. Permission to reproduce copyrighted items must be secured from the copyright owner.

Suggested citation:

Pollock, D.W., 2016, User guide for MODPATH Version 7—A particle-tracking model for MODFLOW: U.S. Geological Survey Open-File Report 2016–1086, 35 p., <http://dx.doi.org/10.3133/ofr20161086>.

ISSN 2331-1258 (online)

Cover Image: Diagram showing MODPATH three-dimensional perspective view of pathlines and time points for simulation 1A (see fig. 18, p. 27).

Preface

This report describes MODPATH, a particle-tracking model designed to work with output from the U.S. Geological Survey (USGS) groundwater flow model commonly referred to as MODFLOW. The performance of the program has been tested for a variety of test problems of simplified systems. Future applications might reveal errors that were not detected in the test simulations. Users should notify the USGS of any errors found in the report or the model program.

Although the computer program has been written and used by the USGS, no warranty, expressed or implied, is made by the USGS or the U.S. Government as to the accuracy and functionality of the program and related program material. Nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the USGS in connection therewith. MODPATH is available online from the USGS at the following address: <http://water.usgs.gov/ogw/modpath/>. Input instructions for MODPATH and additional information for using the program are included with the distribution. The distribution can be downloaded from the link shown above. Users can check the link shown above for updates to the computer program and for supporting documentation.

Contents

Preface	iii
Abstract	1
Introduction.....	1
Purpose and Scope	1
Method.....	2
Particle Tracking for Structured Grids	2
Particle Tracking for Unstructured Grids	8
Deformed Vertical Grids.....	13
Backward Tracking	15
Transient Flow	15
Solute Retardation	16
MODPATH Overview.....	16
Time Concepts.....	16
Flow Boundaries	17
Terminating Particles	19
Model Input and Output.....	20
Flow-System Files.....	20
Name File.....	21
Simulation File.....	21
Listing File	21
Debug Files	21
Particle Coordinate Files	21
Examples.....	22
Example 1 – Structured Grid.....	23
Simulation 1A – Forward-Tracking Pathlines with Time-Series Simulation	24
Simulation 1B – Forward-Tracking Endpoint Simulation.....	27
Example 2 – Unstructured Grid.....	28
Simulation 2A – Backward-Tracking Pathline and Time-Series Simulation	31
Simulation 2B – Backward-Tracking Endpoint Simulation.....	33
Summary	34
References Cited.....	35

Figures

1. Schematic of finite-difference cell showing the volumetric flow components across the cell faces.....	2
2. Schematic showing the computation of the exit point and travel time for the case of two-dimensional flow in the x-y plane.....	5
3. Diagrams showing additional combinations of velocities (v) between pairs of cell faces	7
4. Unstructured quadtree and quadpatch rectangular grids.....	8
5. Cell connections and face flows in a smoothed, unstructured grid	9
6. Definition of subcells for particle tracking with unstructured grids.....	10
7. Schematic of pathline produced using the subcell method	12
8. Schematic representation of an inclined aquifer with variable thickness	14

9.	Schematic representation of time concepts used by MODPATH.....	17
10.	Diagram showing definition of the cell face index, IFACE.....	18
11.	Conceptual model for hypothetical groundwater flow system.....	22
12.	Structured grid for example 1, used in the MODFLOW-2005 simulation.....	23
13.	MODPATH basic data file for example 1.....	24
14.	MODPATH name file.....	24
15.	MODPATH simulation file for simulation 1A.....	25
16.	MODPATH starting locations file for simulation 1A.....	25
17.	MODPATH map view of pathlines and time points for simulation 1A.....	26
18.	MODPATH three-dimensional perspective view of pathlines and time points for simulation 1A.....	27
19.	MODPATH simulation file for simulation 1B.....	27
20.	MODPATH starting locations file for simulation 1B.....	28
21.	MODPATH map view of starting locations for showing discharge locations for simulation 1B.....	28
22.	MODPATH unstructured grid for example 2.....	29
23.	Edited excerpt of the MODPATH unstructured grid file for example 2.....	30
24.	MODPATH name file for example 2.....	31
25.	MODPATH simulation file for simulation 2A.....	31
26.	MODPATH starting locations file for example 2A.....	32
27.	MODPATH pathlines generated by backward tracking from the well in simulation 2A.....	32
28.	Enlarged view of MODPATH pathlines for simulation 2A.....	33
29.	Backward-tracking endpoints for MODPATH simulation 2B.....	34
30.	Three-dimensional view of particle-tracking results for MODPATH simulations 2A and 2B.....	34

User Guide for MODPATH Version 7—A Particle-Tracking Model for MODFLOW

By David W. Pollock

Abstract

MODPATH is a particle-tracking post-processing program designed to work with MODFLOW, the U.S. Geological Survey (USGS) finite-difference groundwater flow model. MODPATH version 7 is the fourth major release since its original publication. Previous versions were documented in USGS Open-File Reports 89–381 and 94–464 and in USGS Techniques and Methods 6–A41.

MODPATH version 7 works with MODFLOW-2005 and MODFLOW–USG. Support for unstructured grids in MODFLOW–USG is limited to smoothed, rectangular-based quadtree and quadpatch grids.

A software distribution package containing the computer program and supporting documentation, such as input instructions, output file descriptions, and example problems, is available from the USGS over the Internet (<http://water.usgs.gov/ogw/modpath/>).

Introduction

MODPATH is a particle-tracking post-processing program designed to work with MODFLOW, the U.S. Geological Survey (USGS) finite-difference groundwater flow model. MODPATH version 7 is the fourth major release since its original publication (Pollock, 1989; Pollock, 1994; Pollock, 2012). MODPATH-7 works with both MODFLOW-2005 (Harbaugh, 2005) and MODFLOW–USG (Panday and others, 2013). Input and output file formats also have been updated to minimize the differences between structured and unstructured grid simulations and to make it easier for other applications to prepare the input files and to process the particle coordinate output files.

Purpose and Scope

The purpose of this report is to (1) describe the particle-tracking method used by MODPATH, (2) provide an overview of input and output data, and (3) present simple example simulations that illustrate the workflow associated with MODPATH particle-tracking simulations. Detailed descriptions of input data files, output file formats, and other aspects of model implementation that may change as MODPATH evolves are not presented in this report. Such information is included with the software distribution that is available over the Internet.

Method

Particle Tracking for Structured Grids

MODFLOW-2005 is based on a three-dimensional network of rectangular grid cells that is referred to in this report as a structured grid. Each structured grid cell has six faces that connect it to either a neighboring cell or a boundary of the system. Figure 1 shows a structured grid cell and the components of inflow and outflow across its cell faces (Q_x, Q_y, Q_z). The flow rate components shown in figure 1 are computed by MODFLOW and saved as output in the MODFLOW budget file for use by other programs. The flow data in the MODFLOW budget file are used by MODPATH to construct the groundwater velocity distribution that forms the basis for particle-tracking calculations.

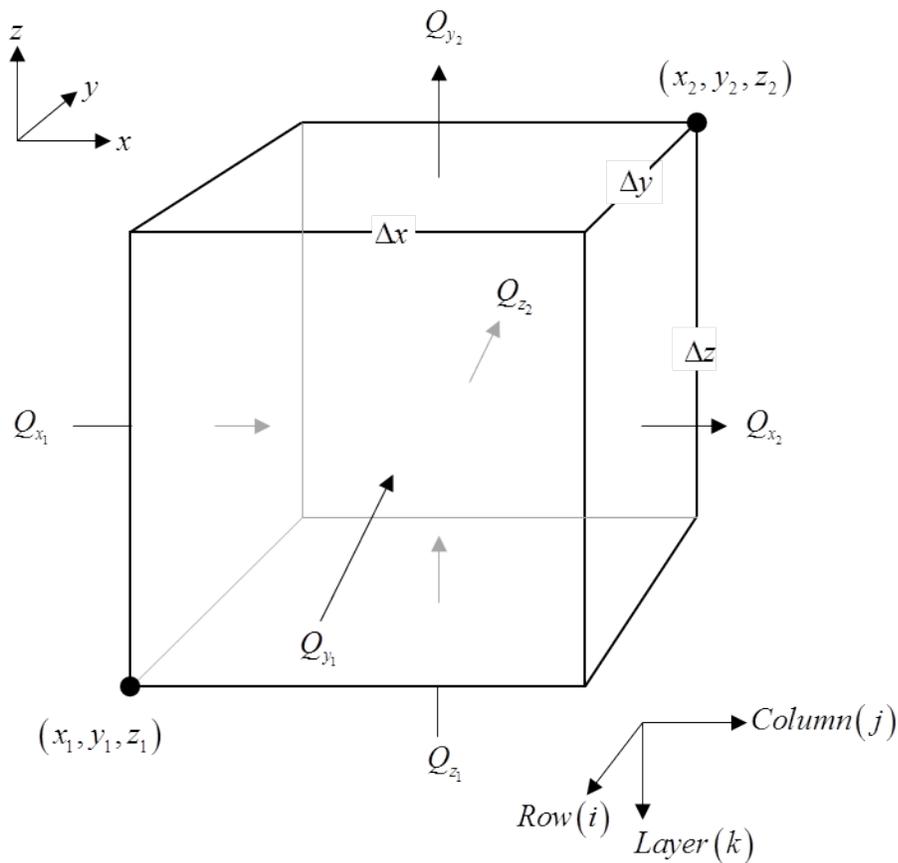


Figure 1. Schematic of finite-difference cell showing the volumetric flow components across the cell faces.

The principal velocity components must be defined at every point in the flow system in order to compute particle paths. Starting with the face flow rates shown in figure 1, the average linear velocity component across each face in cell (i, j, k) can be obtained by dividing the volumetric flow rate across the face by the cross-sectional area of the face and the porosity (n) of the material contained in the cell:

$$v_{x_1} = \frac{Q_{x_1}}{(n\Delta y\Delta z)}, \quad v_{x_2} = \frac{Q_{x_2}}{(n\Delta y\Delta z)} \quad (1a, 1b)$$

$$v_{y_1} = \frac{Q_{y_1}}{(n\Delta x\Delta z)}, \quad v_{y_2} = \frac{Q_{y_2}}{(n\Delta x\Delta z)} \quad (1c, 1d)$$

$$v_{z_1} = \frac{Q_{z_1}}{(n\Delta x\Delta y)}, \quad v_{z_2} = \frac{Q_{z_2}}{(n\Delta x\Delta y)}. \quad (1e, 1f)$$

Velocity components at points within the cell can be obtained by interpolating the velocity components at the cell faces. MODPATH uses simple linear interpolation to compute the velocity components within the cell,

$$v_x = A_x(x - x_1) + v_{x_1} \quad (2a)$$

$$v_y = A_y(y - y_1) + v_{y_1} \quad (2b)$$

$$v_z = A_z(z - z_1) + v_{z_1}, \quad (2c)$$

where A_x , A_y , and A_z are the components of the velocity gradient within the cell:

$$A_x = \frac{(v_{x_2} - v_{x_1})}{\Delta x}, \quad A_y = \frac{(v_{y_2} - v_{y_1})}{\Delta y}, \quad \text{and} \quad A_z = \frac{(v_{z_2} - v_{z_1})}{\Delta z}. \quad (3a, 3b, 3c)$$

The velocity component equations are independent of one another; v_x depends only on x , v_y depends only on y , and v_z depends only on z .

As a particle moves through a grid cell, its x component of velocity changes in time. The time rate of change in v_x can be expressed as

$$\frac{dv_x}{dt} = \left(\frac{dv_x}{dx} \right) \left(\frac{dx}{dt} \right). \quad (4)$$

By definition, $\frac{dx}{dt} = v_x$ and $\frac{dv_x}{dx} = A_x$; therefore, equation 4 can be rewritten as

$$\frac{dv_x}{dt} = A_x v_x, \quad (5)$$

and then rearranged to the form

$$\frac{dv_x}{v_x} = A_x dt. \quad (6)$$

Integrating equation 6 over the time interval t_1 to t yields

$$\ln \left[\frac{(v_x)_t}{(v_x)_{t_1}} \right] = A_x (t - t_1) . \quad (7)$$

Taking the exponential of each side of equation 7, substituting equation 2a for $(v_x)_t$, and rearranging yields the solution for the x coordinate of the particle as a function of time:

$$x_t = x_1 + \frac{1}{A_x} \left[(v_x)_{t_1} e^{A_x(t-t_1)} - v_{x_1} \right] . \quad (8a)$$

Analogous equations can be developed for the y and z coordinates of the particle:

$$y_t = y_1 + \frac{1}{A_y} \left[(v_y)_{t_1} e^{A_y(t-t_1)} - v_{y_1} \right] \quad (8b)$$

and

$$z_t = z_1 + \frac{1}{A_z} \left[(v_z)_{t_1} e^{A_z(t-t_1)} - v_{z_1} \right] . \quad (8c)$$

In equations 7 and 8, the subscripts t_1 and t refer to the location and velocity components of a particle at that specific value of time. The velocity components of the particle at time t_1 are known functions of the particle's coordinates; therefore, the coordinates of the particle at any future time, t , can be computed directly from equations 8a, 8b, and 8c.

The direct integration method described above can be implemented in a simple algorithm that allows the time at which a particle exits a cell, t_e , and its exit location, $(x, y, z)_{t_e}$, to be determined directly, given any known starting location within the cell. Figure 2 illustrates the method applied to a two-dimensional cell.

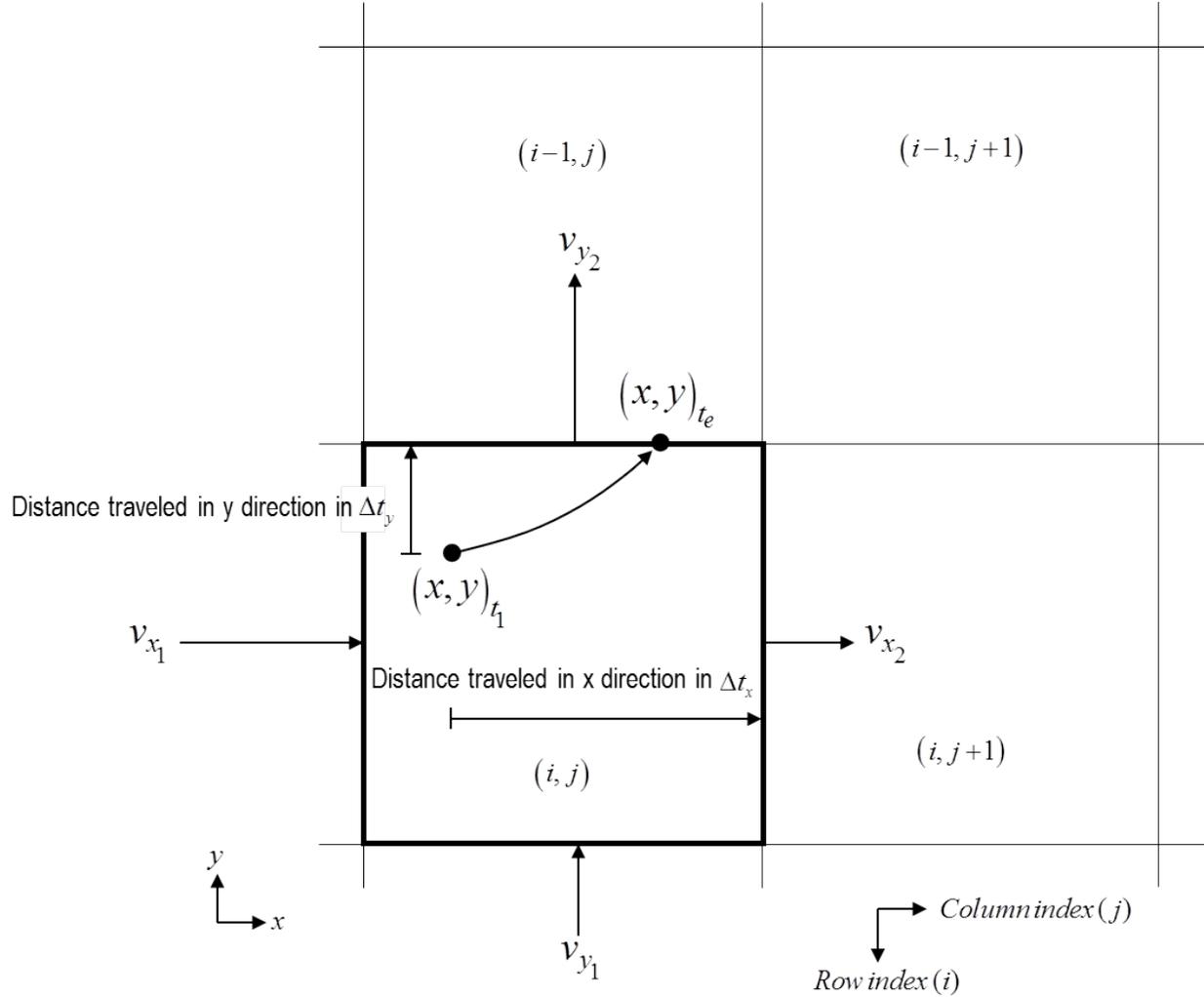


Figure 2. Schematic showing the computation of the exit point and travel time for the case of two-dimensional flow in the x-y plane.

At time t_1 , the cell contains a particle at a known location, $(x, y)_{t_1}$. The first step in computing the exit location is to determine which cell faces are potential exit faces for the particle. In this example, the velocities at the cell faces indicate that the particle potentially could exit the cell either along face x_2 or face y_2 . Consider the x direction first. The velocity v_x can be calculated at any location in the cell from equation 2a. Because v_x is equal to v_{x_2} at face x_2 , equation 7 can be used to compute the length of time required for the particle to move from $(x, y)_{t_1}$ to some point on face x_2 :

$$\Delta t_x = \frac{1}{A_x} \ln \left[\frac{v_{x_2}}{(v_x)_{t_1}} \right]. \quad (9a)$$

An analogous computation can be made for the time of travel in the y direction to some point on face y_2 :

$$\Delta t_y = \frac{1}{A_y} \ln \left[\frac{v_{y_2}}{(v_y)_{t_1}} \right]. \quad (9b)$$

If $\Delta t_x < \Delta t_y$, the particle must exit somewhere along face x_2 and the exit time is given by $t_e = t_1 + \Delta t_x$. Otherwise, if $\Delta t_y < \Delta t_x$, then the particle must exit somewhere along face y_2 and the exit time is given by $t_e = t_1 + \Delta t_y$. Once the exit time is known, the coordinates of the exit point can be computed by setting $t = t_e$ in equations 8a and 8b and solving for $(x, y)_{t_e}$. The sequence of calculations described above is repeated, cell by cell, until the particle either reaches a cell with no potential exit face or meets some other specified termination condition. The method often is referred to as semianalytical because particle paths are computed by a series of analytical solutions one cell at a time. For steady-state flow conditions, particles are moved from one cell to the next by a series of analytical solutions. Because flow rates change from one cell to the next, so do the parameters of the analytical solution within each cell. Although it is not necessary to compute intermediate particle locations within a cell, particle locations can be determined at any arbitrary point in time within a cell by substituting the appropriate value of time into equations 8a–8c and by computing the particle coordinates. Particle paths are built by repeatedly applying the algorithm described above as particles move from cell to cell.

The preceding example considered a specific case in which all of the velocity components at the cell faces were nonzero and in the positive x and y directions. Of course, those conditions will not always exist. Figure 3 illustrates the other possible situations that can occur in the direction of any of the three coordinates. Figure 3A shows the case in which v_{x_1} and v_{x_2} are in opposite directions and flow is into the cell through both faces x_1 and x_2 .

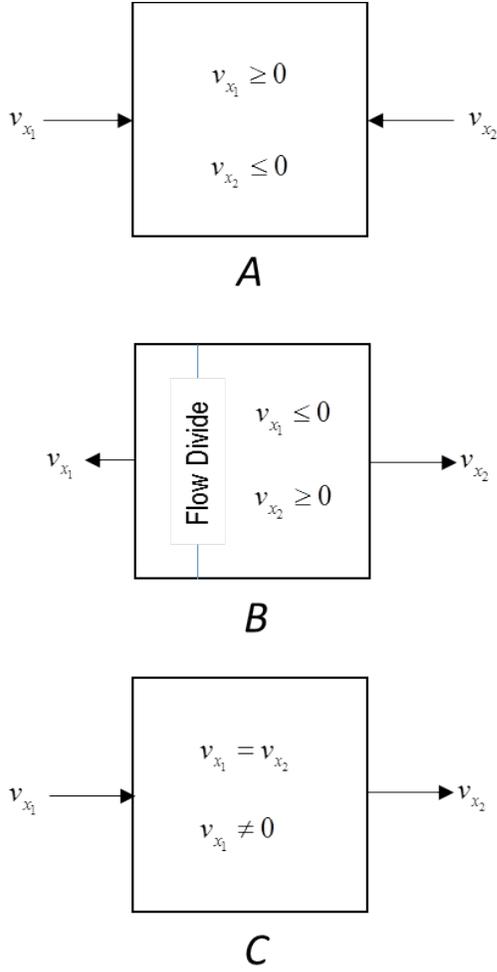


Figure 3. Diagrams showing additional combinations of velocities (v) between pairs of cell faces.

For this case, it is apparent that once a particle enters the cell, it cannot leave the cell in the x direction. When this condition exists in all three directions, it indicates that a strong sink is present and no outflow from the cell can occur across the cell faces. Figure 3B shows a second alternative in which v_{x_1} and v_{x_2} are in opposite directions and flow is out of the cell through faces x_1 and x_2 . This condition implies that a local flow divide exists for the x direction somewhere within the cell. For this situation, the potential exit face in the x direction is determined by checking the sign of v_x . If the sign is negative, the particle has the potential to leave the cell across face x_1 . If the sign is positive, the particle has the potential to exit across face x_2 . Once the potential exit face has been determined, the particle-tracking calculations for the x direction can proceed. Lastly, the case in which v_x is constant and nonzero (fig. 3C) must be considered special because the analytical expression in equation 9a is undefined and cannot be evaluated. In such situations, equation 9a is bypassed and the transit time in the x direction is computed from the following simple relation:

$$\text{If } v_{x_1} > 0, \text{ then } \Delta t_x = \frac{x_2 - x_{t_1}}{v_{x_1}}. \text{ Or, if } v_{x_1} < 0, \text{ then } \Delta t_x = \frac{x_1 - x_{t_1}}{v_{x_1}}. \quad (10a, 10b)$$

Particle Tracking for Unstructured Grids

The semianalytical particle-tracking algorithm described above requires rectangular cells with uniform velocity components at the cell faces. The method can be extended to rectangular-based unstructured grids. MODPATH supports smoothed quadtree and quadpatch grids like those shown in figure 4.

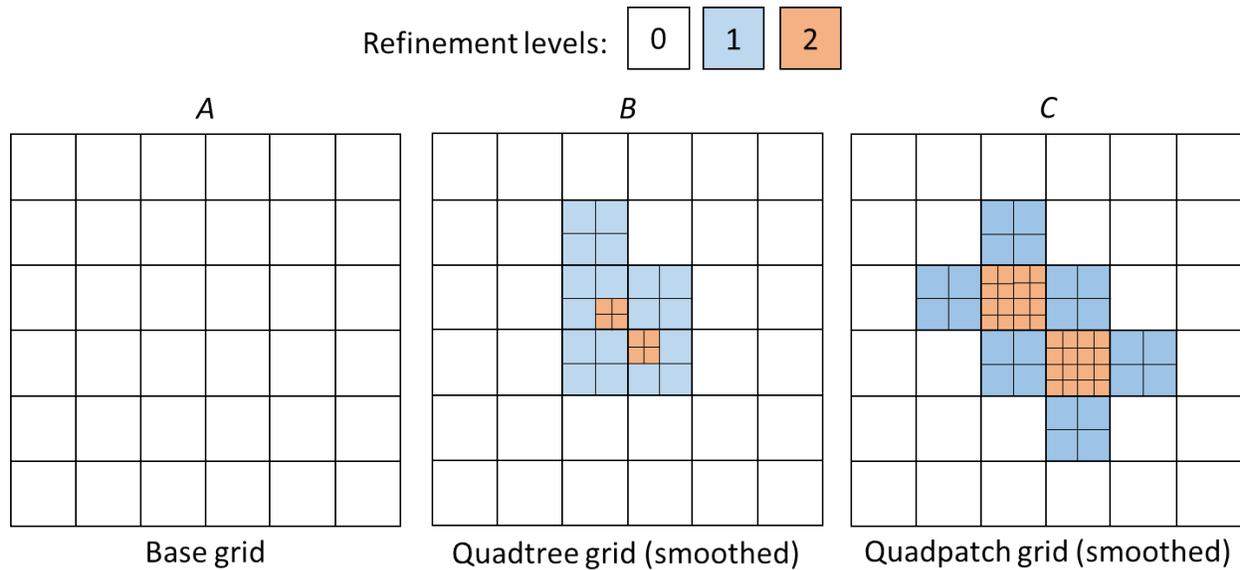


Figure 4. Unstructured quadtree and quadpatch rectangular grids.

Quadtree and quadpatch grids both are based on a traditional rectangular structured grid that is referred to as the base grid. Refinement is accomplished by successively subdividing base-grid cells by a factor of 2 in each horizontal direction. MODPATH does not allow vertical refinement within a model layer. Quadtree refinement allows cells to be subdivided repeatedly and, therefore, provides a high degree of flexibility in grid refinement (fig. 4B). The refinement method for quadpatch grids is simpler than that of quadtree grids, but provides somewhat less flexibility in grid refinement. For quadpatch grids, a level of refinement is specified for each base-grid cell. The base-grid cell is then completely subdivided to that level of refinement (fig. 4C). The refinement level of a cell indicates the degree of refinement relative to the base grid. Refinement level 0 refers to an undivided base-grid cell; for level 1, the cell is one-half of the base-grid cell size in both horizontal directions; for level 2, the cell is one-quarter of the base-grid cell size in each horizontal direction, and so on. The concept of refinement level applies to both quadtree and quadpatch grids. MODPATH also requires that quadtree and quadpatch grids be smoothed. A smoothed quad-refined grid is one in which the contrast in refinement level between adjacent cells is not allowed to be greater than 1.

Figure 5 is an expanded view showing face connections and face flows for a typical unstructured grid cell.

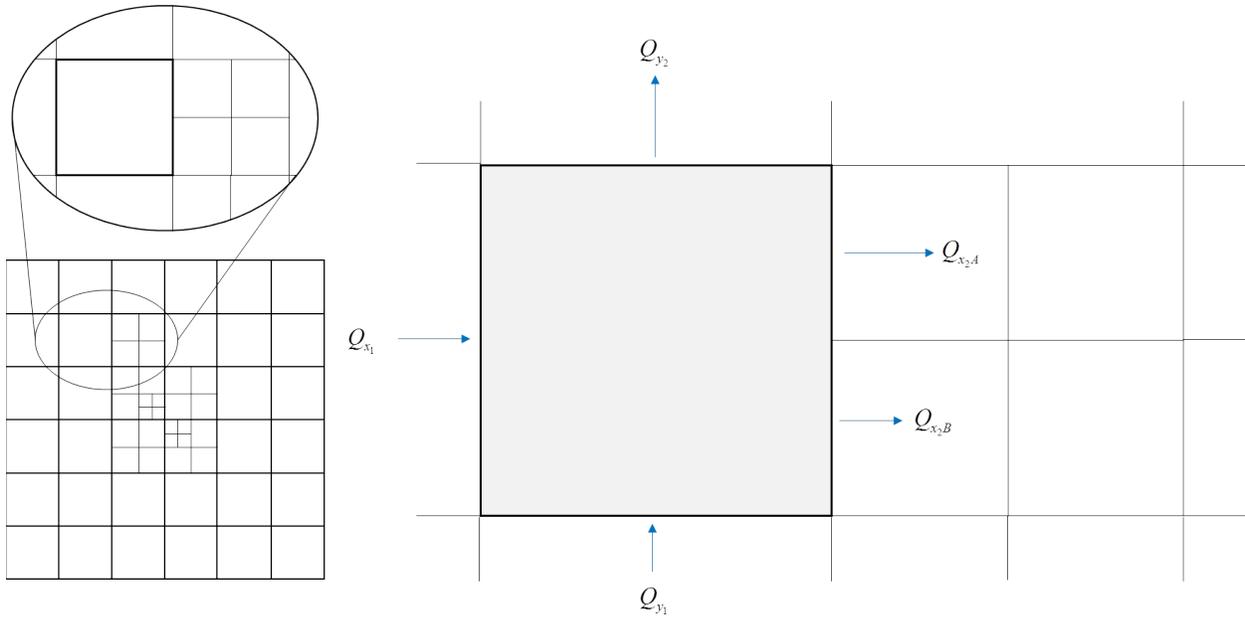


Figure 5. Cell connections and face flows in a smoothed, unstructured grid.

One of the fundamental characteristics of unstructured grids is that cells can be connected to multiple neighboring grid cells across a single cell face. Inspection of figure 5 clearly shows that the simple linear velocity interpolation method required for the analytical solution will not work for the shaded cell. In order to implement the analytical algorithm for such a cell, MODPATH subdivides the cell into multiple subcells so that each subcell satisfies the requirements of the analytical method. The subcell solution method is described in Pollock (2016). A brief synopsis of the method is presented below.

Figure 6 shows the shaded cell in figure 5 subdivided into four subcells labeled *A*, *B*, *C*, and *D*.

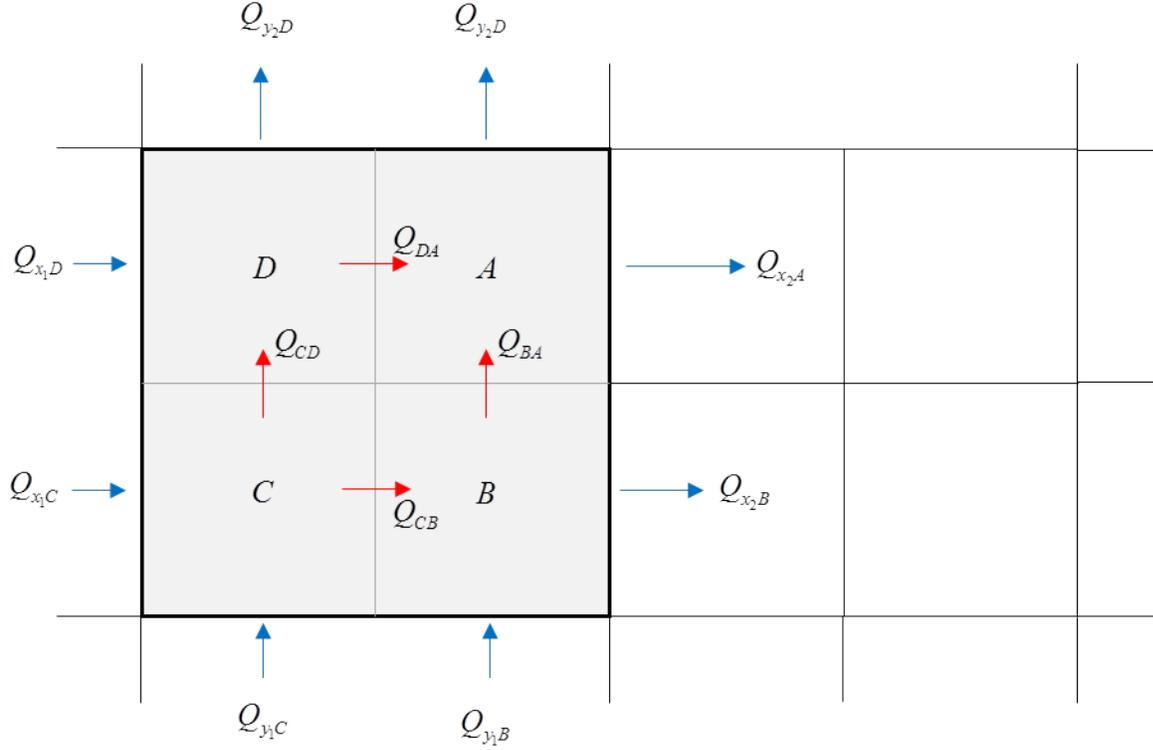


Figure 6. Definition of subcells for particle tracking with unstructured grids.

A water balance equation can be written for each subcell. The four subcells have a total of 12 face flows. The eight flow rates shown around the perimeter of the shaded cell are known from data in the MODFLOW–USG budget file. Flows Q_{x_2A} and Q_{x_2B} correspond directly to flow rates computed by MODFLOW–USG. Face flows Q_{x_1D} , Q_{x_1C} , Q_{y_1C} , Q_{y_1B} , Q_{y_2D} , and Q_{y_2A} are equal to one-half of the original face flow component computed by MODFLOW–USG for their respective faces. The other four face flows shown in red are the flow rates between the subcells, which are not known from the MODFLOW solution.

To determine the four internal face flows, it is necessary to solve a simultaneous system of four flow equations that represent the water balance equations for the four subcells. That process involves solving a system of equations for heads at the subcell centers. The four internal face flows can be obtained from the head values using the finite-difference form of Darcy’s Law:

$$Q_{DA} = -C(h_A - h_D) \quad (11a)$$

$$Q_{CB} = -C(h_B - h_C) \quad (11b)$$

$$Q_{BA} = -C(h_A - h_B) \quad (11c)$$

$$Q_{CD} = -C(h_D - h_C) . \quad (11d)$$

The hydraulic conductance (C) is defined as $C = \frac{KA}{L}$, where K is the hydraulic conductivity, A is the cross-sectional area perpendicular to the direction of flow, and L is the distance between nodes.

To see how the system of subcell water balance equations is structured, it is useful to look at one of the subcell equations in detail. The water balance equation for subcell A can be written as

$$\left[Q_{x_2A} - Q_{DA} \right] + \left[Q_{y_2A} - Q_{BA} \right] = Q_{SRC_A} + Q_{STO_A} . \quad (12)$$

The terms on the left side of equation 12 represent the net outflow of water from subcell A due to flow across the subcell faces. The term Q_{SRC_A} represents the net source of water from internal sources and sinks within subcell A . The term Q_{STO_A} represents the net source of water due to changes in storage under transient-flow conditions. As with previous implementations of the MODPATH semianalytical tracking algorithm, internal sources and sinks and flow from storage changes are assumed to be uniformly distributed within the grid cell. Therefore, Q_{SRC_A} is equal to one-quarter of the total flow due to storage changes in the full grid cell. Grouping the known flow terms on the right side gives

$$-Q_{DA} - Q_{BA} = Q_{SRC_A} + Q_{STO_A} - Q_{x_2A} - Q_{y_2A} . \quad (13)$$

Substituting Darcy's Law equations (11a and 11c) into equation 13 gives an equation in terms of subcell heads:

$$2C h_A - C h_B - C h_D = Q_{S_A} . \quad (14)$$

To simplify notation, the term Q_{S_A} is defined to be the sum of all the known flow terms on the right side of equation 13. The same process can be repeated for the other subcells to obtain the following system of equations:

$$\begin{bmatrix} 2C & -C & 0 & -C \\ -C & 2C & -C & 0 \\ 0 & -C & 2C & -C \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_A \\ h_B \\ h_C \\ h_D \end{bmatrix} = \begin{bmatrix} Q_{S_A} \\ Q_{S_B} \\ Q_{S_C} \\ 0 \end{bmatrix} . \quad (15)$$

Q_{S_B} and Q_{S_C} are the known flow terms grouped on the right side for subcells B and C . They are defined in the same way as Q_{S_A} . The water balance equation for subcell D has been replaced with the equation $h_D = 0$ because at least one specified head value is required to make the system of equations uniquely defined and solvable. The choice of "0" for the head in subcell D is arbitrary. Although the specified head value could be estimated from heads output by MODFLOW-USG, for the purposes of this method, the subcell heads do not need to have any relation to the actual heads in the system. The only purpose of the subcell heads is to provide an internally consistent set of heads that can be used to compute the interior flow rates using the Darcy's Law relations (equations 11a to 11d). Any specified head value would work. Similarly, the value of hydraulic conductance also can be assigned an arbitrary value and does not need to be computed from the MODFLOW data. It may seem counterintuitive that subcell flows can be computed from arbitrary values of conductance and specified head. However, inspection of the Darcy's Law equations (11a to 11d) shows that the effect of increasing the value of conductance by some factor simply results in a solution of head values for which the head differences between adjacent cells are reduced by the same factor. The product of conductance and head differences yields an identical set of flow rates between subcells for any combination of conductance and

specified head. We only are interested in the flows produced by the product of conductance and head difference, not their individual values. The matrix equation shown above can be solved efficiently using a direct Gaussian elimination solution procedure.

Once the subcell flows have been determined, each subcell has a fully defined set of face flows. The semianalytical particle-tracking procedure then can be applied to each subcell. Particles are moved from one subcell to another until they reach cell boundaries or meet a specified criterion that indicates particle tracking should stop (fig. 7).

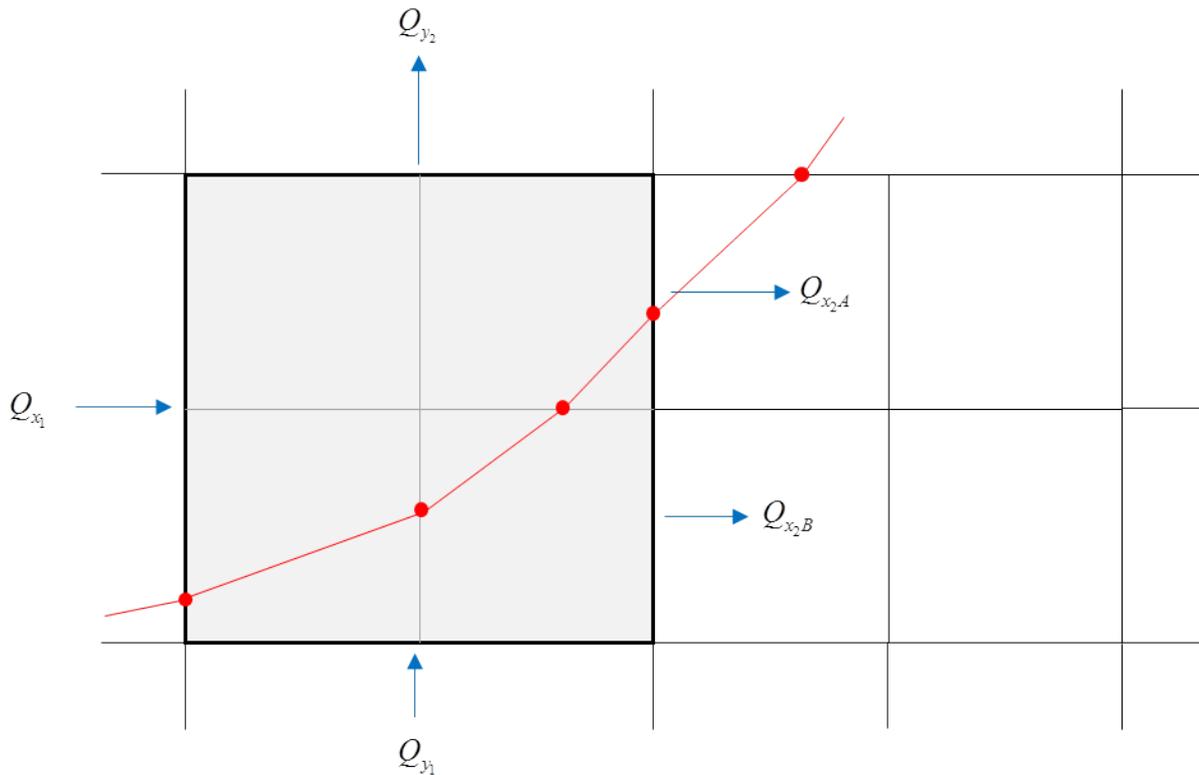


Figure 7. Schematic of pathline produced using the subcell method.

Although the subcell method can be extended to deal with refinement contrasts between cells that are greater than 1, the amount of computer memory required for subcell divisions larger than two by two make the method impractical to implement. Larger refinement contrasts between cells also may result in larger errors in flows generated by MODFLOW-USG. Restricting MODPATH to smoothed quadtree and quadpatch grids assures that no cell is connected to more than two neighboring cells across any of the horizontal faces or more than four neighboring cells across the top or bottom faces. The two-by-two subcell method described above can be applied to any smoothed quadtree or quadpatch grid.

In theory, the accuracy of the semianalytical particle-tracking method applied to smoothed quadtree and quadpatch grids should be similar to that of standard structured grids. However, errors associated with intercell flow rates produced by MODFLOW-USG for unstructured grids are larger and more difficult to quantify than those associated with MODFLOW simulations based on structured grids. Many of the errors arise as the result of the ad hoc application of Darcy's Law to compute flow across cell faces that are connected to multiple adjacent cells. Intercell flow rates generated by unstructured grid simulations are

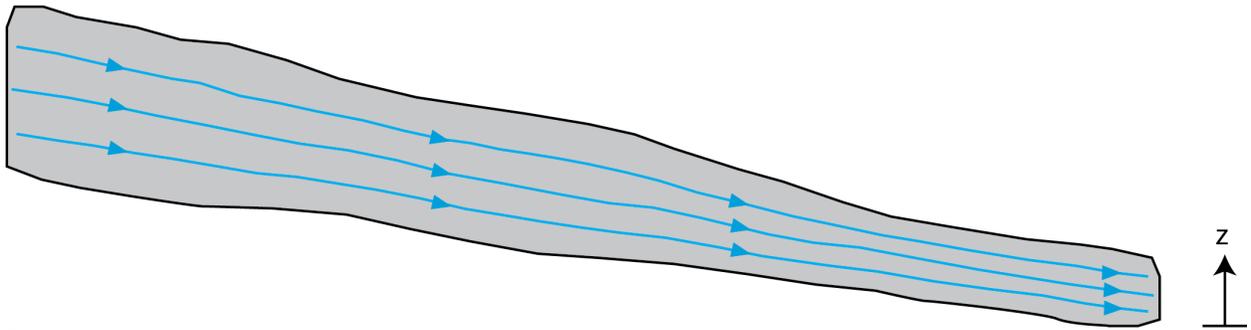
especially subject to spurious oscillations from cell to cell. Although the oscillations in flow occur in both the horizontal and vertical directions, they are much more significant in the vertical direction. In MODPATH, errors in computed particle paths based on unstructured grids directly reflect the erroneous oscillations in intercell flow rates. MODFLOW–USG attempts to reduce oscillation errors by applying a computational correction known as the ghost node correction. The ghost node correction implemented in MODFLOW–USG is described in detail by Panday (2013). The correction is extremely effective in dampening oscillations in the horizontal direction. The use of ghost nodes also reduces oscillations in the vertical direction, but unacceptable errors often still occur near cells that contain internal stresses, such as wells. In order to reduce these errors to an acceptable level when using MODPATH with unstructured grids, follow these two guidelines:

- Use the ghost node correction in MODFLOW–USG for all cell connections (both horizontal and vertical) that involve a contrast in refinement level.
- Whenever possible, use the same horizontal grid in all model layers to eliminate any possibility of vertical-flow oscillations due to contrasts in refinement level between layers.

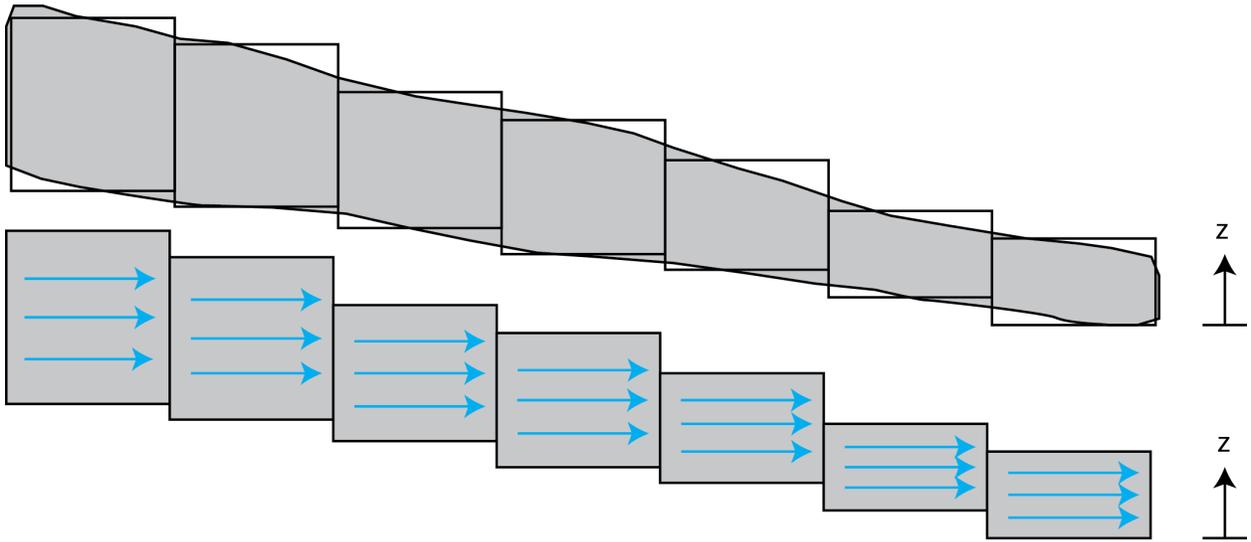
Deformed Vertical Grids

The development presented in the previous section was based on the assumption that the flow domain was discretized into a three-dimensional grid with horizontal layers and uniform thickness. In practice, however, many three-dimensional finite-difference simulations use a deformed grid in the vertical direction to allow grid cells to conform to stratigraphic units that vary in thickness and that are not perfectly horizontal. Deformed grid layers also arise due to water-table layers. The particle-tracking algorithm described above can be used to compute approximate paths for deformed grids. Figure 8 shows how a simple confined aquifer having variable thickness and elevation can be represented by a vertically deformed finite-difference grid.

A. Dipping, variable thickness confined aquifer



B. Approximation using horizontal rectangular cells



C. Grid and path lines in local coordinates

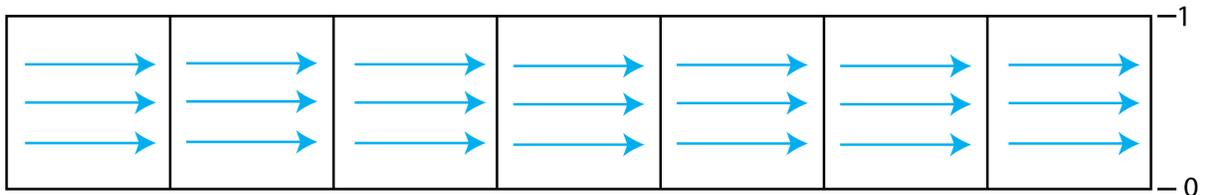


Figure 8. Schematic representation of an inclined aquifer with variable thickness.

Cells are assumed to be horizontal and rectangular with top and bottom elevations equal to top and bottom elevations of the cell at its node. A local coordinate, z_L , can be defined for each cell:

$$z_L = \frac{z - z_1}{z_2 - z_1}, \quad (16)$$

where z_1 and z_2 are the elevations of the bottom and top of the cell, respectively. According to equation 16, the local z coordinate varies from 0 at the bottom of the cell to 1 at the top. When a particle is transferred laterally from one cell to another, its local z coordinate remains the same.

Using this approach, if a particle leaves a cell at a position halfway between the top and bottom of the cell, it is assumed to enter the neighboring cell halfway between the top and bottom of that cell, regardless of how the thickness or absolute elevation of the layer changes from one cell to the next. This procedure is illustrated schematically for the case of lateral flow in a confined aquifer of variable thickness and dip (figs. 8A and 8B). When all layers are constant in thickness and horizontal, this approach reduces to the algorithm developed above for true rectangular grids (fig. 8C).

The advantage of a stratigraphic three-dimensional grid is that complex hydrogeologic systems can be simulated with fewer layers than using a grid with horizontal layers. The principal disadvantage is that spatial discretization errors are introduced that are difficult to quantify, especially with respect to particle-path computations. Nevertheless, experience has shown that the method described above produces results that are internally consistent with water budgets generated by MODFLOW for models that use deformed vertical grids.

Backward Tracking

Particles can be tracked backwards in the “upstream” direction by running the particle-tracking algorithm in reverse. Backward tracking is accomplished by reversing the sign of all the velocity components and then applying the algorithm just as described for forward tracking.

Transient Flow

Transient finite-difference flow simulations consist of a series of discrete time steps during which flow rates remain constant and storage changes within cells contribute an additional component to the internal source/sink term for each cell. The particle-tracking method described above can be extended to transient simulations by taking advantage of the fact that transient simulations behave as a series of steady-state flow periods. For each time step, particle paths are computed until the end of the time step is reached. A new velocity distribution is then calculated for the next time step and the computation of particle paths is resumed. The computation of paths forward or backward, boundary conditions, and the path termination criteria are handled the same as for steady-state flow.

Even though the mechanical aspects of implementing particle tracking is the same for transient and steady-state simulations, transient-flow conditions often affect computed particle paths in ways that are different than for steady-state flow, and are often unappreciated. In steady-state flow, the process of changing the value of porosity does not affect the spatial trajectory of the particle. Only the travel time of the particle is affected. As porosity is reduced, the particle speeds up and the travel time to its ultimate termination point decreases. The path itself is not changed. In contrast, under transient-flow conditions, both the particle trajectory and its travel time may change when the porosity is adjusted. As with steady-state flow, changing the porosity causes the particle to speed up or slow down. But, under transient conditions, the change in speed causes the particle also to experience a different history of groundwater flow direction over the course of its journey. As a result, path trajectories can differ significantly from one transient MODPATH simulation to the next, simply because of the difference in porosity.

Solute Retardation

The effect of linear sorption on the apparent velocity of a reactive solute (v^*) can be examined using the particle-tracking method presented in the previous sections by dividing the velocity components by a retardation factor, R :

$$v_x^* = \frac{v_x}{R}, \quad v_y^* = \frac{v_y}{R}, \quad v_z^* = \frac{v_z}{R}. \quad (17)$$

For nonreactive solutes that do not interact with the surrounding rock, the retardation factor is equal to 1 and the magnitude of the apparent solute velocity is equal to that of the groundwater velocity. For solutes affected by linear sorption, the retardation factor is greater than 1 and the magnitude of the apparent solute velocity is less than that of the groundwater velocity. The retardation factor is a function of the porosity, the density of the geologic material, and the sorption coefficient. A detailed discussion of the retardation factor can be found in Domenico and Schwartz (1990). As implemented by MODPATH, the retardation factor is simply a scaling factor for the magnitude of the velocity that is specified by the user. The retardation factor does not affect the direction of the velocity.

MODPATH Overview

Time Concepts

MODPATH defines the following time concepts (fig. 9):

- **Simulation time.**—This is the value of time associated with the MODFLOW simulation. Simulation time starts at 0 and increases throughout the course of a MODFLOW simulation. The range of simulation time is defined by the stress period and time-step data in the MODFLOW discretization file.
- **Tracking time.**—This is defined for each MODPATH simulation relative to specified reference value of simulation time. Tracking time is “0” at the specified reference simulation time. Tracking time measures the accumulated time during a particle-tracking analysis. The value of tracking time is always positive, regardless of whether particles are tracked in the forward or backward direction.

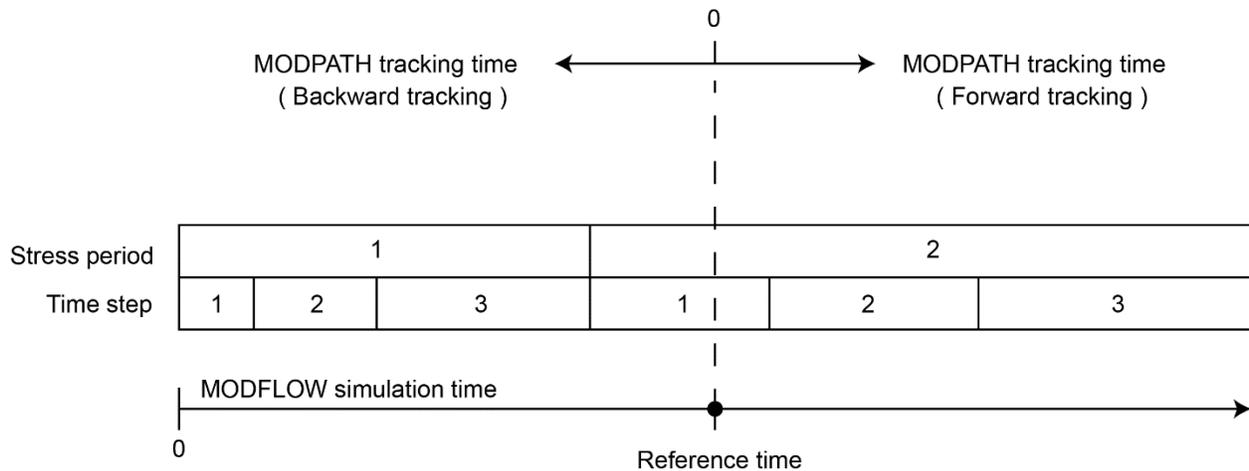


Figure 9. Schematic representation of time concepts used by MODPATH.

The concept of tracking time allows MODPATH to track particles backward in time through either steady-state or transient-flow fields, simply by reversing the sign of the velocity used in the tracking calculation. That approach greatly simplifies the particle-tracking computation by allowing MODPATH to use the same set of logic and code to simulate the entire range of forward- and backward-tracking scenarios under both steady-state and transient-flow conditions.

Flow Boundaries

It is common for MODFLOW simulations to use stress packages to represent boundary flows. In MODFLOW, stress packages simply add a component of flow to the grid cells in which they are active. In MODFLOW, stress-package flows behave as internal sources or sinks of water. However, in cases where stress-package flows are intended to represent flows that enter across system boundaries, the velocity field used by MODPATH will more accurately represent the system if those flows are assigned to the appropriate boundary faces of cells and used to compute velocity components across boundary cell faces. A good example is recharge represented with the recharge package. Areal recharge commonly represents inflow to the system across the water table. Consequently, the interpolated velocity field in a water-table cell will be more accurately represented in MODPATH by assigning the recharge flow to the top face of the cell and using it to compute a vertical component of velocity across the top face of the cell. Another common example is the river package, which often is used to represent shallow streams that sit at or are near the top of grid cells that represent an unconfined aquifer. As with recharge, the velocity in those cells also will be more accurately represented by treating the river package flows as if they occurred across the top face of the cell. In other cases, such as for the well package, the stress-package flows are intended to represent internal sources and sinks. In those cases, no additional information is required by MODPATH because MODPATH automatically represents them as internal, uniformly distributed sources or sinks. In the case of boundary stress-package flows, MODPATH requires additional information indicating which faces are the flow boundary faces.

Boundary face information is provided to MODPATH by assigning each of the six cell faces an index number as illustrated in figure 10. Faces 1 and 2 are perpendicular to the x direction, faces 3 and 4 are perpendicular to the y direction, face 5 is the bottom of the cell, and face 6 is the top of the cell. MODPATH refers to cell faces using a variable named IFACE that is

assigned to each stress-package flow. The value of IFACE indicates the face across which the flow occurs. In the case of true internal sources or sinks, such as wells, the IFACE value is equal to 0.

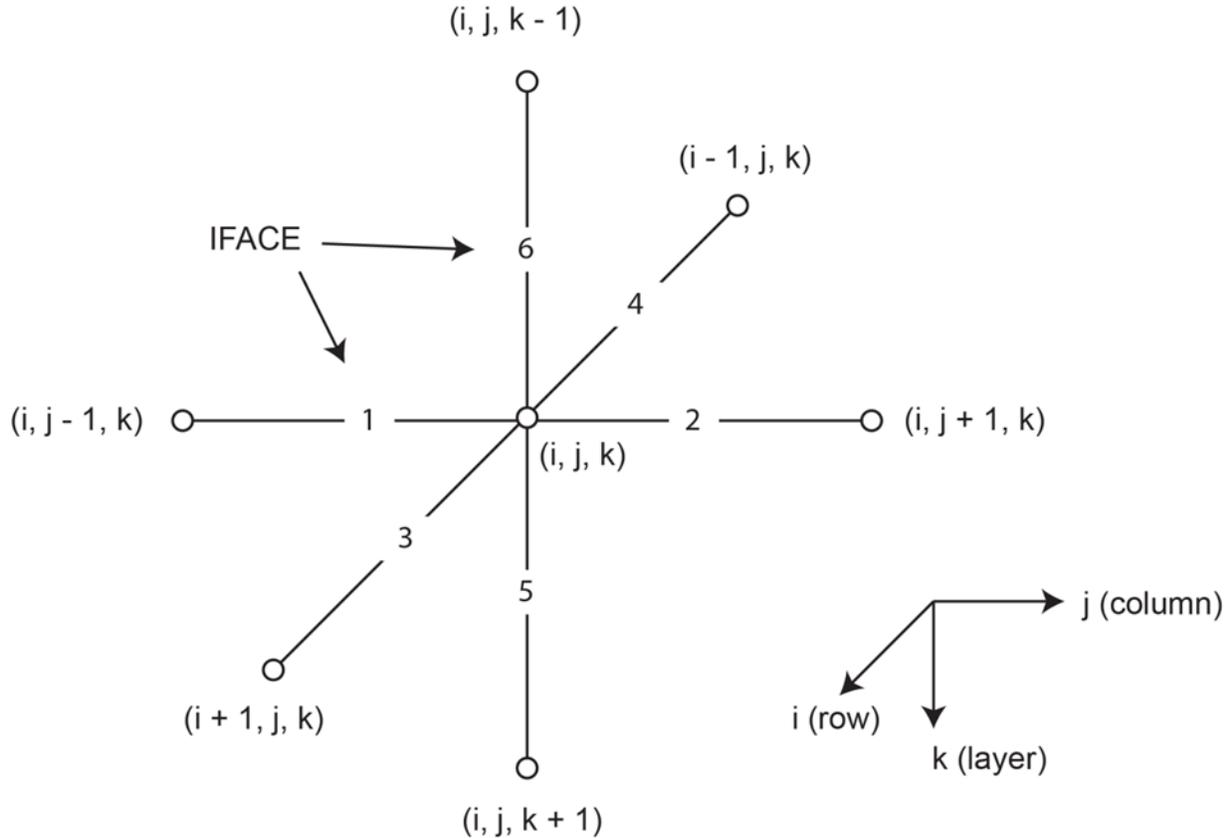


Figure 10. Diagram showing definition of the cell face index, IFACE.

The additional IFACE data are provided to MODPATH in one of two ways, depending on the type of stress package. If a MODFLOW stress package saves IFACE values in the compact budget file as auxiliary data (Harbaugh, 2005), then MODPATH reads IFACE values from the compact budget file. However, some stress packages do not support saving IFACE values in the compact budget file. The recharge and evapotranspiration packages are the two most widely used packages that do not support auxiliary IFACE values. For those types of packages, MODPATH allows the user to specify the budget label and a default IFACE value in the MODPATH basic data file. The specified IFACE value is applied to all of the stresses for that package; IFACE information for any number of stress packages may be specified this way. If a stress package that saves IFACE values in the compact budget file also has a default IFACE value specified in the MODPATH basic data file, the IFACE values in the compact budget file are used and the specified default value is ignored. If no information about IFACE is provided in either the compact budget file or the MODPATH basic data file, an IFACE value of 0 is assumed and all stress periods are treated as internally distributed sinks or sources. Whenever possible, IFACE values should be saved as auxiliary data in the compact budget file to provide the most flexibility in defining flow boundary conditions.

Terminating Particles

Particles are stopped whenever they reach points of termination or whenever the cumulative tracking time equals the maximum allowed value. A particle terminates when:

- it reaches an external boundary face or an internal sink or source cell that captures the particle,
- it enters a cell with a zone code that designates the cell as a termination location,
- the cumulative tracking time has reached the maximum allowed value, or
- it encounters an abnormal condition that will not allow tracking to continue.

MODPATH continues to track particles until all particles have terminated.

Particles always terminate when they reach a cell face that represents a flow boundary for the active grid, as described in the preceding section. Particles also may terminate when they enter a cell containing an internal sink or source. For the forward tracking case, cells that contain internal sinks can be characterized as either strong sink cells or weak sink cells. A strong sink cell is one that contains an internal sink and has inflow from adjacent cells but no outflow to other cells. In contrast, weak sink cells have both inflow and outflow across their cell faces. Weak internal sinks remove some of the inflow from adjacent cells, but not all of it. Analogous definitions exist for strong and weak source cells. For weak sink and weak source cells, an arbitrary decision must be made by the user about when to terminate particles. MODPATH provides options to allow particles either to pass through or stop when they enter cells containing weak sinks or weak sources.

Sometimes it is useful to stop particles at an arbitrary location that does not correspond to a natural termination point. MODPATH provides that capability by allowing zone numbers to be assigned to grid cells and by identifying a specific zone number that indicates particle termination locations. An example of such a situation would be a case where the desire is to determine the recharge locations at the water table that contribute water to a deep aquifer. This determination could be achieved by assigning a zone number to the cells representing the deep aquifer, indicating they are termination points. Particles that start at the water table would then stop when they reach the deep aquifer because the cells are designated termination locations. The results could be processed to produce a recharge map for the deep aquifer.

Particle tracking is stopped whenever the maximum allowed value for tracking time has been reached. That maximum value may be specified by the user, or it may be when the cumulative tracking time has reached the time specified for the end of the MODFLOW simulation (if particles are tracked forward in time) or the beginning of the MODFLOW simulation (if particles are tracked backward in time). MODFLOW allows for a mix of steady-state and transient stress periods. MODPATH provides a variety of options for how to handle the situation when a particle-tracking simulation reaches the end or beginning of a MODFLOW simulation. For forward-tracking simulations, MODPATH forces the particle-tracking simulation to stop when the end of the MODFLOW simulation is reached and the final stress period is transient. Under those conditions, there is no basis for extending the particle tracking beyond the end of the simulation. If the final stress period is steady state, however, the option is provided to either stop the particle-tracking simulation or extend it indefinitely in time using the velocity distribution of the final time step. If a steady-state time step is extended, particle tracking continues until all particles have terminated. Providing the option to extend the simulation if a steady-state condition exists at the end of the simulation is appropriate because it is reasonable to assume that steady-state flow at the end of the simulation could continue beyond that point in

time. The same logic is applied for backward-tracking simulations, except in such cases, the decision to stop or extend the particle-tracking simulation is based on whether steady-state or transient-flow conditions exist for the first MODFLOW stress period.

Sometimes particles encounter abnormal conditions that require them to be terminated. The most common cause of abnormal termination occurs in transient-flow systems where cells are allowed to go dry. Under those conditions, particles are sometimes stranded in cells that become inactive after having gone dry. Particles are terminated in such cases and given a status code that indicates they were stranded.

Model Input and Output

MODPATH obtains data from a set of input files that is a combination of MODFLOW input and output files plus additional data files specific to MODPATH. MODPATH generates a number of output files that summarize simulation results. The following sections provide only an overview of the data required by MODPATH. Detailed descriptions of the input and output files are included with the software distribution.

Flow-System Files

MODPATH requires the following input files that contain the basic information about the groundwater flow system:

- MODPATH basic data file
- MODFLOW discretization file
- MODPATH unstructured grid file
- MODPATH time discretization file
- MODFLOW head output file
- MODFLOW cell-by-cell budget output file

The MODPATH basic data file consists of a combination of data contained in the MODFLOW basic (BAS) package data file and the MODFLOW layer property flow (LPF) package data file. It also contains additional data, such as porosity, that is specific to MODPATH and not required by MODFLOW. The MODPATH basic data file may also refer to additional ancillary data files that contain various types of array data.

For structured grid simulations, spatial and time discretization is obtained directly from the MODFLOW discretization (DIS) file. In the case of unstructured grids, the MODFLOW-USG unstructured discretization file (DISU) does not contain all of the information needed by MODPATH to define the spatial characteristics of the grid. To deal with that problem, MODPATH defines its own unstructured spatial discretization file (MPUGRID), as well as a separate time discretization file (TDIS).

Head output from MODFLOW is used by MODPATH to compute saturated thickness for water-table layers and to provide information about the distribution of dry cells. MODPATH requires that the head output be saved as a binary file.

The flow-rate data used by MODPATH to compute groundwater velocity components are obtained from the MODFLOW budget file. MODPATH requires that MODFLOW saves budget data for all packages in a single file. Although both standard and compact budget output styles are supported, it is strongly recommended that budget output be saved using the compact option. Unlike the standard MODFLOW budget file, the compact budget file can contain IFACE values

that allow flow boundary conditions to be represented more accurately by assigning stress-package flows to specific cell faces (as described in the Flow Boundaries section).

Name File

The name file provides information about file names and data file types that MODPATH needs to open and manage flow-system input files. The MODPATH name file serves a similar function as the MODFLOW name file, although its structure is different.

Simulation File

The MODPATH simulation file contains input data that define the options for a specific MODPATH simulation, such as the following:

- Output file names
- Tracking direction (forward or backward)
- Simulation type and output options
- Particle-starting location data
- Options for terminating particles

It is common to generate several MODPATH simulation files for a single MODFLOW flow simulation in order to examine the groundwater flow from a variety of perspectives. In such instances, the MODPATH flow-system files generally will not change from run to run, but the simulation file will change to reflect changes in particle-tracking options.

Listing File

The listing file is an output file that summarizes model input and the results generated for each MODPATH simulation. The MODPATH listing file is similar in style to the listing file produced by MODFLOW.

Debug Files

Debug files record output that provides information about program flow and detailed accounting of particle-tracking calculations that can be helpful for troubleshooting problems when they occur. The debug files include a log file that records the program flow and an optional trace file that can be used to record detailed output on a cell-by-cell basis for a single particle throughout the course of a simulation.

Particle Coordinate Files

Particle coordinate files record the movement of particles throughout the course of a simulation and form the basis for input data for post-processing programs that analyze and display the results.

Four types of simulations are supported by MODPATH:

- Endpoint simulation
- Pathline simulation
- Time-series simulation
- Combined pathline and time series

The particle-tracking calculations performed by MODPATH are the same for each of the four simulation types. The only difference between them is the type of output generated by the simulation. For an endpoint simulation, only information about the initial and final location and time data are recorded in an endpoint file. This text file is generated for all MODPATH simulations and contains a one-line data record for each particle. Intermediate particle coordinates computed during the simulation are not saved as output. Endpoint simulations produce the minimum amount of output possible for a MODPATH simulation. Endpoint simulations are most useful when the objective is to map recharge locations to discharge locations for a large number of particles. For pathline simulations, coordinates along the path of each particle are recorded in a pathline file. The pathline file contains the starting coordinates of a particle and the coordinates at every intermediate point computed during the simulation. Pathline files contain all of the information necessary for a graphical post-processing program to display the pathline track. Pathline simulations also produce an endpoint file. For time-series simulations, only the locations of particles at specified points in time are recorded in a time-series file. Time-series simulations produce a set of particle location records that are stacked in time. When points in a time-series file are displayed, the progress of a group of particles is shown as a series of snapshots in time. As with other simulations, time-series simulations also produce an endpoint file. A combined pathline and time-series simulation produces endpoint, pathline, and time-series files.

Examples

The example problems presented below are based on the hypothetical groundwater flow system shown in figure 11. The groundwater flow system consists of two aquifers separated by a thin confining layer.

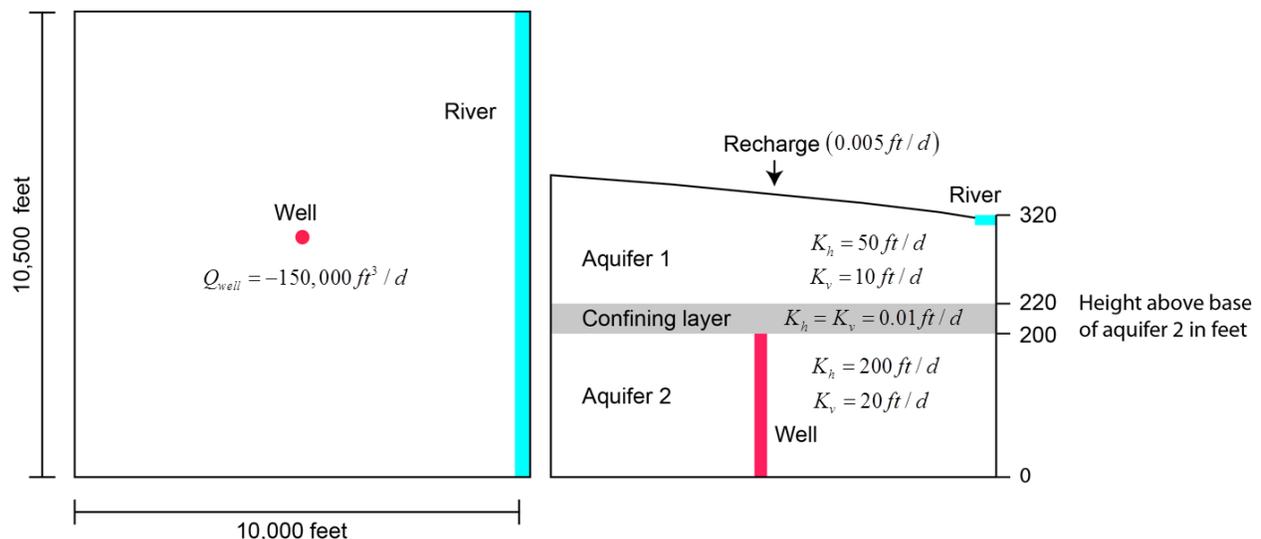


Figure 11. Conceptual model for hypothetical groundwater flow system (ft, foot; ft³, cubic foot; ft/d, foot per day).

All of the examples presented below are for steady-state flow. Example 1 is based on a MODFLOW-2005 simulation using a standard structured grid. Example 2 is based on a MODFLOW-USG simulation using an unstructured grid that is refined in the vicinity of the

well. A complete set of input and output files for the example problems is provided in the software distribution. Only key excerpts are presented in this report.

Example 1 – Structured Grid

Figure 12 shows the areal grid used in the MODFLOW-2005 simulation. The grid consists of 21 rows, 20 columns, and 3 layers. Areal grid cells are uniform square cells, 500 feet per side. The model layers correspond to the hydrogeologic units shown in figure 11.

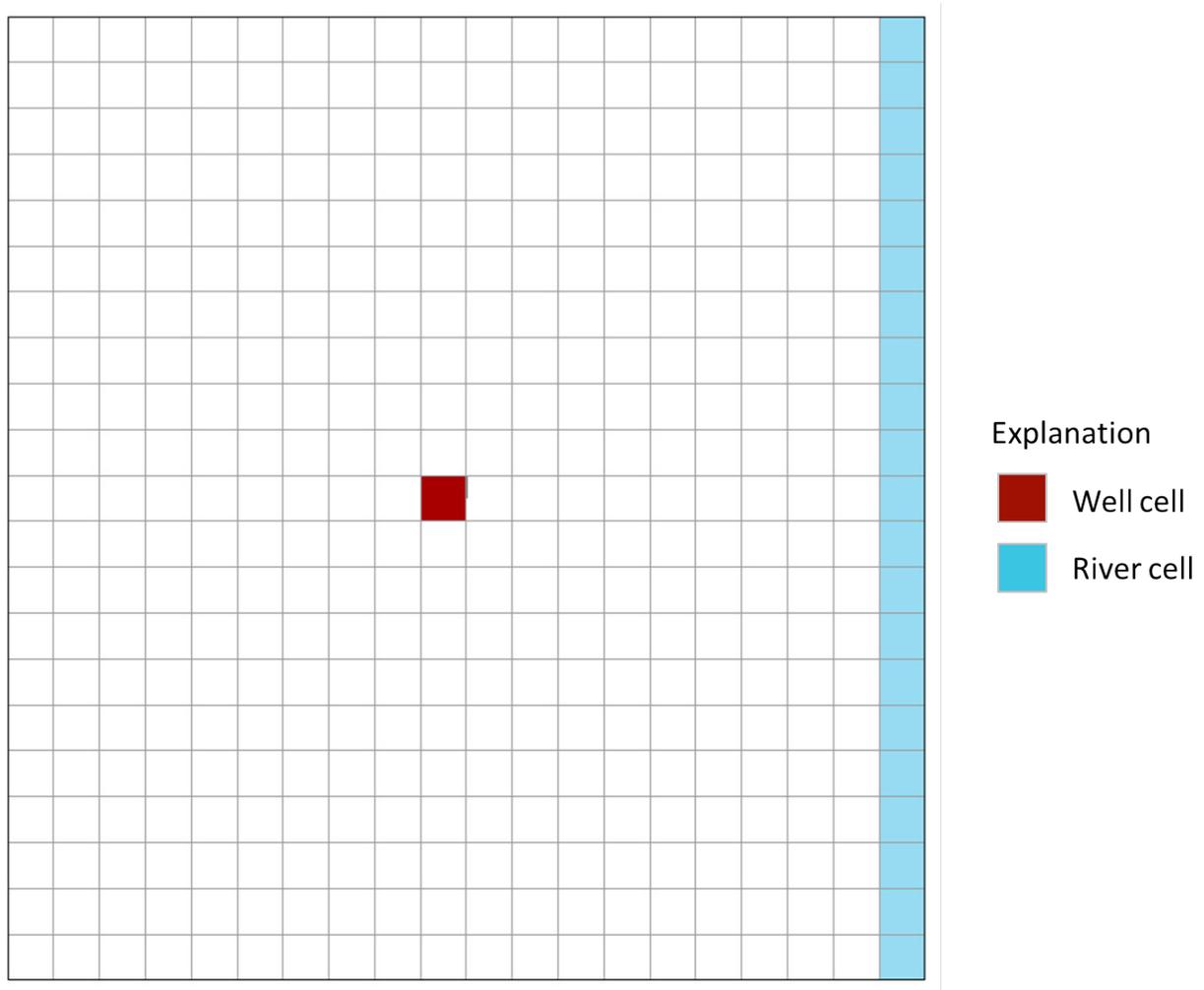


Figure 12. Structured grid for example 1, used in the MODFLOW-2005 simulation.

The well is located in row 11, column 10, and layer 3. The well, river, and recharge are represented by the MODFLOW well, river, and recharge packages. In figure 12, the location of the river cells is shown in light blue and the location of the cell containing the well is shown in dark red. The MODPATH basic data file contains a combination of data, such as IBOUND, porosity, MODFLOW layer-type flags, default IFACE specification, and no-flow/dry-cell head values. It usually is the case that the same MODPATH basic data file is used for several different MODPATH simulations. The MODPATH basic data file for example 1 is shown in figure 13.

No-flow and dry-cell head values	1.0E+20	1.0E+30
	2	
	RECHARGE	
Default IFACE values	6	
	ET	
	6	
Layer-type flags	1	0 0
	CONSTANT	1
IBOUND for model layers	CONSTANT	1
	CONSTANT	1
	CONSTANT	0.1
Porosity for model layers	CONSTANT	0.1
	CONSTANT	0.1

Figure 13. MODPATH basic data file for example 1.

It is essential to represent vertical-flow boundaries accurately in MODPATH in order to produce realistic three-dimensional flow paths. For this problem, both river discharge and recharge are assigned to the top face (face 6) of cells in layer 1. The assignment of face flows is accomplished using the IFACE variable in MODPATH. In the case of the river, IFACE is included as an auxiliary variable in the MODFLOW river package and assigned a value of 6 for each river reach. Saving the MODFLOW budget output in a compact budget file makes IFACE information for the river reaches available for MODPATH. In contrast, IFACE information for the recharge package must be specified in the MODPATH basic data file using a default IFACE value because auxiliary variables are not supported for area-based stress packages, such as the recharge package.

Flow-system files are specified in the MODPATH name file shown in figure 14.

```
MPBAS    example_1.mpbas
DIS      example_1.dis
HEAD     example_1.hed
BUDGET   example_1.bud
```

Figure 14. MODPATH name file.

Each line in the name file designates a specific flow-system file with a key word indicating the file type and the name of the file. For a structured grid, the flow-system files are:

- MPBAS – the MODPATH basic data file
- DIS – the MODFLOW-2005 discretization file
- HEAD – the MODFLOW binary head output file
- BUDGET – the MODFLOW budget output file

Simulation 1A – Forward-Tracking Pathlines with Time-Series Simulation

In simulation 1A, a line of 21 particles is placed at the water table in layer 1 for column 3, rows 1 through 21. A combined pathline and time-series simulation is run that tracks the particles forward to their discharge locations. The particle termination option is set to extend the final time step to assure that all of the particles reach their discharge locations, regardless of travel time. The simulation file and starting locations file for simulation 1A are shown in figure 15 and figure 16, respectively.

```

Name and listing file names { example_1.mpnam
                             example_1a.mplist
Simulation option flags    { 4 1 1 1 1 0
Particle coordinate output file names { example_1a.endpoint7
                                       example_1a.pathline7
                                       example_1a.timeseries7
Number of individual cell budgets { 0
Reference time data           { 1
                               { 0.0
Stop time option             { 2
Time point data              { 1
                               { 500 1000.
                               { 2
Zone data                    { 0
                               { CONSTANT 1
                               { CONSTANT 1
                               { CONSTANT 2
Retardation factor option    { 1
Particle group data          { 1
                               { PG1
                               { 1
                               { 0.0
                               { external example_1a.sloc

```

Figure 15. MODPATH simulation file for simulation 1A.

```

Input style          1
Location style       1
Number of particles, Particle ID option { 21 0
layer, row, column, local X, local Y, local Z, time offset, drupe option {
1 1 3 0.500 0.500 1.000 0.0 0
1 2 3 0.500 0.500 1.000 0.0 0
1 3 3 0.500 0.500 1.000 0.0 0
1 4 3 0.500 0.500 1.000 0.0 0
1 5 3 0.500 0.500 1.000 0.0 0
1 6 3 0.500 0.500 1.000 0.0 0
1 7 3 0.500 0.500 1.000 0.0 0
1 8 3 0.500 0.500 1.000 0.0 0
1 9 3 0.500 0.500 1.000 0.0 0
1 10 3 0.500 0.500 1.000 0.0 0
1 11 3 0.500 0.500 1.000 0.0 0
1 12 3 0.500 0.500 1.000 0.0 0
1 13 3 0.500 0.500 1.000 0.0 0
1 14 3 0.500 0.500 1.000 0.0 0
1 15 3 0.500 0.500 1.000 0.0 0
1 16 3 0.500 0.500 1.000 0.0 0
1 17 3 0.500 0.500 1.000 0.0 0
1 18 3 0.500 0.500 1.000 0.0 0
1 19 3 0.500 0.500 1.000 0.0 0
1 20 3 0.500 0.500 1.000 0.0 0
1 21 3 0.500 0.500 1.000 0.0 0

```

Figure 16. MODPATH starting locations file for simulation 1A.

All of the data are now ready to run MODPATH for simulation 1A. MODPATH can be run from a command window by typing the name of the MODPATH application file (usually mpath7.exe) and by pressing Enter. MODPATH will then prompt the user to enter the name of the simulation file. After entering the simulation file name, MODPATH runs the simulation. When the simulation is complete, summary information is printed to the command window. In the case of simulation 1A, the following output files are generated:

- Listing file
- Endpoint file
- Pathline file

- Time-series file
- Log file

The particle coordinate output files are simple text files that contain spatial and temporal coordinates that define the particle paths. The data provided in those files often are used to create standardized graphics files that can be displayed in a variety of two- and three-dimensional viewers. Figure 17 shows a map view of the pathlines generated by simulation 1A. The points represent 1,000-day time points. The point color signifies the model layer (green for layer 1, gold for layer 2, and red for layer 3). The same data are displayed in a three-dimensional perspective view in figure 18.

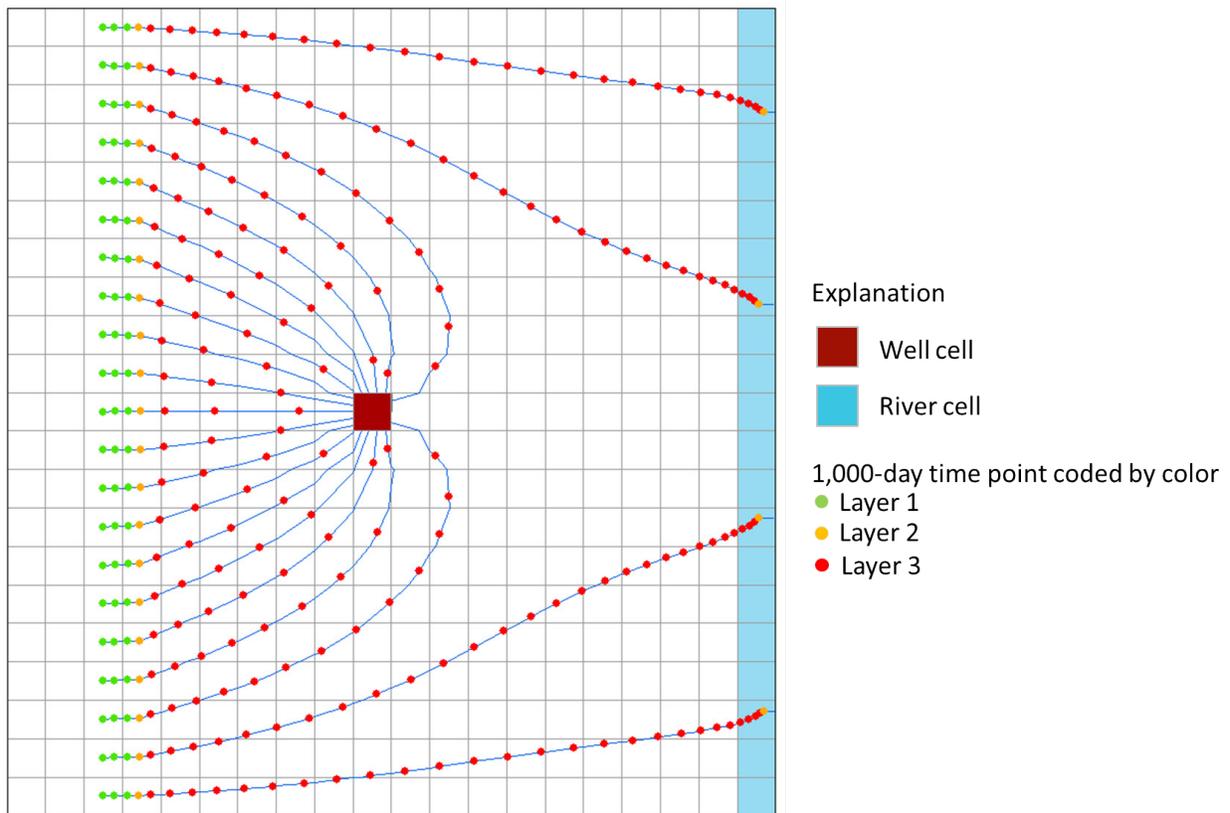


Figure 17. MODPATH map view of pathlines and time points for simulation 1A.

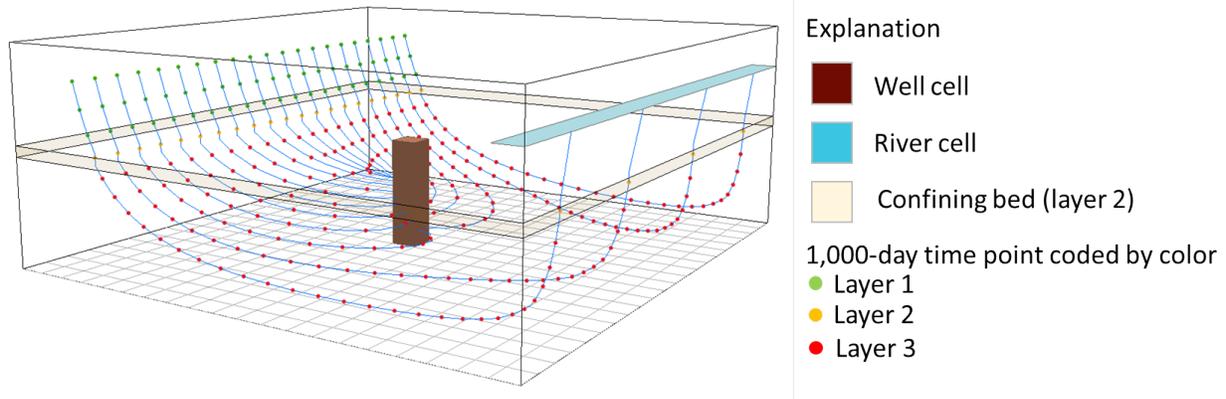


Figure 18. MODPATH three-dimensional perspective view of pathlines and time points for simulation 1A.

Simulation 1B – Forward-Tracking Endpoint Simulation

MODPATH often is used to compute recharge capture areas for wells and other hydrologic features. Forward-tracking endpoint simulations are an efficient way to use MODPATH to compute capture areas for wells. In simulation 1B, a capture area for the well is determined by placing a three-by-three array of particles on the top face of layer 1 and by tracking the particles forward to their discharge points using a forward endpoint simulation. The endpoint file generated by MODPATH contains information about the starting location and the final location of each particle. The capture area for the well can be displayed by plotting the starting locations of all the particles that discharge to the cell containing the well. Figure 19 and figure 20 show the simulation file and starting locations file for simulation 1B, respectively.

```

example_1.mpnam
example_1b.mplist
1 1 1 1 1 0
example_1b.endpoint7
2
1050 1060
1
0.0
2
2
0
CONSTANT 1
CONSTANT 1
CONSTANT 2
1
1
PG1
1
0.0
external example_1b.sloc

```

Figure 19. MODPATH simulation file for simulation 1B.

```

2
1 1
1 1 0
0 0 0 0 0 0 0 0 3 3
1 1 1 1 21 20

```

Figure 20. MODPATH starting locations file for simulation 1B.

The starting locations file used for simulation 1B has a different, more compact structure than the starting locations file for simulation 1A. In simulation 1B, the starting locations are specified using a template and a specified range of cells. The template defines how particles are placed in each cell within the range. That approach allows thousands of particles to be generated automatically with only a few lines of data in the starting locations file. The other data files used in simulation 1A also can be used in simulation 1B. Figure 21 shows the map view of the capture area generated by simulation 1B. The starting locations of particles that discharge to the cell containing the well are shown in red. Starting locations that discharge to the river are shown with open circles.

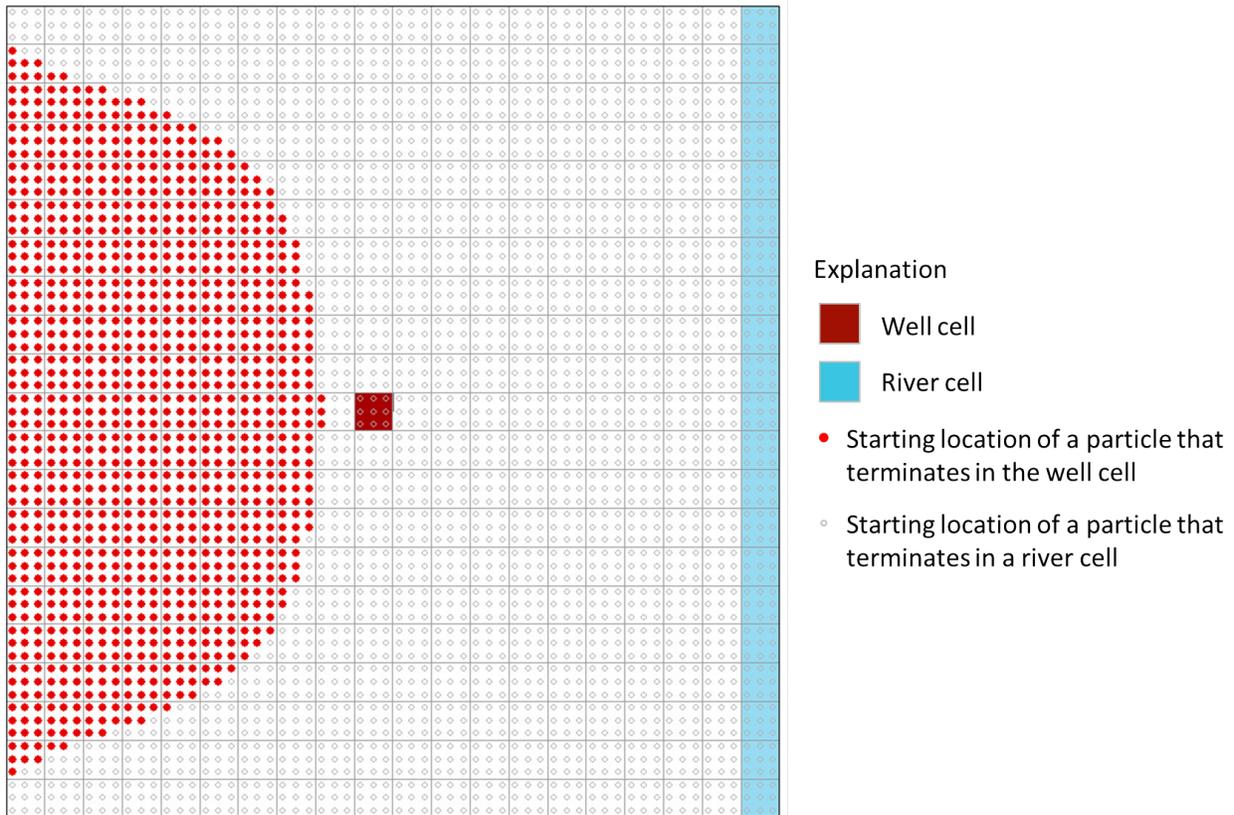


Figure 21. MODPATH map view of starting locations for showing discharge locations for simulation 1B.

Example 2 – Unstructured Grid

Example 2 simulates the groundwater flow system with MODFLOW–USG and uses the unstructured grid capability of MODPATH to examine the flow with particle-tracking simulations. Figure 22 shows the unstructured grid. The unstructured grid is derived from the structured grid used in example 1. The areal grid is refined around the base-grid location that

contains the well in example 1 (row 11, column 10). The base-grid location (row 11, column 10) is refined three levels, so that the unstructured grid contains an eight-by-eight array of 64 cells in place of the original base-grid cell. The grid is smoothed so that it will meet the requirements of MODPATH for unstructured particle tracking. The same areal grid is applied to all model layers. The grid contains a total of 1,953 cells. The well is located in layer 3 in cell 1,623.

One of the major differences encountered with using MODPATH with an unstructured grid is that it is not possible to use the MODFLOW–USG unstructured discretization file to construct the unstructured particle-tracking grid. The MODFLOW–USG file has cell connection information but does not contain information about the location of the cells or the topology of the rectangular cell connections. MODPATH requires its own unstructured grid file that provides that information. Figure 23 shows an edited excerpt of the MODPATH unstructured grid file used for example 2.

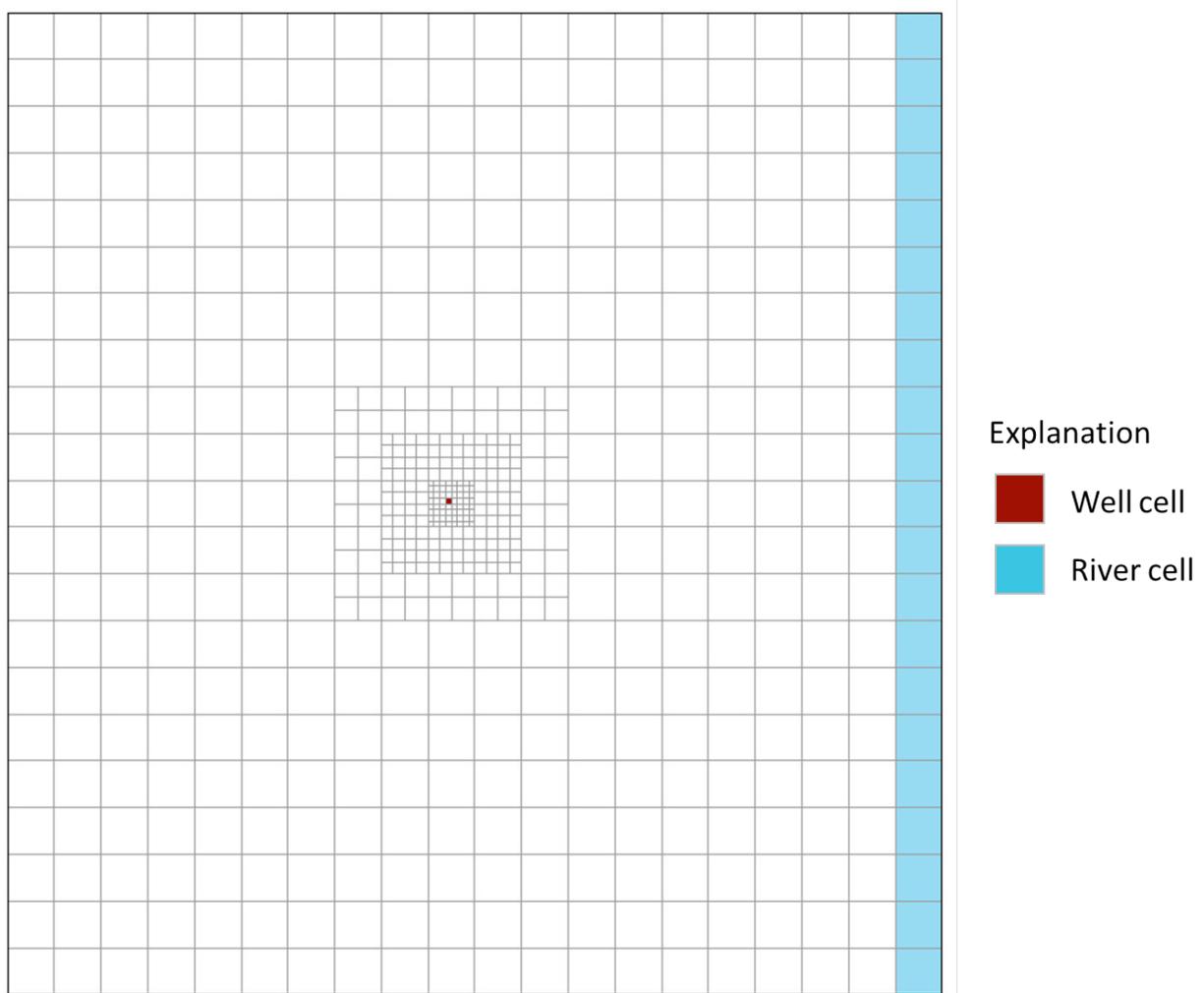


Figure 22. MODPATH unstructured grid for example 2. (Red square indicates the well, which is located in cell 1,623 in layer 3.)

```

Grid-dimension data -----
1953 3 21 20 10350
INTERNAL 1 (FREE) 0 BaseDX
Base-grid cell spacing -----
500 500 500 500 500 500 500 500 500 500 500
500 500 500 500 500 500 500 500 500 500 500
INTERNAL 1 (FREE) 0 BaseDY
500 500 500 500 500 500 500 500 500 500 500
500 500 500 500 500 500 500 500 500 500 500
500
1 1 1 1 0 250 10250 220 400
2 1 1 2 0 750 10250 220 400
3 1 1 3 0 1250 10250 220 400
4 1 1 4 0 1750 10250 220 400
5 1 1 5 0 2250 10250 220 400
6 1 1 6 0 2750 10250 220 400
7 1 1 7 0 3250 10250 220 400
8 1 1 8 0 3750 10250 220 400
9 1 1 9 0 4250 10250 220 400
10 1 1 10 0 4750 10250 220 400
11 1 1 11 0 5250 10250 220 400
12 1 1 12 0 5750 10250 220 400
13 1 1 13 0 6250 10250 220 400
14 1 1 14 0 6750 10250 220 400
15 1 1 15 0 7250 10250 220 400
16 1 1 16 0 7750 10250 220 400
17 1 1 17 0 8250 10250 220 400
18 1 1 18 0 8750 10250 220 400
19 1 1 19 0 9250 10250 220 400
20 1 1 20 0 9750 10250 220 400

Grid-cell size and location data -----
...
1933 3 20 20 0 9750 750 0 200
1934 3 21 1 0 250 250 0 200
1935 3 21 2 0 750 250 0 200
1936 3 21 3 0 1250 250 0 200
1937 3 21 4 0 1750 250 0 200
1938 3 21 5 0 2250 250 0 200
1939 3 21 6 0 2750 250 200
1940 3 21 7 0 3250 250 0 200
1941 3 21 8 0 3750 250 0 200
1942 3 21 9 0 4250 250 0 200
1943 3 21 10 0 4750 250 0 200
1944 3 21 11 0 5250 250 0 200
1945 3 21 12 0 5750 250 0 200
1946 3 21 13 0 6250 250 0 200
1947 3 21 14 0 6750 250 0 200
1948 3 21 15 0 7250 250 0 200
1949 3 21 16 0 7750 250 0 200
1950 3 21 17 0 8250 250 0 200
1951 3 21 18 0 8750 250 0 200
1952 3 21 19 0 9250 250 0 200
1953 3 21 20 0 9750 250 0 200
1 0 1 1 0 1 0 2 21 652
2 1 1 1 0 1 0 1 3 22 653
3 1 1 1 0 1 0 2 4 23 654
4 1 1 1 0 1 0 3 5 24 655
5 1 1 1 0 1 0 4 6 25 656
6 1 1 1 0 1 0 5 7 26 657
7 1 1 1 0 1 0 6 8 27 658
8 1 1 1 0 1 0 7 9 28 659
9 1 1 1 0 1 0 8 10 29 660
10 1 1 1 0 1 0 9 11 30 661
11 1 1 1 0 1 0 10 12 31 662
12 1 1 1 0 1 0 11 13 32 663
13 1 1 1 0 1 0 12 14 33 664
14 1 1 1 0 1 0 13 15 34 665
15 1 1 1 0 1 0 14 16 35 666
16 1 1 1 0 1 0 15 17 36 667
17 1 1 1 0 1 0 16 18 37 668
18 1 1 1 0 1 0 17 19 38 669
19 1 1 1 0 1 0 18 20 39 670
20 1 0 1 0 1 0 19 40 671
...
460 1 1 1 1 1 0 459 461 495 389 1111
461 1 1 1 1 1 0 460 462 496 390 1112
462 1 1 1 1 1 0 461 463 497 391 1113
463 1 2 1 1 1 0 462 464 466 498 392 1114
464 1 1 1 1 1 0 463 465 466 395 1115
465 1 1 1 1 1 0 464 468 467 396 1116
466 1 1 1 1 1 0 463 467 499 464 1117
467 1 1 1 1 1 0 466 470 499 465 1118
468 1 1 1 2 1 0 465 469 470 409 410 1119
469 1 1 1 2 1 0 468 472 471 411 412 1120
470 1 1 1 1 1 0 467 471 500 468 1121
471 1 1 1 1 1 0 470 474 500 469 1122
472 1 1 1 2 1 0 469 473 474 425 426 1123
473 1 1 1 2 1 0 472 476 475 427 428 1124
474 1 1 1 1 1 0 471 475 501 472 1125
475 1 1 1 1 1 0 474 478 501 473 1126
476 1 1 1 2 1 0 473 477 478 441 442 1127
477 1 1 1 2 1 0 476 480 479 443 444 1128
478 1 1 1 1 1 0 475 479 502 476 1129
479 1 1 1 1 1 0 478 482 502 477 1130
480 1 1 1 1 1 0 477 481 482 447 1131
...
1933 1 0 1 1 0 1 1932 1953 1913 1282
1934 0 1 0 1 0 1 1935 1914 1283
1935 1 1 0 1 0 1 1934 1936 1915 1284
1936 1 1 0 1 0 1 1935 1937 1916 1285
1937 1 1 0 1 0 1 1936 1938 1917 1286
1938 1 1 0 1 0 1 1937 1939 1918 1287
1939 1 1 0 1 0 1 1938 1940 1919 1288
1940 1 1 0 1 0 1 1939 1941 1920 1289
1941 1 1 0 1 0 1 1940 1942 1921 1290
1942 1 1 0 1 0 1 1941 1943 1922 1291
1943 1 1 0 1 0 1 1942 1944 1923 1292
1944 1 1 0 1 0 1 1943 1945 1924 1293
1945 1 1 0 1 0 1 1944 1946 1925 1294
1946 1 1 0 1 0 1 1945 1947 1926 1295
1947 1 1 0 1 0 1 1946 1948 1927 1296
1948 1 1 0 1 0 1 1947 1949 1928 1297
1949 1 1 0 1 0 1 1948 1950 1929 1298
1950 1 1 0 1 0 1 1949 1951 1930 1299
1951 1 1 0 1 0 1 1950 1952 1931 1300
1952 1 1 0 1 0 1 1951 1953 1932 1301
1953 1 0 0 1 0 1 1952 1933 1302

```

Figure 23. Edited excerpt of the MODPATH unstructured grid file for example 2.

The MODPATH basic data file is not shown here because it is nearly identical to that used for example 1. The name file used for example 2 is shown in figure 24.

```
MPBAS      example_2.mpbas
MPUGRID    example_2.mpugrid
TDIS       example_2.tdis
HEAD       example_2.hed
BUDGET     example_2.bud
```

Figure 24. MODPATH name file for example 2.

For unstructured grid simulations based on MODFLOW–USG, the spatial and time discretization are split into two files. Spatial discretization is contained in the MODPATH unstructured grid file, example_2.mpugrid (fig. 23). The MODPATH unstructured grid file is designated in the name file by the key word MPUGRID. Time discretization is contained in a separate file, example_2.tdis, designated with the key word TDIS. The time discretization is not shown here. The data format for the time discretization file is identical to time data that appear at the end of the MODFLOW-2005 and MODFLOW–USG discretization files. Separating spatial and time discretization data is useful because it allows the MODPATH unstructured grid file, which is extremely large even for small problems, to be used for multiple flow simulations that have different time discretization.

Simulation 2A – Backward-Tracking Pathline and Time-Series Simulation

Simulation 2A illustrates backward particle tracking from the well cell by placing a ring of 16 particles around faces 1 through 4 at the middepth of layer 3. Figure 25 and figure 26 show the simulation file and starting locations file, respectively.

```
example_2.mpname
example_2a.mplist
4 2 1 1 1 0
example_2a.endpoint7
example_2a.pathline7
example_2a.timeseries7
0
1
0.0
2
1
500 1000.
2
0
CONSTANT 1
CONSTANT 1
CONSTANT 2
1
1
PG1
1
0.0
EXTERNAL example_2a.slloc
```

Figure 25. MODPATH simulation file for simulation 2A.

```

1
2
16 0
1623 0.000 0.125 0.500 0.0 0
1623 0.000 0.375 0.500 0.0 0
1623 0.000 0.625 0.500 0.0 0
1623 0.000 0.875 0.500 0.0 0
1623 1.000 0.125 0.500 0.0 0
1623 1.000 0.375 0.500 0.0 0
1623 1.000 0.625 0.500 0.0 0
1623 1.000 0.875 0.500 0.0 0
1623 0.125 0.000 0.500 0.0 0
1623 0.375 0.000 0.500 0.0 0
1623 0.625 0.000 0.500 0.0 0
1623 0.875 0.000 0.500 0.0 0
1623 0.125 1.000 0.500 0.0 0
1623 0.375 1.000 0.500 0.0 0
1623 0.625 1.000 0.500 0.0 0
1623 0.875 1.000 0.500 0.0 0

```

Figure 26. MODPATH starting locations file for example 2A.

For unstructured grid simulations, particle-starting locations are specified by cell number rather than by layer, row, column. The pathlines generated by backward tracking are shown in figures 27 and 28. As with example 1A, time points are color coded by layer to provide an indication of vertical flow (green for layer 1, gold for layer 2, and red for layer 3).

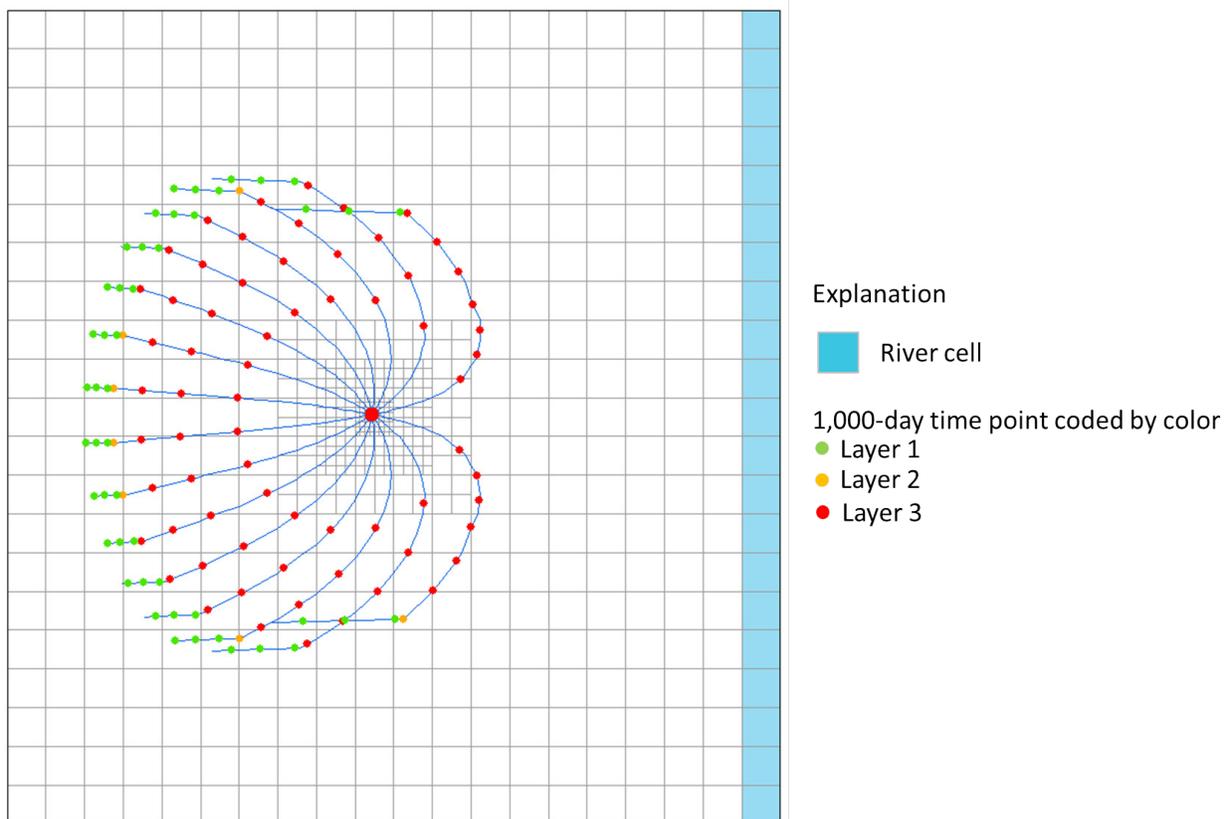


Figure 27. MODPATH pathlines generated by backward tracking from the well in simulation 2A.

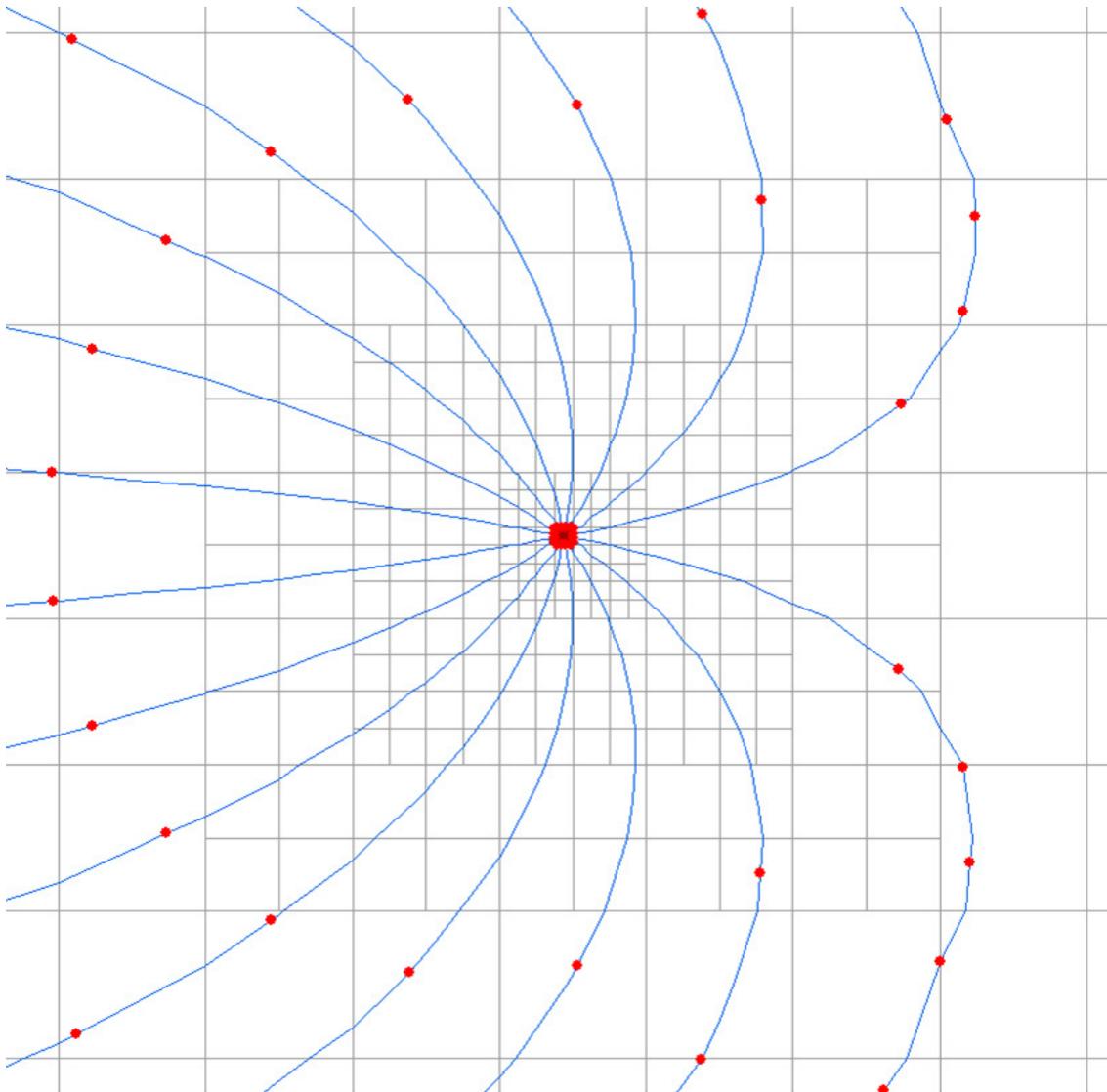


Figure 28. Enlarged view of MODPATH pathlines for simulation 2A.

Simulation 2B – Backward-Tracking Endpoint Simulation

Simulation 2B uses backward particle tracking to generate a capture area for the well. A ten-by-ten array of 100 particles is placed on faces 1 through 4 of cell 1,623, which contains the well. An additional four-by-four array of 16 particles is placed on the top of the cell. Particles are tracked backward using an endpoint analysis. Figure 29 shows the final location of the particles that originated at the well. Figure 30 shows a three-dimensional view of the particles together with the pathlines generated in simulation 2A.

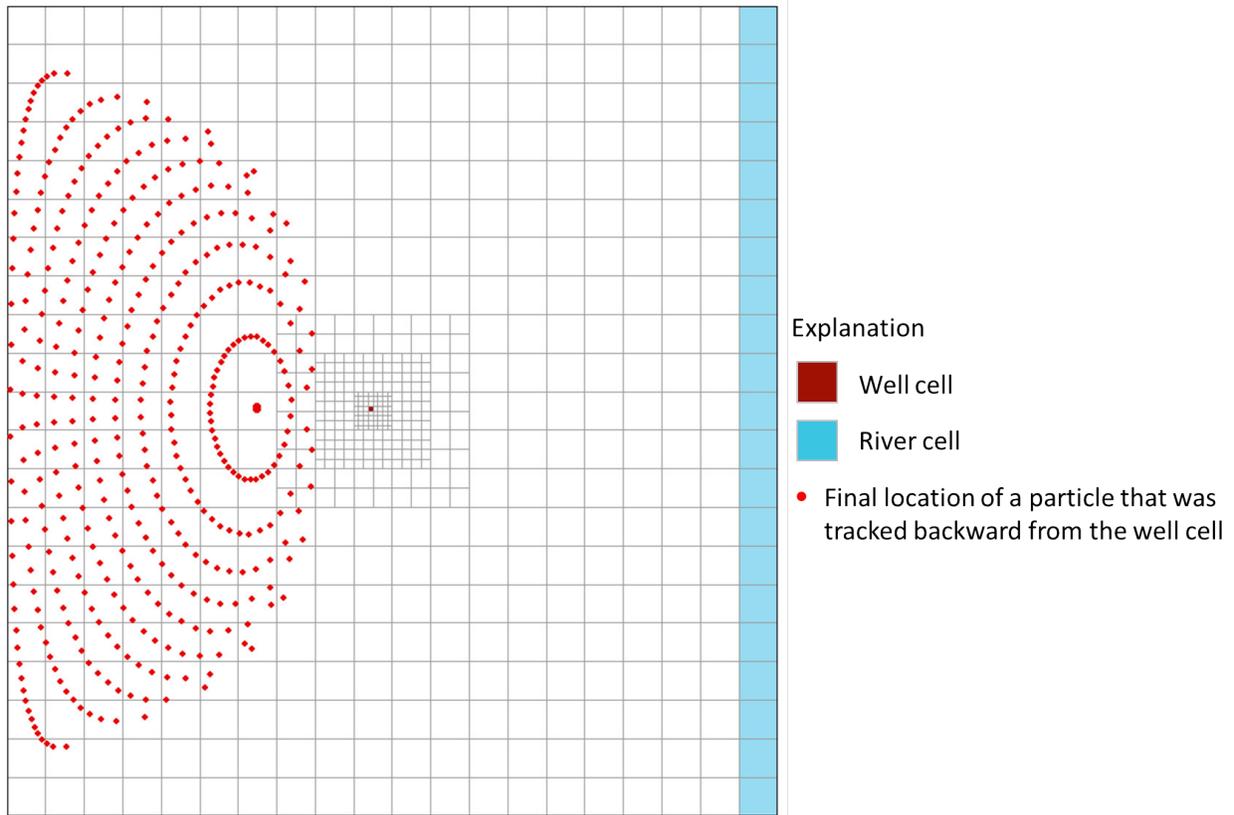


Figure 29. Backward-tracking endpoints for MODPATH simulation 2B.

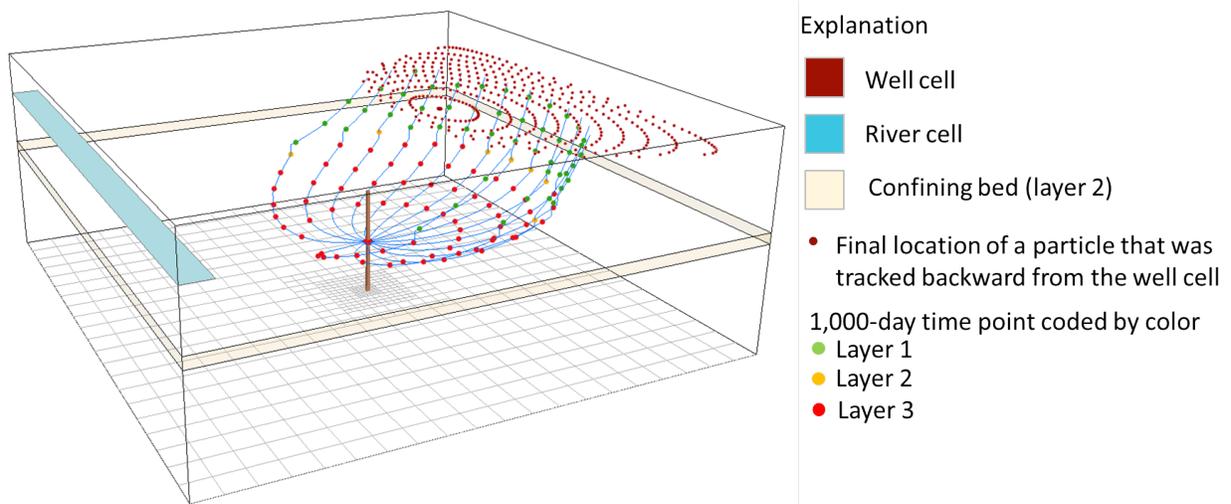


Figure 30. Three-dimensional view of particle-tracking results for MODPATH simulations 2A and 2B.

Summary

This report presents an overview methodology, features, and data requirements of version 7 of the MODPATH particle-tracking model. MODPATH version 7 adds the capability to track particle paths for unstructured grid simulations based on smoothed, rectangular-based quadtree

and quadpatch grids. More detailed information regarding input instructions, output files, and example problems is provided with the software distribution package, which is available online from the U.S. Geological Survey.

References Cited

- Domenico, P.A., and Schwartz, F.W., 1990, *Physical and chemical hydrogeology*: New York, John Wiley and Sons, 824 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 paginated, accessed April 14, 2016, at <https://pubs.er.usgs.gov/publication/tm6A16>.
- Panday, Sorab, Langevin, C.D., Niswonger, R.G., Ibaraki, Motomu, and Hughes, J.D., 2013, MODFLOW-USG version 1: An unstructured grid version of MODFLOW for simulating groundwater flow and tightly coupled processes using a control volume finite-difference formulation: U.S. Geological Survey Techniques and Methods book 6, chap. A45, accessed April 14, 2016, at <https://pubs.er.usgs.gov/publication/tm6A45>.
- Pollock, D.W., 1989, Documentation of a computer program to compute and display pathlines using results from the U.S. Geological Survey modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 89-381, 188 p., accessed April 14, 2016, at <https://pubs.er.usgs.gov/publication/ofr89381>.
- 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., accessed April 14, 2016, at <https://pubs.er.usgs.gov/publication/ofr94464>.
- 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, accessed April 14, 2016, at <https://pubs.er.usgs.gov/publication/tm6A41>.
- Pollock, D.W., 2016, Extending the MODPATH algorithm to rectangular unstructured grids: *Groundwater*, v. 54, no. 1, p. 121–125.

