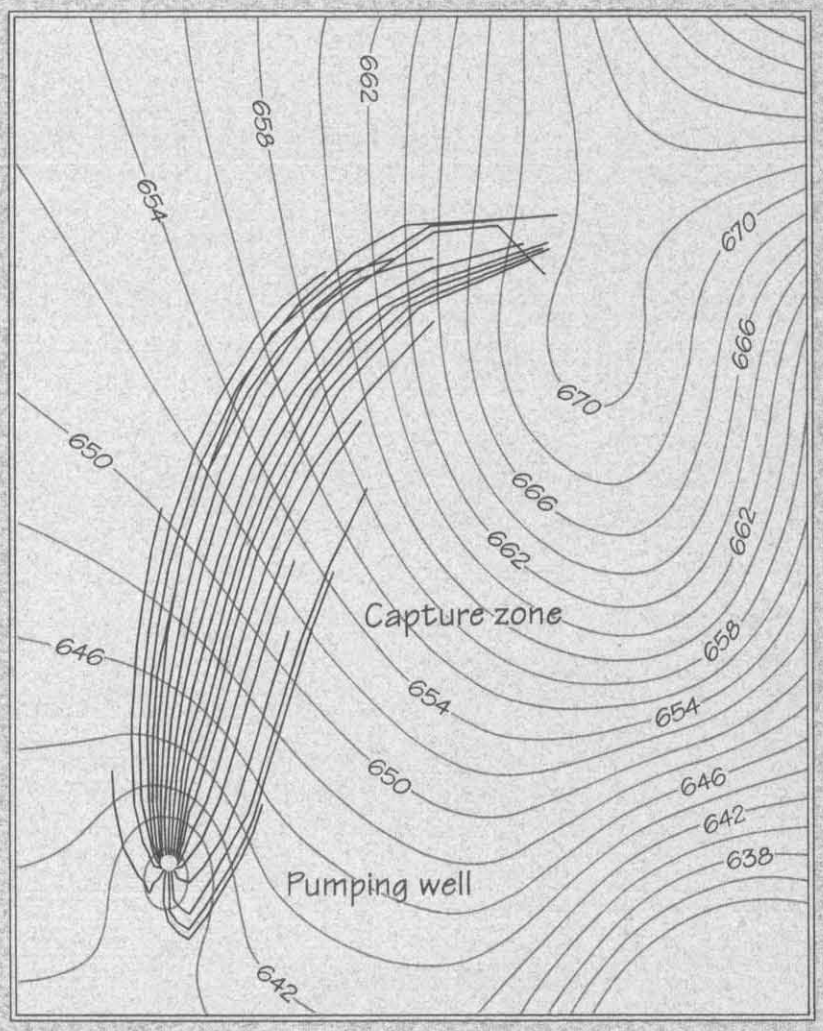
A Computer Model for Calculation of Groundwater Paths and Travel Times in Transient Three-Dimensional Flows

Chunmiao Zheng, Kenneth R. Bradbury, and Mary P. Anderson



Wisconsin Geological and Natural History Survey
Information Circular 70 • 1992

A Computer Model for Calculation of Groundwater Paths and Travel Times in Transient Three-Dimensional Flows

Chunmiao Zheng, Kenneth R. Bradbury, and Mary P. Anderson

Wisconsin Geological and Natural History Survey
Information Circular 70 • 1992

Published by and available from

UWEX University of Wisconsin–Extension
Geological and Natural History Survey
Ronald Hennings, Acting Director and State Geologist
3817 Mineral Point Road, Madison, Wisconsin 53705

1992

ISSN: 0512-0640

CONTENTS

FIGURES

A Computer Model for Calculation of Groundwater Paths and Travel Times in Transient Three-Dimensional Flows

Chunmiao Zheng, Kenneth R. Bradbury, and Mary P. Anderson

ABSTRACT

In this report we describe a computer code for calculation of groundwater flow paths and travel times in transient three-dimensional flow fields. The particle-tracking model uses a velocity interpolator consistent with the block-centered finite-difference representation of flow equations and a fourth-order Runge-Kutta solution capable of automatic step-size adjustment to achieve a predetermined accuracy with minimum computational effort. The model is linked to a modular three-dimensional finite-difference flow model, developed by the U.S. Geological Survey, to allow flexibility in handling a wide range of field problems. Comparisons with analytical solutions verified the accuracy of the particle-tracking code, which can be used to delineate contaminant-capture zones and wellhead-protection areas.

INTRODUCTION

Knowledge of flow paths and travel times is indispensable in studying the movement of contaminants and evaluating the effectiveness of groundwater contamination control (for example, Nelson, 1978). To assess the impact of various hydraulic control strategies and to evaluate alternative aquifer remediation scenarios, it is generally necessary to define the boundaries of contaminant-capture zones on the basis of the calculation of flow paths and travel times. In addition, proper delineation of wellhead-protection zones requires accurate information about the pathlines and travel times of groundwater flow; a wellhead-protection zone program was recently adopted by the U.S. Environmental Protection Agency (for example, Shafer, 1987; Born and others, 1988).

In general, the calculation of flow paths and travel times is carried out through particle tracking, which can be used in steady and transient flows in two or three dimensions. In particle tracking, particles are placed in the flow domain; their positions are monitored as they move through the flow field driven by the hydraulic gradient.

The effect of dispersion on particles typically is neglected in particle-tracking calculations. Thus, the pathlines and travel times calculated with particle tracking reflect

the advective movement of contaminants and only provide an approximation of the actual solute transport process. However, this approximation is generally sufficient to delineate contaminant-capture or wellhead-protection zones because of the large scale involved and the lack of field data about solute chemical properties needed for a detailed transport modeling study.

Particle tracking has been used in several previous studies as a means of deriving two-dimensional groundwater flow paths from the head solution. For instance, Nelson (1978) presented a comprehensive analysis of contaminant arrival times based on a particle-tracking solution. Javendel and others (1984) used a particle-tracking method to define the flow pattern and contaminant front positions. Charbeneau and Street (1979) applied a particle-tracking procedure to generate pathlines and travel times with a finite-element flow model. Shafer (1987) described a numerical technique for particle tracking to delineate well capture zones based on the calculation of groundwater pathlines and travel times.

There are only a few three-dimensional studies in the literature. Feinstein (1986) computed the velocity field by weighing the head differences between neighboring cells determined by a flow model and by tracking particles in steps for a fixed displacement at each step. Mandle and Kontis (1986) employed a basis function approach to interpolate specific discharge at an arbitrary point on the basis of the specific discharge at eight surrounding nodal points. This was accomplished by assigning the specific discharge calculated at cell interfaces by a flow model to nodal points. Groundwater-travel distances and times were then calculated by moving particles over constant time steps.

Both studies used head solutions generated by the U.S. Geological Survey (USGS) modular three-dimensional finite-difference flow model (McDonald and Harbaugh, 1984). However, the velocity interpolating schemes used by these investigators are not consistent with the block-centered finite-difference representation of the three-dimensional flow equation because mass is not conserved in each block and velocity discontinuities occur between blocks of different hydraulic conductivities. Furthermore, there is no documentation regarding the accuracy and efficiency of these particle-tracking procedures.

In this paper we describe a general computer model (referred to as PATH3D) that can calculate groundwater flow paths and travel times in two- or three-dimensional, steady or transient flow fields. This particle-tracking model uses a velocity interpolator consistent with the governing equations used in the USGS modular three-dimensional finite-difference flow model (referred to as MODFLOW in the following discussions) of McDonald and Harbaugh (1984) to which PATH3D is linked for the hydraulic head solution. The model also uses an automatic step-size-adjustment

procedure, which has not been attempted in previous studies. The use of such a procedure makes it possible to achieve a predetermined accuracy with minimum computational effort.

VELOCITY FIELD

The governing equation for the advective movement of particles in groundwater flow systems can be written as

$$\vec{p} = \vec{p}_o + \int_{t_1}^{t_2} \vec{v}(x, y, z, t)\, dt, \qquad (1)$$

where $\vec{p}$ is the particle position vector (x, y, z);
$\vec{p}_o$ is the starting position;
t is time; t_1 and t_2 are starting and ending times, respectively; and
$\vec{v}(x, y, z, t)$ is the seepage velocity vector.

Typically, analytical expressions for the velocity field ($\vec{v}$) are not available. A numerical flow model is often used to solve for hydraulic heads, which are then used to calculate velocity values at an arbitrary point at a given time. MODFLOW was chosen as the flow model for PATH3D because of its wide availability and its flexibility in stimulating complex hydrogeological systems.

MODFLOW is a block-centered finite-difference model. Figure 1 depicts the grid system used in MODFLOW and PATH3D. In MODFLOW the head at the center of cell (i, j, k) is solved on the basis of the heads in the six adjacent cells in the three directions, where i, j, and k indicate the numbers of row, column, and layer, respectively. Flux into cell (i, j, k) along the row direction (the x direction) from cell (i, j-1, k) is determined from the heads at cells (i, j-1, k) and (i, j, k) (see fig. 2):

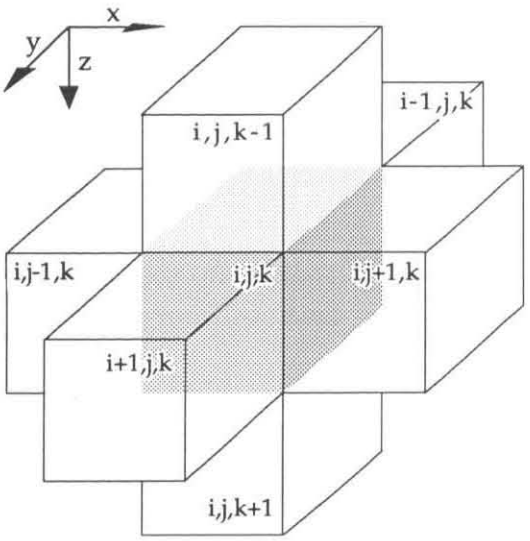


Figure 1. *Grid system used in the finite-difference flow model (adapted from McDonald and Harbaugh, 1984).*

$$q_{i,j-1/2,k} = -K_{i,j-1/2,k} \frac{(h_{i,j,k} - h_{i,j-1,k})}{\Delta x_{i,j-1/2,k}}, \qquad (2)$$

where $q_{i,j-1/2,k}$ is the flux, or the specific discharge, through the interface between cells (i, j-1, k) and (i, j, k);

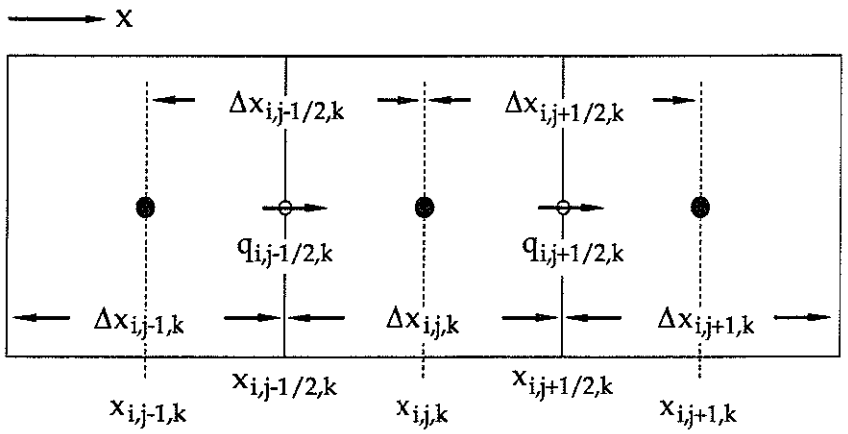


Figure 2. *Calculation of the seepage velocity based on the block-centered finite-difference formulation.*

$$K_{i,j-1/2,k}=\left[\frac{1}{\Delta x_{i,j-1,k}+\Delta x_{i,j,k}}\left(\frac{\Delta x_{i,j-1,k}}{K_{i,j-1,k}}+\frac{\Delta x_{i,j,k}}{K_{i,j,k}}\right)\right]^{-1},$$

the equivalent hydraulic conductivity between cells (i, j-1, k) and (i, j, k);
$h_{i,j,k}$ and $h_{i,j-1,k}$ are heads at nodal points (i, j, k) and (i, j-1, k); and
$\Delta x_{i,j-1/2,k}$ is the distance between nodal points (i, j-1, k) and (i, j, k).

The x component of the seepage velocity (u) at interface (i, j-1/2, k) can then be obtained from

$$u_{i,j-1/2,k}=\frac{q_{i,j-1/2,k}}{\theta}=-\frac{K_{i,j-1/2,k}}{\theta}\frac{(h_{i,j,k}-h_{i,j-1,k})}{\Delta x_{i,j-1/2,k}}, \tag{3}$$

where θ is the porosity of aquifer materials. Similarly, at the interface between cells (i, j, k) and (i, j+1, k), the x component of the seepage velocity is

$$u_{i,j+1/2,k}=\frac{q_{i,j+1/2,k}}{\theta}=-\frac{K_{i,j+1/2,k}}{\theta}\frac{(h_{i,j+1,k}-h_{i,j,k})}{\Delta x_{i,j+1/2,k}}. \tag{4}$$

The x component of the seepage velocity at an arbitrary point (x, y, z) within cell (i, j, k) is computed from the following equation (see also Farmer, 1987, p. 976):

$$u(x) = (1-\alpha)u_{1,j-1/2,k}+\alpha u_{i,j+1/2,k}, \tag{5}$$

where $\alpha=\frac{(x-x_{i,j,k})}{\Delta x_{i,j,k}}+\frac{1}{2}$.

$x_{i,j,k}$ is the x coordinate of the nodal point (i, j, k), relative to the origin set at the upper top left corner of cell (1, 1, 1); α is a number that varies linearly from 0 to 1 between interfaces (i, j-1/2, k) and (i, j+1/2, k). Thus, u(x) is equal to $u_{i,j-1/2,k}$ at the left-side interface, $u_{i,j+1/2,k}$ at the right-side interface, and a linear interpolation of $u_{i,j-1/2,k}$ and $u_{i,j+1/2,k}$ between the two interfaces. The boundary conditions can be handled readily with this velocity scheme (see fig. 3).

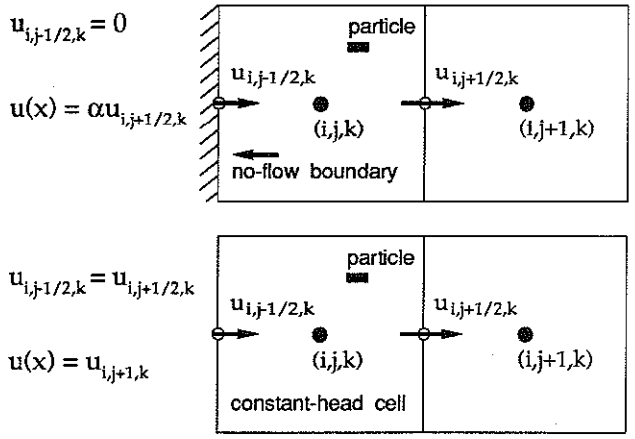


Figure 3. *Particle is located at (x,y,z) within cell (i,j,k).*

The y and z components of the seepage velocity can be calculated analogously. In cases where a particle is located in a water-table cell, the calculation of the z component of the seepage velocity is illustrated in figure 4. If no recharge or evapotranspiration occurs, the water table is a no-flow boundary and a particle placed on the water table would move along the water table. If the net recharge rate is negative (that is, there is discharge out of the aquifer), a particle placed near the water table will travel upward and leave the aquifer. Otherwise, the particle will move downward into the aquifer.

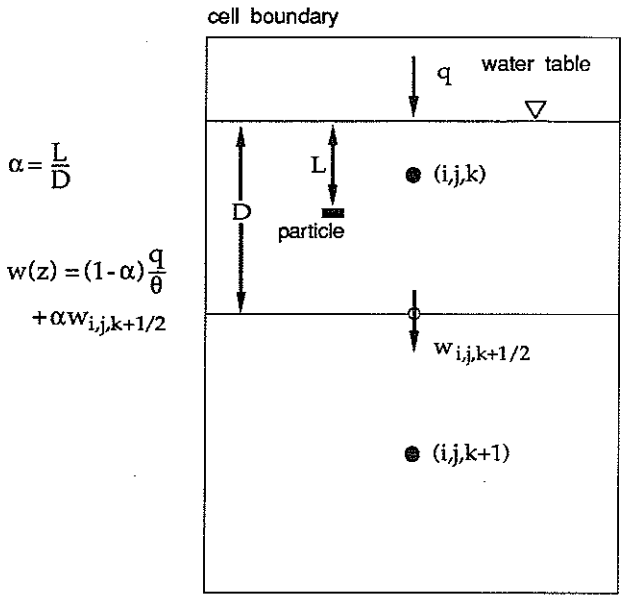


Figure 4. *Calculation of the vertical component (z) of the seepage velocity for the particle located at a water-table cell.*

Russell and Wheeler (1983) showed that the block-centered finite-difference method is also the lowest-order (zero-order) mixed finite-element method, which implies that the velocity solution is of the same truncation error as the head solution. They also showed that the x component of the velocity is continuous in the x direction only and is discontinuous in the y and z directions. Similarly, the y and z components of the velocity are continuous only in the y and z directions, respectively. The velocity interpolation scheme described above is consistent with these observations.

Thus far, we have assumed that the aquifer materials are of uniform porosity. If the porosity is not uniform, the components of the specific discharge instead of the seepage velocity can be interpolated at the particle location, and the specific discharge is converted into the seepage velocity. For example, the specific discharge for the x component [q(x)] within cell (i, j, k)

$$q(x)=(1-\alpha)q_{1,j-1/2,k}+\alpha q_{i,j+1/2,k} \tag{6}$$

and

$$u(x)=\frac{q(x)}{\theta_{i,j,k}}, \tag{7}$$

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

In steady-state flows, the head distribution is constant over time so that the velocity field need be evaluated only once during particle tracking. Under transient conditions, however, when heads change, so does the velocity field. The simulation time in MODFLOW is divided into stress periods, or time intervals during which all external stresses (such as the well pumping rates) are constant (see McDonald and Harbaugh, 1984). Each stress period is, in turn, divided into one or more time steps as illustrated in figure 5. The same time-discretization convention is used in PATH3D. The velocity field is updated at each time step of the head solution. Between two time steps, the velocity field is held constant.

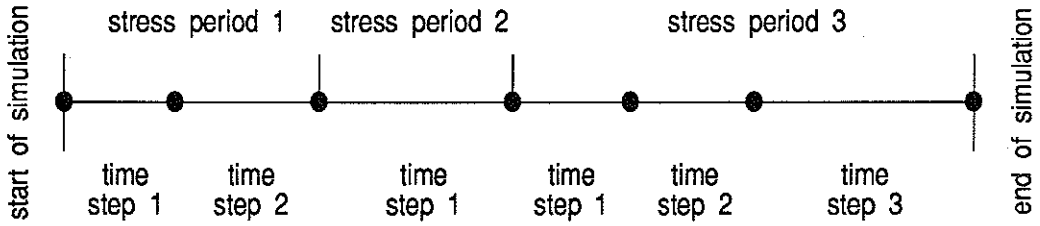


Figure 5. *Discretization of simulation time (adapted from McDonald and Harbaugh, 1984).*

SOLUTION FOR PARTICLE TRACKING

If the velocity field can be evaluated using the scheme described above, equation 1 can then be solved using a first-order Euler's method with very small tracking step sizes (for example, Konikow and Bredehoeft, 1978; Prickett and others, 1981) or a higher-order Runge-Kutta method, which permits larger tracking step sizes (for example, Charbeneau and Street, 1979; Nelson, 1979). The formula for the first-order Euler method is

$$\begin{cases} x_{n+1} = x_n + \Delta t \cdot u(x_n, y_n, z_n, t_n) \\ y_{n+1} = y_n + \Delta t \cdot v(x_n, y_n, z_n, t_n) \\ z_{n+1} = z_n + \Delta t \cdot w(x_n, y_n, z_n, t_n) \end{cases}, \qquad (8)$$

where Δt is the time step size;
(x_n, y_n, z_n) and $(x_{n+1}, y_{n+1}, z_{n+1})$ are the positions of the particle at tracking steps n and n+1, respectively; and
u, v, and w are the components of the seepage velocity in the x, y, and z directions, respectively.

Note that the tracking step is different from the time step used in the head solution. One way to distinguish them is to visualize the tracking steps as "small" steps and the time steps as "large" steps. A particle may take a number of small steps to move from one large step to another.

Equation 8 advances a particle from position (x_n, y_n, z_n) to $(x_{n+1}, y_{n+1}, z_{n+1})$ over a time interval (Δt), but it only uses the velocity calculated at the beginning of the time interval (t_n). Thus, unless Δt is very small, the solution is not accurate. For greater accuracy and efficiency, a higher-order solution like the fourth-order Runge-Kutta method is commonly used.

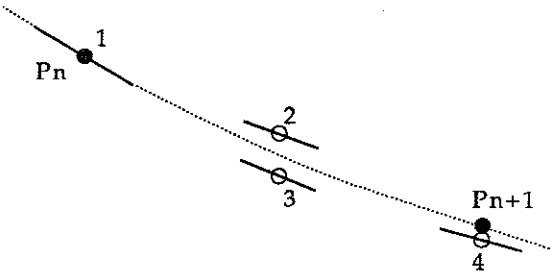


Figure 6. *Fourth-order Runge-Kutta solution. In each step, the velocity is evaluated four times: once at the initial point (P_n), twice at trial midpoints, and once at trial endpoint. From these velocities the final position of the particle (shown as a filled dot, P_{n+1}) is calculated (modified from Press and others, 1986).*

The basic idea of the fourth-order Runge-Kutta method is to evaluate the velocity four times for each step: once at the initial point, twice at two trial midpoints, and once at a trial endpoint (fig. 6). From velocity values evaluated at these four points, the position for the next step $(x_{n+1}, y_{n+1}, z_{n+1})$ is determined. This process may be expressed as follows:

$$\begin{cases} x_{n+1} = x_n + 1/6(k_1 + 2k_2 + 2k_3 + k_4) \\ y_{n+1} = y_n + 1/6(l_1 + 2l_2 + 2l_3 + l_4) \\ z_{n+1} = z_n + 1/6(m_1 + 2m_2 + 2m_3 + m_4) \end{cases}, \qquad (9)$$

where $k_1 = \Delta t u(x_n, y_n, z_n, t_n)$
$k_2 = \Delta t u(x_n + k_1/2, y_n + l_1/2, z_n + m_1/2, t_n + \Delta t/2)$
$k_3 = \Delta t u(x_n + k_2/2, y_n + l_2/2, z_n + m_2/2, t_n + \Delta t/2)$
$k_4 = \Delta t u(x_n + k_3, y_n + l_3, z_n + m_3, t_n + \Delta t)$

$l_1 = \Delta t v(x_n, y_n, z_n, t_n)$
$l_2 = \Delta t v(x_n + k_1/2, y_n + l_1/2, z_n + m_1/2, t_n + \Delta t/2)$
$l_3 = \Delta t v(x_n + k_2/2, y_n + l_2/2, z_n + m_2/2, t_n + \Delta t/2)$
$l_4 = \Delta t v(x_n + k_3, y_n + l_3, z_n + m_3, t_n + \Delta t)$

$$m_1 = \Delta t w(x_n, y_n, z_n, t_n)$$
$$m_2 = \Delta t w(x_n + k_1/2, y_n + l_1/2, z_n + m_1/2, t_n + \Delta t/2)$$
$$m_3 = \Delta t w(x_n + k_2/2, y_n + l_2/2, z_n + m_2/2, t_n + \Delta t/2)$$
$$m_4 = \Delta t w(x_n + k_3, y_n + l_3, z_n + m_3, t_n + \Delta t).$$

It is obviously important to select an appropriate time step size (Δt) for either Euler's method or the fourth-order Runge-Kutta method. If the step size taken is too large, the particle may deviate from the actual path and the results will be inaccurate. On the other hand, if the step size taken is too small, it may take so long for the particle to move a certain distance that the solution is inefficient. In most previous studies in which particle tracking was used for calculating pathlines and travel times, a constant step size was used (for example, Shafer, 1987). However, as pointed out by Press and others (1986), some adaptive control over steps should be implemented to monitor errors and make frequent changes in step sizes to achieve a predetermined accuracy with minimum computational effort. Press and others (1986) noted that the resulting gain in efficiency can sometimes be a factor of 10, 100, or more.

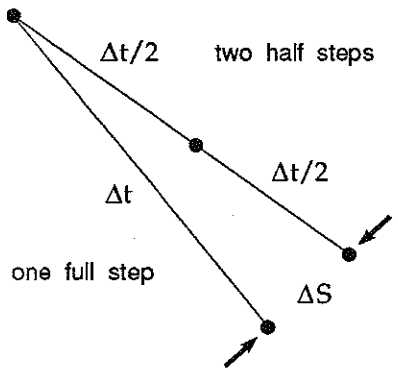


Figure 7. *Step-doubling as a means for adaptive step-size control in the fourth-order Runge-Kutta solution.*

An automatic step-size adjustment procedure developed by Press and others (1986) was modified and implemented in the particle-tracking model. The procedure is based on step doubling. A time step (Δt) is always taken twice, once as a full step, and once as two half steps (fig. 7). If Δt is small enough, the resulting difference in the particle locations, denoted by ΔS, would by small. Because the basic solution is accurate to the fourth-order, ΔS can be scaled as $(\Delta t)^5$:

$$\left\{\frac{\Delta t_o}{\Delta t}\right\}^5 = \frac{\Delta S_o}{\Delta S}. \qquad (10)$$

If a given step size (Δt) results in an error (ΔS), the step size (Δt_o) that would have resulted in some other error (ΔS_o) can be estimated from the relationship

$$\Delta t_o = f_s \Delta t \left(\frac{\Delta S_o}{\Delta S}\right)^{0.2}, \qquad (11)$$

where f_s is a safety factor that is a few percent smaller than unity (for example, 0.9).

Let ΔS_o denote some predetermined accuracy. Then, if ΔS is larger than ΔS_o, equation 11 tells how much to decrease the step size when the model makes another try (the present step Δt is rejected). If ΔS is smaller than ΔS_o, equation 11 tells how much to increase the step size for the next step (the present step Δt is successful).

Note that the accuracy indicator ΔS_o is actually a vector. Its three components are ΔX_o, ΔY_o, and ΔZ_o in the row, column, and layer directions. The three components can be linked into one error criterion (ε) by the following equation:

$$\begin{cases} \Delta X_o = \varepsilon \cdot XMAX \\ \Delta Y_o = \varepsilon \cdot YMAX, \\ \Delta Z_o = \varepsilon \cdot ZMAX \end{cases} \tag{12}$$

where ε is a dimensionless factor; XMAX, YMAX, and ZMAX are the maximum lengths of the flow domain in the row, column, and layer directions, respectively.

Thus, in the program input, one only needs to enter an error criterion (ε), and the model will calculate the maximum allowed errors in the x, y, and z directions according to equation 12. In cases where the aquifer is long and thin, or ZMAX is much smaller relative to XMAX and YMAX, the maximum error allowance in the z direction will be much smaller than that in the x and y directions.

Under many circumstances, it may be more convenient to track particles backwards (for example, downgradient to upgradient in the flow field, as in the delineation of well capture zones). This can be done readily in steady-state flow fields by using negative time step sizes (set ending time t_2 smaller than starting time t_1). In transient flows, however, backward tracking requires the reversing of the head solution generated by a flow model and is not conceptually straightforward. The current version of PATH3D does not support backward tracking in transient flow fields.

When a particle enters a cell containing a sink such as a well, a drain, or a river reach, the velocity components on the six faces of the model cell are generally directed inward toward the nodal point. If this is the case, the particle cannot leave the cell again. Therefore, the movement of this particle is terminated at the cell containing the sink. However, if wells, drains, or rivers do not create complete sinks at cells containing them, that is, if there are outward velocity components on one or more of the cell faces, particles entering these cells will continue to travel and may leave these cells again. If it is more desirable to terminate the movement of a particle as soon as it encounters any well, drain, or river cell, subroutine MOVE can be easily modified to do so.

Stagnation zones may exist in the flow field where the groundwater seepage velocity is theoretically equal to zero. A particle approaching a stagnation zone will be removed at a point where all the velocity components of the particle at that point are smaller than a tiny number (10^{-30}). This number can be changed to accommodate specific circumstances by modifying the parameter statement (TINY=10^{-30}) in subroutine MOVE.

COMPUTER PROGRAM

The source code for the particle-tracking model PATH3D was written in standard FORTRAN 77 with a modular structure. The code consists of a main program and three subprograms, each of which contains several subroutines. The code and a user's guide are being distributed by S.S. Papadopulos and Associates, Inc., of Rockville, Maryland, U.S.A., which will also provide necessary technical support and upgrades to registered users.

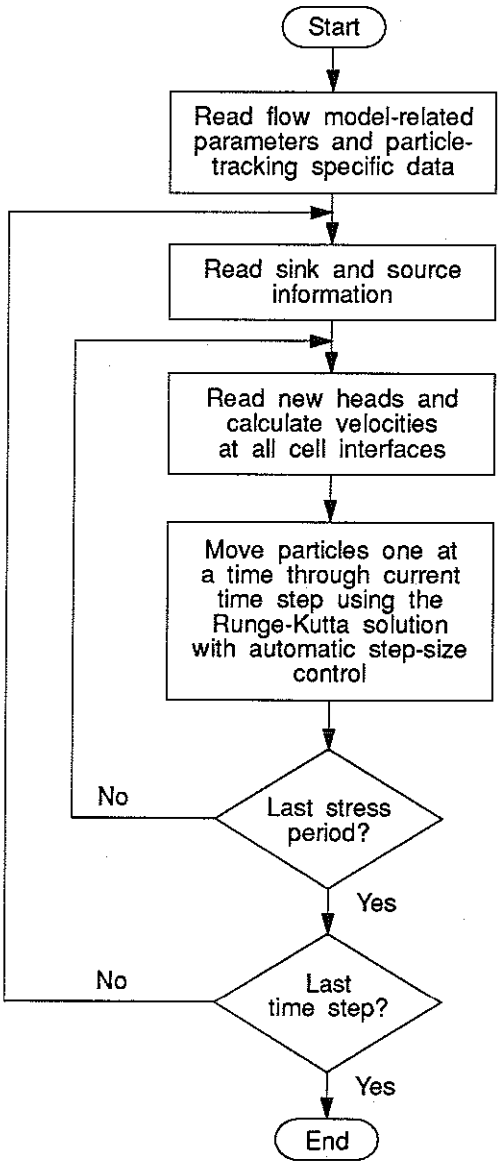


Figure 8. *Flow chart for the main program.*

Main Program P3DMAIN2

The main program P3DMAIN2 (2 denotes the current version number) controls the overall execution of the program by calling subroutines to execute specific tasks in the following order. Figure 8 provides a simplified flow chart for the main program.

1. Set the length of the Y array, a one-dimensional array in which all data arrays are stored.
2. Open input and output files, and assign them to certain units.
3. Allocate space in the Y array for individual data arrays. If the Y array is not dimensioned large enough, terminate the program.
4. Read and prepare information that is constant throughout the entire simulation.
5. For each stress period:
   (a) Read and prepare stress-period timing information if the simulation is transient.
   (b) Read sink or source information that may change each stress period. The sinks or sources may include wells, drains, rivers, recharge, evapotranspiration, or general-head boundaries.
   (c) For each step:
      (1) Read the head solution saved in

an unformatted or formatted file by the flow model.

(2) Calculate the length of the current time step and determine whether it is covered by the tracking-time interval. If yes, proceed. Other wise, go to the next time step. Note that if the head solution is steady state, the length of the current time step is set to be equal to the length of the tracking-time interval.

(3) Compute the velocity components at nodal interfaces for all cells.

(4) Move all particles via a series of tracking steps from the beginning of the current time step to the end of the time step using the Runge-Kutta solution with automatic step-size control.

6. End program.

Subprogram LNKMOD3D

It is necessary to use PATH3D in conjunction with a flow model to obtain the head solution. The current version is linked to MODFLOW (McDonald and Harbaugh, 1984). The primary function of subprogram LNKMOD3D is to extract parameters necessary for particle tracking from input files for MODFLOW and to read hydraulic heads solved and saved by MODFLOW. The subprogram includes the following subroutines: GETDIM, GETDAT, GETSS, READH, U2DINT, U1DREL, U2DREL, and IGNORE.

GETDIM reads the numbers of layers, rows, columns, and stress periods used in the head solution and calculates the location of each individual array in the one-dimensional Y array. GETDAT reads model discretization and hydraulic parameters from input files for MODFLOW as well as particle-tracking parameters from an input file that is set up exclusively for PATH3D. Subroutine GETSS reads information related to sinks or sources, which may change every stress period, from input files for MODFLOW. These sinks or sources may be rivers, drains, wells, recharge, evapotranspiration, or general-head boundaries. READH is a subroutine used to input heads solved by MODFLOW and saved in an unformatted or formatted file.

U2DINT, U1DREL, and U2DREL are utility subroutines modified from McDonald and Harbaugh (1984), which are called by subroutines GETDAT and GETSS to read a two-dimensional integer array, a one-dimensional real array, or a two-dimensional real array, respectively. IGNORE is similar to U2DREL except that it is used to read data arrays that are entered in the input files for MODFLOW but are not needed by PATH3D.

In addition to MODFLOW, PATH3D can be used with any other block-centered finite-difference flow model. The subprogram LNKMOD3D can be easily modified to link PATH3D to other flow models.

Subprogram VELOCITY

Subprogram VELOCITY includes three subroutines: VFACE, VPOINT, and LOCATE. Subroutine VFACE is used to calculate the components of the specific discharge at interfaces between neighboring cells along the x, y, and z directions and to store them in three-dimensional arrays (QX, QY, and QZ, respectively). If the recharge or evapotranspiration option is used in the head solution, VFACE determines the effective (net) recharge rates and takes them as the vertical component of the specific discharge at the water table. VFACE also takes into account the constant-head or general-head boundaries if they are present in the head solution. QX, QY, and QZ are then passed to subroutine VPOINT to calculate the specific discharge at a given point between two cell interfaces on the basis of the interpolation scheme described previously. The specific discharge is then divided by the effective porosity to obtain the seepage velocity at the same point.

Subroutine LOCATE is used to determine the location of an arbitrary point (x, y, z) in an irregular finite-difference grid defined by the column, row, and layer number (or the j, i, and k indices). To do so, LOCATE uses a simple bisection searching algorithm (Press and others, 1986). LOCATE also checks whether the particle location is outside the model edges, above the water table, or at an inactive cell.

Subprogram TRACKING

This subprogram is the core of the particle-tracking program. It moves particles through the flow field over a time period by using a fourth-order Runge-Kutta solution with an automatic step-size control procedure. The subprogram contains three subroutines: RK4, RKASC, and MOVE.

Subroutine RK4 is a direct implementation of equation 9. It advances a particle over a time increment using the fourth-order Runge-Kutta solution. Subroutine VPOINT is called several times to evaluate the velocity components needed for the Runge-Kutta solution.

Subroutine RKASC monitors the accuracy of the Runge-Kutta solution and adjusts the size of each tracking step to achieve the accuracy specified by the user with minimum computational effort. For any given time increment, RKASC always takes one full step and two half steps by calling RK4 and then compares the resulting error (the difference in the particle position scaled by a factor). If the error is larger than the error criterion input by the user, a smaller step size will be determined and the procedure will be repeated until the calculated error is smaller than the error criterion. RKASC will then return with the new particle position and an estimate of the step size for the next tracking step. If a particle is about to exit the

model edge, enter an inactive cell, or leave the water table, RKASC will try to determine the time required for the particle to travel from its current position to the exit point and then will move the particle to the location of the exit point.

Given a time period from t_1 to t_2, subroutine MOVE calls subroutine RKASC and advances a particle from its starting position at t_1 to a new position at t_2 through a series of tracking steps. As one particle moves step by step, its intermediate positions and travel times will be saved according to the options chosen by the user. If the particle is trapped at a sink or stagnation point, or if the particle is leaving the active flow domain from model edges or the water-table cells, the particle will be removed. Otherwise, the procedure will be repeated for the next particle until all particles have been processed. If the flow field is steady state, subroutine MOVE is called by the main program only once to move all particles from the start to the end of the tracking-time interval specified by the user. However, if the flow field is transient, MOVE will be called by the main program more than once, depending on the number of time steps used in the head solution. MOVE will be called every time step to advance particles from the beginning to the end of the time step, until the end of either the specified tracking-time interval or the head simulation time.

VALIDATION AND APPLICATION

The accuracy of the particle-tracking program can be evaluated by using it to solve relatively simple problems for which analytical solutions of flow paths and travel times are available. Several examples used for this purpose are discussed in this section. These examples also illustrate how the program can be applied to delineate contaminant-capture zones or wellhead-protection zones.

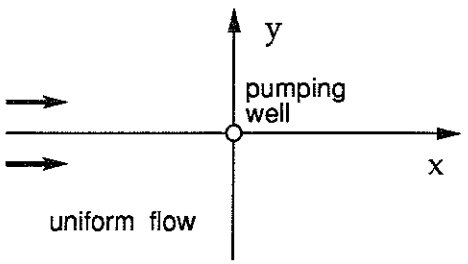


Figure 9. *Flow field examined in example 1.*

Example 1

The first example is shown in figure 9. The aquifer is assumed to be confined and infinite in areal extent; a pumping well is situated in the originally uniform flow field. The steady-state solutions for the hydraulic head (h) and stream function (ψ) can be derived by superposition of a uniform flow on a sink (the pumping well):

$$h(x,y) = Ix + \frac{Q_w}{2\pi Kb} \ln\left(\sqrt{x^2+y^2}/r_w\right) + h_w \qquad (13a)$$

and

$$\psi(x,y) = -KIy - \frac{Q_w}{2\pi b}\tan^{-1}(y/x) + \frac{Q_w}{2b}, \qquad (13b)$$

where I is the gradient of the uniform flow field (dh/dx);
Q_w is the pumping rate (positive);
K is the hydraulic conductivity;
b is the thickness of the confined aquifer;
r_w is the well radius; and
h_w is the head at the radius of the well.

The flow domain was discretized into 41 columns and 21 rows with a uniform spacing of 1 m by 1 m for each cell. The head and stream function values at each cell were calculated from equation 13, given the following set of parameters:

$I = -0.01$ m/m;
$Q_w = 1.0$ m^3/day;
$K = 10.0$ m/day;
$b = 1.0$ m;
$r_w = 0.01$ m; and
$h_w = 10$ m.

A small value of r_w was used to represent a point sink. The flow paths, also the pathlines under the steady-state condition, were obtained by contouring the stream function distribution at nodal points and are illustrated in figure 10 as solid lines. Dots show the flow paths as calculated by the particle-tracking program with the error criterion (ε) set to 10^{-4} and a uniform porosity (θ) equal to 0.2. The particles were placed downgradient and tracked backwards. Each dot along a pathline represents the location of a particle at a particular tracking step. Note that the particles travel at unequal steps to achieve maximum efficiency within a predetermined accuracy. The figure shows that the flow paths solved by the particle-tracking program are virtually identical to those obtained from the analytical solution.

The particle travel times can also be compared with the analytical solution. Take particle 10, which moves along the x-axis, as an example. Because the flow domain is symmetrical about the x-axis, the y component of the seepage velocity is zero along the x-axis. Hence, the travel time of particle 10 can be readily computed from

$$t = \int_{x_s}^{x_e} \left(-\frac{K}{\theta}\frac{\partial h}{\partial x}\right)^{-1} dx = \int_{x_s}^{x_e} \left[-\frac{K}{\theta}\left(I + \frac{Q_w}{2\pi K b}\frac{1}{x}\right)\right]^{-1} dx, \qquad (14)$$

where x_s and x_e are the x coordinates of the particle at the starting and ending points, respectively.

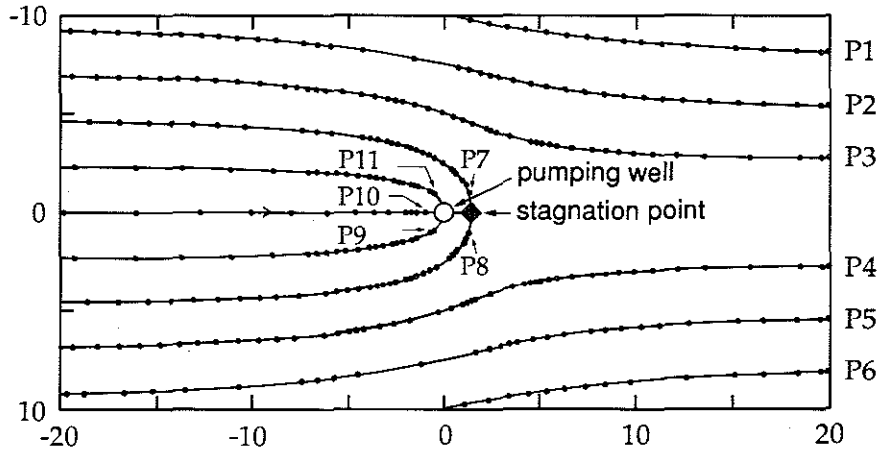


Figure 10. *Flow paths calculated by the analytical solution (solid lines) and PATH3D (filled dots). P1 to P11 indicate the initial particle locations. Particles were placed downgradient and tracked backward. Distances shown in the figure are in meters.*

Substituting x_s and x_e with -0.9 m and -19.9 m, the value of t directly integrated from equation 14 is 31.1 days, or less than 1 percent different from the value calculated by PATH3D (30.8 days).

Note that particles 7 and 8 move along the dividing streamlines; that is, they define the boundary of the capture zone for the pumping well. It is more appropriate to delineate the wellhead-protection zone on the basis of the boundary of the well capture zone. For this simple problem, the analytical solution is available. However, under more complicated conditions, the particle-tracking program may be the only tool to aid the proper delineation of contaminant-capture zones or wellhead-protection zones.

Example 2

Example 2 is intended to demonstrate the accuracy of the particle-tracking program as applied to a heterogeneous aquifer. The cross section of a layered aquifer is shown in figure 11. The aquifer is bounded by no-flow boundaries at the left, right, and bottom sides. The top side is a specified-head boundary representing a linear water table having a slope of 1/100. The aquifer was divided into 51 columns, 1 row, and 21 layers, with a uniform spacing of 2 m by 1 m by 2 m for each cell. The porosity is assumed to be 0.2 and the hydraulic conductivity values are 20 m/day for the top and bottom layers and 1 m/day for the middle layer. The flow path of the particle as computed using PATH3D under the steady-state condition is shown in figure 11. The particle was tracked backward, as in the last example. The

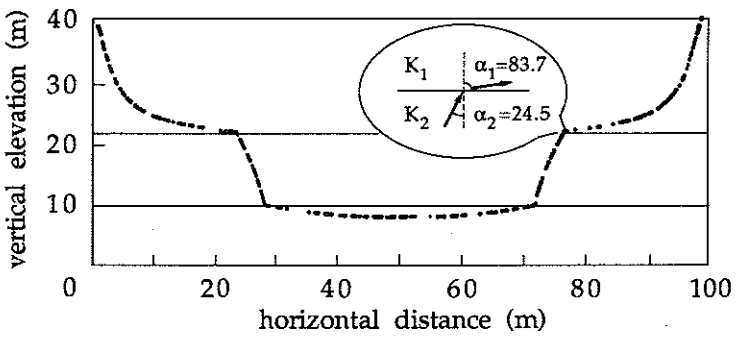


Figure 11. *Refraction of a flowline in a layered aquifer.*

error criterion was set to 10^{-4}, as in the first example.

Note that at the interface between the two layers of different hydraulic conductivities (K_1 and K_2), the pathline sharply changes direction. According to the tangent law (see Freeze and Cherry, 1979, p. 172), the ratio of the hydraulic conductivity in the upper layer to the hydraulic conductivity in the lower layer (K_1/K_2) must equal $\tan(\alpha_1)/\tan(\alpha_2)$ (see the enlarged part of fig. 11). In this example, $\tan(\alpha_1)/\tan(\alpha_2) = \tan(83.7)/\tan(24.5) = 19.9$, which is close to $K_1/K_2 = 20$. Thus, this example illustrates that PATH3D can correctly simulate the advective movement of water particles in heterogeneous media.

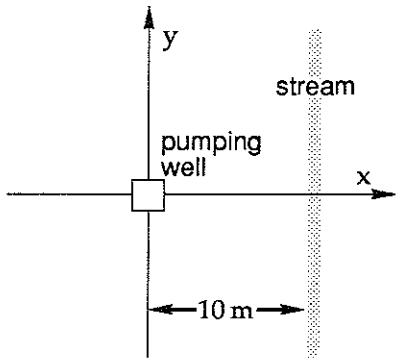


Figure 12. *Configuration of the flow domain used in example 3.*

Example 3

The third example is used to examine the accuracy of PATH3D in solving transient problems. The configuration of the flow domain used in this example is shown in figure 12. The confined aquifer is 10 m thick. The flow field is bounded by a fully penetrating stream on the right side and assumed to be infinite in the left half-space. Under the assumed initial condition, the potentiometric surface of the aquifer is horizontal and at the same level as the stream stage. A pumping well represented by a rectangular cell (1 m by 1 m) is 10 m from the stream channel. After the well starts pumping at a given rate of 0.1 m^3/min, the particles initially placed at the edge of the stream begin to move toward the well. The other hydraulic parameters used in the calculation are transmissivity (T) = 0.02 m^2/min; storage coefficient (S) = 0.002; porosity (θ) = 0.4.

The head field was simulated by setting the no-flow boundaries at the left, upper, and lower sides far away from the pumping well so that the flow field can be approximately treated as infinite in the half-domain. The stream was modeled as a constant-head boundary. A period of 20,000 minutes was simulated with 20 time steps. The flow paths as shown in figure 13 were calculated by PATH3D with the error criterion set to 10^{-4}. It took particles 1, 2, and 3 approximately 14,257 min, 9,860 min, and 8,120 min, respectively, to travel from the stream to the well. Because the

flow field is symmetrical about the x-axis, the travel times for particles 4 and 5 were the same as for 1 and 2, respectively.

The flow paths and travel times in this example can also be obtained by integration of the governing ordinary differential equation. For simplicity, take particle 3, which travels along the x-axis, as an example. The head distribution along the x-axis can be derived by using the Theis solution and by simulating the stream using an image well. The velocity distribution can then be evaluated on the basis of the head solution:

$$\frac{dx}{dt} = -\frac{K}{\theta}\frac{\partial h}{\partial x} = \frac{K}{\theta}\frac{Q_w}{4\pi T}\left[\frac{dW(u_1)}{du_1}\frac{\partial u_1}{\partial x} - \frac{dW(u_2)}{du_2}\frac{\partial u_2}{\partial x}\right]$$

$$= \frac{K}{\theta}\frac{Q_w}{4\pi T}\left[-\frac{2}{x}e^{-u_1} - \frac{2}{(20-x)}e^{-u_2}\right], \qquad (15)$$

where $u_1 = \frac{x^2 S}{4Tt}$;

$u_2 = \frac{(20-x)^2 S}{4Tt}$;

$W(u_1)$ is the well function of the pumping well; and
$W(u_2)$ is the well function of the image well.

Equation 15 was evaluated using the first-order Euler's method with a small time step equal to 3 min to obtain the travel time of particle 3, which is equal to 8,364 min, about 3 percent different from 8,120 min calculated by PATH3D.

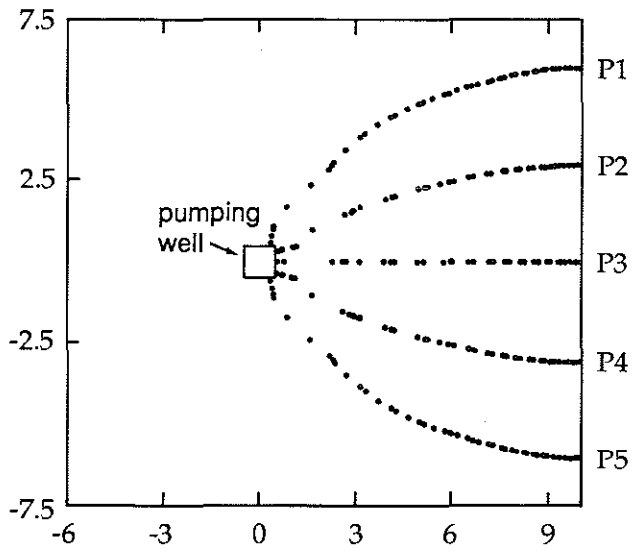


Figure 13. *Flow paths in the transient flow field as calculated by PATH3D. (P1 to P5 indicate the initial particle locations.)*

Example 4

The final example was designed to verify the accuracy of the particle-tracking model to solve three-dimensional problems. The configuration of the flow domain examined in this example is shown in figure 14. The boundary conditions for the unconfined aquifer are no-flow along up-

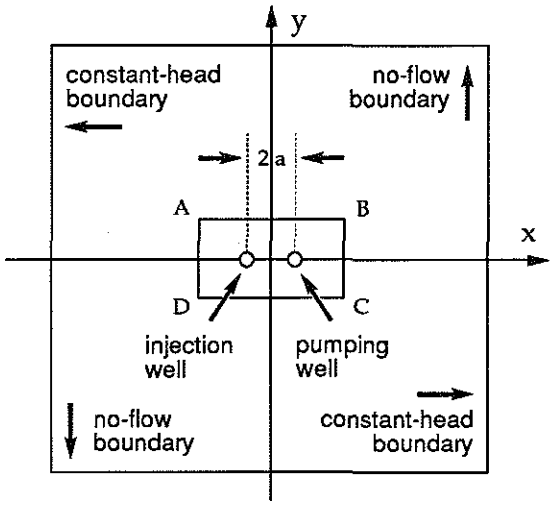


Figure 14. *Flow domain and boundary conditions for example 4.*

per and lower sides and constant-head along the left and right sides. Given that the constant-head at the left and right boundaries is 90.0 m and 89.5 m, respectively, the average gradient along the x direction is -0.5 m/131 m = -3.8×10^{-3}, where 131 m is the distance between the left and right boundaries. A pair of injection/pumping wells is placed near the center of the flow domain to contain and clean up contaminants between the two wells. The lengths of the well screens (1 m) are shallow in relation to the aquifer thickness (90 m). The flow field is assumed to be steady-state.

The flow domain was discretized into 31 columns, 21 rows, and 6 layers with 35 percent of the cells distributed within region ABCD to focus on the flow field near the wells. Given that the pumping and injection rate is 50 m^3/day, porosity is 0.2, and hydraulic conductivity is 100 m/day, the containment zone created by the pair of injection/pumping wells, as calculated by PATH3D, is illustrated in figure 15. Figure 15A is the projection of the containment zone on the water table; figure 15B depicts the three-dimensional view of the containment zone by contouring the elevation of particles that travel along the dividing surface between the capture zone and the background aquifer. The error criterion for this example was set to 10^{-6}.

By treating the injection and pumping wells as a point source and point sink superposed on a uniform flow field (h = Ix), Wilson (1984) presented analytical solutions for calculating the maximum depth and half-width (L) of the containment zone,

$$\left(\frac{L}{a}\right)^2 \sqrt{1+\left(\frac{L}{a}\right)^2} = -\frac{2Q}{\pi K I a^2} \qquad (16)$$

and for calculating the characteristic flushing time (t), or the shortest travel time between the injection well and pumping well,

$$t = 2\theta \int_0^a \left[\frac{1}{\pi} \left| \frac{Q(x^2+a^2)}{(x^2-a^2)^2} \right| - KI \right]^{-1} dx, \qquad (17)$$

where Q is the recirculation rate (positive);

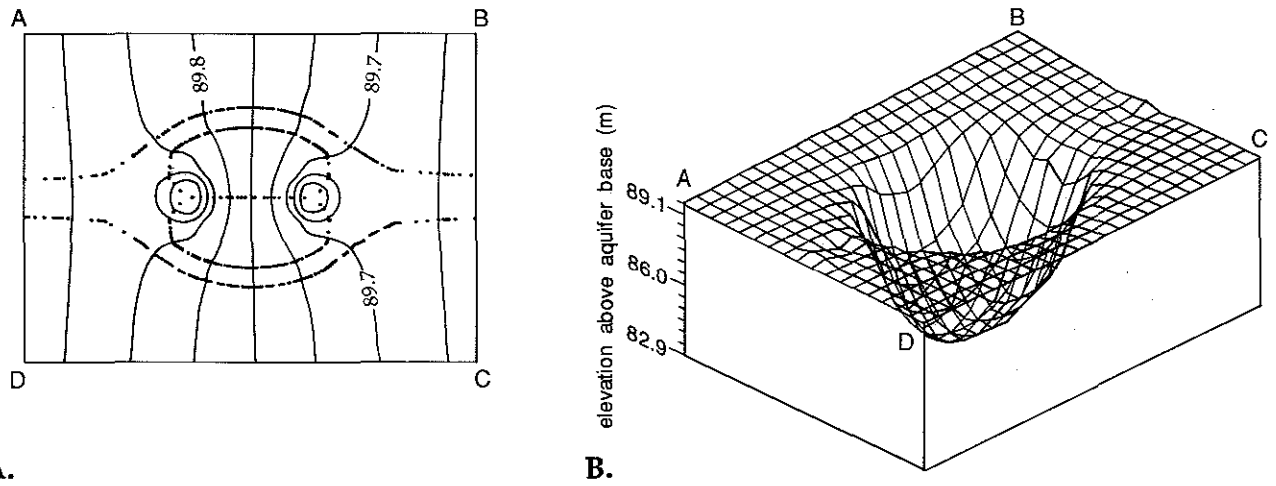


Figure 15. *A: Projection of the containment zone on the water table. (See fig. 14 for the location of ABCD in the flow domain.) B: Three-dimensional view of the containment zone.*

a is half the distance between the injection and pumping well; and
I is the gradient of the uniform flow field.

Equation 16 was solved directly and the value of L was found to be 6.98 m. Equation 17 was evaluated using numerical integration and the result is t=1.05 days. These values are close to those calculated by PATH3D, which yields L=7.05 m and t=1.03 days. Thus, this example indirectly verifies the feasibility and accuracy of the particle-tracking model in delineating contaminant-capture zones in three-dimensional flow fields.

CONCLUSIONS

We evaluated the accuracy of PATH3D by using it to solve relatively simple problems for which analytical solutions were available. The four examples presented show excellent agreement between the numerical and analytical results. However, these examples are idealized and do not cover all the situations. Users should check the accuracy of the results with available data and common sense.

The model can be applied to a wide range of field problems, such as delineation of contaminant-capture zones or wellhead-protection zones, calculation of groundwater resident times, or evaluation of groundwater contamination potential. However, it is important to point out that the appropriate use of this program requires accurate information about the flow field and hydraulic heads. The model can give meaningful results only if the flow field is correctly simulated.

ACKNOWLEDGMENTS

We wish to thank Professor John L. Wilson at New Mexico Institute of Technology, Socorro, New Mexico, and Daniel T. Feinstein at Geraghty and Miller, Inc., of Milwaukee, Wisconsin, for many helpful discussions during the development of this program. The senior author is also very grateful for the support provided by S.S. Papadopulos and Associates, Inc., during the revision and modification of the report and the computer codes.

REFERENCES

Born, S.M., D.A. Yanggen, A.R. Czecholinski, R.J. Tierney, and R.G. Hennings, 1988, Wellhead-protection districts in Wisconsin: An analysis and test applications: Wisconsin Geological and Natural History Survey Special Report 10, 75 p.

Charbeneau, R.J., and R.L. Street, 1979, Modeling groundwater flow fields containing point singularities: Streamlines, travel times and breakthrough curves: *Water Resources Research*, vol. 15, no. 6, 1445-1450 p.

Farmer, G.L., 1987, Moving point techniques, *in* Bear, J., and M.Y. Corapcioglu [eds.], Advances in Transport Phenomena in Porous Media: NATO ASI Series 128, Nijhoff, Boston, p. 955-1004.

Feinstein, D.T., 1986, A three-dimensional model of flow to the sandstone aquifer in northeastern Wisconsin with discussion of contamination potential: University of Wisconsin, Madison, Master's thesis, 240 p.

Freeze, R.A., and J.A. Cherry, 1979, *Groundwater:* Prentice-Hall, Englewood Cliffs, N.J., 604 p.

Javendel, I., C. Doughty, and C.F. Tsang, 1984, Groundwater Transport: Water Resources Monograph Series 10, American Geophysical Union, Washington, D.C, 228 p.

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

Mandle, R.J., and A.L. Kontis, 1986, Directions and rates of ground-water movement in the vicinity of Kesterson Reservoir, San Joaquin Valley, California: U.S. Geological Survey Water-Resources Investigations Report 86-4196, 57 p.

McDonald, J.M., and A.W. Harbaugh, 1984, A modular three-dimensional finite-difference flow model: U.S. Geological Survey Open-File Report 83-875, Reston, Virginia, 528 p.

Nelson, R.W., 1978, Evaluating the environmental consequences of groundwater contamination: *Water Resources Research*, vol. 14, no. 3, p. 409-450.

Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, 1986, *Numerical Recipes: The Art of Scientific Computing:* Cambridge University Press, Cambridge, 818 p.

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

Russell, T.F., and M.F. Wheeler, 1983, Finite element and finite difference methods for continuous flows in porous media, *in* The Mathematics of Reservoir Simulation: 1, Frontier Series, SIAM, Philadelphia, p. 35-106.

Shafer, J.M., 1987, Reverse pathline calculation of time-related capture zones in nonuniform flow: *Ground Water*, vol. 25, no. 3, p. 283-289.

Wilson J.L, 1984, Double-cell hydraulic containment of pollutant plumes: National Water Well Association, Proceedings of the Fourth National Symposium on Aquifer Restoration and Ground Water Monitoring, p. 65-70.

Zheng, C., H.F. Wang, M.P. Anderson, and K.R. Bradbury, 1988, Analysis of interceptor ditches for control of groundwater pollution: *Journal of Hydrology*, vol. 98, p. 67-81.

Cover: *Plan view of 20 travel paths from the water table to a pumping well in an anisotropic, heterogeneous, three-dimensional groundwater flow system. The flow paths, which outline the capture zone of the pumping well, were generated by PATH3D. The contour lines, which represent the hydraulic head distribution, were generated by the U.S. Geological Survey modular groundwater flow model (MODFLOW).*

ISSN: 0512-0640