PATH3D 3.2

a ground-water path and travel-time simulator

By C. Zheng

First edition: July 1989
First revision: June 1990
Second revision: July 1991
Third revision: October 1992

S. S. Papadopulos & Associates, Inc.

DISCLAIMER

S.S. Papadopulos & Associates Inc. (SSP&A) makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of fitness for any particular purpose. Further, SSP&A reserves the right to revise this publication and software, and to make changes from time to time in the content hereof without obligation of SSP&A to notify any person of such revisions or changes.

TABLE OF CONTENTS

Chapter 1
Introducing PATH3D

1.1 ABOUT PATH3D

PATH3D is a general particle tracking program for calculating groundwater paths and travel times in steady-state or transient, two- or three-dimensional flow fields. The program can be used to simulate the movement of groundwater flows and the advection of contaminant solutes in groundwater. The program is particularly useful for delineating contaminant capture zones or well-head protection zones in order to evaluate the effectiveness of groundwater pollution control under complicated hydrogeological conditions. PATH3D can serve as a valuable extension to groundwater flow models which in themselves do not provide quantitative information on flow paths and travel times, or as a practical alternative to contaminant transport models in cases where they are not feasible due to budget constraints or the lack of chemical data.

PATH3D was originally developed at the University of Wisconsin-Madison and the Wisconsin Geological and Natural History Survey, motivated by the need for a well-documented and carefully-verified three-dimensional particle tracking program for solving a wide range of field problems (see Zheng, 1988). Since then, the program has been substantially modified and improved. PATH3D is being distributed by S. S. Papadopulos & Associates, Inc., which will also provide necessary technical support and future upgrades to the registered users.

1.2 KEY FEATURES

PATH3D includes two major segments: 1) a velocity interpolator which converts the hydraulic heads calculated at discretized nodal points into a velocity field in which the velocity in a given time step is determined as function of position (x, y, z); and 2) a numerical solver for tracking the movement of fluid particles in groundwater flow systems. In the first segment, PATH3D uses a velocity scheme which is consistent with the block-centered finite-difference representation of the three-dimensional groundwater flow equation and conserves mass locally within each finite-difference cell. In the second segment, PATH3D uses a fourth-order Runge-Kutta solution capable of automatic stepsize adjustment to achieve pre-determined accuracy with minimum computational effort.

PATH3D uses the head solution of the U. S. Geological Survey modular three-dimensional finite-difference ground-water flow model, referred to as MODFLOW in this manual, (McDonald and Harbaugh, 1988). This allows flexibility in handling a wide range of field problems, including the presence of wells, drains, rivers, recharge, and evapotranspiration, and permits representation of a variety of boundary conditions. The input files and the resulting head file of MODFLOW are directly used by PATH3D. However, PATH3D can be readily modified to work in conjunction with any other block-centered finite-difference flow model, if desired.

PATH3D generates several output files, containing: 1) the x, y, z coordinates and the components of the seepage velocity at different travel times along the path of each individual particle; 2) the x, y, z coordinates of all particles at selected times; and 3) the initial and final positions of those particles captured by various sinks, or sources in the case of reverse tracking. These output files

can be easily imported into most commercial graphic packages for plotting the pathlines of individual particles, the positions of displacement fronts of captures zones at desirable times, and the distribution of captured particles. Several post-processing programs are included with PATH3D to facilitate the conjunctive use of PATH3D with a graphic package such as Golden Software's SURFER.

1.3 SYSTEM REQUIREMENTS

This software package comes with two floppy disks: *Program Disk* and *Example Disk*. The program disk contains the source codes and run-time versions of PATH3D and related programs. The example disk includes the input and output files of the application examples discussed in Chapter 4 of this manual.

If the software is intended for use in IBM-compatible personal computers, the executable files included in the program disk are ready to run. The regular version was compiled with the Microsoft FORTRAN 77 Compiler Version 5.0, and will run on any PC with or without a math co-processor. The 32-bit extended-memory version, denoted by a suffix of *EM32*, was compiled with the Lahey EM-32 FORTRAN 77 Compiler Version 4.0 and is intended to run on PCs equipped with 80386 or 80486 CPU. Both versions were compiled with the dynamic memory allocation feature and will use all the memory that is available.

The software can also be used in non-IBM-compatible personal computers or mainframes. To do this, you need to transfer the DOS-formatted files in the two disks into formats used in other types of PCs or mainframes and re-compile the source codes with a FORTRAN 77 compiler. Since PATH3D was written exclusively in the standard FORTRAN 77 language, little or no modification is required to compile the source codes with most FORTRAN 77 compilers.

1.4 QUICK START

If you are already familiar with the general concepts and procedures of particle tracking, and have a set of MODFLOW data files available, you may skip Chapter 2 *"Understanding PATH3D"* and refer directly to Sections 3.5.1 and 3.5.3 in Chapter 3 *"Using PATH3D"*.

The first thing you need to do is create a new input file which contains the particle tracking control parameters and the initial positions of particles to be tracked as explained in Section 3.5.1. Next, you need to add three new data arrays at the end of the BCF input file for the MODFLOW Block-Centered-Flow package (See Section 3.5.3). Then, you are ready to run PATH3D and obtain the desired outputs.

Chapter 2
Understanding PATH3D

2.1 INTRODUCTION

Groundwater flow paths, also the pathways of contaminant solutes in groundwater if the effect of dispersion is neglected, can be determined using graphic, stream functions or particle tracking methods. Among the three methods, particle tracking is most powerful and can be used for both steady-state and transient flows in two or three dimensions. The principle of particle tracking is to place infinitely small imaginary fluid particles in the flow domain and keep track of their positions as they move through the flow field in a series of small time steps, driven by the hydraulic gradients.

PATH3D is a general computer code designed to perform particle tracking calculations accurately and efficiently. This chapter gives a brief overview of algorithms and methodologies used in PATH3D in order to provide a better understanding of the unique features and capacities of PATH3D.

2.2 VELOCITY SCHEME

PATH3D uses the head solution of the U.S. Geological Survey modular three-dimensional finite-difference flow model, referred to as MODFLOW in this manual, (McDonald and Harbaugh, 1988). A velocity scheme consistent with

MODFLOW, or any other block-centered finite-difference flow model, is implemented in PATH3D. This velocity scheme is explained below.

Fig. 2.1 depicts the model grid system used in MODFLOW and also PATH3D. In MODFLOW the head at the center of cell (i, j, k) is calculated based on the heads in the six adjacent cells in the three directions, where I, J, and K indicate the numbers of rows, columns and layers, respectively.

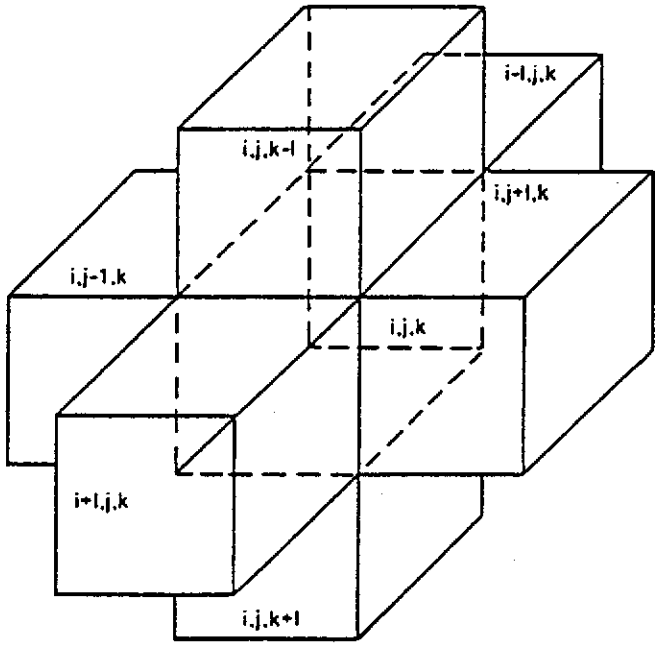


Fig.2.1. Grid system used in MODFLOW and PATH3D

Flux into cell (i, j, k) along the row direction, or 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. 2.2) as:

$$q_{i,j-1/2,k} = -K_{i,j-1/2,k}\frac{\left(h_{i,j,k} - h_{i,j-1,k}\right)}{\Delta x_{j-1/2}} \qquad (2.1)$$

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);

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

$\Delta x_{j-1/2}$ is the distance between nodal points (i, j-1, k) and (i, j, k), and

Δx_j is the cell width along the row direction at column j.

Similarly, at the interface between cells (i, j, k) and (i, j+1, k), the specific discharge $q_{i,j+1/2,k}$, is:

$$q_{i,j+1/2,k} = -K_{i,j+1/2,k}\frac{\left(h_{i,j+1,k} - h_{i,j,k}\right)}{\Delta x_{j+1/2}} \qquad (2.2)$$

The x component of the specific discharge at an arbitrary point (x, y, z) within cell (i, j, k) can then be computed from the following equation:

$$q_x(x,y,z) = (1-\alpha)\, q_{i,j-1/2,k} + \alpha\, q_{i,j+1/2,k} \qquad (2.3)$$

where

$\alpha = \frac{\left(x - x_j\right)}{\Delta x_j} + \frac{1}{2}$, is the interpolation factor;

x is the x coordinate at arbitrary point (x,y,z);

x_j is the x coordinate at noal point (i,j,k).

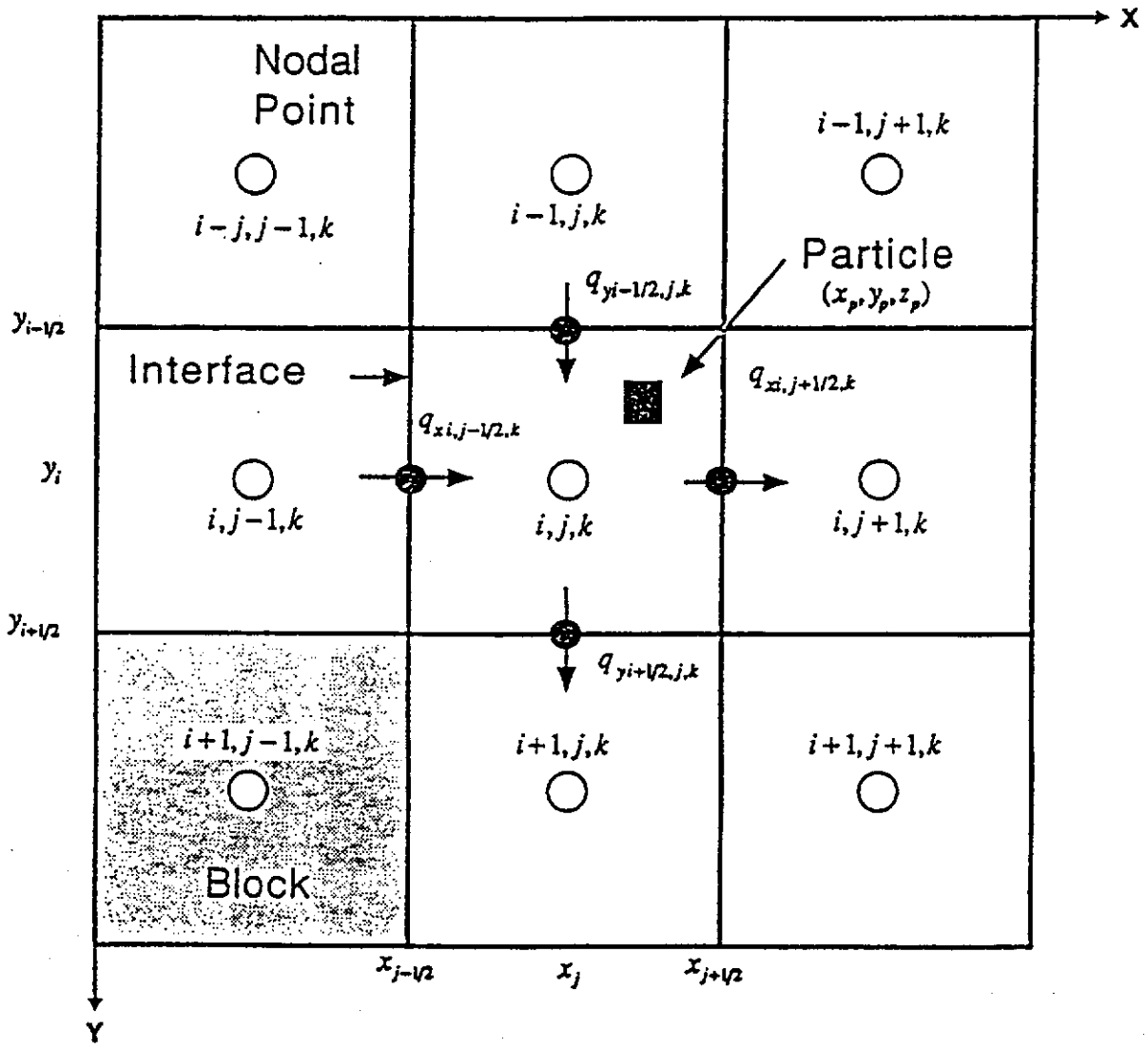


Fig.2.2. Velocity interpolation scheme based on the block-centered finite-difference formulation

The x component of the seepage velocity, or linear velocity, v_x, is computed from:

$$v_x(x,y,z) = \frac{q_x(x,y,z)}{\theta_{i,j,k}} \tag{2.4}$$

where $\theta_{i,j,k}$ is the porosity value for cell (i,j,k). Since α is a number which varies linearly from 0 to 1 between interfaces (i, j-1/2, k) and (i, j+1/2, k), v_x is equal to $v_{x\,i,j-1/2,k}$ at the left interface and $v_{x\,i,j+1/2,k}$ at the right interface, and follows a linear interpolation between $v_{x\,i,j-1/2,k}$ and $v_{x\,i,j+1/2,k}$ between the two interfaces. The effect of sorption can be incorporated into the particle tracking calculation by multiplying the porosity valuve with a retardation factor, R, i.e.,

$$v_x(x,y,z) = \frac{q_x(x,y,z)}{R\theta_{i,j,k}} \tag{2.5}$$

Specified-flux boundary conditions can be handled readily with this velocity scheme. Assume that the left interface of cell (i,j,k) is along a specified-flux boundary, the specific discharge at the left interface is equal to the value of the specified flux, q_o. As a result, equation (2.3) becomes:

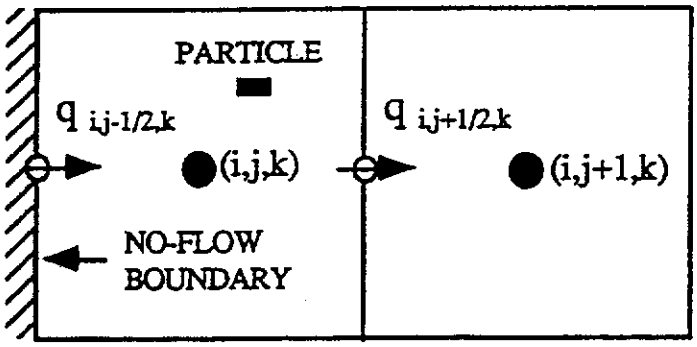
$$q_x(x,y,z) = (1-\alpha)\, q_o + \alpha\, q_{i,j+1/2,k} \tag{2.6}$$

The no-flow boundary is a special case of the specified-flux boundaries where q_o is equal to zero (see Fig. 2.3a):

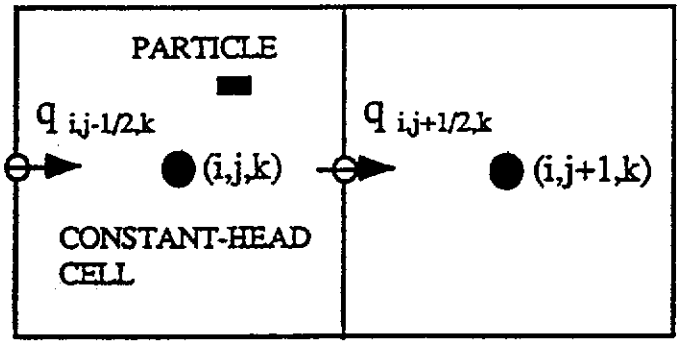
$$q_x(x,y,z) = \alpha\, q_{i,j+1/2,k} \tag{2.7}$$

The treatment of constant-head boundary conditions is not as straightforward. If a constant-head cell is inside the active flow domain, the specific discharges at all cell interfaces are known from the flow model, and

therefore, no special treatment is necessary. However, if the constant-head cell is on the edge of the active flow model, the specific discharge at the edge interface cannot be calculated. In PATH3D, this specific discharge is set equal to that at the other interface in the same direction where the specific discharge is known from the flow model (see Fig. 2.3b). The general-head-dependent boundary conditions are handled in the same fashion.



(a) at no-flow boundary where $q_{i,j-1/2,k} = 0$



(b) at constant-head boundary where $q_{i,j-1/2,k} = q_{i,j+1/2,k}$

Fig.2.3. Treatment of boundary conditions in the velocity scheme

The y and z components of the seepage velocity are calculated in an analogous way. In cases where a particle is located in a water table cell, the interpolation of the z component of the seepage velocity is handled in the same way as the specified-flux boundary condition. The specific discharge at the particle location is interpolated from the net recharge rate at the water table and the specific discharge at the lower cell interface (see Fig. 2.4):

$$q_z(x,y,z) = (1-\alpha)\, q_R + \alpha\, q_{i,j,k+1/2} \tag{2.8}$$

where q_R is the net recharge rate (including the effects of recharge and evapotranspiration) and $\alpha = L/D$ where L is distance between the particle and the water table and D is the saturated thickness of cell (i,j,k).

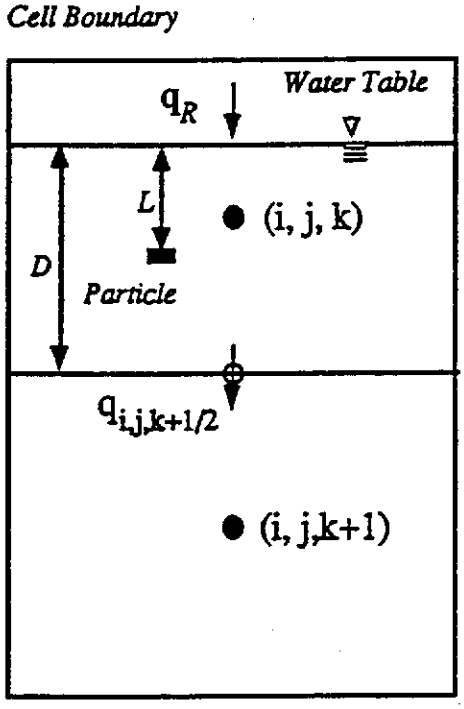


Fig.2.4. Calculation of the vertical component of the seepage velocity for a particle located within the water table cell

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 the head distribution changes, so does the velocity field. The simulation time in MODFLOW is divided into stress periods, or, time intervals during which all external stresses (say, the well pumping rates) are constant (see McDonald and Harbaugh, 1988). Each stress period is, in turn, divided into one or more time steps as illustrated in Fig. 2.5. 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.. Thus, whereas the MODFLOW solution provides head values corresponding to the end of each time step, these values are assumed to prevail throughout the duration of the time step in the PATH3D calculations.

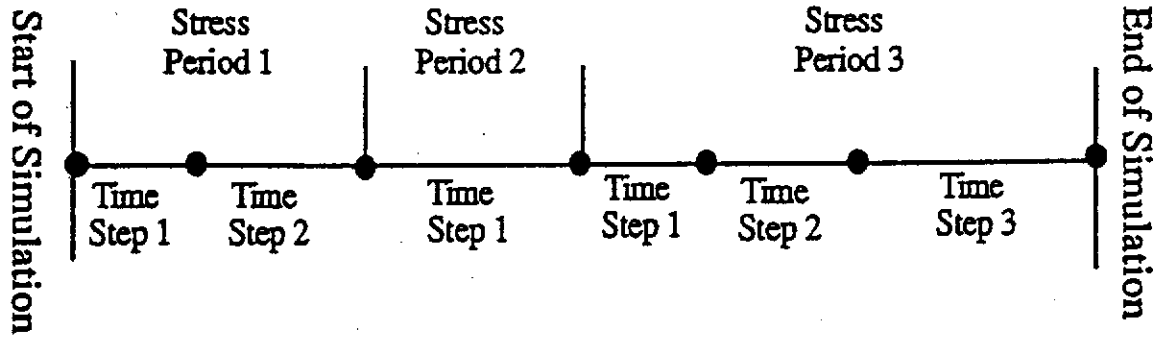


Fig.2.5. Discretization of simulation time in MODFLOW and PATH3D

2.3 TRACKING ALGORITHM

PATH3D uses a fourth-order Runge-Kutta method for its particle tracking solution. The basic idea of a fourth-order Runge-Kutta method is to evaluate the velocity four times for each tracking step: once at the initial point, twice at two trial midpoints, and once at a trial end point (Fig. 2.6). From velocity values evaluated at these four points, the position of the particle at the beginning of the next tracking step (x_{n+1}, y_{n+1}, z_{n+1}) is determined. This process may be expressed as follows:

$$\begin{cases} x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ y_{n+1} = y_n + \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4) \\ z_{n+1} = z_n + \frac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4) \end{cases} \tag{2.9}$$

where

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

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

$$m_1 = \Delta t\, v_x(x_n, y_n, z_n, t_n)$$

$$m_2 = \Delta t\, v_x(x_n + k_1/2, y_n + l_1/2, z_n + m_1/2, t_n + \Delta t/2)$$

$$m_3 = \Delta t\, v_x(x_n + k_2/2, y_n + l_2/2, z_n + m_2/2, t_n + \Delta t/2)$$

$$m_4 = \Delta t\, v_x(x_n + k_3, y_n + l_3, z_n + m_3, t_n + \Delta t)$$

where v_x, v_y, and v_z are the components of the seepage velocity along the x, y, z directions, respectively; and Δt is the stepsize of a tracking step. If Δt is positive, the tracking is forward along the flow direction. Otherwise, the tracking is backward along the reverse flow direction. Note the difference between the tracking step of a particle and the time step of the head solution. A particle generally takes a series of tracking steps to move from the beginning to the end of each time step of the head solution.

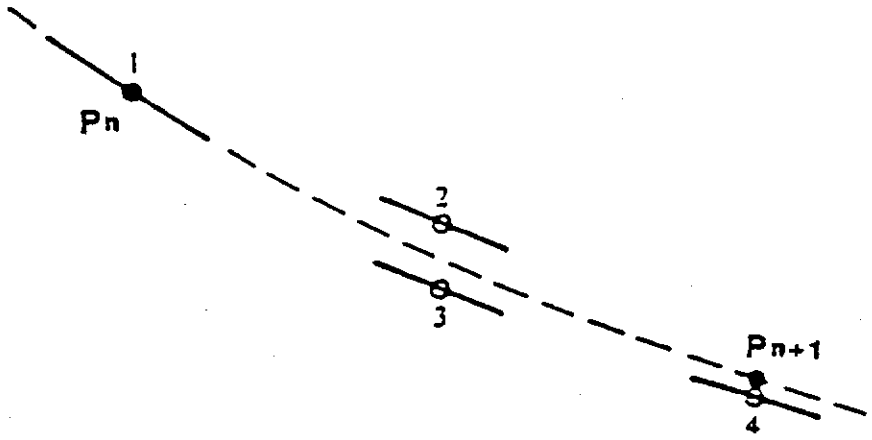


Fig.2.6. The fourth-order Runge-Kutta method. In each step, the velocity is evaluated four times: once at the initial point, twice at trial midpoints, and once at a trial endpoint. From these velocities the final position (shown as a filled dot) is calculated.

It is obviously important to select an appropriate tracking stepsize, Δt. If the stepsize is taken too large, then the particle may deviate from the actual travel path and the results are inaccurate. On the other hand, if the stepsize is taken too small, then a large number of calculations may be needed to determine the particle movement over a given distance, and the solution is inefficient.

To avoid this problem, an automatic stepsize adjustment procedure was implemented in PATH3D to monitor the errors of the fourth-order Runge-Kutta solution and make frequent changes in stepsize in order to achieve a predetermined accuracy with minimum computational effort. A tracking step Δt is always taken twice, once as a full step, and once as two half steps (see Fig. 2.7). If Δt is small enough, then the resulting difference in the particle locations, denoted by ΔS, would be small. Since the basic solution is accurate to the fourth-order, ΔS can be scaled as $(\Delta t)^5$:

$$\left[\frac{\Delta t_o}{\Delta t}\right]^5 = \frac{\Delta S_o}{\Delta S} \tag{2.10}$$

where ΔS_o and ΔS are the differences in particle locations, determined by the forgoing procedure, associated with the stepsizes Δt_o and Δt.

Equation (2.10) can be used to estimate the stepsize Δt_o which will give an acceptable position discrepancy, ΔS_o, given an initial calculation of ΔS using an initial stepsize Δt . In doing this, a "safety factor", f_s, having a value slightly smaller than unity (e.g., 0.9), can be incorporated, i.e.,

$$\Delta t_o = f_s \Delta t \left(\frac{\Delta S_o}{\Delta S}\right)^{0.2} \tag{2.11}$$

Equation (2.11) is utilized in PATH3D to determine the required adjustment in stepsize in each tracking step, after an initial calculation of ΔS has

been made with the stepsize Δt. If ΔS is larger than the required accuracy, ΔS_o, the tracking stepsize is reduced to Δt_o and the calculation is repeated for that tracking step. If ΔS is less than ΔS_o, the calculation based on the stepsize Δt is acceptable, and the initial stepsize for the following tracking step is taken as Δt_o.

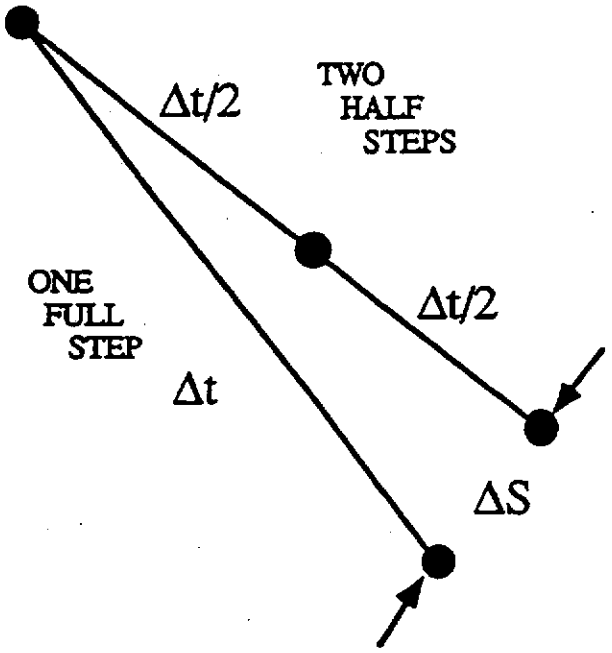


Fig.2.7. Procedure for adaptive stepsize control

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

$$\begin{cases} \Delta X_o = \varepsilon \times XMAX \\ \Delta Y_o = \varepsilon \times YMAX \\ \Delta Z_o = \varepsilon \times ZMAX \end{cases} \tag{2.12}$$

where ε is a dimensionless factor; and 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 enters an error criterion ε, and the program calculates the maximum allowed errors in the x, y and z directions according to equation (2.12).

2.4 SPECIAL CONSIDERATIONS

2.4.1. Boundary and Water Table

When a particle is about to exit from one of the model edges, approach an inactive cell, or leave the water table, PATH3D determines the time required for the particle to travel from its current position to the exit point, and move the particle to the location of the exit point before removing it from the flow domain.

If a particle is initially placed above the water table or moves above the water table during the tracking process, PATH3D will check whether there is upward gradient at the water table. If there is, that is, if the net recharge rate is negative, the particle will be removed from the flow domain. Otherwise, the particle will be placed back on the water table. Therefore, if the net recharge rate

is zero, a particle placed on the water table will move along the step-shaped water table, as illustrated in Fig. 2.8.

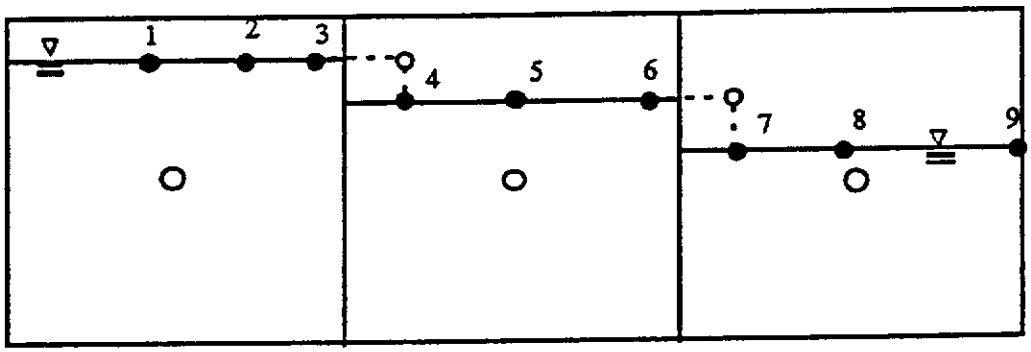


Fig.2.8. Diagram showing the movement of a particle along the water table where the net recharge rate is zero. Numbers indicate the steps of the tracking process. When the particle moves above the water table due to the discrete nature of the head solution, the particle is placed back on the water table, as illustrated at steps 4 and 7.

2.4.2 Sink and Stagnation Zone

When a particle enters a sink such as a well, a drain or a river reach, the velocity components at the six interfaces of the model cell containing the sink are generally inward toward the nodal point. If this is the case, then the particle cannot leave the cell again. However, if the size of the sink cell is not sufficiently small or the flow rate of the sink is not large enough compared to the rate of the regional flow, then, it is possible that there are outward velocity components at one or more of the cell interfaces. When this happens, a particle entering the sink cell may continue to travel and leave the cell again.

PATH3D provides several options for the removal of particles at sink cells through the use of flag IOPSS in the particle tracking input file. The first option is to remove only those particles which have entered a sink with inward gradients at all interfaces. The second option is to remove particles as soon as they enter a sink cell even if the sink cell does not have inward gradients at all of its interfaces. A third alternative option (available for wells only) is to determine the location of the dividing streamline within the sink cell based on the analytical solution (see Bear, 1979):

$$y'/x' = \pm\tan(2\pi y' U / Q_w)\begin{cases} + \text{ for } y' > 0 \\ - \text{ for } y' < 0 \end{cases} \tag{2.13}$$

where x' and y' represent the local coordinate system whose origin is at the well location and whose x-axis is along the direction of the regional flow (see Fig. 2.9). Q_w is the volumetric flow rate of the sink or source. U is the volumetric regional flow rate per unit width of the aquifer. The value of U at a sink cell is estimated in PATH3D from the following equation:

$$U_{i,j,k} = \sqrt{\left(U_{xi,j,k}\right)^2 + \left(U_{yi,j,k}\right)^2} \tag{2.14}$$

where

$$U_{xi,j,k} = H_{i,j,k}\frac{q_{xi,j-1/2,k} + q_{xi,j+1/2,k}}{2}$$

$$U_{yi,j,k} = H_{i,j,k}\frac{q_{yi-1/2,j,k} + q_{yi+1/2,j,k}}{2}$$

and $H_{i,j,k}$ is the saturated thickness of cell (i,j,k). The angle of the regional flow (β) is approximated according to:

$$\beta = \arctan\left(\frac{U_{yi,j,k}}{U_{xi,j,k}}\right) \tag{2.15}$$

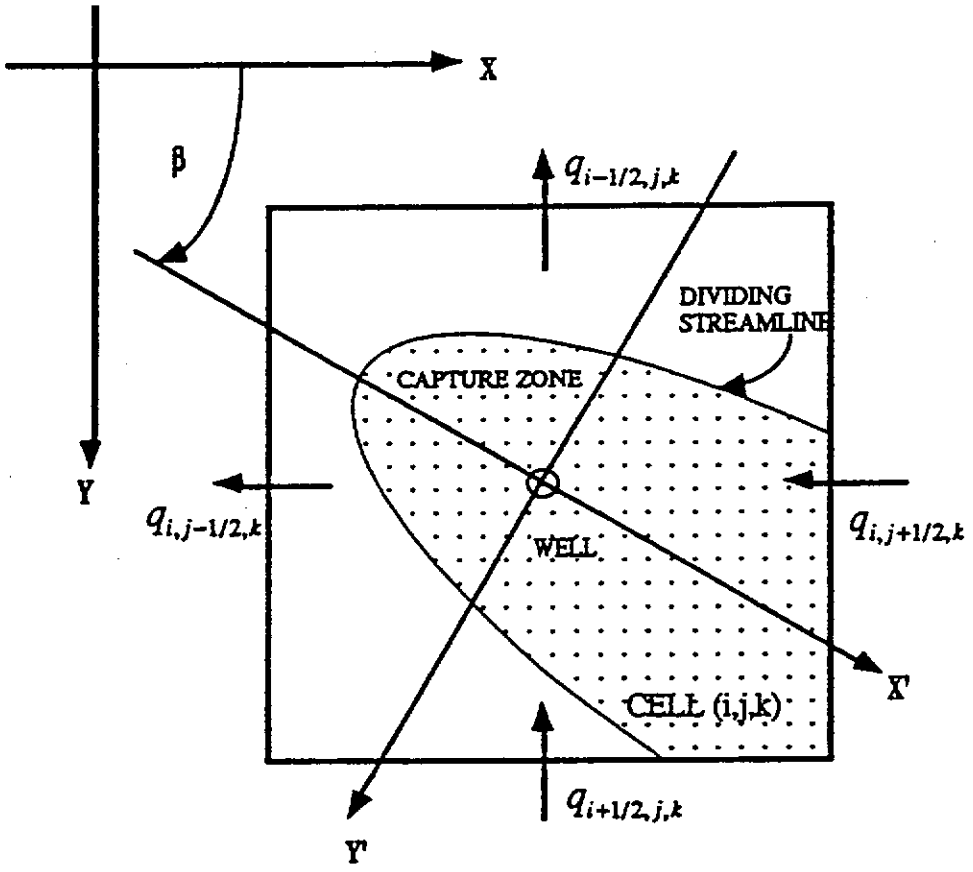


Fig.2.9. Determination of the dividing streamline inside a weak sink cell

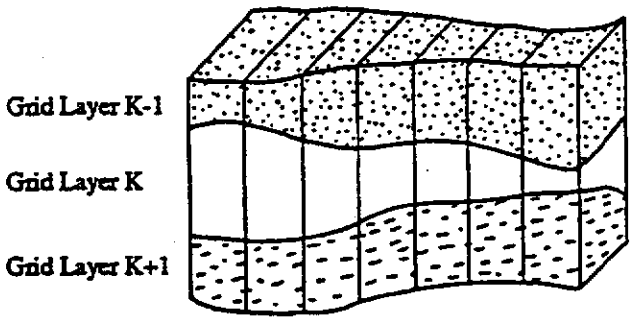
With the dividing streamline determined from equation (2.14), the particles inside the dividing streamline are removed whereas those particles outside the dividing streamline are allowed to travel out of the sink cell if the flag IOPSS is entered in the input file properly.

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 from the flow field if it reaches a point where all the velocity components of the particle are less than a very small number (say, 10^{-30}). This number may be changed to accommodate specific circumstances by modifying the statement `TINY = 1.E-30` in the main program and recompiling the source code.

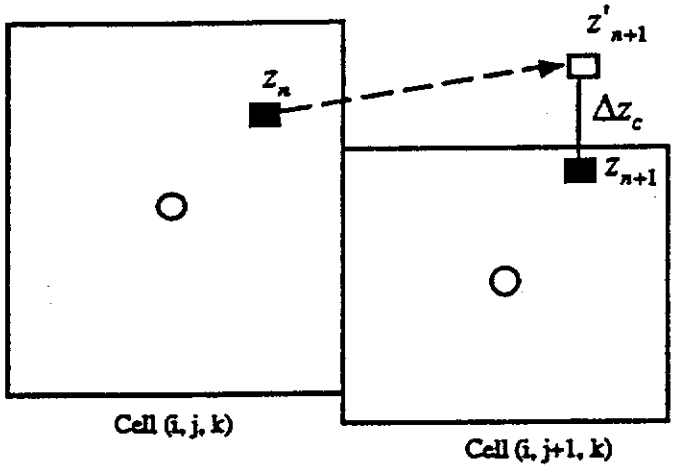
2.4.3 Distorted Vertical Discretization

Distorted vertical discretization is frequently used in flow simulation in order to accommodate natural stratigraphic irregularities (e.g., Fig. 2.10a) using a reasonable number of layers. The use of distorted model grids can lead to certain numerical discretization errors, as it departs from the assumptions implied in the standard finite difference theory. These numerical errors become more profound in particle tracking. For this reason, large distortion of model grids should be avoided, if possible at all.

In the presence of distorted model grids, a correction procedure as illustrated in Fig. 2.10b is used to adjust the vertical coordinates of the moving particle according to the degree of distortion in grid. As a particle moves from a cell (i, j, k) to its neighboring cell (i, j+1, k), the new z-coordinate of the particle based on the normal tracking procedure without considering the grid distortion would be:



(a) Aquifer cross section with distorted grid superimposed (Modified from McDonald and Harbaugh, 1988).



(b) Correction procedure for particle tracking in distorted grid.

Fig. 2.10. Illustration of a distorted model grid and the correction procedure for adjusting the particle's vertical coordinates to account for the effect of grid distortion.

$$z'_{n+1} = z_n + \bar{v}_{z,n}\Delta t \tag{2.16}$$

where z'_{n+1} is the z-coordinate at the new tracking step without applying any correction procedure, as indicated by the open square in Fig. 2.10b; $\bar{v}_{z,n}$ is the weighted vertical velocity component as determined from the fourth-order Runge-Kutta solution. To take the grid distortion into account, equation (2.16) can be rewritten as,

$$z_{n+1} = z'_{n+1} + \Delta z_c = z_n + \bar{v}_{z,n}\Delta t + \Delta z_c \tag{2.17}$$

where z_{n+1} is the final z-coordinate of the particle at the new tracking step after adjusted by a correction factor, Δz_c The determination of the correction factor, Δz_c, is illustrated in Fig. 2.11. First, assume that a particle moves from cell (i, j, k) to the adjacent cell (i, j+1, k) in one step in a regular, non-distorted grid as shown in Fig. 2.11a. If the vertical velocity component is zero, then, the particle's z-coordinate at the new tracking step, z_{n+1}, is equal to that at the old tracking step, z_n, (see Fig. 2.11a). When the grid is distorted to a certain degree as shown in Fig. 2.11b, z_{n+1} should also be adjusted accordingly since no particle should cross faces AB and CD by virtue of finite-difference formulation of the flow equation. The adjusted z_{n+1} can be determined by setting the percentage of change in the particle's z-coordinates relative to the cell's nodal points to be equal to the percentage of change in the cell's thicknesses. In other words, the following relationship can be established:

$$\frac{z_n - z_{i,j,k}}{z_{n+1} - z_{i,j+1,k}} = \frac{\Delta z_{i,j,k}}{\Delta z_{i,j+1,k}} \tag{2.18}$$

where $z_{i,j,k}$ and $z_{i,j+1,k}$ are the z-coordinates of the block-centered nodal points of cells (i, j, k) and (i, j+1, k), respectively; $\Delta z_{i,j,k}$ and $\Delta z_{i,j+1,k}$ are the thicknesses

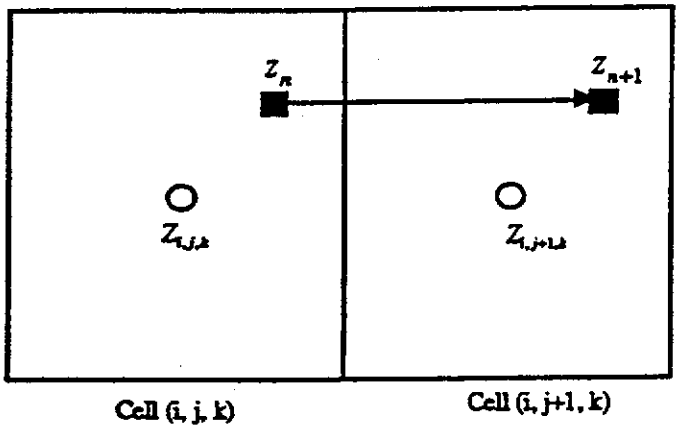
of cells (i, j, k) and (i, j+1, k), respectively. The correction factor, Δz_c, is then taken as the difference between z_{n+1} and z_n (also see Fig. 2.11b):

$$\Delta z_c = z_{n+1} - z_n = \frac{\Delta z_{i,j+1,k}}{\Delta z_{i,j,k}}\left(z_n - z_{i,j,k}\right) + z_{i,j+1,k} \tag{2.19}$$

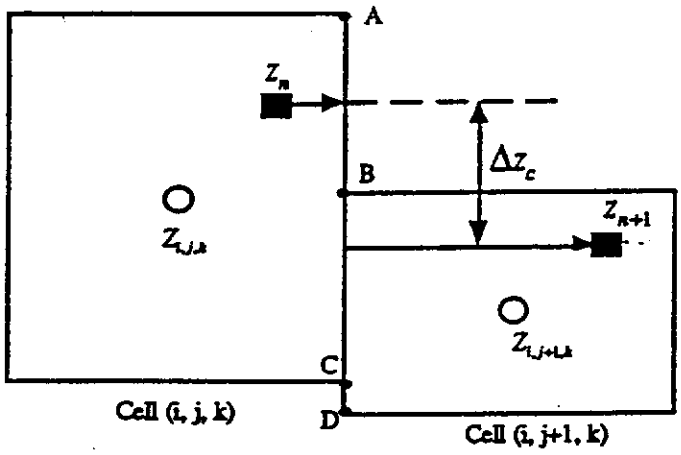
By substituting equation (2.19) into equation (2.17), the equation for computing the z-coordinate of moving particles in a distorted grid is obtained:

$$z_{n+1} = \bar{v}_{z,n}\Delta t + \frac{\Delta z_{i,j+1,k}}{\Delta z_{i,j,k}}\left(z_n - z_{i,j,k}\right) + z_{i,j+1,k} \tag{2.20}$$

Equation (2.20) reflects the movement of particles caused by the vertical velocity component and at the same time accounts for the effect of the grid distortion.



(a) Particle movement in one step assuming no vertical velocity in a regular grid.



(b) Particle movement in one step assuming no vertical velocity in a distorted grid.

Fig. 2.11. Schematic diagram illustrating the determination of the correction factor

2.5 PROGRAM DESIGN

The source code of PATH3D was written in the standard ANSI FORTRAN 77 language. The code consists of a main program and a number of subroutines, which are loosely grouped into several "packages", dealing with velocity interpolation, particle tracking calculation, and data input/output, respectively. The functions of the main program and each of the packages are explained in this section. For more information on the PATH3D program design, refer to the extensively commented source code included in the Program Disk.

2.5.1 Main Program P3DMAIN3

The main program P3DMAIN3, where 3 denotes the current version number, controls the overall execution of the entire program. It does so by calling subroutines to execute specific tasks in the following order. A simplified flow chart for the main program is provided in Fig. 2.12.

1. Set the length of the Y array, a one-dimensional array in which all data arrays are stored. If the Y array is not dimensioned large enough, the user may redimension the Y array by modifying the following parameter statement :

   ```
   PARAMETER (LENY=80000)
   ```

   in the main program and recompiling the source code.
2. Open input and output files, and assign them to certain units.
3. Allocate space in the Y array for individual data arrays.
4. Read and prepare information which are constant throughout the entire simulation.

5. For each stress period:

(1) Read and prepare stress-period timing information if the simulation is transient.

(2) 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-dependent boundaries.

(3) For each time step:

   (a) Read the head solution saved in an unformatted or formatted file by the flow model.

   (b) Calculate the length of the current time step and determine whether it is covered by the tracking time interval. If yes, proceed with particle tracking. Otherwise, go to the next time step. (Note that if the head solution is steady state, the length of the current time step is set equal to the length of the tracking time interval).

   (c) Compute the components of specific discharges at all cell interfaces in the row, column and layer directions.

   (d) 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 stepsize control.

6. End program.

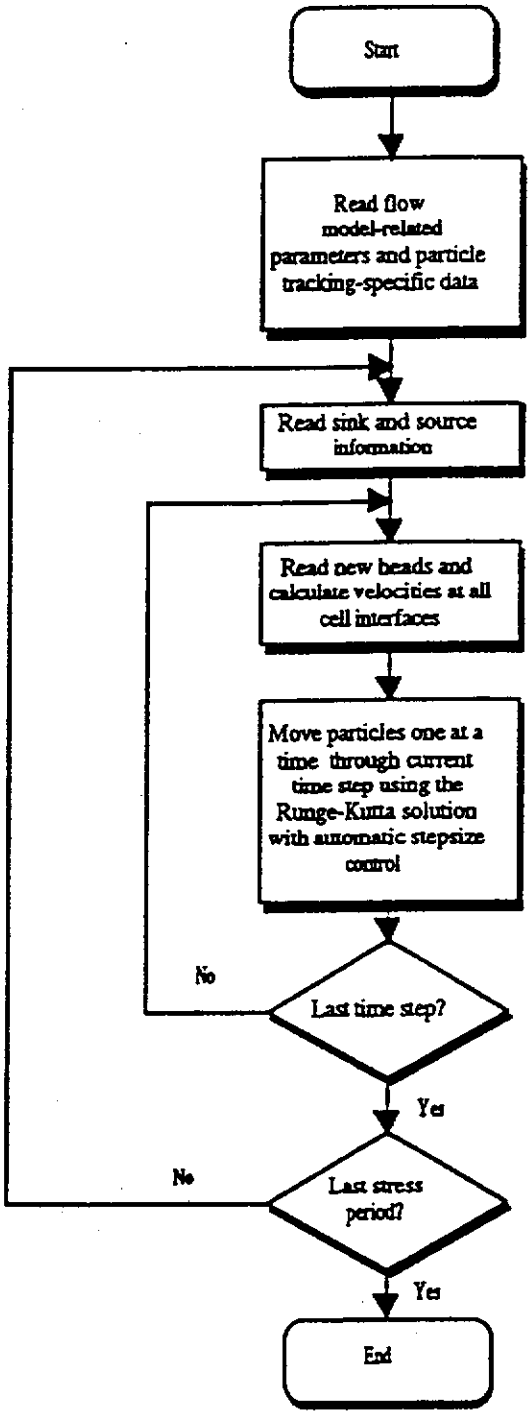


Fig.2.12. Simplified flowchart for the main program

2.5.2 *Package P3DVELO3*

This package includes two subroutines: VFACE and VPOINT . 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 store them in three three-dimensional arrays QX, QY and QZ, respectively. If the recharge and/or evapotranspiration options are used in the flow simulation, 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 effect of the constant-head boundaries if they are present in the head solution. QX, QY and QZ are then passed to subroutine VPOINT whenever needed during the tracking process to compute the specific discharge at a given point based on the interpolation scheme described previously. The specific discharge is then divided by the effective porosity, and retardation factor if sorption is to be considered, to obtain the seepage velocity at the same point.

2.5.4 *Package P3DTRCK3*

This package contains the core elements 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 stepsize control procedure. The package includes these subroutines: MOVE, RKASC, RK4, LOCATE, and WELLCP.

Given a time period from TIME1 to TIME2, subroutine MOVE calls subroutine RKASC and advances a particle from its starting position at TIME1 to the new position at TIME2 through a series of tracking steps. 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 and the length of the tracking time interval. MOVE will be called every time step to advance particles throughout that time step, until the end of the specified tracking time interval or until the end of the head simulation time.

Subroutine RK4 is a direct implementation of equation (2.9). It advances a particle over a time increment using the fourth-order Runge-Kutta solution. It calls subroutine VPOINT 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 stepsize for each tracking step to achieve the error criterion specified by the user.

Subroutine LOCATE is used to determine the location of an arbitrary point (x, y, z) in a finite-difference grid defined by the rows, column and layer numbers (or the J, I and K indices). LOCATE also checks whether the particle location is outside the model edges, above the water table, or at an inactive cell.

Function WELLCP is used to determine the position of the dividing streamline inside a weak sink or source cell based on the analytical solution.

2.5.2 Package P3DDATA3

This package performs such basic tasks as allocation of memory space for data arrays and input of flow model and particle tracking parameters. It includes the following subroutines: GETDIM, GETDAT, GETSS, READH, GENPTC, and GENPTR.

Subroutine 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. Subroutine GETDAT reads model discretization and hydraulic parameters from input files created for MODFLOW, as well as particle tracking parameters from an input file which is set up exclusively for PATH3D. Subroutine GETSS reads sink/source

information at the beginning of every stress period from input files created for MODFLOW. READH is a subroutine used to input hydraulic heads solved by MODFLOW and saved in an unformatted or formatted file. These subroutines need to be modified if PATH3D is to be used in conjunction with a flow model other than MODFLOW.

Subroutines GENPTC and GENPTR are called by subroutine GETDAT to generate initial particles within any finite difference cell if this option of specifying initial particle locations is chosen by the user. GENPTC places particles around the radius of a sink or source, while GENPTR distributes particles randomly inside the cell.

2.5.4 Package P3DUTIL3

This package contains several utility modules used to open input and output files (OPENFL), read and write one- or two-dimensional integer or real data arrays (IARRAY, RARRAY, IPRINT and RPRINT). IARRAY and RARRAY are compatible to U2DINT and U1DREL/U2DREL used in MODFLOW and can be used interchangeably. But IARRAY and RARRAY have more input options available. For more information, refer to Appendix B.

Chapter 3
Using PATH3D

3.1 INTERFACE WITH FLOW MODEL

The current version of PATH3D is designed to interface with the U.S. Geological Survey modular three-dimensional finite-difference ground-water flow model (MODFLOW) (McDonald and Harbaugh, 1988) for the head solution. It is assumed that the users of PATH3D are familiar with MODFLOW and can use MODFLOW to construct a flow model.

Prior to running PATH3D, MODFLOW should be used first to simulate the head distribution. PATH3D then reads the head file saved by MODFLOW and other related hydraulic and discretization parameters from input files created for MODFLOW in order to perform particle tracking and calculate flow paths and travel times. It is important to point out that the appropriate use of PATH3D requires accurate information on the distribution of hydraulic heads. Only if the flow field is correctly simulated can PATH3D give meaningful results.

3.2 DEFINITION OF PARTICLE LOCATIONS

In MODFLOW as well as PATH3D, an aquifer is discretized into a mesh of blocks, or cells, the locations of which are described in terms of rows (I), columns (J) and layers (K) (see Fig. 3.1).

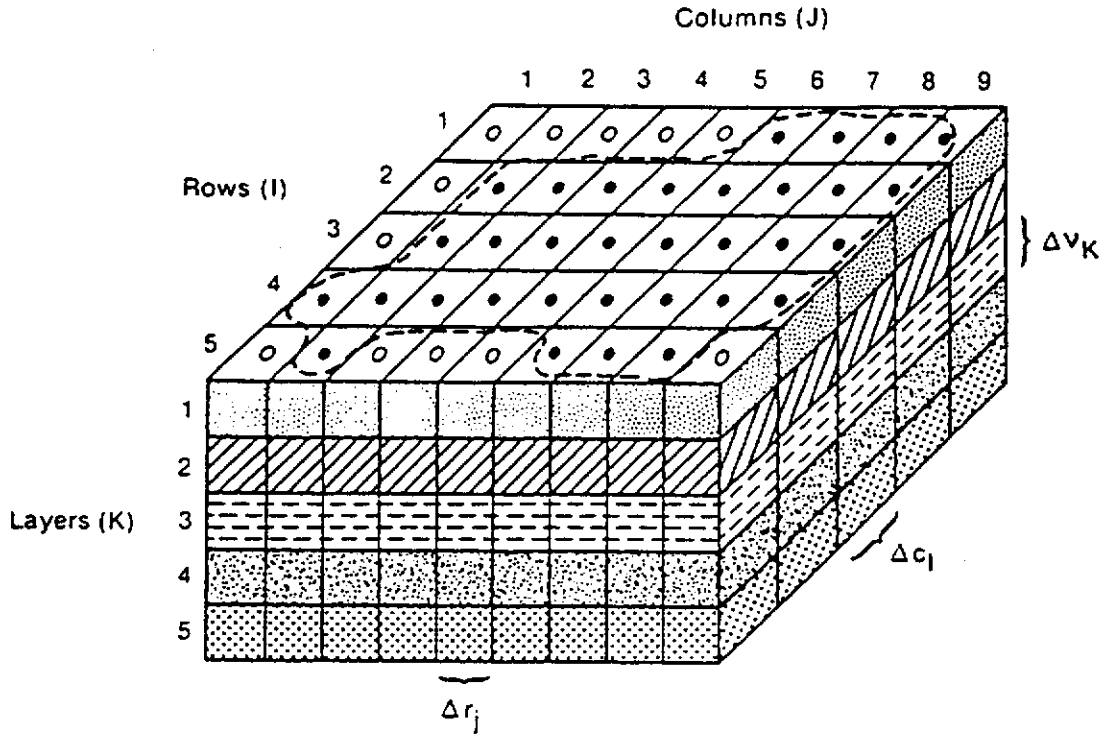


Fig. 3.1. Space discretization convention used in MODFLOW and PATH3D

To define the position of a particle in the finite-difference grid, the Cartesian coordinate system, instead of the I, J, K indexing system, is used. The coordinate origin is set at the upper top left corner of the cell at the first column, row, and layer, or cell (1, 1, 1), and the x, y, and z coordinate axes are oriented along the directions in which the J, I, and K indices increase (see Fig. 3.2). It is important to note that the origin (x=0, y=0, z=0) is set at the upper top left corner of cell (1, 1, 1), not at the block center of cell (1, 1, 1).

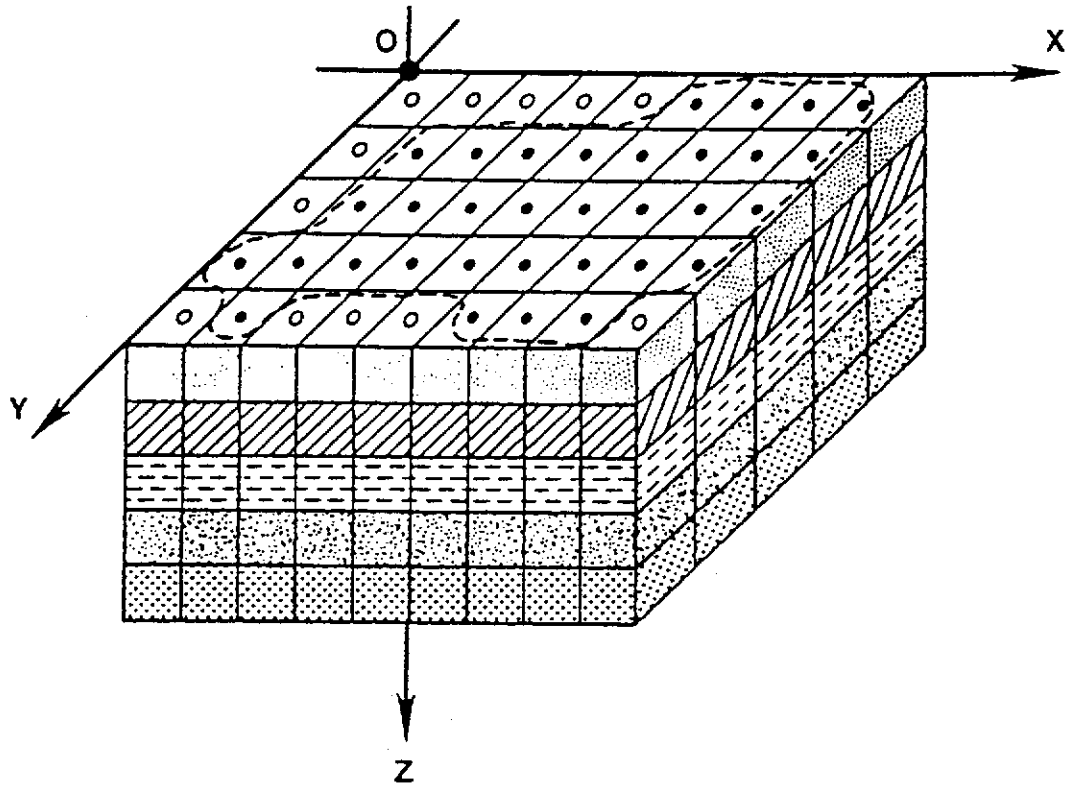


Fig. 3.2. Cartesian coordinate system for defining particle locations

3.3 UNITS OF MODEL VARIABLES

The unit of time must be consistent for all variables that involve time. For example, if years is the chosen time unit, hydraulic conductivity, travel time, etc., must all be expressed using years for their time unit. Likewise, the unit of length must also be consistent for all variables involving length.

3.4 SPACE REQUIREMENT

All data arrays used in the program are stored in the one-dimensional Y array. The length of the Y array is set prior to program compilation. If the Y array is not dimensioned larger enough, then modify the first parameter statement

```
PARAMETER (LENY=80000)
```

in the main program and recompile the source code. The total space needed for a specific problem is approximately 10 to 15 times the number of nodes in the flow model, plus six times the total number of particles used for tracking. The exact size of the Y array can be known by running the program, which will always print out the required Y array dimension even if the Y array is dimensioned too small.

3.5 INPUT INSTRUCTIONS

3.5.1 Particle tracking input file

This file contains the information needed for the particle tracking solution. Input data should be entered according to the following formats:

Record 1:	NCOUNT	NTRKOP	NFRONT	INIPOS	IOPSS	INCAPT	
Format:	6I10						
Record 2:	TIME1	TIME2	EPS	DT	MAXSTP	ISAV	IPRT
Format:	F10.0	F10.0	F10.0	F10.0	I10	I10	I10

Record 3: TFRONT(1), TFRONT(2), ..., TFRONT(NFRONT)
Format: 8F10.0
(Record 3 should be omitted if NFRONT=0; This record can occupy more than one line if the value of NFRONT is greater than 8.)

If INIPOS=1, enter record 4 in the following form:
Record 4: X Y Z
Format: 3F10.0
(Record 4 is repeated for each particle to be initialized)

If INIPOS=2, enter record 4 in the following form:
Record 4: J I K NPCELL
Format: 4I10
(Record 4 is repeated for each finite difference cell in which particles are to be initialized)

Explanation:

NCOUNT is the maximum number of particles that can be used in the current simulation run. The actual number of particles to be entered can be smaller or equal to NCOUNT, but must not exceed it.

NTRKOP is the option flag for generating a default output for plotting.
NTRKOP>0, the ID number, x, y, z coordinates, travel times, and J, I, K cell indices of each particle along its path will be saved in a default output file, named P3DPLOT.DAT, whenever the tracking step is an even multiple of NTRKOP. This file can be used for plotting path lines (see Appendix A);
NTRKOP=0, the default P3DPLOT.DAT file is not generated.

NFRONT is the number of specific times at which the positions of all active particles are to be saved in a default output file, named P3DFRONT.DAT, for the purpose of plotting front positions of capture zones (see Appendix A). Set NFRONT=0 if this feature is not needed.

INIPOS is the option flag indicating how to specify the initial positions of the tracking particles.
INIPOS=1, initial positions are entered by specifying the x, y, z coordinates of each individual particle;
INIPOS=2, initial positions are entered by specifying the column (J), row (I), and layer (K) indices of the finite difference cells in which particles are to be initialized as well as the number of particles to be placed in each cell.

IOPSS is the option flag indicating how to remove particles at a sink cell.
IOPSS=1, particles are removed only at a sink cell with inward gradients at *all* of its cell interfaces;
IOPSS=2, particles are removed *as soon as* they enter the sink cell even if it does not have complete inward gradients at its cell interface;
IOPSS=3, this option is the same as IOPSS=1 except when particles enter a well cell without complete inward gradients at its cell interfaces, the dividing streamline of the well is determined based on the analytical solution and the particles are removed if they are inside the dividing streamline;
IOPSS=4, this option is the same as IOPSS=2 except when particles enter a well without complete inward gradients at its cell interfaces, the dividing streamline of the well is determined based on the analytical solution and the particles are removed if they are inside the dividing streamline.

Note: the IOPSS option works similarly in the case of reverse tracking except that particles are removed at source cells with outward gradients instead of sink cells.

INCAPT is the option flag indicating whether to save, in a default file, named P3DCAPT.DAT, the final and initial positions of those particles captured by sinks in the case of forward tracking or sources in the case of reverse tracking.
INCAPT>0, save the initial and final positions of captured particles in the default output file P3DCAPT.DAT (see Appendix A);
INCAPT≤0, the default P3DCAPT.DAT file is not created.

TIME1 is the time to begin particle tracking.

TIME2 is the time to stop particle tracking. If TIME2<TIME1, particles will be tracked backward in the steady state flow field. Currently, PATH3D does not support backward tracking in transient flows. Thus, TIME2 must be greater than TIME1 for transient simulations. Note that if TIME2 is set to a large value like 10^{30}, PATH3D will keep track of the particles until they reach a boundary, or, until the end of the head simulation time in the case of transient simulations.

EPS is the error criterion for the particle tracking solution. The smaller EPS is taken, the more accurate the results will be. But smaller EPS values may require considerably more calculation time than larger EPS values. Values between 10^{-3} to 10^{-4} are normally adequate. However, it is suggested that several values for EPS be tested to find the most efficient EPS value which gives accurate results. A typical procedure is as follows: try a large EPS value first, then decrease it systematically and compare the outputs of two runs. Repeat the procedure until the results between the two runs are approximately the same. The EPS value of the final run is an accurate and efficient choice.

DT is the first guessed stepsize to initialize the particle tracking solution. PATH3D will automatically adjust the stepsize according to the error criterion specified by the user. A reasonable DT may speed up the solution, but will not affect the results. If TIME2 is entered smaller than TIME1, DT will be automatically assigned the negative sign even if it is entered as positive.

MAXSTP is the maximum number of tracking steps allowed. In cases where the error criterion (EPS) is chosen too small, or, when a particle is near areas where the velocity changes dramatically, the program may adjust the stepsize to such a small value that a solution would take a very large number of very small steps. MAXSTP is used to avoid this situation. When the number of tracking steps for a particular particle exceeds MAXSTP, the movement of the particle is terminated. A value equal to 500 is normal adequate.

ISAV is the interval for saving intermediate results in the standard output file. The results of the first and final steps are always saved regardless of the value of ISAV.

ISAV>0, the program will save tracking results whenever the tracking step is an even multiple of ISAV;

ISAV≤0, the program will not save intermediate results except for the first and final tracking steps.

IPRT is the option flag for saving intermediate results in the standard output file.

IPRT=1, the program will print out the X, Y, Z coordinates of the intermediate positions and the travel times at each position;

IPRT=2, the program will print out the X, Y, Z coordinates, travel times, as well as the seepage velocity components (V_x, V_y, V_z) at each position;

IPRT=3, the program will print out the X, Y, Z coordinates, travel times, and the J, I, K cell indices in the finite difference grid corresponding to the particle positions.

TFRONT is an array containing times at which the particle positions are to be saved in a default output file, P3DFRONT.DAT, for the purpose of plotting front positions of the capture zones. The number of elements in array TFRONT must be equal to NFRONT. If NFRONT=0, this array should be omitted, and the default file P3DFRONT.DAT will not be generated.

X, Y, Z are the Cartesian coordinates of a particle to be initialized for tracking. Note that X, Y and Z are *actual distances* along the row, column and layer directions relative to the origin with positive axes in the directions of increasing column, row and layer numbers (remember that the Z axis positive downward). The initial position of particles are entered as the X, Y, Z coordinates only when the flag INIPOS=1 (see Fig. 3.3).

J, I, K are the column, row and layer indices of a finite difference cell where one or more particles are to be initialized for tracking. The initial position of particles are entered as the J, I, K cell indices only when the flag INIPOS=2 (see Fig. 3.4).

NPCELL is the number of particles to be placed in a finite difference cell and also an option flag indicating how to distribute the particles in the cell. NPCELL is needed only if the flag INIPOS=2.

NPCELL=1, only one particle is to be initialized at cell (J, I, K) and the particle is placed at the nodal point (see Fig. 3.4);

NPCELL>1, multiple particles are to be initialized at cell (J, I, K) and the particles are placed in a circle around the nodal point with equal intervals. The effective radius of the circle is calculated as $R_e = \sqrt{\Delta X \Delta Y} / 4.81$ (see Fig. 3.4). All particles in the same cell have the same Z coordinate equal to that of the nodal point);

NPCELL<0, the number of particles to be initialized at cell (J, I, K) is equal to the absolute value of NPCELL and the particles are distributed within cell (J, I, K) randomly as determined by a random number generator (see Fig. 3.4). Particles in the same cell have different Z coordinates.

Content of the particle tracking input file with INIPOS=1:

```
30        1        0        1        1        0
 0  -5000.00    1.E-3       1.      200        1        1
 5        1        5              X,Y,Z
 5        1       15              X,Y,Z
 5        1       25              X,Y,Z
```

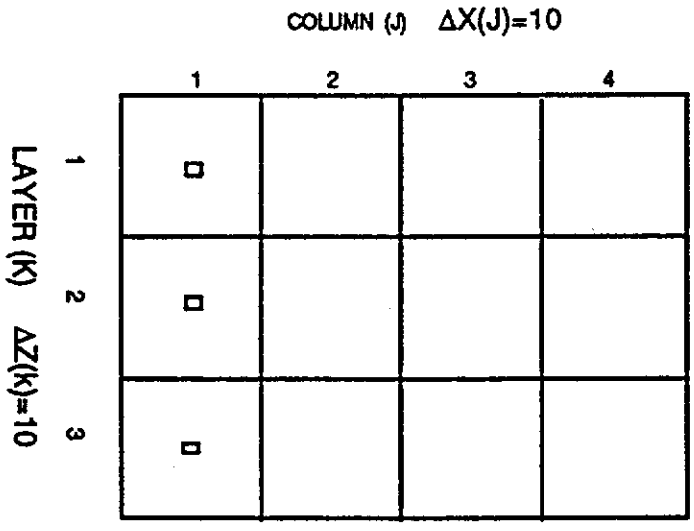


Fig.3.3. Placement of initial particles with option INIPOS=1

Content of the particle tracking input file with INIPOS=2:

```
    30        1        2        2        1        0
     0  -5000.00   1.E-3       1.      200        1        1
-500.     -1500        TFRONT
     1        1        1        1        J, I, K, NPCELL
     1        2        1        8        J, I, K, NPCELL
     1        3        1       -8        J, I, K, NPCELL
```

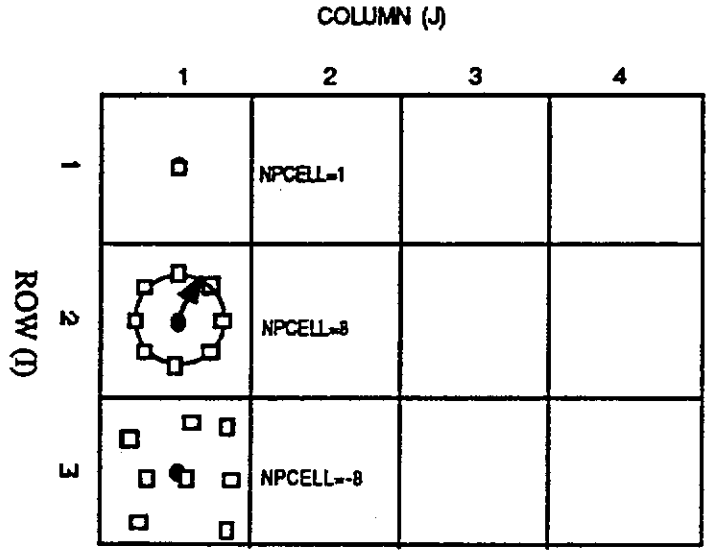


Fig.3.4. Placement of initial particles with option INIPOS=2

3.5.2 BAS input file:

The input file for the MODFLOW Basic (or BAS) package is also needed by PATH3D. No modification is necessary for its use by PATH3D. See McDonald and Harbaugh (1988) for detailed input instructions on the BAS input file.

3.5.3 BCF input file:

The input file for the MODFLOW Block-centered-Flow (or BCF) package is also used by PATH3D. However, three additional data arrays should be added at the end of this input file. These arrays are:

(1) ARRAY: HTOP
READER: RARRAY
(Note: the real array reader RARRAY is compatible with U1DREL/U2DREL as used in MODFLOW and can be used interchangeably. However, RARRAY has more input options available. Refer to Appendix B for more information).

(2) ARRAY: DZ
READER RARRAY
(one array for each layer in the grid)

(3) ARRAY: PRSITY
READER: RARRAY
(one array for each layer in the grid)

Explanation:

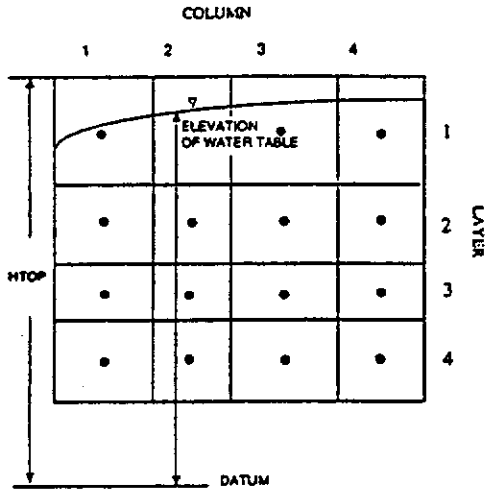
HTOP is the top elevation of cells in the first grid layer relative to the same datum as the hydraulic heads. HTOP is used in PATH3D as reference to the location of particles in the finite-difference grid system in the vertical direction. If the first grid layer is unconfined, HTOP can be set most conveniently to a uniform elevation slightly above the water table (see Fig.

3.5a). If the first grid layer is confined, then, HTOP is equal to the bottom elevation of the confining unit overlying the first grid layer (see Fig. 3.5b). Note that HTOP should be entered properly even for inactive cells in order for particles beneath the inactive cells to be located correctly.

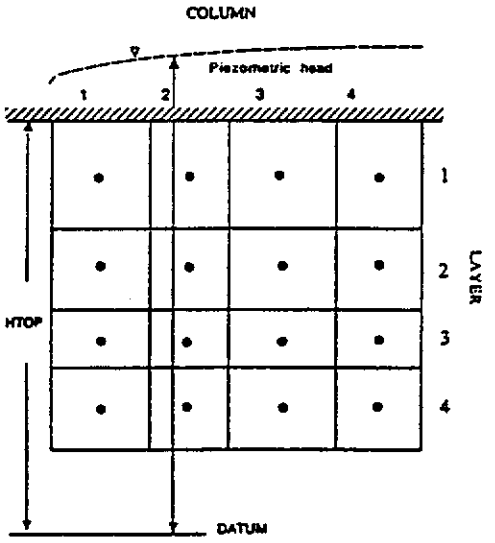
DZ is the cell thickness in each layer. The thickness of the first layer should be entered as the difference between HTOP and its bottom elevation. Under most circumstances, the grid is discretized into horizontal layers so that HTOP for the first layer and DZ within each layer are uniform (see Fig. 3.6a). However, if a distorted grid is used in the vertical direction, both HTOP and DZ can be different within the same layer (see Fig. 3.6b). Note that DZ should be entered properly even for inactive cells in order for particles beneath the inactive cells to be located correctly.

It should be pointed out that in MODFLOW, an aquitard between two aquifers may be implicitly represented by vertical conductance between the two aquifers so that the aquitard itself is not present as a layer in the flow model. It is advisable to avoid this "quasi-three-dimensional" approach and to have every layer explicitly represented inside the particle tracking area in order to fully define the flow domain through which a particle has to travel. Nevertheless, a quasi-3D model may not always be avoided. If this is the case, then, the thickness of the aquitard, which is not present in the flow model, should be added to the layer immediately above the aquitard when entering DZ in order to have particles located properly and total travel time computed correctly.

PRSITY is the effective porosity of the aquifer materials in each cell. If it is necessary to include the effect of sorption on particle tracking, enter PRSITY as product of porosity and retardation factor.

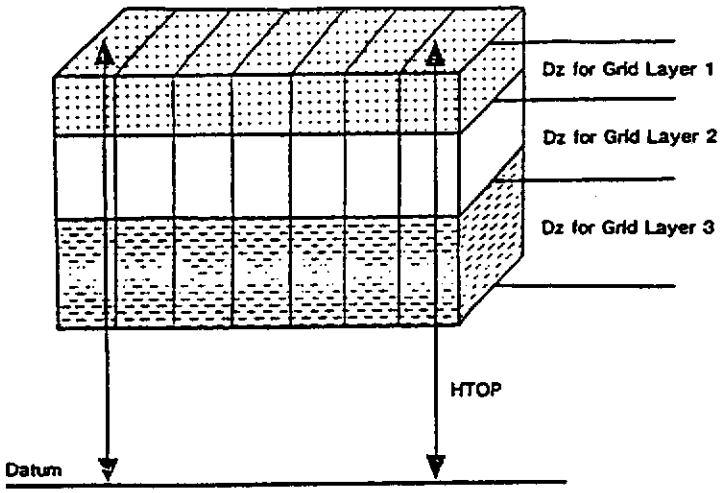


(a) Array HTOP for unconfined aquifer

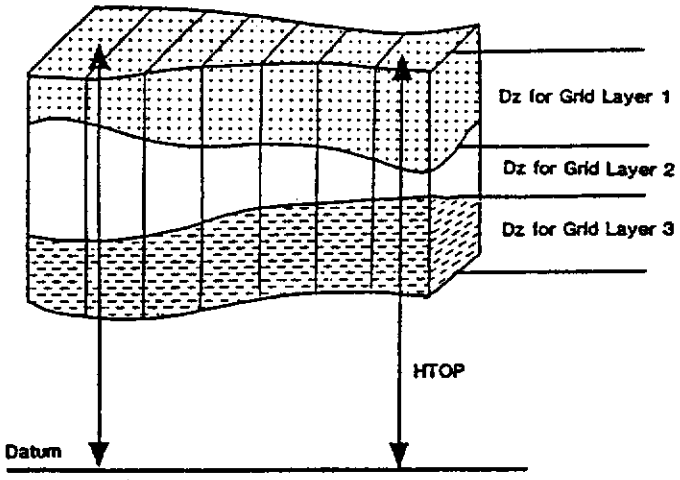


(b) Array HTOP for confined aquifer

Fig. 3.5. Illustration of arrays HTOP for unconfined and confined aquifer layers



(a) Horizontal layers



(b) Distorted layers

Fig.3.4 Arrays HTOP and DZ for different vertical discretization schemes

3.5.4 RCH and EVT input files:

If the recharge or evapotranspiration option is used in the head simulation, the input files for the MODFLOW RCH and EVT packages are also required by PATH3D in order to calculate the vertical components of the seepage velocity at water table cells. No modification to these input files is necessary.

3.5.5 WEL, DRN, RIV and GHB and STR input files:

If wells, drains, rivers or general-head boundaries are simulated in the head solution, input files for the MODFLOW WEL, DRN, RIV or GHB packages are also needed by PATH3D. PATH3D will read these files to specify the locations of sinks/sources. No modification is needed for these files to be read by PATH3D.

The STR package was an add-on package to MODFLOW for simulating streams with variable stages (see Prudic, 1989). If this package is used in flow simulation, the input file is also needed by PATH3D. No modification is necessary.

3.5.6 Head input file:

PATH3D can directly read the unformatted head file, created during a run of the MODFLOW model using OUTPUT CONTROL options (see the MODFLOW documentation). In the case of transient simulations, the heads should be saved EVERY time step during the MODFLOW run since PATH3D reads heads sequentially beginning from the first step. PATH3D can also read a formatted head file. When running the program, PATH3D will ask the user to enter the name of the head file, and to specify whether it is unformatted or formatted. If it is formatted, PATH3D will then prompt the user to enter the format of the head file. The format should be a valid FORTRAN format specifier

included within a parenthesis such as (10F8.3). PATH3D reads heads row by row for each layer, proceeding sequentially from the first layer to the last layer.

3.6 FILE USAGE

PATH3D presets several unit numbers for various input and output files through the following parameter statement in the main program:

```
PARAMETER (INBAS=1,IOUT=6,INTRK=2,INH=3,INPLOT=4,IFRONT=7,ICNFG=8,ICAPT=9)
```

where

INBAS	is unit number for the input file of MODFLOW BAS package;
IOUT	is unit number for the standard output file;
INTRK	is unit number for the particle tracking input file;
INH	is unit number for the head input file;
ICNFG	is unit number for the model grid configuration file generated by PATH3D for creating head contour maps (see Appendix A);
INPLOT	is unit number for the optional P3DPLOT.DAT output file. This unit is used only if the flag NTRKOP≥0;
IFRONT	is unit number for the optional P3DFRONT.DAT output file. This unit is used only if the flag NFRONT≥0;
ICAPT	is unit number for the optional P3DCAPT.DAT output file. This unit is used only if the flag INCAPT≥0;

These units can be changed, if necessary, by modifying the above parameter statement and recompiling the source code. Note that if the unit numbers for MODFLOW individual packages that are needed by PATH3D conflict with a unit number shown above, the IUNIT array entries in the MODFLOW BAS package input file should be reassigned to different numbers.

3.7 RUNNING PATH3D

There are two ways to start PATH3D. The first method is simply type the name of the executable file. The program will prompt the user for the names of various input and output files. An example is given below, where "C:\>" is the command prompt and "P3D" is the name of the executable file of the PATH3D program.

```
C:\>P3D
          ++++++++++++++++++++++++++++++++++++++++++++++++++++++
          +                                                    +
          +                      P A T H 3 D                   +
          +     A Ground-Water Path and Travel-Time Simulator  +
          +                       (V. 3.00)                    +
          +                                                    +
          ++++++++++++++++++++++++++++++++++++++++++++++++++++++

Enter a name for output file:
P3D.OUT
Enter the name of particle tracking input file:
PATH.INP
Enter the name of head file:
HEAD.UFM
Enter U if the head file is UNFORMATTED;
Otherwise, enter format of the head file:
U
Print out heads for checking? (Enter Y or N)
N
Enter the name of BAS input file:
BAS.DAT
Enter the name of BCF input file:
BCF.DAT
Enter the name of WEL input file:
WEL.DAT
Enter the name of GHB input file:
GHB.DAT
Enter the name of RCH input file:
RCH.DAT

                    STRESS PERIOD NO.    1
                    ------------------------
```

```
TIME STEP NO.    1
-----------------
Particle No.    5 in progress

TIME STEP NO.    2
-----------------
Particle No.    5 in progress

Stop - Program terminated.
```

The second method is to run PATH3D in the batch mode by creating a response file which contains the names of input and output files in the order required by PATH3D. The content of such a response file (P3D.FIL) for the example above should be as follows:

```
P3D.OUT
path.inp
head.ufm
u
n
bas.dat
bcf.dat
wel.dat
ghb.dat
rch.dat
```

Then, at the command prompt, type:

```
C:\>P3D < P3D.FIL
```

Chapter 4
Applying PATH3D

This chapter is intended to demonstrate some of the capabilities and accuracy aspects of the PATH3D program by applying it to four example problems with varying complexities. It is also intended to familiarize the users with the preparation, setup and post-processing of a PATH3D run.

The first example illustrates the use of PATH3D to calculate groundwater flow paths and travel times in a heterogeneous cross section. The second example demonstrates the use of PATH3D to solve a transient problem in two-dimensional plan view. The third example shows the use of PATH3D to delineate the well capture zone in a three-dimensional flow system. The final example evaluates the use of PATH3D to define the frontal position of the capture zone in a spherical flow field.

The input and output files of the these four examples are included in *Example Disk* accompanying this manual. It is recommended that the user spend some time studying and running these examples to become familiar with PATH3D before applying it to their own problems.

4.1 FLOW PATH IN HETEROGENEOUS CROSS SECTION

The configuration of the flow field used in this example is illustrated in Fig. 4.1. The cross section 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. The flow domain was divided into 50 columns and 20 layers with a

uniform spacing of 2 feet in each direction The cross section was treated as a single row with a uniform width of 1 foot in the direction perpendicular to the figure. The aquifer was assumed to have a hydraulic conductivity of 20 ft/d except between layers 10 and 15 where a conductivity value of 1 ft/d was given.

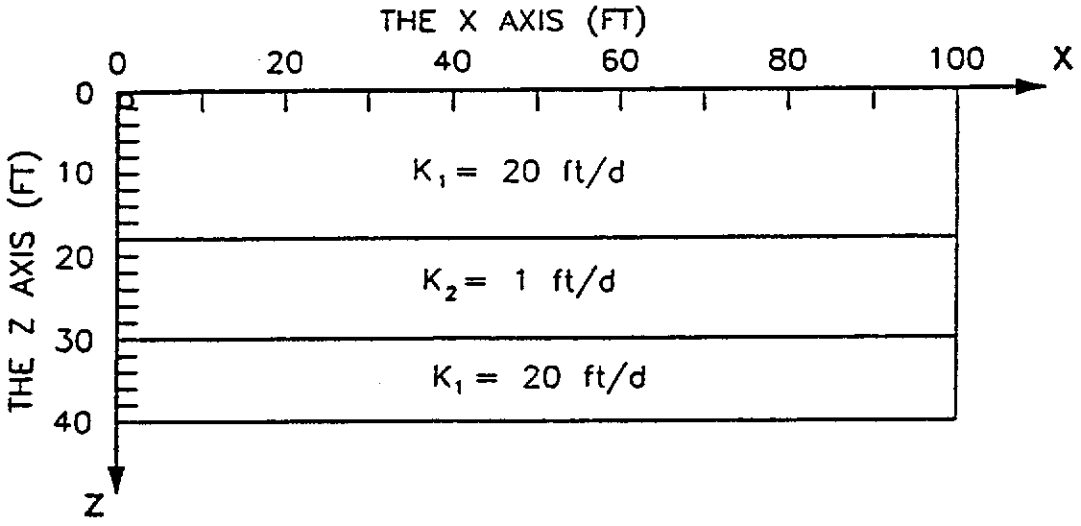


Fig. 4.1. Configuration of the flow field used in the first example

To simulate the distribution of hydraulic heads with MODFLOW, four input files were created. BAS.DAT for the Basic package; BCF.DAT for the Block-Centered-Flow package; SIP.DAT for the solution package; and OPC.DAT for the Output Control options. By setting the flags in file OPC.DAT to appropriate values, an unformatted head file, named HEAD.UFM, was generated after running MODFLOW.

Both BAS.DAT and HEAD.UFM can be used directly by PATH3D. BCF.DAT must be modified by adding HTOP, DZ and PRSITY arrays at the end of file. HTOP is the top elevation of the first model layer, and is equal to 40 feet in this example, since the datum for the heads is set at the aquifer bottom. DZ is

the thickness of cells in each layer and is equal to 2 feet. PRSITY is the porosity of aquifer materials and is assumed to have a uniform value of 0.2. The modified BCF.DAT is shown here:

```
        1        0                                  */ISS,IBCFCB
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  */LAYCON
        0  1.00                                    */TRPY
        0 2.000                                    */DELR
        0 1.000                                    */DELC
        0 20.00                                    */HY   for Layer:  1
        0 38.00                                    */BOT  for Layer:  1
        0 10.00                                    */VC   for Layer:  1
        0 40.00                                    */TRAN for Layer:  2
        0 10.00                                    */VC   for Layer:  2
        0 40.00                                    */TRAN for Layer:  3
        0 10.00                                    */VC   for Layer:  3
        0 40.00                                    */TRAN for Layer:  4
        0 10.00                                    */VC   for Layer:  4
        0 40.00                                    */TRAN for Layer:  5
        0 10.00                                    */VC   for Layer:  5
        0 40.00                                    */TRAN for Layer:  6
        0 10.00                                    */VC   for Layer:  6
        0 40.00                                    */TRAN for Layer:  7
        0 10.00                                    */VC   for Layer:  7
        0 40.00                                    */TRAN for Layer:  8
        0 10.00                                    */VC   for Layer:  8
        0 40.00                                    */TRAN for Layer:  9
        0 .9524                                    */VC   for Layer:  9
        0 2.000                                    */TRAN for Layer: 10
        0 .5000                                    */VC   for Layer: 10
        0 2.000                                    */TRAN for Layer: 11
        0 .5000                                    */VC   for Layer: 11
        0 2.000                                    */TRAN for Layer: 12
        0 .5000                                    */VC   for Layer: 12
        0 2.000                                    */TRAN for Layer: 13
        0 .5000                                    */VC   for Layer: 13
        0 2.000                                    */TRAN for Layer: 14
        0 .5000                                    */VC   for Layer: 14
        0 2.000                                    */TRAN for Layer: 15
        0 .9524                                    */VC   for Layer: 15
        0 40.00                                    */TRAN for Layer: 16
        0 10.00                                    */VC   for Layer: 16
        0 40.00                                    */TRAN for Layer: 17
        0 10.00                                    */VC   for Layer: 17
        0 40.00                                    */TRAN for Layer: 18
        0 10.00                                    */VC   for Layer: 18
```

```
        0 40.00                                    */TRAN for Layer: 19
        0 10.00                                    */VC   for Layer: 19
.........0 40.00 ..................................*/TRAN for Layer: 20
        0    40.    Add new arrays starting here     */HTOP
        0  2.00                                     */DZ-1
        0  2.00                                     */DZ-2
        0  2.00                                     */DZ-3
        0  2.00                                     */DZ-4
        0  2.00                                     */DZ-5
        0  2.00                                     */DZ-6
        0  2.00                                     */DZ-7
        0  2.00                                     */DZ-8
        0  2.00                                     */DZ-9
        0  2.00                                     */DZ-10
        0  2.00                                     */DZ-11
        0  2.00                                     */DZ-12
        0  2.00                                     */DZ-13
        0  2.00                                     */DZ-14
        0  2.00                                     */DZ-15
        0  2.00                                     */DZ-16
        0  2.00                                     */DZ-17
        0  2.00                                     */DZ-18
        0  2.00                                     */DZ-19
        0  2.00                                     */DZ-20
        0  0.20                                     */PRSITY-1
        0  0.20                                     */PRSITY-2
        0  0.20                                     */PRSITY-3
        0  0.20                                     */PRSITY-4
        0  0.20                                     */PRSITY-5
        0  0.20                                     */PRSITY-6
        0  0.20                                     */PRSITY-7
        0  0.20                                     */PRSITY-8
        0  0.20                                     */PRSITY-9
        0  0.20                                     */PRSITY-10
        0  0.20                                     */PRSITY-11
        0  0.20                                     */PRSITY-12
        0  0.20                                     */PRSITY-13
        0  0.20                                     */PRSITY-14
        0  0.20                                     */PRSITY-15
        0  0.20                                     */PRSITY-16
        0  0.20                                     */PRSITY-17
        0  0.20                                     */PRSITY-18
        0  0.20                                     */PRSITY-19
        0  0.20                                     */PRSITY-20
```

Next, a new input file was set up for entering particle tracking control parameters as well as the initial position of particles to be tracked. The content of this input file, named PATH.INP, is shown below:

```
      1         1         0         1         1         0
      0  -5000.00     1.E-3        1.       200         1         1
   1.00        1.      1.00
```

Since in this input file TIME2 (-5000) is smaller than TIME1 (0), the particle will be tracked backward, from downgradient to upgradient. The initial position of the particle to be tracked is placed at x=1.0, y=1.0, z=1.0. The y coordinate must be given a value between 0 and 1 because the cross section is treated as one single row with a uniform width of 1 foot.

Note that in this example, the initial stepsize is set to 1.0. This value is automatically converted to -1.0 because TIME2 is less than TIME1. The negative travel time indicates that the particle is tracked backward. The error criterion is set equal to 10^{-3}. Running PATH3D with input files PATH.INP, HEAD.UFM, BAS.DAT, and BCF.DAT, an output file, P3D.OUT, as shown next, is obtained. Note that because in file PATH.INP flag ISAV=1, results at every tracking step are printed. Because option IPRT is set equal to 1, only the x, y, z coordinates and travel times are printed.

```
                  ++++++++++++++++++++++++++++++++++++++++++++++++++++
                  +                                                  +
                  +                    P A T H 3 D                   +
                  +   A Ground-Water Path and Travel-Time Simulator  +
                  +                    (V. 3.00)                     +
                  +                                                  +
                  ++++++++++++++++++++++++++++++++++++++++++++++++++++

EXAMPLE NO. 1:
2-D CROSS SECTION
FLOW MODEL CONSISTS OF     20 LAYERS     1 ROWS    50 COLUMNS
NUMBER OF STRESS PERIOD(S) IN SIMULATION =         1
TIME UNIT USED IN THE MODEL IS DAYS
I/O UNITS:  11  0  0  0  0  0  0  0 19  0  0 22
```

```
FLOW FIELD IS STEADY-STATE (ISS NOT 0)
NUMBER OF PARTICLES FOR TRACKING =         1
OPTION FOR GENERATING PLOT DATA FILE IS ENABLED
     11296 ELEMENTS OF THE Y ARRAY USED OUT OF      42000

(echo of input data for flow model deleted)

                TRACKING START TIME (TIME1) =     .0000000
                TRACKING END TIME (TIME2) =    -5000.000
                ERROR CRITERION (EPS) =     .1000000E-02
                INITIAL TRACKING STEPSIZE =     1.000000
                MAXIMUM TRACKING STEPS ALLOWED =         200
                INTERVAL FOR SAVING INTERMEDIATE RESULTS =   1
                OPTION FOR PRINTING INTERMEDIATE RESULTS =   1

                        PARTICLE STARTING POSITIONS
               PARTICLE NO.       X            Y            Z
                     1        1.0000       1.0000       1.0000

                          STRESS PERIOD NO.    1
                ------------------------------------------------

TIME STEP NO.    1
---------------

HEADS FOR CURRENT TIME STEP READ UNFORMATTED ON UNIT   4

                ================================================
                PARTICLE COORDINATES AND TRAVEL TIMES
                ================================================

PARTICLE NO.        1
      X             Y             Z             TIME
   1.000000      1.000000      1.000000       .0000000
   1.447983      1.000000      2.377378      -.8865356
   1.689094      1.000000      3.484905      -1.688833
   2.260835      1.000000      5.822359      -3.908461
   2.865899      1.000000      7.810176      -6.423677
   3.650099      1.000000      9.852192      -9.630926
   4.779618      1.000000      11.88647      -13.83143
   5.998296      1.000000      13.38876      -17.83507
   8.062387      1.000000      14.94112      -23.43160
   10.94374      1.000000      16.15647      -29.68211
   14.55102      1.000000      17.07000      -36.10947
   18.97880      1.000000      17.74929      -42.72905
   21.16464      1.000000      17.98953      -45.69310
   21.30883      1.000000      18.11040      -47.27416
   21.35938      1.000000      18.22014      -48.71148
   21.56215      1.000000      18.65786      -54.46072
   22.37314      1.000000      20.33887      -77.45772
```

```
   23.05778        1.000000        21.86279        -99.49593
   24.63166        1.000000        26.15693        -165.0653
   25.26358        1.000000        28.38137        -201.4057
   25.56163        1.000000        29.74449        -223.9082
   25.68972        1.000000        30.00900        -228.2942
   26.61413        1.000000        30.24702        -232.4786
   27.92472        1.000000        30.55466        -238.2462
   33.57249        1.000000        31.51129        -261.3168
   41.33440        1.000000        32.24084        -290.0969
   52.85384        1.000000        32.43695        -330.4456
   70.82515        1.000000        30.66946        -397.2947
   73.19823        1.000000        30.12231        -407.5370
   73.70692        1.000000        29.98158        -409.9894
   73.74242        1.000000        29.83048        -412.6070
   73.88410        1.000000        29.22473        -423.0772
   74.56139        1.000000        26.59618        -464.9581
   75.57233        1.000000        23.71146        -509.8444
   76.43431        1.000000        21.57964        -541.4313
   77.62540        1.000000        19.02223        -577.0383
   77.96504        1.000000        18.33121        -586.5191
   78.10599        1.000000        18.02254        -590.4717
   79.27317        1.000000        17.89060        -592.1342
   80.38274        1.000000        17.76472        -593.6342
   82.10529        1.000000        17.54512        -596.0491
   84.95253        1.000000        17.08352        -600.3389
   88.02973        1.000000        16.38130        -605.5984
   90.76443        1.000000        15.37993        -611.0862
   93.04468        1.000000        14.00929        -616.6804
   94.78898        1.000000        12.24710        -621.8956
   96.13236        1.000000        10.06135        -626.6640
   97.45743        1.000000        6.555615        -631.8723
   98.14637        1.000000        4.037750        -634.5739
   98.75540        1.000000        1.421205        -636.4404
   99.37796        1.000000       -.7439667E-08    -637.3491
REMOVED: LAST POSITION EXIT FROM MODEL EDGE

>>ALL PARTICLES HAVE ALREADY BEEN REMOVED
```

Since flag NTRKOP is set to a non-zero value in PATH.INP, a default output, called P3DPLOT.DAT, is also generated. This file contains the particle ID number, the x, y, z coordinates, and travel times. P3DPLOT.DAT can be used directly to illustrate the flow paths as calculated by PATH3D with a plotting program such as SURFER© by Golden Software, Inc. The flow paths for this example, as created by TOPO, a part of SURFER, is shown in Figure 4.2. It is

worth pointing out that in TOPO the vertical axis is taken from bottom to top, but the vertical axis of particle coordinates in PATH3D is oriented downward from top to bottom. Therefore, to display particle coordinates with TOPO, the y or z coordinates saved in file P3DPLOT.DAT must be transformed properly (see Appendix A). In this example, the z coordinates are transformed as (40-z).

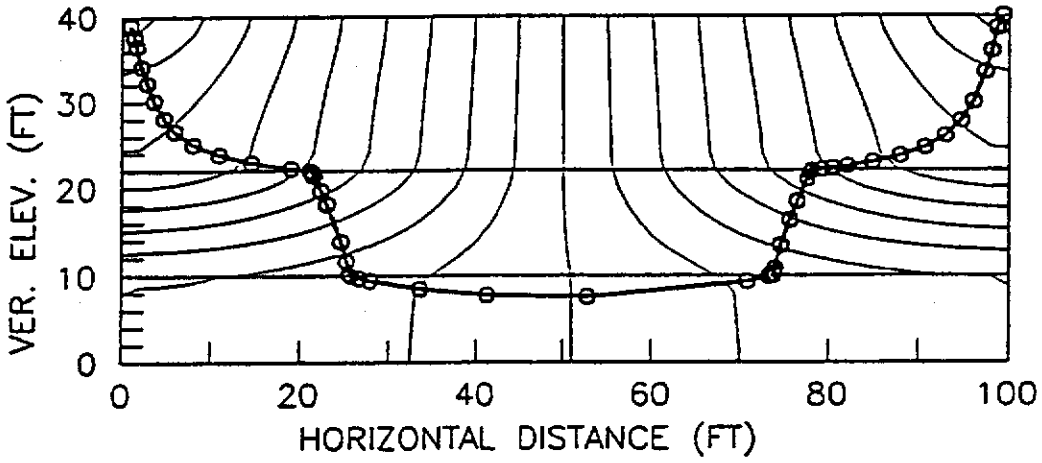


Fig.4.2. Flow path along the cross section as calculated by PATH3D. Each dot denotes an intermediate position of the particle at a particular tracking step.

In Figure 4.2, each dot indicates the intermediate position of the particle at a particular tracking step. Note that the particle travels at uneven intervals in order to achieve the user-specified error criterion with minimum computational effort. The particle takes more steps where the head gradients change rapidly, but moves with fewer steps through the areas where the gradients do not vary much. Also notice that the pathline changes its direction sharply at the interface between the low- and high-conductivity layer. The reflection of the pathline preserves the

tangent law, thereby indicating that PATH3D can correctly simulate flow paths in heterogeneous media.

4.2 FLOW PATHS IN TRANSIENT FLOW FIELD

The configuration of the flow domain used in this example is shown in Fig. 4.3. The confined aquifer is 10 feet thick and is bounded by a fully penetrating stream on the right side. The left, top and bottom boundaries are placed far away from the pumping well so that the aquifer may be viewed as an infinite half-domain. A period of 20,000 minutes was simulated with 20 time steps to obtain the transient head solution. A total number of 5 particles were placed on the stream edge. Since the confined aquifer is 10 feet thick, and the head datum is set at the aquifer bottom, HTOP is equal to 10 feet for this example, and DZ is also 10 feet. Note that although the z coordinates of the initial positions do not affect the tracking results, because the flow field is only two-dimensional in the horizontal plane, they still must be between 0 and 10 feet. Otherwise, PATH3D will remove the particles and print out the message that the particles are outside the model edges.

The flow paths through the transient flow field as simulated by PATH3D are illustrated in Fig. 4.4. The maximum length of the flow domain in the y direction is 2149 feet. To display the particle coordinates saved in file P3DPLOT.DAT with TOPO, the y coordinates were transformed with (2149-y). It takes particle No.1, No.2 and No.3 approximately 14499, 9959 and 8648 minutes, respectively, to travel from the river to the well. Since the flow field is symmetrical about the x axis, the travel times for particle No.4 and No.5 are the same as for No.2 and No.1, respectively. The travel times can also be calculated

with the semi-analytical method based on the Theis solution. The difference with those calculated by PATH3D is negligible.

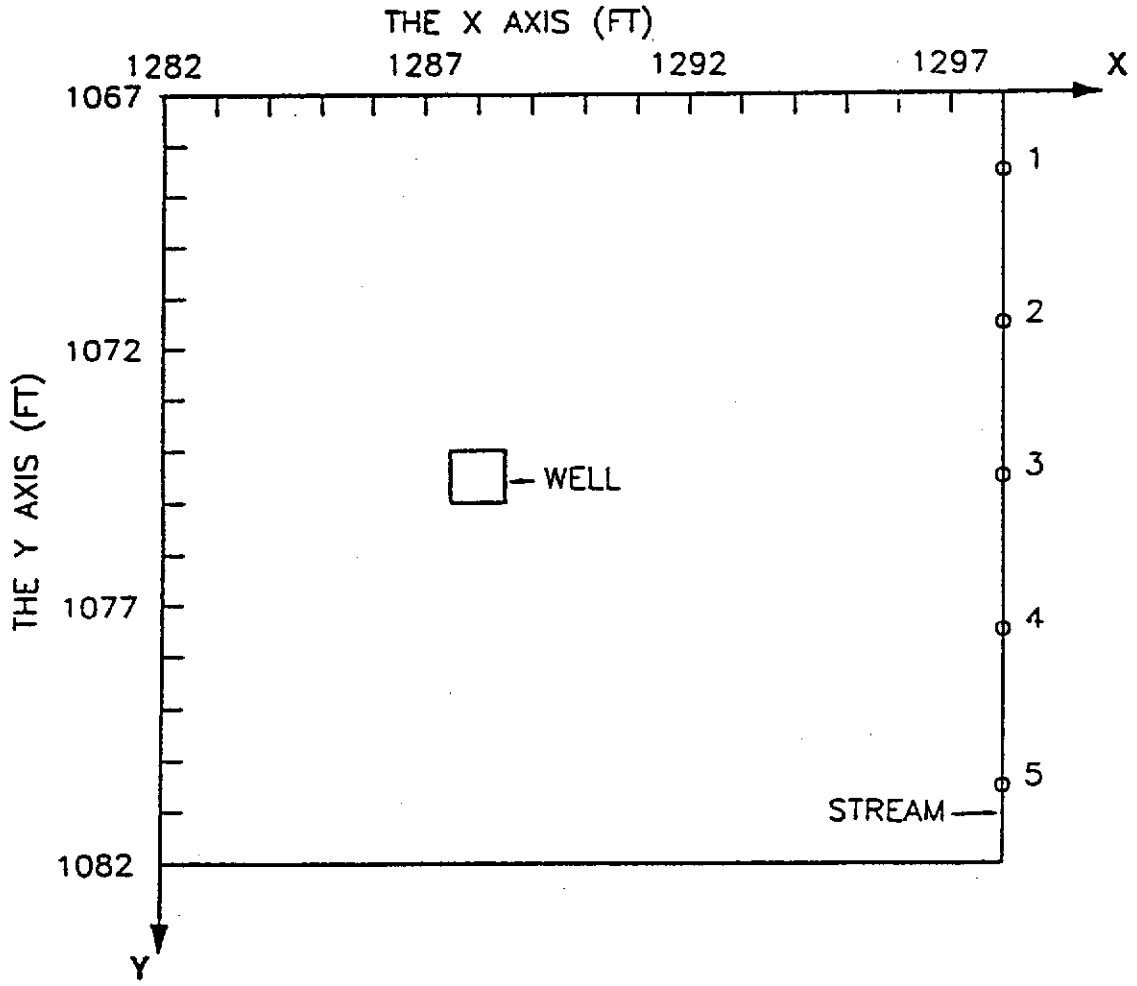


Fig.4.3. Configuration of the flow domain used in the second example

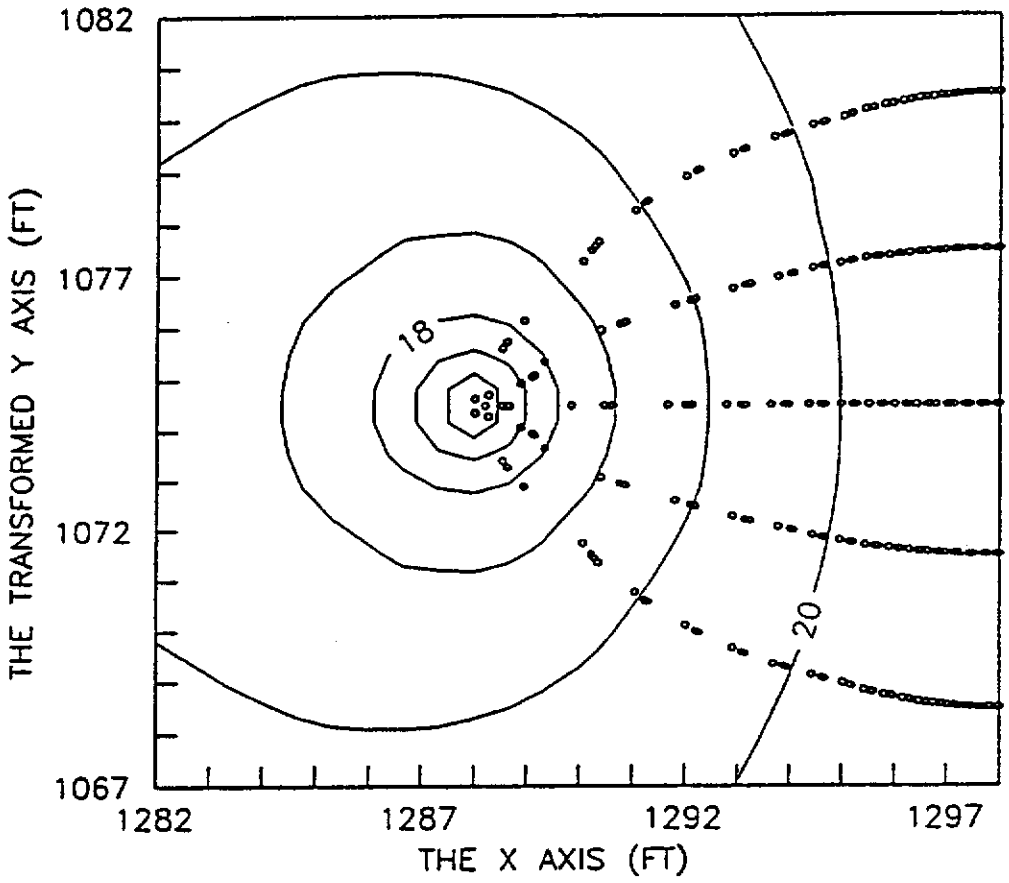


Fig.4.4. Flow paths through the transient head field examined in the second example

4.3 CAPTURE ZONE IN 3D FLOW FIELD

The configuration of the flow domain in this example is shown in Fig. 4.5. The boundary conditions for the unconfined aquifer are no-flow along the north and south sides, and constant-head along the east and west sides to simulate a regional gradient. In the vertical domain, the aquifer is bounded by a no-flow boundary at the bottom. The aquifer is discretized into 31 columns, 21 rows and

6 layers. DZ is equal to 1, 1, 2, 4, 6, and 10 feet, respectively, with a total thickness of 24 feet. Because the datum for the heads is set at 66 ft below the aquifer bottom, HTOP is equal to 90 feet in this example.

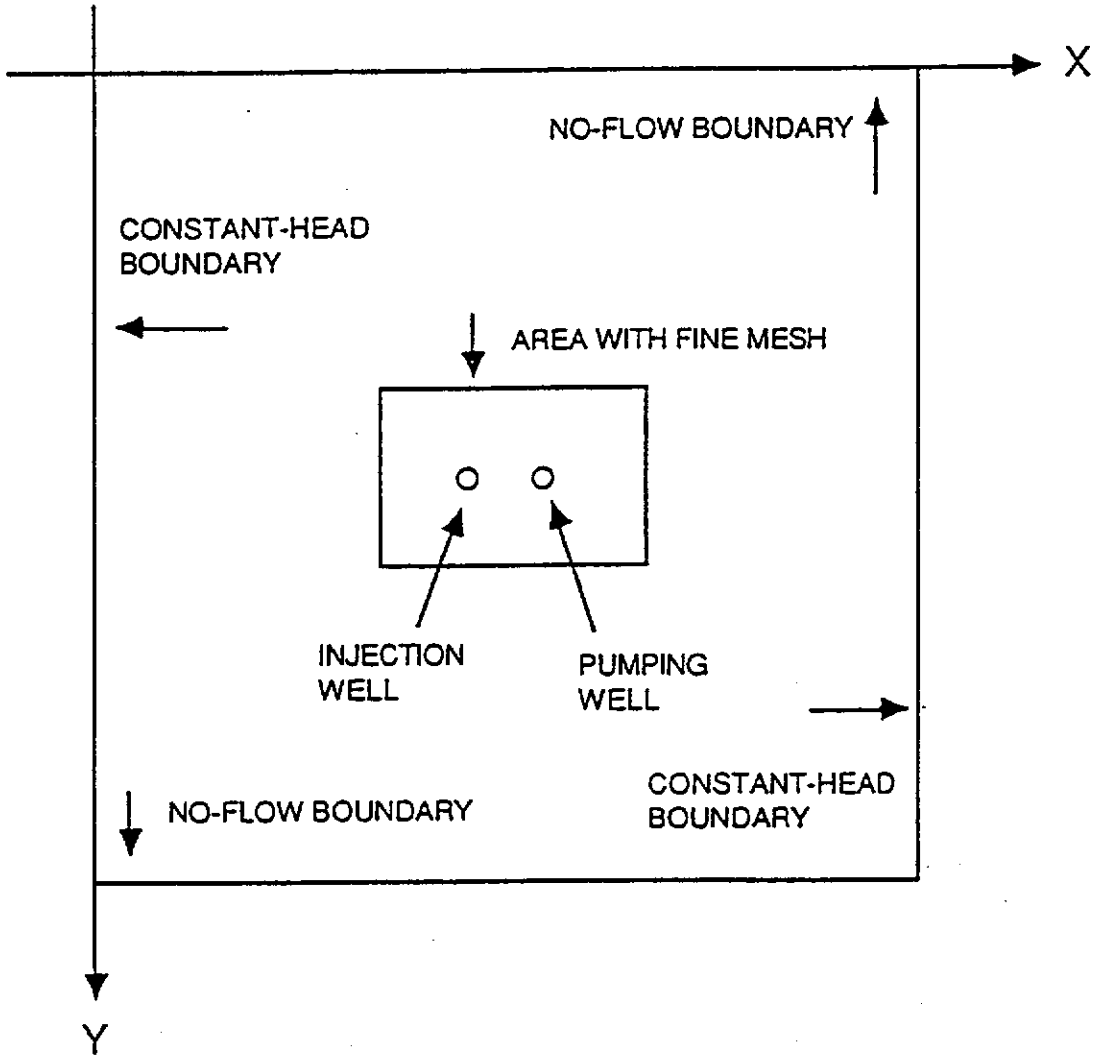


Fig.4.5. Configuration of flow field examined in the third example

A pair of injection and extraction wells were placed near the center of the flow domain in order to contain and clean up contaminants in the vicinity of the wells. The wells are assumed to penetrate into the water table only 1 foot. To delineate the capture zone created by the wells, 8 particles were placed surrounding the cell which represents the injection well in the flow model. The initial z coordinates for all six particles are set at 0.5 ft, or at the midpoint of the first layer.

Fig. 4.6 shows the projection of the capture zone on the water table and Fig. 4.7 the projection on a vertical section passing through the two wells. Fig. 4.8 gives a three-dimensional view of the capture zone.

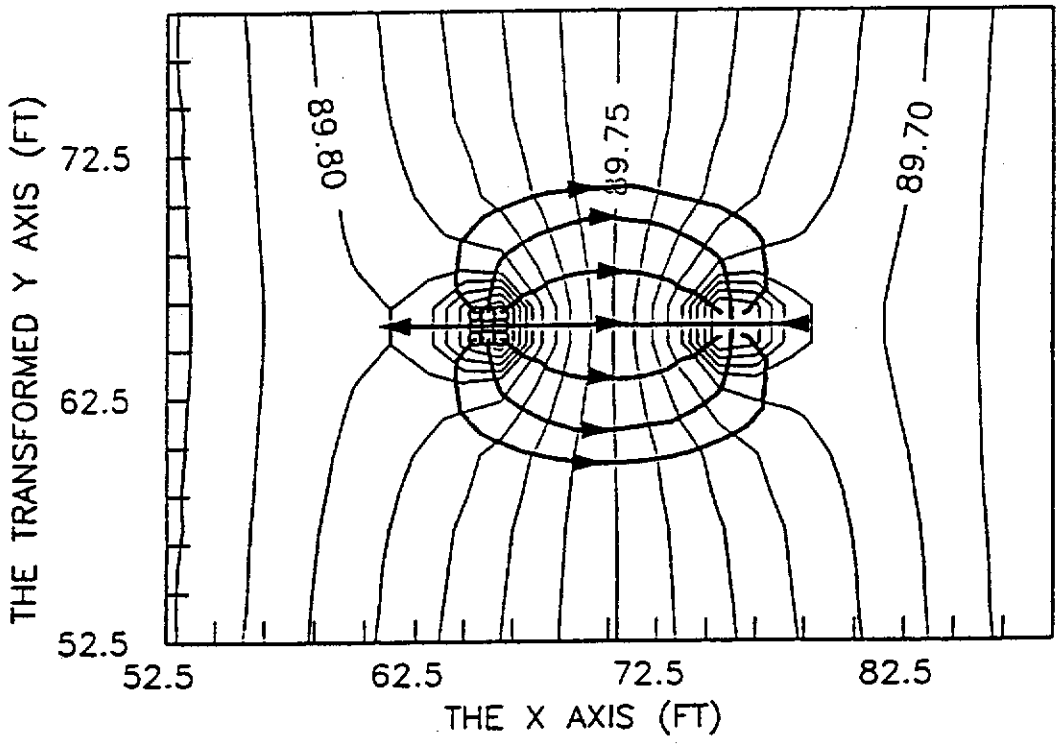


Fig.4.6. Projection of flow paths on the water table

If the aquifer is approximately treated as an infinite flow domain and the wells as a point sink and source, the maximum depth of the capture zone and the minimum travel time between the two wells can be calculated by an analytical solution. The results of this analytical solution and PATH3D are nearly identical, demonstrating the accuracy of PATH3D in solving three-dimensional problems.

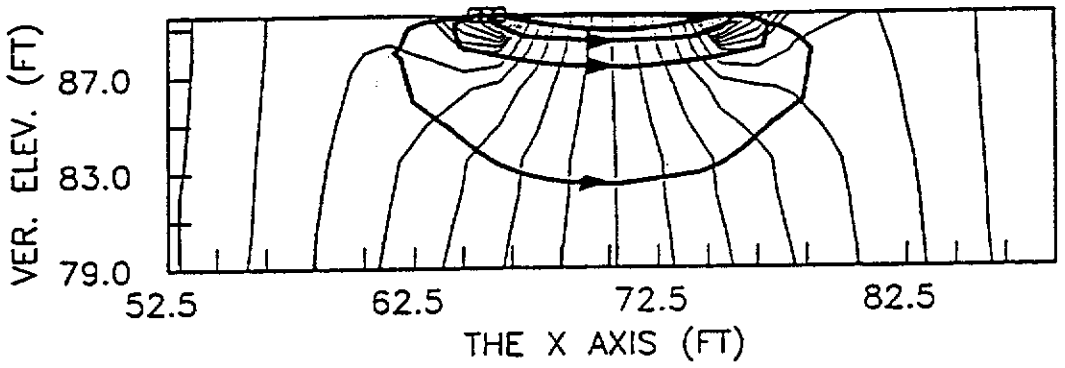


Fig.4.7. Projection of flow paths on the cross section

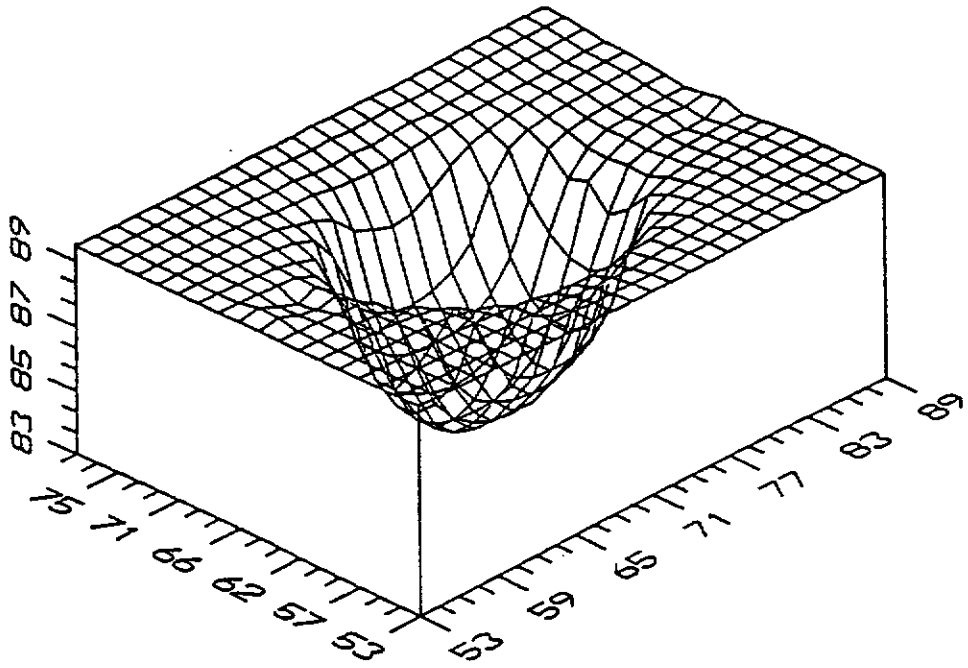


Fig.4.8. A three-dimensional view of the capture zone delineated by flow paths

4.4 FRONT POSITION IN SPHERICAL FLOW FIELD

The flow field considered in this example is spherical three-dimensional with a point source located at the origin. Since the flow field is symmetric, only the portion with positive x, y, and z coordinates was modeled with a grid of 25 columns by 25 rows by 25 layers. The specific discharges at cell interfaces were directly calculated from the analytical solution:

$$q_R = \frac{Q}{4\pi R^2}$$
$$q_x = q_R \cos\theta \sin\phi$$
$$q_y = q_R \sin\theta \sin\phi$$
$$q_z = q_R \cos\phi$$

with

$$R = \sqrt{x^2 + y^2 + z^2}$$
$$\theta = \arctan(y / x)$$
$$\phi = \arccos(z / R)$$

where Q is the flow rate of the point source, set equal to 1 ft^3/day for this example.

A total number of 676 particles were placed surrounding the point source at a radius equal to 0.1 ft. Fig. 4.9 shows the position of the displacement front at time equal to 20,000 days, given a porosity n=0.2. The radius of the displaced front as calculated by PATH3D is approximately 28.8 ft, nearly identical to that calculated from the following analytical solution:

$$R = \left(\frac{3Qt}{4\pi n}\right)^{1/3}$$

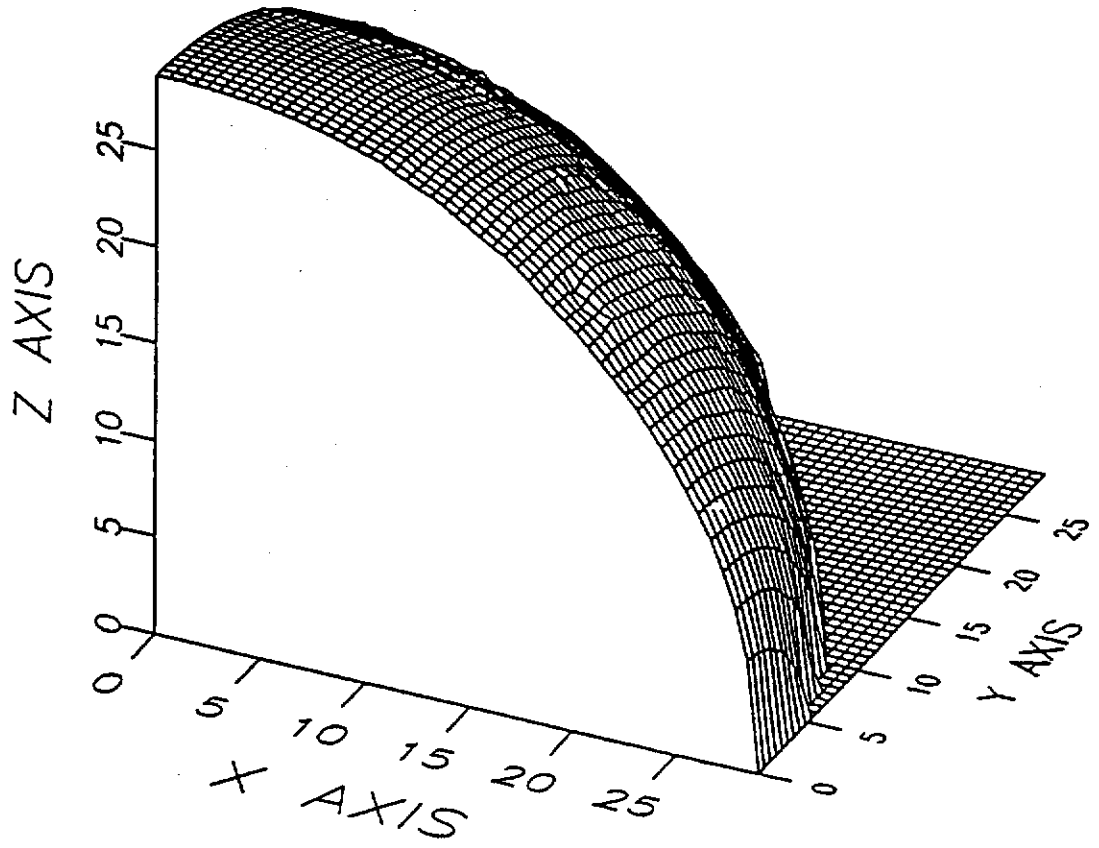


Fig.4.9. A three-dimensional view of the front position at time equal to 20,000 days in the spherical flow field

REFERENCES

Bear, J. 1972. Dynamics of fluids in porous media. Elsevier Publishing Company. 764pp.

McDonald, M.G. and A.W. Harbaugh. 1988. A modular three-dimensional finite-difference ground-water flow model. USGS TWI Chapter 6-A1. 586pp.

Prudic, D.E. 1989. Documentation of a computer program to simulate stream-aquifer relations using a modular, finite-difference ground-water flow model. USGS Open-File Report 88-729. 113pp.

Zheng, C. 1988. New solution and model for evaluation of groundwater pollution control. Ph.D Thesis, University of Wisconsin-Madison, Madison. 149pp.

Appendix A
Post-Processing Programs

PATH3D comes with several utility programs which can be used to process output files generated by PATH3D and the unformatted head file generated by MODFLOW for plotting purposes. These post-processing programs include POSTMODF, P3DPLOT, P3DFRONT, and P3DCAPT, and are explained in this appendix.

A.1 POSTMODF

POSTMODF can be used to extract the simulated hydraulic heads within a user-specified window at any desired time step from the unformatted head file created by MODFLOW, and save them in such a form that they can be utilized directly by most commercially available graphic software packages to generate contour maps or other types of plots.

POSTMODF operates on the unformatted head file generated after running MODFLOW, and the model grid configuration file, P3DCNFG.DAT, generated after running PATH3D. The structures and contents of these two files are shown as below:

Unformatted Head File (also see McDonald and Harbaugh, 1988):

For each time step saved:

For each layer of the three-dimensional head matrices:

Record 1: KSTP,KPER,PERTIM,TOTIM,TEXT,NCOL,NROW,ILAY

Record 2: (((HNEW(J,I,ILAY),J=1,NCOL),I=1,NROW)

where

KSTP	is the number of the time step at which the heads are saved;
KPER	is the number of the stress period at which the heads are saved;
PERTIM	is the elapsed time since beginning of the current stress period;
TOTIM	is the total elapsed time at which the heads are saved;
TEXT	is a character string set equal to "HEAD";
NCOL	is the total number of columns;
NROW	is the total number of rows;
ILAY	is the layer at which the heads are saved; and
HNEW	is the simulated heads.

P3DCNFG.DAT (saved in free format):

Record 1:	NLAY, NROW, NCOL
Record 2:	(DELR(J), J=1,NCOL)
Record 3:	(DELC(I),I=1,NROW)
Record 4:	((HTOP(J,I), J=1,NCOL),I=1,NROW)
Record 5:	(((DZ(J,I,K), J=1,NCOL),I=1,NROW),K=1,NLAY)
Record 6:	HNOFLO

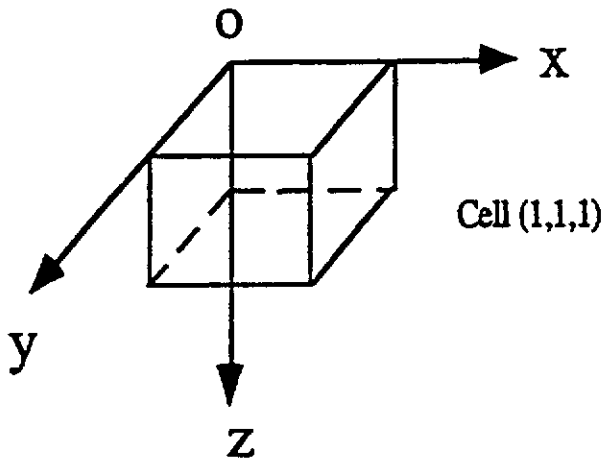
where

NLAY	is the total number of layers;
DELR	is the cell width along the row direction;
DELC	is the cell width along the column direction;
HTOP	is the top elevation of the first model layer;
DZ	is the thickness of each cell in the grid; and
HNOFLO	is the value used in MODFLOW for indicating inactive cells.

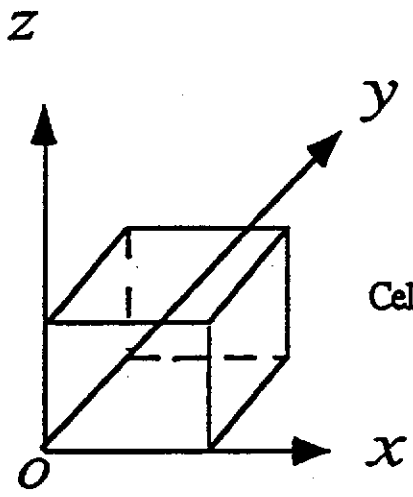
POSTMODF is an interactive program. To run it, simply type the name of its executable file. The program will prompt the user for the various input items. The user can select the heads at the desired step by specifying either the time step and stress period numbers, or the total elapsed time, whichever is more convenient. The user can also define a "window", within which the graphic data files are desired, by specifying the *starting* and *ending* column (J), row (I) and layer (K) indices of the window.

For example, to generate a data file for a cross sectional contour map along the 5th column, from row 20 to row 40 and from layer 1 to 10, enter the starting (J, I, K) indices as 5, 20, 1, and the ending (J, I, K) indices as 5, 40, 10. Similarly, to generate a data file for a cross sectional contour map along the 5th row, from column 20 to column 40 and from layer 1 to 10, enter the starting (J, I, K) indices as 20, 5, 1, and the ending indices as 40, 5, 10. Moreover, to generate a data file for a contour map on the 5th layer, from column 20 to column 40 and from row 1 to row 10, enter the starting (J, I, K) indices as 20, 1, 5 and at the lower right corner as 40, 10, 5. Finally, to generate a data file for a contour map on the water table, i.e., the cells in the uppermost active layers instead of a specific layer, from column 20 to column 40 and from row 1 to row 10, enter the starting (J, I, K) indices as 20, 1, 0 and the ending indices as 40, 10, 0.

It should be pointed out that in PATH3D, the origin of the coordinate system is set at the upper, top, left corner of cell (1, 1, 1) and the positive x, y and z coordinates are in the directions of increasing column, row, and layer indices, respectively (see Fig. A-1). However, in the output files generated by POSTMODF, the origin is transformed to the lower, bottom, left corner of cell (1, NROW, NLAY), as is customary in most graphic programs. As a result, the y and z axes used in PATH3D are in the opposite directions of those used in POSTMODF whereas the x axis is in the same direction (see Fig. A-1).



(a) The coordinate system used in PATH3D



(b) The coordinate system used in POSTMODF

Fig. A-1. Transformation of coordinates from PATH3D to POSTMODF

Therefore, if the contour map is on a layer or water table (i.e., the x-y plane), the horizontal axis of the map is along the direction of increasing column (J) indices, and the vertical axis is along the direction of decreasing row (I) indices. If the contour map is on a cross section along a row (i.e., the x-z plane), the horizontal axis of the map is along the direction of increasing column (J) indices, and the vertical axis is along the direction of decreasing layer (K) indices. If the contour map is on a cross section along a column (i.e., the y-z plane), the horizontal axis of the map is along the direction of decreasing (I) indices, and the vertical axis is along the direction of decreasing layer (K) indices. All the necessary transformations are done by POSTMODF automatically.

For output, POSTMODF writes data files in one of the two formats, referred to as the ".GRD" format and the ".DAT" format, according to the convention used by Golden Software's SURFER® graphic package. The .GRD format as listed below writes the head within a user-defined window of *regular* model mesh spacing to an output file, directly useable for generating contour maps by a contouring program such as the TOPO and SURF programs included in SURFER®. The .DAT format as listed below writes the head at each node with the nodal coordinates within the user-defined window to the output file. This format is useful for generating data files of *irregular* model mesh spacing to be used by a gridding program such as the GRID program included in SURFER® in order to be contoured.

.GRD file format (saved in free format):

```
DSAA
NX, NY, XMIN, XMAX, YMIN, YMAX, HMIN, HMAX
((HWIN(JJ,II),JJ=1,NX),II=NY,1,-1)
```

where

DSAA is the character string identifying the .GRD format;

NX is the number of nodal points in the horizontal direction of the window;

NY is the number of nodal points in the vertical direction of the window;

XMIN is the minimum nodal coordinate in the horizontal direction of the window;

XMAX is the maximum nodal coordinate in the horizontal direction of the window;

YMIN is the minimum nodal coordinate in the vertical direction of the window;

YMAX is the maximum nodal coordinate in the vertical direction of the window;

HMIN is the minimum head value within the window;

HMAX is the maximum head value within the window; and

HWIN is the simulated head within the mapping window.

.DAT file format (saved in free format):

For each active cell inside the mapping window:

X, Y, HXY

where

X is the nodal coordinate in the horizontal direction of the window;

Y is the nodal coordinate in the vertical direction of the window;

HXY is the simulated head at the node defined by (X,Y).

A.2 P3DPLOT

P3DPLOT is used to translate the default output file, P3DPLOT.DAT, generated after running PATH3D, into new files appropriate for use by graphic packages such as TOPO included in Golden Software's SURFER® graphic package.

The P3DPLOT.DAT file includes a header containing the following items saved in free format:

```
NCOL, NROW, NLAY, RX, XTRANS, RY, YTRANS, RZ, ZTRANS
```

followed by a certain number of records containing the particle number (NP), the X, Y, Z coordinates, the travel times (TIME), the column, row and layer indices (J, I, K), saved as follows in free formats:

```
NP, X, Y, Z, TIME, J, I, K
```

where RX, RY and RZ are character flags indicating how the X, Y, Z coordinates in the P3DPLOT.DAT file should be transformed for plotting purposes, and XTRANS, YTRANS, ZTRANS are constants for coordinate transformation in the x, y, and z axes, respectively. The transformation is done in the P3DPLOT program according to:

```
if RX='+'   X=X-XTRANS
      '-'   X=XTRANS-X
if RY='+'   Y=Y-YTRANS
      '-'   Y=YTRANS-Y
if RZ='+'   Z=Z-YTRANS
      '-'   Z=ZTRANS-Z
```

By default, when PATH3D generates the P3DPLOT.DAT file, it sets RX, RY and RZ to '+', '-', and '-', respectively; and XTRANS to the total length of the model grid along the rows (x axis), YTRANS to the total length of the model grid along the columns (y axis) and ZTRANS to the top elevation of cell (1,1,1). As a result, the X,Y,Z coordinates in the P3DPLOT.DAT file, which are relative to the origin at the upper, top, left corner of cell (1,1,1), are transformed to the coordinate system with the origin at the lower, bottom, left corner of cell (1, NROW, NLAY), as used by TOPO, (see Fig. A.1). If necessary, RX, RY, RZ and XTRANS, YTRANS, ZTRANS can be edited to suit a particular need.

P3DPLOT creates, after proper coordinate transformation, several output files, including:

1) PATHXYZ.DAT, which contains four columns in the following order:

```
X, Y, Z, NP
```

2) PATHXY.BLN (if NCOL>1 and NROW>1), which contains X, Y coordinates for drawing pathline segments in the xy plane (or projections in the xy plane if the flow field is three-dimensional). Each segment is proceeded by an integer N, followed by N lines of X, Y coordinates:

```
N1          there are N1 points in segment 1
X, Y        the first point in segment 1
X, Y        the second point in segment 1
...
X, Y        the N1th point in segment 1
......
NN          there are NN points in segment N
X, Y        the first point in segment N
X, Y        the second point in segment N
...
X, Y        the NNth point in segment N
```

THXZ.BLN (if NCOL>1 and NLAY>1), which contains X, Z dinates for drawing pathline segments in the xz plane (or projections in xz plane if the flow field is three-dimensional). Each segment is eeded by an integer N, followed by N lines of X, Z coordinates.

THYZ.BLN (if NROW>1 and NLAY>1), which contains Y, Z dinates for drawing pathline segments in the yz plane (or projections in yz plane if the flow field is three-dimensional). Each segment is eeded by an integer N, followed by N lines of Y, Z coordinates.

PATHXYZ.DAT can be directly used to "post" the intermediate locations cles along their pathlines on the head contour maps through the "Post" in TOPO or a similar feature in other graphic programs. Make sure the iate columns are selected. PATHXY.BLN, PATHXZ.BLN and Z.BLN can be directly used to plot the continuous pathlines or ons of pathlines on head contour maps in their respective planes through Line" feature in TOPO or a similar feature in other programs.

P3DFRONT

P3DFRONT is used to translate the default output file, P3DFRONT.DAT, d after running PATH3D, into a new file appropriate for use by graphic s such as TOPO.

The P3DFRONT.DAT file includes the same header as in the DT.DAT file described in the previous section, followed by a certain of records containing the particle number (NP), the X, Y, Z coordinates, l times (TIME), and the front number (IFRONT), saved as follows in nats:

```
NP, X, Y, Z, TIME, IFRONT
```

P3DFRONT creates, after proper coordinate transformation, an output file named FRONTXYZ.DAT, which contains five columns in the following order:

```
X, Y, Z, TIME, IFRONT
```

FRONTXYZ.DAT can be directly used to "post" the frontal positions of capture zones at selected times on the head contour maps through the "Post" feature in TOPO or a similar feature in other graphic programs,

A.4 P3DCAPT

P3DCAPT is used to translate the default output file, P3DCAPT.DAT, generated after running PATH3D, into a new file appropriate for use by graphic packages such as TOPO.

The P3DCAPT.DAT file includes the same header as in the P3DPLOT.DAT or P3DFRONT.DAT file described in the previous sections, followed by a certain number of records containing the particle number (NP), the final coordinates of the captured particles (X, Y, Z), the times at which the particles are captured (TIME), the initial coordinates of those captured particles (XI, YI, ZI), and a character string indicating the type of sinks (TYPE), saved as follows in free formats:

```
NP, X, Y, Z, TIME, XI, YI, ZI, TYPE
```

P3DFRONT creates, after proper coordinate transformation, an output file named CAPTXYZ.DAT, which contains eight columns in the following order:

```
X, Y, Z, XI, YI, ZI, TIME, TYPE
```

CAPTXYZ.DAT can also be directly used to "post" the initial and final positions of the captured particles on the head contour maps through the "Post" feature in TOPO or a similar feature in other graphic programs.