METHOD AND COMPUTER PROGRAMS TO IMPROVE PATHLINE RESOLUTION NEAR WEAK SINKS REPRESENTING WELLS IN MODFLOW AND MODPATH GROUND-WATER-FLOW SIMULATIONS

By Frederick J. Spitz

U.S. GEOLOGICAL SURVEY

Open-File Report 00-392

West Trenton, New Jersey

2001

DEPARTMENT OF THE INTERIOR

GALE A. NORTON, *Secretary*

U.S. GEOLOGICAL SURVEY

Charles G. Groat, *Director*

For additional information write to:

District Chief
U.S. Geological Survey
Mountain View Office Park
810 Bear Tavern Road
Suite 206
West Trenton, NJ 08628

Copies of this report can be purchased from:

U.S. Geological Survey
Branch of Information Services
Box 25286, Building 810
Denver, CO 80225-0286

PREFACE

This report documents computer programs to improve pathline resolution near weak sinks representing wells when using MODPATH, a particle-tracking simulation package for MODFLOW, the modular finite-difference ground-water-flow model. Information on obtaining and installing the computer programs can be found in appendix 1 of this report.

The computer programs have been tested successfully on models of hypothetical and actual ground-water systems; however, future applications of the programs could reveal problems that were not detected or anticipated in the test simulations. Users are requested to notify the U.S. Geological Survey (USGS) if problems are found. Correspondence should be sent to

U.S. Geological Survey
810 Bear Tavern Road, Suite 206
West Trenton, NJ 08628
Telephone: (609) 771-3900
Internet: http://nj.usgs.gov

CONTENTS

ILLUSTRATIONS

CONVERSION FACTORS

Multiply	By	To obtain
foot (ft)	0.3048	meter
foot per day (ft/d)	0.3048	meter per day
square foot (ft^2)	0.0929	square meter
square foot per day (ft^2/d)	0.0929	square meter per day
cubic foot (ft^3)	0.0283	cubic meter
cubic foot per second (ft^3/s)	0.0283	cubic meter per second
cubic foot per day (ft^3/d)	0.0283	cubic meter per day
million gallons per day (Mgal/d)	3785.2	cubic meter per day

ILLUSTRATIONS--Continued

TABLES

METHOD AND COMPUTER PROGRAMS TO IMPROVE PATHLINE RESOLUTION NEAR WEAK SINKS REPRESENTING WELLS IN MODFLOW AND MODPATH GROUND-WATER-FLOW SIMULATIONS

By Frederick J. Spitz

ABSTRACT

This report documents a method for converting weak sinks representing wells to strong sinks when using the particle-tracking postprocessor MODPATH for the ground-water-flow model MODFLOW. A weak sink is a model cell (representing a well, for example) that does not discharge at a sufficiently large rate to capture all of the flow entering the cell; thus, some of the flow leaves the cell across one or more of the cell faces. Because of this limitation of model discretization, flow paths to weak sink cells cannot be uniquely defined, as it is impossible to know whether a specific water particle discharges to the sink or passes through the cell. Creating a finer discretization of the cell by using the nested rediscretization method can eliminate this ambiguity by converting the cell representing the well into a strong sink. The method is presently limited to applications involving well sinks, backtracking particle endpoint analysis, and steady-state conditions. Software for applying the method consists of five Fortran programs that are operated manually. Program input is a combination of data files and interactive keyboard entry. Program output consists of a composite endpoint file. Application of the method to a sample problem demonstrates that the particle-tracking results obtained can be more accurate than those obtained by using a coarsely discretized model alone. This method is a relatively simple alternative to more involved telescopic mesh refinement approaches and does not require additional information. For certain types of problems involving identification of recharge source areas, this method can provide improved pathline resolution with modest effort. This method could be adapted for use in simulating other types of weak sinks, such as stream cells, but each type of sink boundary has unique considerations that may have to be treated differently than those described in this report.

INTRODUCTION

The nested rediscretization method is a set of computer programs designed to help eliminate weak sinks representing wells when using MODPATH (Pollock, 1994), a particle-tracking package designed for use with the U.S. Geological Survey's modular finite-difference ground-water-flow model, MODFLOW (Harbaugh and McDonald, 1996a, 1996b). Particle-tracking programs are postprocessors to ground-water-flow models because they use the cell-by-cell flow distribution calculated with the model to calculate the ground-water velocity distribution throughout the ground-water system, which then is used to determine flow paths (pathlines) of infinitely small particles. Three-dimensional water-particle flow paths are calculated in MODPATH by using the flow-field results from MODFLOW to track particles from user-defined starting locations in the model grid, cell by cell, until the particles reach a boundary, reach an internal sink/source, or satisfy some other termination criterion.

The effects of spatial grid discretization limit the level of detail and accuracy of model and particle-tracking results. For example, spatial discretization effects in MODFLOW cause the local flow field around a well to be represented insufficiently. This discrepancy is the result of a linear velocity interpolation scheme that cannot reflect that the closer flow is to a well in the center of a cell, the greater the flow velocity. Also, the well is not precisely located within the cell. Similarly, MODPATH has limitations resulting from discretization, such as those relating to weak internal sinks. A weak sink is a cell (representing a well, for example) that does not discharge at a sufficiently large rate to capture all of the flow entering the cell; thus, some of the flow leaves the cell across one or more of the cell faces (fig. 1). As a result of these conditions, particle flow paths calculated within weak sink cells cannot be uniquely defined because it is impossible to know whether a specific water particle discharges to the weak sink or passes through the cell.

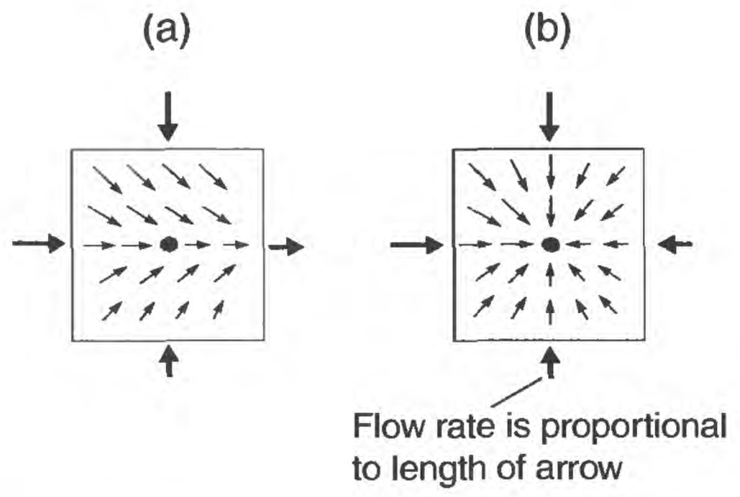


Figure 1. Model cells representing (a) a weak sink and (b) a strong sink.

In MODPATH, the problem of weak sinks is resolved by asking the user to arbitrarily decide how particles will be treated whenever they enter cells with internal sinks: either pass through a weak sink, stop at a weak sink, or stop at a weak sink where discharge to the sink is larger than a specified fraction of the total inflow to the cell. If these options are not satisfactory, the user could resolve the problem by modifying the model so that no weak sinks occur. For example, weak sinks can be converted to strong sinks by reducing the grid cell size in the vicinity of the well. This approach would require additional work and, possibly, recalibration of the model because of the effects of a finer discretization on boundary conditions. For manually prepared models, modifying the model in this manner can be a difficult process. Models prepared with graphical-user interfaces (for example, Shapiro and others, 1997) can be modified with less difficulty if the data used to generate the model are stored in a grid-independent fashion. Limitations in the amount of overall cell reduction are still possible with this approach, however.

Various approaches have been used by previous investigators to resolve the weak sink problem to improve the accuracy of pathlines in particle-tracking simulations (Spitz and others, in press). For example, the flow velocity distribution in the vicinity of the well can be refined mathematically so that all particles that should be captured by the well converge toward the well. A submodel around the weak sink could be designed by using telescopic mesh refinement, but this option may not be practical because of the amount of additional work required, especially if the weak sinks are scattered. Other complications with this approach include submodel boundaries that cut across original model boundaries, the need for recalibration in the submodel when using head-dependent sources and sinks in the original model, and the need to rerun the original model (and generate new boundary conditions for the submodel) if additional detail is incorporated into the submodel. Recently developed procedures and software involving telescopic mesh refinement with MODFLOW could assist with model rediscretizing (Leake and Claar, 1999). In this report the terms “original model” and “submodel” are used to refer to the larger, encompassing model and the smaller, embedded model, respectively. In previous studies, other terms such as “regional model” and “local model” have been used to represent the respective models.

The purpose of this study, conducted in cooperation with the New Jersey Department of Environmental Protection and begun in 1996, is to develop an approach to resolve some of the weak sink limitations of MODPATH for cells representing wells. The approach was to achieve this purpose without requiring rediscretion of the original model or modification of the original model concept by incorporating new data. The result, called the nested rediscretization method, can be useful when MODFLOW models with coarse discretization, such as regional-scale models, are used to determine capture areas of wells. Advantages of the method presented here are that it does not require additional information beyond that specified in the original MODFLOW and MODPATH input files, it is easy to use, and it provides an unambiguous determination of how particles are treated. Disadvantages of the method described in this report are that it is applicable only to weak sinks representing wells in steady-state models. If it is important to know which particle enters a sink, then more detail must be given than is provided in the model. This information could be obtained by refining the model.

This report documents the development of the nested rediscretization method, describes the computer programs and their operation, and presents an application of the method to a hypothetical sample problem. The computer programs have been tested successfully on available models of real-world flow systems that consist of surficial aquifer systems with stream boundaries and large (500-2,000 ft) grid spacing. These real-world systems include glacial valley-fill aquifers (Nicholson and others, 1996) and Coastal Plain aquifers in New Jersey (Nicholson and Watt, 1997; L.J. Kauffman, U.S. Geological Survey, written commun., 2000) and Louisiana (R.B. Winston, U.S. Geological Survey, written commun., 2000).

Special thanks are given to the following USGS colleagues for their insightful technical comments related to this work: Robert Nicholson, Daryll Pope, Anthony Navoy, and Herbert Buxton; Leon Kauffman and Richard Winston, who reviewed this report; and David Pollock, Paul Barlow, Dennis Risser, and Arthur Baehr, who reviewed related articles.

METHOD DEVELOPMENT

The nested rediscretization method involves refining the discretization of the original model at the weak sink to make it a strong sink by creating a submodel of the cell representing the well, coupling the submodel with the original model, and performing a linked MODPATH analysis with both models. A separate MODFLOW simulation is run with the submodel. Flow across the cell faces calculated in the original model are used as constant-flux boundary conditions in the submodel (fig. 2). The well in the original model is replaced by one or more constant-head cells in the submodel to allow for at least one constant head in the submodel. The actual horizontal location of the well can be specified and more than one constant-head cell is allowed in the vertical direction (only one per submodel layer) to represent the well. A constant-head cell is required to remove the constant flux of water entering through the boundaries of the submodel. Tests conducted during development of the method have shown the center cell of the submodel to be the most numerically stable position for a constant-head cell. In order to maintain a center cell at each rediscretization, only odd-numbered submodel dimensions are used. For example, the initial rediscretization is from 1 layer, 1 row, and 1 column (the original model cell) to 3 layers, 3 rows, and 3 columns (27-cell submodel); the next finer rediscretization, if necessary, is 5 layers, 5 rows, and 5 columns (125-cell submodel). The level of rediscretization also can be user-defined. Rediscretization can be continued to a total of 99 layers, 99 rows, and 99 columns (970,299-cell submodel).

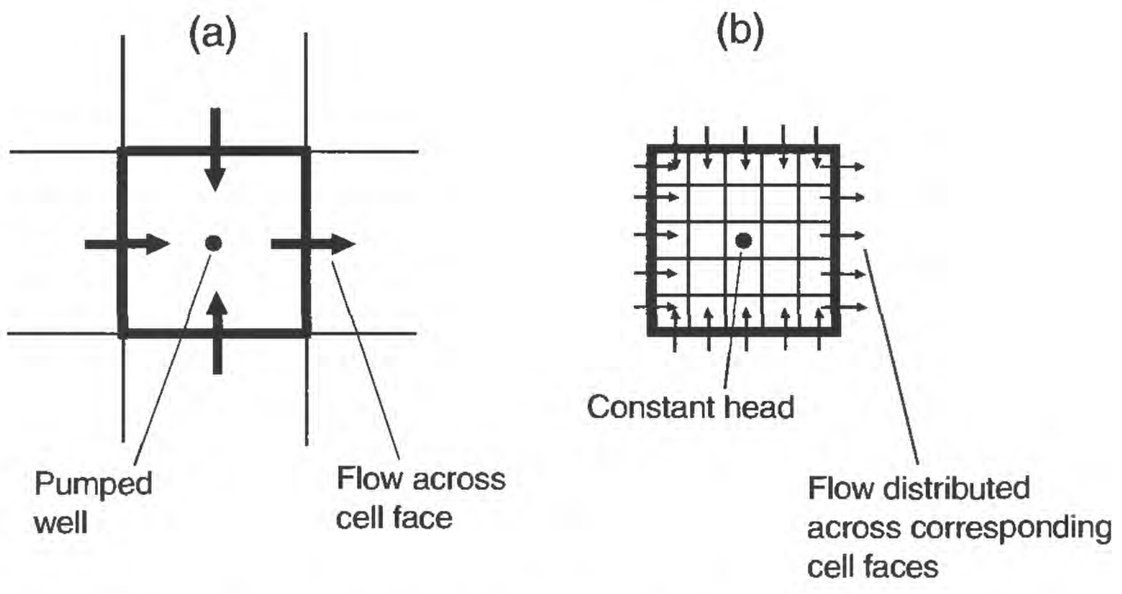


Figure 2. Plan view of model grid around cell representing pumped well in (a) the original model and (b) the submodel.

The value of hydraulic head assigned to the constant-head cell(s) also affects the stability of the submodel solution. Although assigning a value based on heads in neighboring cells of the original model would be advantageous, this value might be too low and could cause dry cells to result in the submodel. In the absence of dry cells, however, the value assigned does not affect particle flow patterns in the submodel. Accordingly, the value assigned is set equal to the simulated head in the cell in the original model.

The dimensions of the cells in the submodel are calculated from the dimensions of the cell in the original model divided by the level of rediscretization. A Well Package file (WEL) that contains the flow across boundary cell faces is created for the submodel. The flow across any particular submodel cell face is calculated from the flow across the corresponding face in the original model divided by the level of rediscretization. If the cell in the original model represents ground water under unconfined conditions, the elevation of the top of the submodel is set equal to the water-table elevation in the original model cell. If the cell in the original model is confined, the elevation of the top of the submodel is set equal to the elevation of the top of the cell in the original model cell. Hydraulic properties of cells in the submodel are derived from the properties of the original model cell. In addition, vertical anisotropy can be set for the cells in the submodel.

In applying the method for capture-area analysis, particles are started from the faces of the constant-head cell(s) of the submodel and tracked backward to the boundaries of the submodel. Backward particle tracking from the well is used instead of forward tracking from the water table to minimize the number of particles needed. The method could be modified for forward tracking, but this option is currently not available. Particles are not tracked from inside the constant-head cell(s) for the same reason, as well as to minimize pathline redundancy. Upon termination in the submodel, particles are transferred to equivalent locations in the original model and particle tracking is continued to completion (fig. 3). Particles that terminate at the top boundary of the submodel, which also is the top boundary of the original model, are not transferred. Similarly, particles that terminate at the bottom boundary of the submodel, which also is the bottom boundary of the original model, are not transferred.

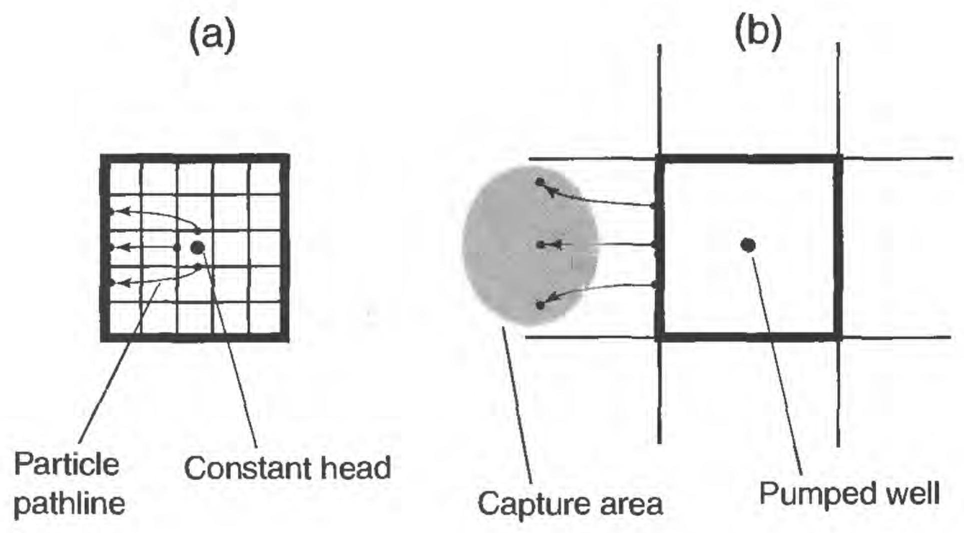


Figure 3. Plan view of particle locations and pathlines in (a) the submodel and (b) the original model.

Particle coordinate transfer in the horizontal direction is straightforward and is done by adding an offset distance (from the original model origin to the submodel origin) to the global x- and y-coordinates in the submodel. Particle coordinate transfer in the vertical direction is more complicated because of the MODPATH grid geometry code (IGRID). Vertical coordinates are handled either by using consistent global coordinates between the models (IGRID=1; true rectangular three-dimensional grid) or by converting local z-coordinates in the submodel to local z-coordinates in the original model (IGRID=0; grid with variable thickness and (or) nonhorizontal layers).

TECHNICAL CONSIDERATIONS AND LIMITATIONS

Certain aspects of the nested rediscretization method must be understood in order to correctly interpret the results obtained by using the method. These aspects are related to the (1) design of the original model, (2) discretization of the original model, and (3) application of the method.

For the first case, the effect of constant-head cells in the original model on particle pathlines should be examined to ensure that these cells are not exerting an unrealistic artificial constraint on flow volume, direction, or travel time. These effects may overshadow results obtained by using the method. Certain boundary conditions in the original model also need to be evaluated. For example, a gaining stream in the same cell as the well may falsely prevent the well in the submodel from containing any part of the capture area.

For the second case, wells commonly may be represented in the original model as if they are screened over the entire thickness of the layer, which generally is greater than the actual length of the screened interval. Rediscretization can be done or continued beyond the point at which the well is converted to a strong sink in the submodel if the thickness of the constant-head cell(s) is greater than the length of the screened interval. In the opposite case, the well may need to be screened over more than one submodel layer. (The choice of well placement in the submodel will affect the delineation of capture area and calculation of travel time.) Also, a single model cell can represent more than one well in the original model, precluding a discrete determination of the capture area of individual wells. In these instances, the total withdrawal from the group of wells is represented by an equivalent composite withdrawal extracted at the center of the cell, and provides a composite capture area. Finally, because the model location of the well and the associated capture area are always affected by the scale of the original model discretization (even in the submodel), they may not be accurate. Resolving all the limitations related to the original model discretization would require the redevelopment, and possibly recalibration, of a large part of the model with a finer horizontal and vertical discretization.

For the third case, the method is designed to be run manually for each well in the original model to verify that each submodel runs and converges to an accurate solution without error messages. The method could be modified for rediscretizing the well and adjacent cells, both horizontally and vertically, but this would require extensive computer programming. A limitation of the current method design is that boundary effects occur at the edges of capture areas determined from submodels representing a well screened over multiple layers in the original model. The method also is designed for steady-state conditions. Design for transient conditions would require substantial additional programming. Steady-state conditions provide the stable extent of the capture area and durations of particle travel time under a given set of pumping conditions. If a constant head is placed in any cell other than the center cell of the submodel, numerical problems may occur. For example, a submodel boundary cell could contain a constant head and a constant flux. The user can avoid this situation by further rediscretizing the submodel so that the constant head can be located away from the boundary. Even if no problems occur for this case, submodel results should be evaluated to determine whether results are reasonable. When the method is applied to very weak sinks, the iterative effort required to achieve the necessary rediscretization can be substantial, whereas the capture area that is defined can be very small. A practical solution may be to establish a minimum withdrawal for applying the method to a specific model or simply to backtrack particles from the exact location of the well screen in the original model.

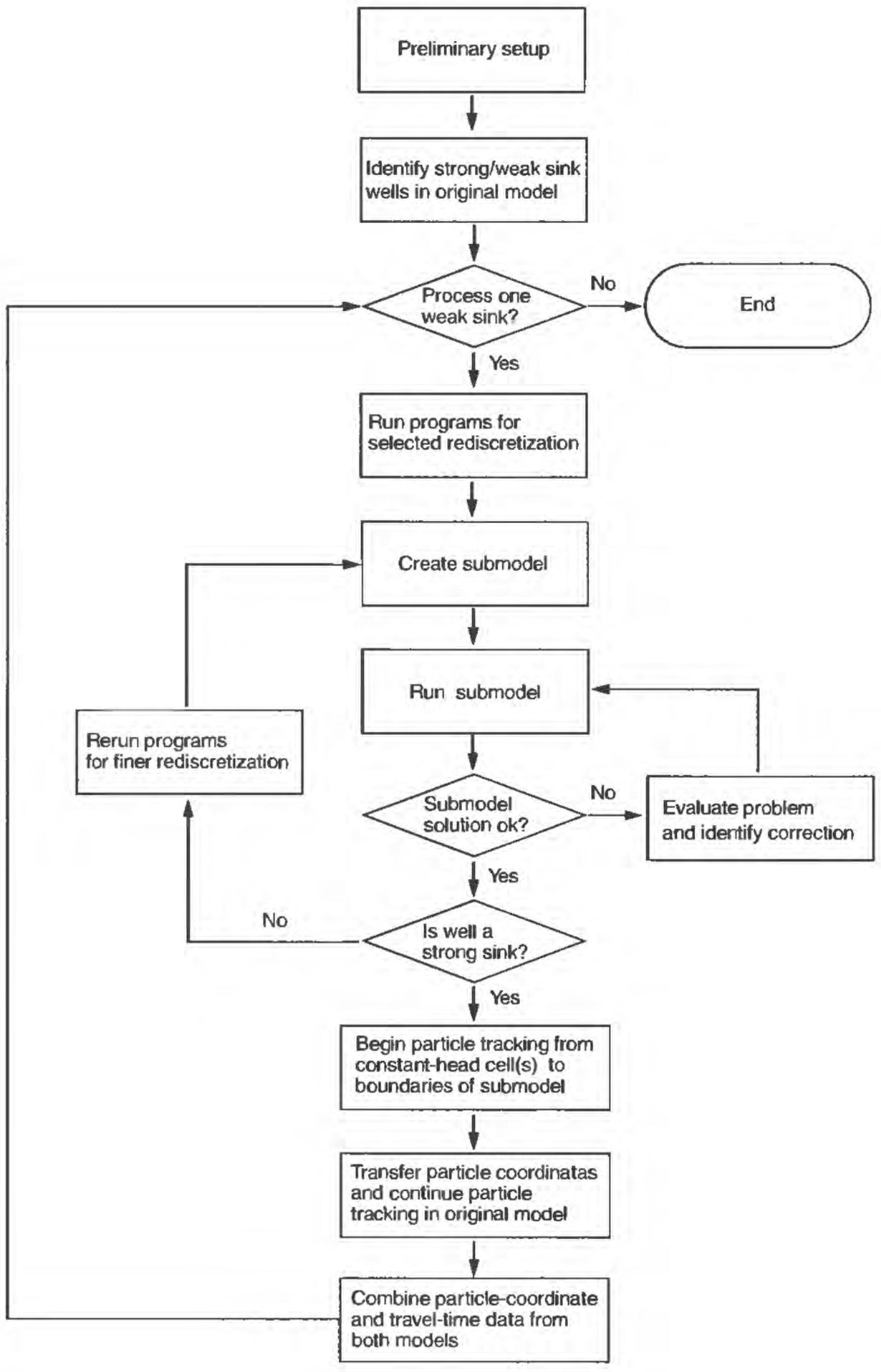
Also for the third case, the potential for well interference must be evaluated. For example, consider two weak sink wells, one upgradient from the other. If it is specified that particles stop at weak sinks in the model domain, the size of capture area of the downgradient weak sink will be underestimated. If particles are allowed to pass through other weak sinks (as is specified for the nested rediscretization method), overlapping capture areas are possible. To resolve the overlap case, the overlapping capture area of the upgradient well can be subtracted from the capture area of the downgradient well, so long as the capture area of the upgradient well does not lie outside the capture area of the downgradient well.

COMPUTER PROGRAMS

The software consists of five Fortran programs that are operated manually. The programs have been compiled and tested on a UNIX mainframe computer and compiled only on a Windows NT personal computer. Not all possible MODFLOW and MODPATH applications have been tested with the computer programs. Future applications of the programs may lead to problems requiring modification and recompilation of the source codes. Instructions on obtaining and installing the computer programs and associated files are given in appendix 1. The associated files include two generic MODPATH response files (a special output file that records interactive input from the keyboard). The names of the programs and associated files are shown in italics throughout this report.

Organization and Structure

The determination of capture areas of wells by using particle tracking and the nested rediscretization method is shown schematically in figure 4. The first step is to identify weak and



Preliminary setup
Identify strong/weak sink wells in original model
Process one weak sink?
No
End
Yes
Run programs for selected rediscretization
Create submodel
Run submodel
Rerun programs for finer rediscretization
Submodel solution ok?
No
Evaluate problem and identify correction
Yes
No
Is well a strong sink?
Yes
Begin particle tracking from constant-head cell(s) to boundaries of submodel
Transfer particle coordinatas and continue particle tracking in original model
Combine particle-coordinate and travel-time data from both models

Figure 4. Nested rediscretization method for determining capture areas of wells.

strong sinks in the particular model domain. If particle tracking is desired at this point for either type of well, MODPATH then can be run. If use of the nested rediscretization method is chosen, it can be run as described below.

The default rediscretization of the cell containing the weak sink well is 3 layers, 3 rows, and 3 columns (27-cell submodel). (For each rediscretization, the flow to the constant-head cell(s) representing the well in the submodel is equal to the discharge rate of the well in the original model.) Input files and particle-tracking files are created for the submodel. The number of particles tracked is set by the user in *fine.rsp*. The submodel then is run and the model solution is checked for potential problems and errors. If these discrepancies occur, adjustments (for example, changes to model solver parameters) may be necessary to the submodel and (or) original model. If adjustments can be made or if no problems occur, the well in the submodel then is checked for sink strength. If the well is still a weak sink, modifications are made for a finer submodel rediscretization and the previous steps are repeated until the well becomes a strong sink. When this condition is met, particles are started from the well in the submodel. Then, depending on whether particles are transferred back to the original model (by converting particle ending locations in the submodel to corresponding particle starting locations in the original model), particle tracking is continued to completion. If particles are separated into two endpoint files, one for each model, a composite endpoint file is created that includes cumulative travel times from recharge locations to discharge at the well. See the MODPATH documention (Pollock, 1994) for more information about the input starting locations file and output particle-coordinate endpoint file.

Fortran Code

The program *identify.f* identifies strong and weak sink wells in the original model by (1) finding the packages that are used in the model, (2) reading the model binary (unformatted) cell-by-cell flow file, (3) separating the weak from the strong sinks, and (4) creating files listing the weak and strong sinks. The program is dimensioned for a model with 300 rows, 300 columns, and 100 layers simulating 1,000 wells.

The program *rediscretize.f* makes use of an original, coarsely discretized MODFLOW model to create input files for a finely discretized submodel of a weak sink cell by (1) finding the packages that are used in the model, (2) reading the model cell-by-cell flow and head files, and (3) creating a MODFLOW name file and Basic, Block-Centered Flow, Well, Output Control Option, and Solver Packages for the submodel. The program then (4) reformats the original model MODPATH name and main files for the submodel. The program is dimensioned for a model with 300 rows, 300 columns, and 100 layers, and a level of rediscretization of 99 layers, 99 rows, and 99 columns.

The program *transfer.f* converts MODPATH endpoint locations resulting from particle tracking in the submodel to starting locations for particle tracking in the original model. This is accomplished by (1) finding the packages that are used in the model, (2) determining the global coordinates of the submodel origin, (3) converting coordinates of particles in the submodel to coordinates in the original model, and (4) creating files listing particles that terminate in the submodel and (or) particles to be continued in the original model. The program is dimensioned for a model with 300 rows, 300 columns, and 100 layers, a level of rediscretization of 99 layers, 99 rows, and 99 columns, and 50,000 tracked particles.

The program *combine.f* combines MODPATH endpoint locations from the submodel with endpoint locations from the original model to create a composite endpoint file. Tracking time for particles transferred from the submodel to the original model is accumulated to provide a total tracking time for these particles. The program is dimensioned for 50,000 tracked particles.

The utility program *utility.f* performs various tasks, including checking the submodel solution and particle-tracking results as well as evaluating the strength of the sink in the submodel. The strength of the sink is determined by checking the cell-by-cell flow file for the constant-head cell(s) for outflow across cell faces. Once the weak sink has been converted to a strong sink, the program interactively prorates the number of particles to be tracked from each inflow face of the cell on the basis of the cell water budget. When the program is executed, the user is queried for which task to perform. Output from the program is to the computer monitor only. The program is dimensioned for a model simulating 1,000 wells and a level of rediscretization of 99 layers, 99 rows, and 99 columns.

Input and Output Files

The names of input and output files for the Fortran programs are shown in table 1. The prefix enclosed in brackets is unique to the user's original model. The original model files used as input files to the programs must exist in order for the programs to operate.

Program Compatibility

The computer programs developed for the nested rediscretization method were designed for the versions of MODFLOW and MODPATH that were available at the time of method development. The programs are neither backward compatible with earlier versions of these codes nor forward compatible with updates to these codes, such as MODFLOW-2000 (Harbaugh and others, 2000). Therefore, model and particle-tracking files may need to be modified to be compatible with these versions of the codes. The programs do not accommodate working with cross-sectional models. Such applications may lead to program termination or errors that require modification of the programs. Additionally, MODFLOW and MODPATH were not executed in batch mode; batch files for these codes will interfere with the procedure for the method.

The computer programs can accommodate free- or fixed-format model input. The programs are designed for working with standard and not compact MODFLOW binary budget and head files. Not all MODFLOW packages have been tested with the programs. The Drain (DRN), River (RIV), Recharge (RCH), and Strongly Implicit Procedure (SIP) Packages were tested. The Block-Centered Flow (BCF) and General-Head Boundary (GHB) Packages were tested partially in terms of parameters. The Evapotranspiration (EVT), Stream (STR), Reservoir (RES), Slice Successive Overrelaxation (SOR), and Preconditioned Conjugate Gradient (PCG) Packages are accounted for in the computer programs, but were not tested. The programs are designed for working with standard (not compact or binary) MODPATH endpoint files; they are not designed for working with pathline or time-series files. The MODFLOW name file must be compatible with the MODPATH name file as well as with the MODPATH main data file.

Operation

For detailed instructions on using MODFLOW, refer to Harbaugh and McDonald (1996a, 1996b). For detailed instructions on using MODPATH, refer to Pollock (1994). User input for the computer programs is a combination of data files and interactive keyboard entries in response to menu queries. Data output (MODPATH particle endpoint file) can be used, for example, to delineate capture areas as part of a wellhead-protection program in the model domain by using MODPATH-PLOT (Pollock, 1994) or a geographic-information-systems-

Table 1. Computer program input and output files

Computer program	Input file	Output file	File description
identify.f	*[model].pnm*		Name file[1]
	[model].bud		Binary cell-by-cell flow file[2]
		strongwells	Strong sink wells
		weakwells	Weak sink wells
rediscretize.f	*[model].fnm*		Name file[3]
	[model].bas		Basic Package[3]
	[model].bcf		Block-Centered Flow Package[3]
	[model].slv		Solver Package[3]
	[model].ghb		General-Head Boundary Package[3]
	[model].data		Ancillary data file[1,3]
	[model].bud		Binary cell-by-cell flow file[2]
	[model].bhd		Binary head file[2]
	[model].pnm		Name file[1]
	[model].mdf		Main data file[1]
		fine.fnm	Name file[3]
		fine.bas	Basic Package[3]
		fine.oc	Output Control Option Package[3]
		fine.bcf	Block-Centered Flow Package[3]
		fine.slv	Solver Package[3]
		fine.wel	Well Package[3]
		fine.pnm	Name file[1]
		fine.mdf	Main data file[1]
		fine.ept	Dummy endpoint file[4]
		coarse.ept	Dummy endpoint file[4]
transfer.f	*[model].bhd*		Binary head file[2]
	[model].pnm		Name file[1]
	[model].mdf		Main data file[1]
	fine.ept		Endpoint file[4]
		separ.ept	Endpoint file[4]
		coarse.stl	Starting locations file[1]
combine.f	*coarse.stl*		Starting locations file[1]
	separ.ept		Endpoint file[4]
	coarse.ept		Endpoint file[4]
		outpoint	Composite endpoint file[4]
utility.f	*fine.lst*		Summary output file[2]
	summary.pth		Summary output file[4]
	fine.bud		Binary cell-by-cell flow file[2]

[1] MODPATH input file
[2] MODFLOW output file
[3] MODFLOW input file (*[model].ghb* used only if included)
[4] MODPATH output file

based plotting program such as MODTOOLS (Orzol, 1997). Interactive menu responses, input files, and output files for the sample problem presented farther on in this report are listed in appendixes 2, 3, and 4, respectively.

Five Fortran programs and two MODPATH response files must be used to follow the procedure shown in figure 4. The response files, *fine.rsp* and *coarse.rsp* (references the name file *coarse.pnm*), are for the submodel and original model, respectively. To create *coarse.pnm*, make a copy of the original model MODPATH name file and edit the line containing the keyword "ENDPOINT" in the copied file so that the associated file name is *coarse.ept*.

Setup Instructions

This section provides the necessary setup information for using the computer programs. Certain requirements must be met for the proper operation of the programs and apply mainly to the associated MODFLOW and MODPATH files. For example, the MODFLOW output binary cell-by-cell flow data must be written to a file. MODFLOW output binary head data also must be written to a file for a well screened in any layer containing the water table.

Special input files called name files are used in MODFLOW and MODPATH. Name files specify the data files being used in a model or particle-tracking simulation and provide the information needed to manage these files. Each line in the name file must contain three items for each file used in the simulation: a character string keyword signifying the type of data file, a unique integer Fortran unit number to be used when reading from or writing to the file, and a file name. Lines in the MODPATH name file containing keywords not accounted for by the computer programs need to be converted into comment lines by adding the "#" character to column 1 of the file (for example, "PATHLINE," "TIME-SERIES," "TIME," "LOCATIONS," and so on). Full pathnames are necessary to access files located in other computer directories or folders. Because binary files are opened in the programs by using the MODPATH name file, any ancillary MODFLOW binary files must be specified in this name file. (The DATA (BINARY) keyword entry will be ignored by MODPATH.) If drawdown output is desired, the user must interactively add entries for drawdown files to the appropriate submodel name files. The STR and RES Packages are currently not accounted for in MODPATH, but entries are still required in the MODPATH name file if these packages are used in the model.

Although MODFLOW and MODPATH are designed to work independently, operation of the computer programs described in this report requires the Fortran unit numbers in the respective name files to be consistent; therefore, editing of the model name files may be necessary. Associated data-file internal unit numbers also may need to be adjusted. Fortran unit numbers from 79 to 99 are reserved for use by the computer programs. Fortran unit 77 should be connected to the MODFLOW GHB Package, if used, and unit 78 should be connected to the MODPATH main data file.

When the computer programs are used with MODFLOW, solution problems may occur in the submodel when the SIP Package is used. The program *rediscretize.f* includes parameter adjustments to enhance solution stability, including adjustments to the maximum number of iterations (MXITER=150), head-change convergence criterion ($\mathrm{HCLOSE}=\mathrm{HCLOSE}_{input}/10$), and iteration parameter seed flag (IPCALC=1). If problems still occur with the package, however, the parameter HCLOSE in the submodel or the iteration parameter seed (WSEED) in the original model could be adjusted manually and the model could be rerun. Alternatively, the PCG or SOR Package could be used instead of the SIP Package for the submodel.

In MODPATH, individual flow terms for model stress packages can be assigned as distributed internal sources and sinks or assigned to specific faces of the cell by using the variables IFACE/ITOP (Pollock, 1994, p. 3-8) . When the computer programs are used with MODPATH, stress-package data files should be edited so that only wells are assigned as internal sinks. Other model stresses representing sinks should be assigned to specific faces of the cell. This modification is particularly important when the WEL Package also is used to simulate other model flows, such as boundary flows. If the GHB Package is used, the IFACE value is used to determine which cell face transmits the cell-by-cell flow.

The total number of particles tracked and their starting array locations at the constant-head cell(s) representing the well in the submodel should be set by the user in *fine.rsp*. This file is designed for one submodel layer to represent the screened interval; if multiple layers are needed, this file will need to be modified. Starting locations for particles can be generated from an existing data file by MODPATH or, as in the case of the sample problem, prorated internally from inflow cell faces by using *utility.f*. Prorating means that the particles are distributed on the cell faces according to the percentage of total inflow contributed by each face. Prorating is the preferred choice because using an equal distribution of particles on all cell faces may result in a bias toward longer travel times. Prorating is done interactively in *utility.f* once the weak sink has been converted to a strong sink (there is no outflow across cell faces). The user is queried for an approximate total number of particles to be tracked and the program determines the distribution of the particles on the cell faces. No particles should be used on a face with zero flow or a face adjacent to another constant-head cell. (Because the particles are distributed in square arrays on the cell faces, the actual total number of particles tracked may differ from the number entered by the user.)

Procedure

This section provides the steps that must be followed to run the computer programs. The setup instructions given in the last section must be completed before proceeding with these instructions. Additional details on the procedure can be found in other sections of this report. For example, all of the output files from the programs can be found in table 1.

Identify weak sink wells and strong sink wells in the original model:

Run *identify.f*. This program will query the user for the dimensions of the original model and the name of the MODPATH input name file. This program creates *weakwells* and *strongwells*, which list these wells.

Repeat the following steps for each selected weak sink listed in *weakwells*:

● Create and run submodel for one weak sink

(1) Run *rediscretize.f*. This program will query the user for the dimensions of the original model, the level of rediscretization (for example, parameter NRZ=3 is for a submodel with 3 layer, 3 rows, and 3 columns (27 total cells), NRZ=5 is for a submodel with 5 layers, 5 rows, and 5 columns (125 total cells), and so on), and vertical anisotropy. The program also queries the user for the name of the MODFLOW input name file for the original model; the name of the MODPATH input name file for the original model; layer, row, and column indices for the cell containing the weak sink well in the original model; and the top layer, bottom layer, row, and column

indices of the constant head(s) in the submodel (for example, for NRZ=3, the center cell would be 2,2,2,2).

(2) Run MODFLOW for the submodel. MODFLOW will query the user for the name of the input name file for the submodel. (Reply with *fine.fnm.*)

(3) Does the submodel solution converge? Does the volumetric budget for the submodel indicate that outflow to the constant-head cell(s) representing the well approximately equals the withdrawal in the original model? (If not, then submodel solver parameters could be made more stringent.) Run *utility.f* (option 1) to identify other common problems. If problems occur, evaluate and correct them; then repeat steps 2 and 3.

(4) Run *utility.f* (option 2) to evaluate whether the weak sink has been converted to a strong sink. This option will query the user for the level of rediscretization. If the well is still a weak sink, then repeat steps 1 through 4 using a finer level of rediscretization until the well becomes a strong sink. (At the point at which the well becomes a strong sink, *utility.f* will also query the user for the approximate total number of particles to be tracked.)

● Begin particle tracking in the submodel

(1) Edit the submodel layer, row, and column indices of the constant head(s) in *fine.rsp*. Also, edit the distribution of particles per cell face in this file on the basis of prorating output from *utility.f*. Run MODPATH to begin particle tracking in the submodel. MODPATH will query the user for the name of the submodel input response file. (Reply with *fine.rsp.*)

(2) Run *utility.f* (option 3) to identify common particle-tracking problems. If problems occur, evaluate and correct them; then repeat steps 1 and 2.

● Transfer particles and continue tracking in the original model

(1) Run *transfer.f* to convert appropriate particle endpoint locations in the submodel into corresponding particle starting locations in the original model. This program will query the user for the dimensions of the original model, the number of particles tracked, the level of rediscretization, and the layer, row, and column indices of the model cell containing the weak sink. This program creates *coarse.stl*, which is the file of particle starting locations.

(2) Skip the remaining steps if all of the particles terminate in the submodel domain--that is, *coarse.stl* is empty--or the final step if no particles terminate in the submodel domain--that is, *separ.ept* is empty. For the former case, rename *fine.ept* to *outpoint*; for the latter case, rename *coarse.ept* to *outpoint* after step 4.

(3) Run MODPATH to continue the particle tracking, which is now done in the original model. MODPATH will query the user for the name of the original model input response file. (Reply with *coarse.rsp.*) As a result of simulation in the submodel, a few particles in *coarse.stl* may not have valid starting locations in the original model; these particles will be discarded.

(4) Run *utility.f* (option 3) to identify common particle-tracking problems. If problems occur, evaluate and correct them, then repeat steps 3 and 4.

● Combine endpoint files

Run *combine.f* to merge the endpoint file from the submodel with the endpoint file from the original model. This program will query the user for the dimensions of the original model and the number of particles tracked. The program creates *outpoint*, which is a composite endpoint file of particle data listed in original model coordinates. (Rename *outpoint* to a new file for each well in *weakwells* to avoid overwriting during repeat processing.)

Error Checking

The user should review computer program output any time a composite endpoint file is not created properly. The utility program *utility.f* (options 1 and 3) checks MODFLOW and MODPATH results for common problems during processing. The program does this by searching for key phrases in the files *fine.lst* or *summary.pth*. As examples, the submodel should not have variable-head cells that are converted to dry cells, solution convergence problems (or other run-time errors), or water-budget errors. If these problems occur, adjustments in the original model and (or) submodel may be necessary. Not all the key phrases indicate problems; some merely bring information to the attention of the user.

The Fortran programs were designed to include other error and warning checks. Error checks are included for inappropriate responses to program queries, the existence of input files, problems opening files, and the necessary sizing of arrays. Warning messages are included when the well is located near certain types of boundary cells. Descriptive information also is output to the computer monitor during execution of each program for debugging purposes. As discussed above, the programs were designed for working with certain MODFLOW packages and files. Packages not accounted for or incomplete cell-by-cell flow files will cause errors when the programs are used. User comment statements added to data files also may cause errors when the programs are run.

SAMPLE PROBLEM

A simple hypothetical problem is presented below to illustrate the use of the computer programs. The problem is identical to the hypothetical steady-state flow problem designed by Pollock (1994, p. 6-2), except for the pumping rate and location. Variations of the problem were used during the development and testing of the computer programs.

Description

The flow system consists of two aquifers separated by a confining layer (fig. 5). The lower aquifer is 200 ft thick, the confining layer is 20 ft thick, and the thickness of the upper aquifer varies according to the water-table elevation. Recharge to the system is uniformly distributed over the water table at a rate of 0.0045 ft/d. Discharge from the system is to a partially penetrating river along the right side of the flow system in the upper aquifer and to a well screened in the upper aquifer.

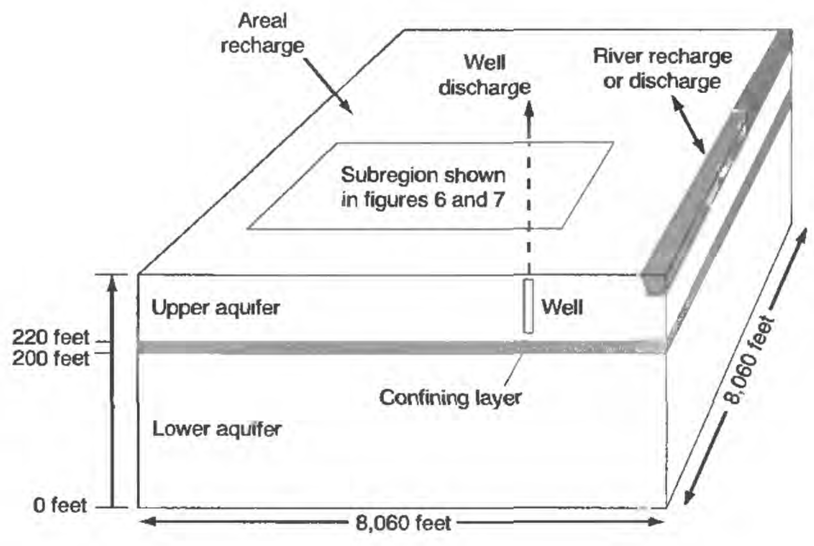


Figure 5. Diagram of simulated hypothetical flow system. (Modified from Pollock, 1994, p. 6-2)

27 columns, and 5 layers. Horizontal grid spacing varies from 40-ft by 40-ft cells to 400-ft by 400-ft cells. The upper aquifer is represented with one model layer. A quasi-three-dimensional representation is used for the confining layer; that is, only vertical flow is simulated in the layer. The lower aquifer is represented by four 50-ft-thick layers. The well is located in layer 1, row 20, column 20, and is discharging at 3,000 ft^3/d (0.023 Mgal/d). The input data files for the original model and response files for particle tracking for this problem are provided in appendix 3. By using program *identify.f*, it can be determined that the well captures only 35 percent of the flow into the cell, and thus is a weak sink.

Analysis of Capture Areas and Pathlines

The capture area of the well is the part of the flow system that contributes water to the well. The capture area of the flow system described above is at the water table and was determined for the well as a weak sink and after the well was converted to a strong sink by using the nested rediscretization method. These areas are shown in figures 6 and 7, respectively. These figures were created by using backward tracking in MODPATH-PLOT. The rediscretization required to make the well a strong sink is 7 layers, 7 rows, and 7 columns (343-cell submodel). The number of particles tracked was prorated from cell faces (7,983 total particles). The input data files for the submodel and particle tracking (excluding particle endpoint files) are provided in appendix 3.

Because the well is represented in the original model as a weak sink, some of the endpoints shown in figure 6 correspond to particles that discharge downgradient from the well rather than to the well itself; thus, the extent of the capture area is overestimated. The net amount of recharge entering the flow system over this area is approximately 8,600 ft^3/d (0.065 Mgal/d), which exceeds the pumping rate by 185 percent. When the well is converted to a strong sink, the capture area is smaller. This capture area, shown in figure 7, is 35 percent of the area defined when the well was a weak sink as determined from output in *weakwells*. The net amount of recharge entering the flow system over this area now equals the pumping rate of 3,000 ft^3/d (0.023 Mgal/d).

The distribution of particle travel times also changes when the well is converted from a weak sink to a strong sink, as shown in table 2. The table shows the maximum travel time for the selected subgroups of particles. After the conversion, particles follow longer flow paths as a result of additional travel time in the submodel.

Table 2. Maximum travel times to a weak sink well with and without using the nested rediscretization method
[Travel times are in years]

Rediscretization level	Percentage of particles				
	20	40	60	80	100
1,1,1 (original model)	1.98	6.25	11.7	18.8	28.9
7,7,7 (submodel)	4.20	8.09	12.7	18.7	28.8

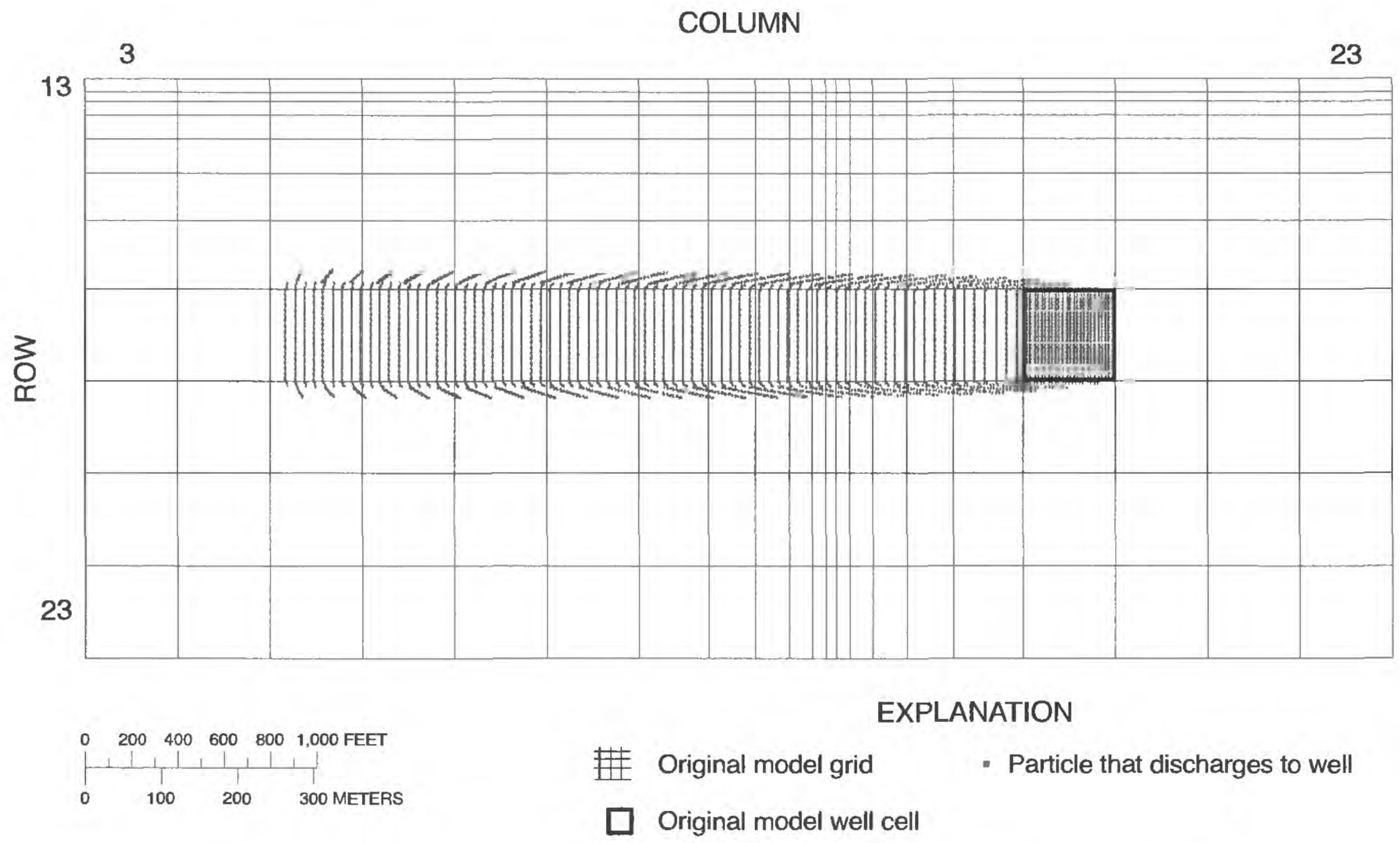


Figure 6. Map view of capture area of the weak sink well in the original model determined with MODPATH. (Location of subregion shown in figure 5.)

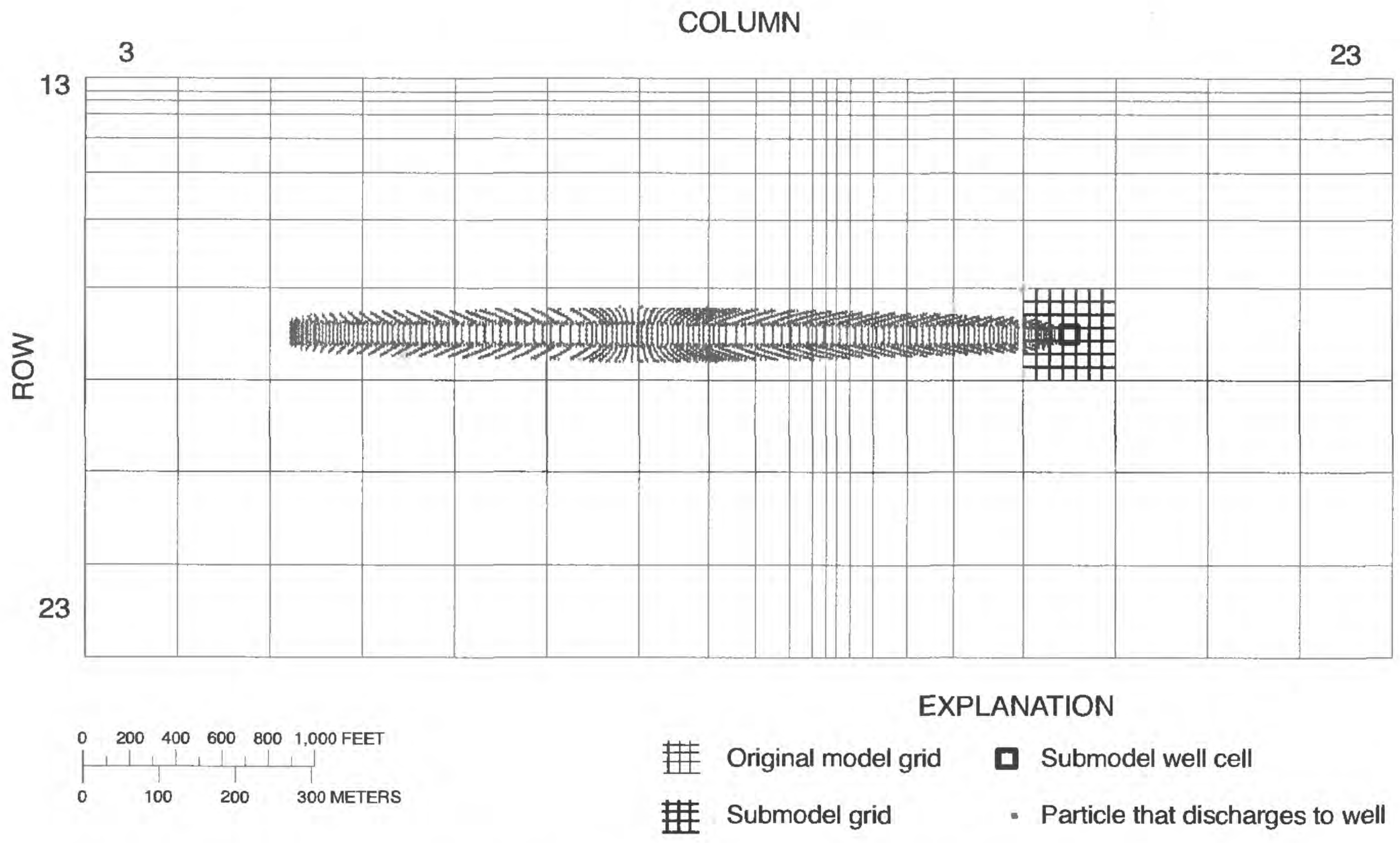


Figure 7. Map view of capture area of the strong sink well in the original model determined with the nested rediscretization method. (Location of subregion shown in figure 5.)

This difference between the original model and submodel decreases to zero as travel time increases.

Some of the changes to the flow system within the cell when the well is converted from a weak sink to a strong sink are shown in figures 8-11. Particle pathlines and time points in the original model are shown in figures 8 and 9; similar information for the submodel is shown in figures 10 and 11. These figures were produced by using forward tracking in MODPATH-PLOT. Flow direction in figures 8-11 (as well as in figures 12 and 13 farther on) is from left to right. Pathlines shown in the figures are the result of judicious selection of particle starting locations at the edge of the cell and do not represent streamlines, which are curves tangent to the flow-velocity vector at every node in the discretized flow field. Time points are discrete points along a pathline that delineate equal intervals of time. The difference in vertical scale between figures 9 and 11 is the result of the choice of starting head in the submodel (that is, the head in the submodel does not correspond to the head in the original model).

When the well is a weak sink (figs. 8 and 9), most of the flow enters the cell at the left and exits at the right. The 35 percent of the flow that is captured by the well is evident in the deceleration of flow in the original model, as shown by the slight decrease in distance between the time points on the pathlines (that is, particles travel a shorter distance in a given time). When the well is converted to a strong sink (figs. 10 and 11), some of the flow that enters the submodel is diverted to the well and accelerates, as shown by the curve in the pathlines and the increase in distance between the time points on the pathlines near the center of the submodel. Simulated heads and flow-velocity vectors in the submodel layers and profiles (not shown) display a horizontal and vertical gradient toward the center cell representing the well in the submodel.

The vertical transfer of particles from the submodel to the original model also can be demonstrated with the sample problem. To best show this transfer, the submodel rediscretization is increased to 15 layers, 15 rows, and 15 columns (3,375-cell submodel); the well discharge is reduced to 800 ft^3/d (0.006 Mgal/d); and the number of particles is reduced to 400. Pathlines for particles started at the center cell of the submodel and backtracked to the submodel-boundary are shown in figure 12; pathlines for particles started at the boundary of the cell representing the well in the original model and backtracked to the water table are shown in figure 13. The particle starting locations in the original model match the transferred particle endpoint locations in the submodel.

The results presented show that capture areas determined by using particle tracking and the nested rediscretization method are smaller than those determined by using particle tracking alone because the method refines the ground-water flow field near the well. Some of the particle pathlines now converge in the cell representing the well, rather than diverging out of the cell. Particle travel times are greater when the method is used because of additional travel time within the submodel. When the method is applied to a well that is already a strong sink (Spitz and others, in press), the capture area remains approximately the same, particle travel times are slightly greater because flow paths are extended in the submodel, and the cone of depression is smoothed. The cone of depression will be exaggerated if the submodel is not created in the water-table layer in the original model because of the assumption that inflow is evenly distributed along the top and bottom faces of the submodel.

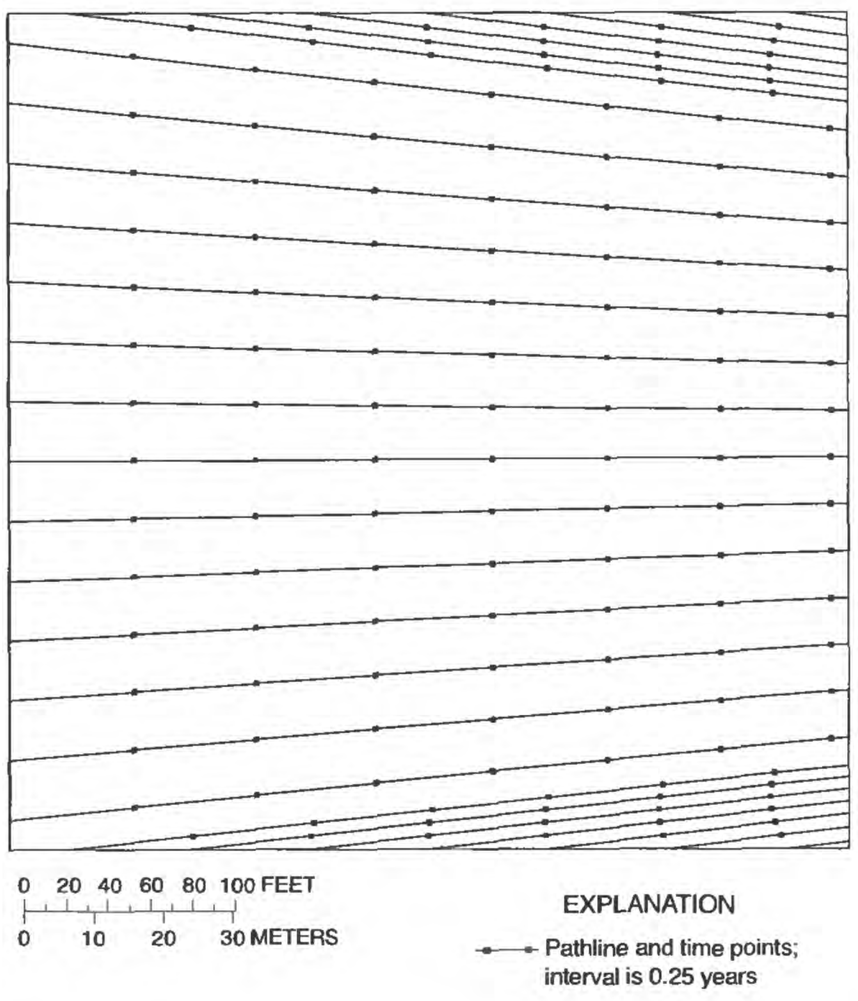


Figure 8. Map view of pathlines inside the weak sink cell in the original model determined with MODPATH.

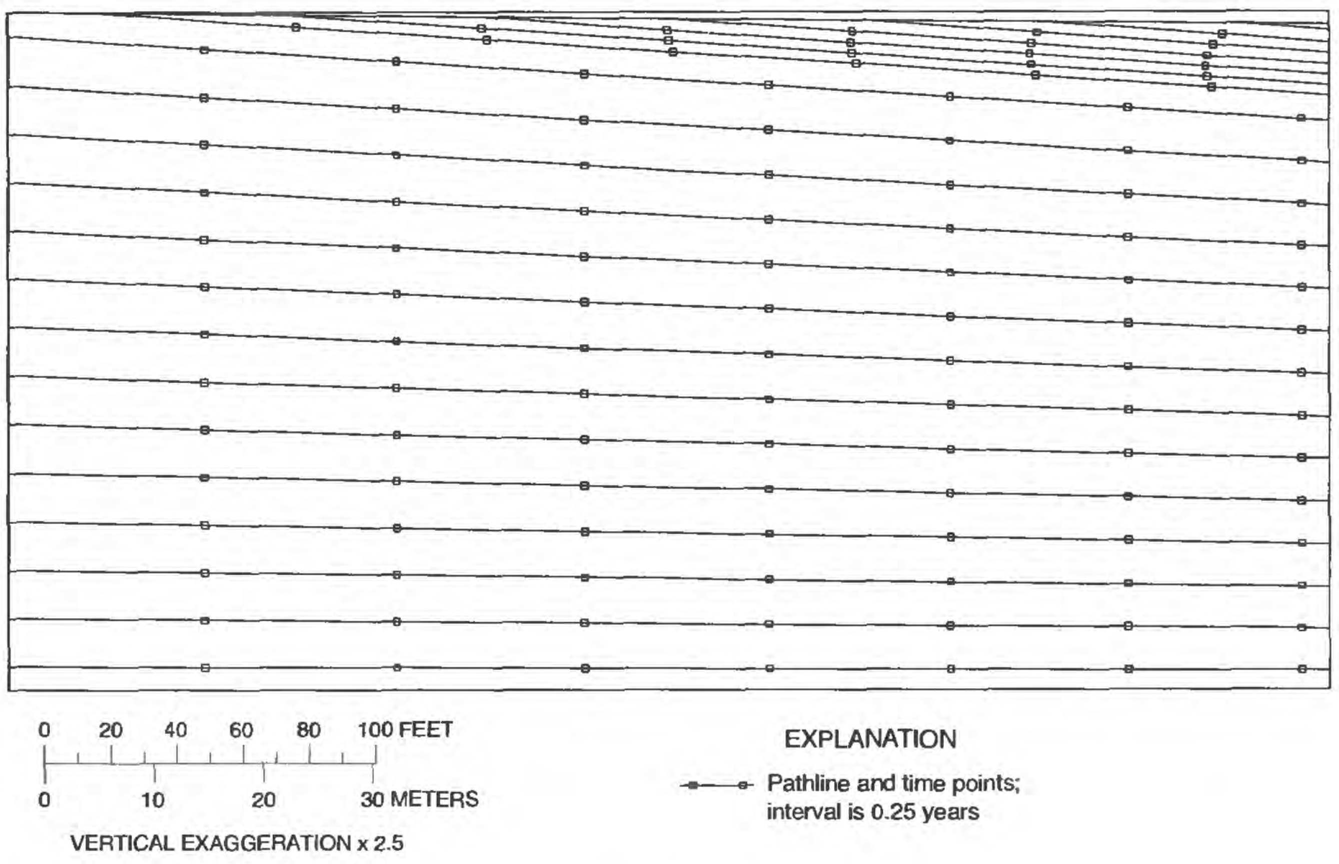

0 20 40 60 80 100 FEET
0 10 20 30 METERS
VERTICAL EXAGGERATION x 2.5
EXPLANATION
Pathline and time points; interval is 0.25 years

Figure 9. Profile view of pathlines inside the weak sink cell in the original model determined with MODPATH.

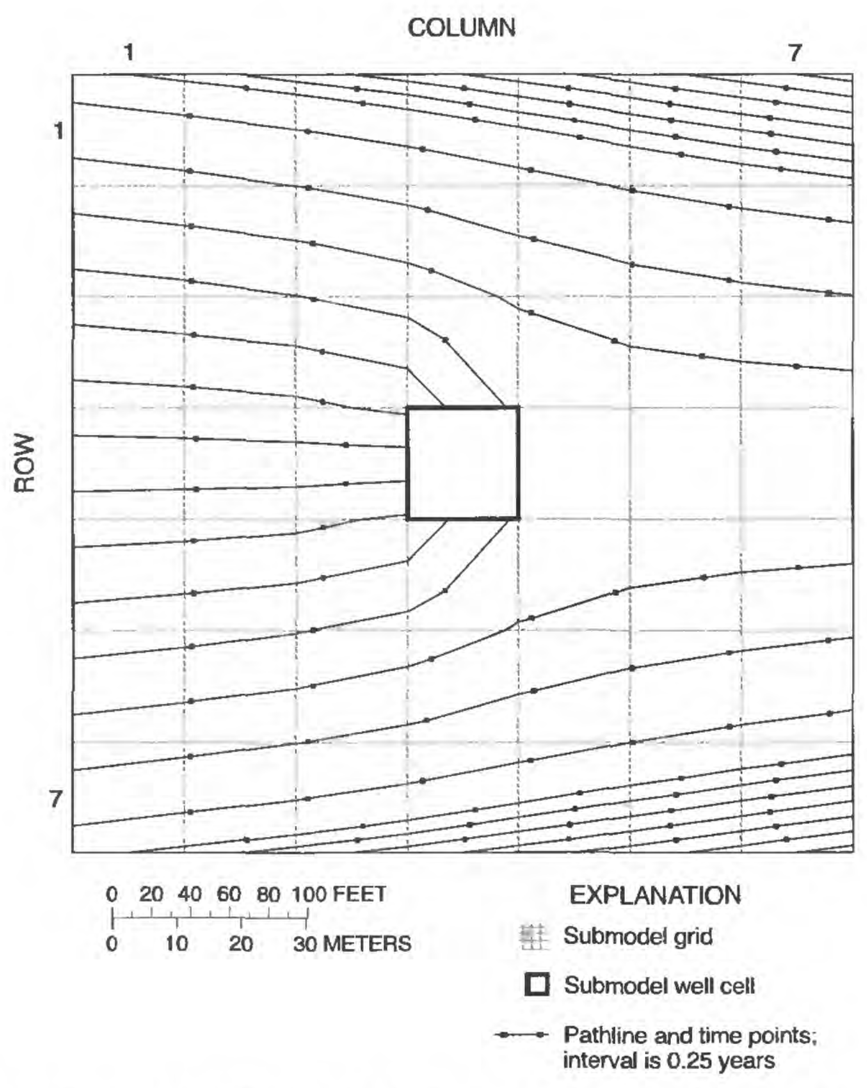


Figure 10. Map view of pathlines inside the submodel determined with the nested rediscretization method. (The view is of layer 4.)

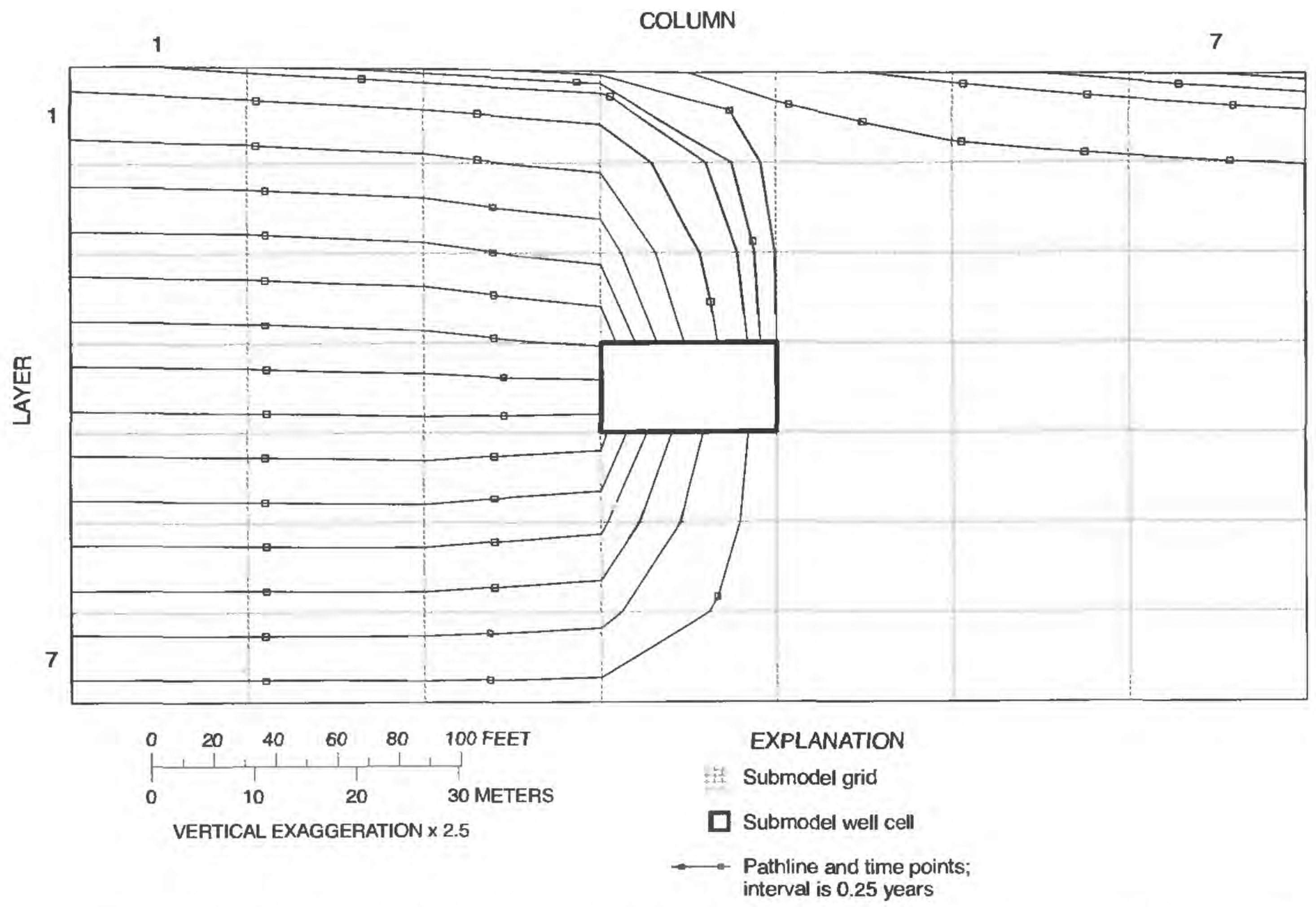


Figure 11. Profile view of pathlines inside the submodel determined with the nested rediscretization method. (The view is along row 4.)

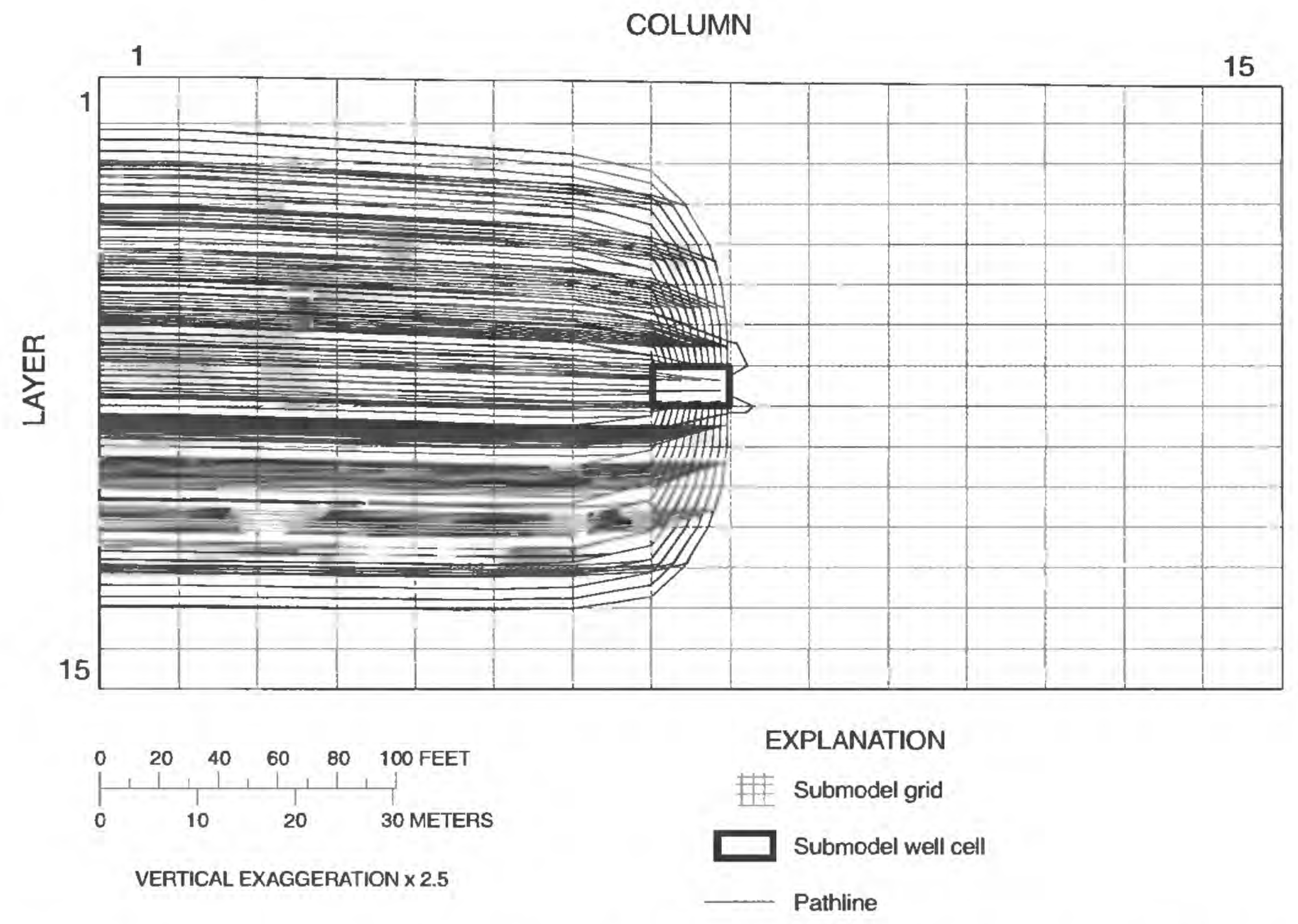


Figure 12. Profile view of pathlines inside the submodel showing endpoint locations used in particle transfer to the original model. (The view is along row 8.)

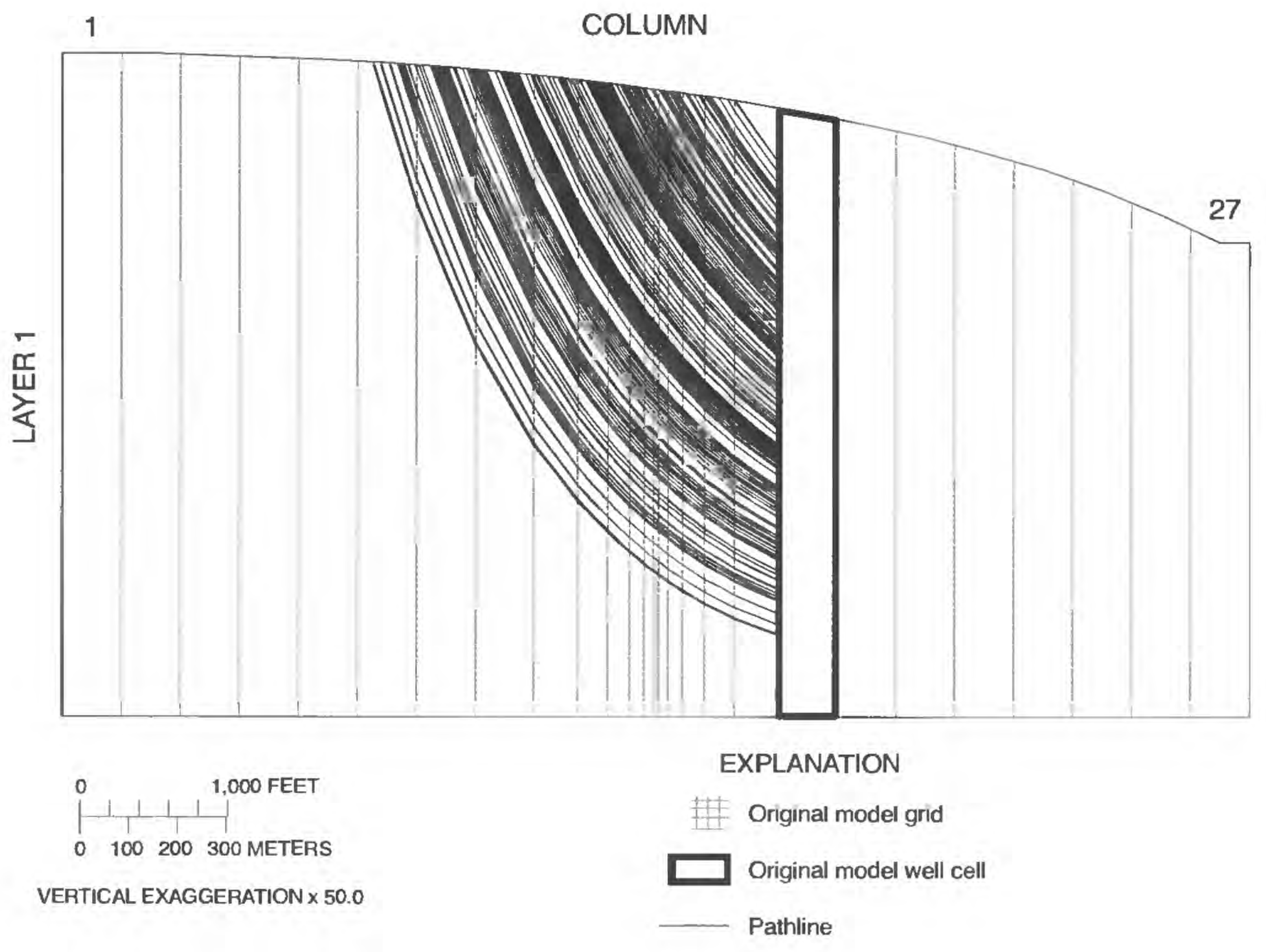


Figure 13. Profile view of pathlines inside layer 1 of the original model showing starting locations generated by particle transfer from the submodel. (The view is along row 20.)

SUMMARY AND CONCLUSIONS

The effects of spatial grid discretization limit the level of detail and accuracy of ground-water model and particle-tracking results. The purpose of this study is to develop an approach to resolve some of the discretization limitations of MODPATH, a particle-tracking package designed for use with the U.S. Geological Survey's modular finite-difference ground-water-flow model, MODFLOW. As a result of insufficient discretization, particle flow paths calculated within weak sink cells cannot be uniquely defined because it is impossible to know whether a specific water particle discharges to the weak sink or passes through the cell. A weak sink is a cell (representing a well, for example) that does not discharge at a sufficiently large rate to capture all of the flow entering the cell; thus, some of the flow leaves the cell across one or more of the cell faces.

The nested rediscretization method documented here is a set of computer programs designed to help eliminate this limitation. The method involves refining the discretization of the original model at the weak sink to make it a strong sink by creating a submodel of the cell representing the well, coupling the submodel with the original model, and performing a linked MODPATH analysis with both models. A separate MODFLOW simulation is run with the submodel. Successive rediscretizations of the cell are applied until the well becomes a strong sink.

Advantages of the method are that it does not require additional information beyond that specified in the original MODFLOW and MODPATH input files, it is easy to use, and it provides an unambiguous determination of how particles are treated. Disadvantages of the method are that it is applicable only to weak sinks representing wells in steady-state models and to backtracking particle endpoint analysis. If it is important to know which particle enters a weak sink, then more detail must be given than is provided in the model. This information could be obtained by refining the model or by specifying the actual location of the well in the cell. Specifying the actual location in the submodel is possible when using this method.

The software consists of five Fortran programs and two MODPATH response files that are operated manually in series. The programs perform different tasks, including identifying strong and weak sink wells in the original model; making use of an original, coarsely discretized MODFLOW model to create input files for a finely discretized submodel of a weak sink cell; converting MODPATH endpoint locations resulting from particle tracking in the submodel to starting locations for particle tracking in the original model; combining MODPATH endpoint locations from the submodel with endpoint locations from the original model to create a composite endpoint file; and error checking the submodel solution and particle-tracking results as well as evaluating the strength of the sink in the submodel. User input for the computer programs is a combination of data files and interactive keyboard responses to menu queries. Data output (MODPATH particle endpoint file) can be used, for example, to delineate capture areas of wells in the model domain.

Technical considerations and limitations of the method that relate to the design of the original model, discretization of the original model, and application of the method are presented. Set-up requirements necessary for the successful operation of the computer programs are described. Because not all possible MODFLOW and MODPATH applications have been tested with the programs, future applications of the programs may lead to problems requiring modification and recompilation of the source codes.

A simple hypothetical problem is presented to illustrate the use of the computer programs. The results show that capture areas determined by using particle tracking and the nested rediscretization method are smaller than those determined by using particle tracking alone because the method refines the ground-water flow field near the well. Some of the particle pathlines now converge in the cell representing the well, rather than diverging out of the cell. Particle travel times are greater when the method is used because of additional travel time within the submodel, which increases the length of the flow path. The difference in travel time created by using the method decreases to zero as travel time increases.

The computer programs have been tested successfully on available models of real-world flow systems, including glacial valley-fill aquifers and Coastal Plain aquifers in New Jersey and Louisiana. The method can be expected to provide the greatest improvement in pathline resolution when used in such existing regional models with coarse discretization; however, it also can be useful in analyzing local flow systems with low-volume pumped wells.

REFERENCES CITED

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

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

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

Leake, S.A., and Claar, D.V., 1999, Procedures and computer programs for telescopic mesh refinement using MODFLOW: U.S. Geological Survey Open-File Report 99-238, 53 p.

Nicholson, R.S., McAuley, S.D., Barringer, J.L., and Gordon, A.D., 1996, Hydrogeology of, and ground-water flow in, a valley-fill and carbonate-rock aquifer system near Long Valley in the New Jersey Highlands: U.S. Geological Survey Water-Resources Investigations Report 93-4157, 159 p.

Nicholson, R.S., and Watt, M.K., 1997, Simulation of ground-water flow in the unconfined aquifer system of the Toms River, Metedeconk River, and Kettle Creek Basins, New Jersey: U.S. Geological Survey Water-Resources Investigations Report 97-4066, 100 p.

Orzol, L.L., 1997, User's guide for MODTOOLS: Computer programs for translating data of MODFLOW and MODPATH into geographic information system files: U.S. Geological Survey Open-File Report 97-240, 86 p.

Pollock, D.W., 1994, User's guide for MODPATH/MODPATH-PLOT, version 3: A particle tracking post-processing package for MODFLOW, the U.S. Geological Survey finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 94-464, 234 p.

REFERENCES CITED--Continued

Shapiro, A.M., Margolin, J., Dolev S., and Ben-Israel, Y., 1997, A graphical-user interface for the U.S. Geological Survey modular three-dimensional finite-difference ground-water flow model (MODFLOW-96) using Argus Numerical Environments: U.S. Geological Survey Open-File Report 97-121, 50 p.

Spitz, F.J., Nicholson, R.S., and Pope, D.A., in press, Description and application of a nested rediscretization method to improve pathline resolution by eliminating weak sinks representing wells: Ground Water.

APPENDIX 1. OBTAINING AND INSTALLING THE COMPUTER PROGRAMS

The source code for the Fortran programs and associated MODPATH response files is available for downloading over the Internet from the USGS, New Jersey District, web site at URL **http://nj.usgs.gov/pub/OFR/00-392/**. The zip file contains the five programs, two response files, a "readme" file, and the sample problem input files from appendix 3 of this report. Updates might occasionally be made to the computer programs. MODFLOW and MODPATH are available for downloading over the Internet from the main USGS web site at URL **http://water.usgs.gov/software** or by anonymous ftp from **water.usgs.gov** in the **pub/software/ground_water** directory.

The programs are installed by copying the executable or Fortran program source code and associated files onto your computer system. The programs have been compiled in Fortran 77 on a Data General[1] UNIX platform and in Lahey Fortran 95 on a personal computer Windows NT platform. No links to special Fortran library subroutines are required. Because these programs were tested on a UNIX platform, personal computer users may encounter run-time errors (as a result of stricter error handling, for example) that were not encountered before. Programs should be compiled as Windows-Console Programs on a Windows NT platform in order to observe error and warning messages. The programs and MODFLOW and MODPATH should be compiled with the same compiler to avoid problems with reading binary files. The two response files should not require computer system modification; however, these files should be obtained in ASCII format to ensure they will work properly on a personal computer.

[1]Use of trade names in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey.

APPENDIX 2. INTERACTIVE MENU RESPONSES FOR SAMPLE PROBLEM

An example of an interactive input session with the computer follows. The procedure is a subset of the complete procedure described on pages 12-14. (Program output is not shown.) This example is for the case in which the well in the submodel has been converted to a strong sink. Program names below are shown in bold italics, program prompts and messages are shown in regular text, and user responses are shown in bold text. A program can be run by typing the program name at the command prompt and pressing the enter key. Following that command, each program writes output to the computer monitor asking the user for specific information. Responses should be entered in lowercase, using commas as field separators, with single quotation marks around full pathnames (if necessary), and decimal values where appropriate. Users may want to embed program names and responses in UNIX shell files or personal computer batch files to automate this procedure.

identify

Enter number of model layers, rows, and columns:

5,27,27

Enter name of modpath name file:

demo.pnm

rediscretize

Enter number of model layers, rows, columns, level of rediscretization, and vertical anisotropy:

5,27,27,7,1.0

Enter name of modflow name file:

demo.fnm

Enter name of modpath name file:

demo.pnm

Enter model layer, row, and column of well:

1,20,20

Enter submodel top layer, bottom layer, row, and column for constant head(s):

4,4,4,4

modflow

Enter the name of the NAME FILE:

fine.fnm

utility

Enter option:

1-Check submodel solution, 2-Check sink strength, 3-Check particle-tracking results

```
1
```

utility

```
 Enter option:
   1-Check submodel solution, 2-Check sink strength, 3-Check particle-tracking results
2
 Enter level of rediscretization:
7
 This cell is now a strong sink, so prorating will be done;
 Enter approximate total number of particles to be tracked:
8140
 Try a different number of particles (y,n=quit)?
n
```

Edit *fine.rsp* according to output from ***utility***.

modpath

```
 MODPATH Version 3.00 (V3, Release 1, 9-94)
 TO READ INPUT FROM AN EXISTING RESPONSE FILE, ENTER FILE NAME:
 ( <CR> = ENTER DATA INTERACTIVELY )
  [ ? = Help ]
fine.rsp
```

utility

```
 Enter option:
   1-Check submodel solution, 2-Check sink strength, 3-Check particle-tracking results
3
```

transfer

```
 Enter number of model layers, rows, columns, level of rediscretization, and particles tracked:
5,27,27,7,7983
 Enter model layer, row, and column of weak sink well:
1,20,20
```

modpath

```
 MODPATH Version 3.00 (V3, Release 1, 9-94)
 TO READ INPUT FROM AN EXISTING RESPONSE FILE, ENTER FILE NAME:
 ( <CR> = ENTER DATA INTERACTIVELY )
  [ ? = Help ]
```

coarse.rsp

utility

```
  Enter option:
    1-Check submodel solution, 2-Check sink strength, 3-Check particle-tracking results
```

3

combine

```
  Enter number of model layers, rows, columns, and particles tracked:
```

5,27,27,7983

APPENDIX 3. INPUT FILES FOR SAMPLE PROBLEM

MODFLOW Data Files

Name File [file: demo.fnm]

```
list             9  demo.lst
bas             10  demo.bas
bcf             11  demo.bcf
wel             12  demo.wel
riv             14  demo.riv
rch             18  demo.rch
oc              22  demo.oc
sip             19  demo.sip
data(binary)    50  demo.bud
data(binary)    60  demo.bhd
```

Basic Package [file: demo.bas]

```
MODPATH SAMPLE PROBLEM

         5        27        27         1         4

         0         0
         0         1
         0         1
         0         1
         0         1
         0         1
     999.9
         0      280.
         0      280.
         0      280.
         0      280.
         0      280.
       1.0         1       1.0
```

Block-Centered Flow Package [file: demo.bcf]

```
         1        50   1.0E+30         0         1         1         1
 1 0 0 0 0
         0       1.0
        11       1.0    (16F5.0)
 400. 400. 400. 400. 400. 400. 400. 400. 300. 200. 150. 100.  60.  40.  60. 100.
 150. 200. 300. 400. 400. 400. 400. 400. 400. 400. 400.
        11       1.0    (16F5.0)
 400. 400. 400. 400. 400. 400. 400. 400. 300. 200. 150. 100.  60.  40.  60. 100.
 150. 200. 300. 400. 400. 400. 400. 400. 400. 400. 400.
         0       50.
         0      220.
         0    0.0005
         0     1250.
         0      0.01
         0     1250.
         0      0.01
         0     1250.
         0      0.01
         0     1250.
```

River Package   [file: demo.riv]

```
    27       50
    27
     1        1        27      280.     3200.      275.          6 :reach 1
     1        2        27      280.     3200.      275.          6 :reach 2
     1        3        27      280.     3200.      275.          6 :reach 3
     1        4        27      280.     3200.      275.          6 :reach 4
     1        5        27      280.     3200.      275.          6 :reach 5
     1        6        27      280.     3200.      275.          6 :reach 6
     1        7        27      280.     3200.      275.          6 :reach 7
     1        8        27      280.     3200.      275.          6 :reach 8
     1        9        27      280.     2400.      275.          6 :reach 9
     1       10        27      280.     1600.      275.          6 :reach 10
     1       11        27      280.     1200.      275.          6 :reach 11
     1       12        27      280.      800.      275.          6 :reach 12
     1       13        27      280.      480.      275.          6 :reach 13
     1       14        27      280.      320.      275.          6 :reach 14
     1       15        27      280.      480.      275.          6 :reach 15
     1       16        27      280.      800.      275.          6 :reach 16
     1       17        27      280.     1200.      275.          6 :reach 17
     1       18        27      280.     1600.      275.          6 :reach 18
     1       19        27      280.     2400.      275.          6 :reach 19
     1       20        27      280.     3200.      275.          6 :reach 20
     1       21        27      280.     3200.      275.          6 :reach 21
     1       22        27      280.     3200.      275.          6 :reach 22
     1       23        27      280.     3200.      275.          6 :reach 23
     1       24        27      280.     3200.      275.          6 :reach 24
     1       25        27      280.     3200.      275.          6 :reach 25
     1       26        27      280.     3200.      275.          6 :reach 26
     1       27        27      280.     3200.      275.          6 :reach 27
```

Recharge Package   [file: demo.rch]

```
     1       50         1
     1        0
     0   0.0045
```

Well Package   [file: demo.wel]

```
     1       50
     1
     1       20        20    -3000.          7
```

Output Control Option   [file: demo.oc]

```
     4        4        60         0
     0        1         1         1
     1        0         1         0
```

Strongly Implicit Procedure Package   [file: demo.sip]

```
   400        5
    1.    .0001         0    0.0005          1
```

MODPATH Data Files

Name File   [file: demo.pnm]

```
main          78  demo.mdf
wel           12  demo.wel
riv           14  demo.riv
rch           18  demo.rch
budget        50  demo.bud
head(binary)  60  demo.bhd
endpoint      75  demo.ept
```

Main Data File   [file: demo.mdf]

```
27 27 5   1  1  0  0   999.9  1.0E+30  0

1 0 0 0 0
1 0 0 0 0
internal  1.0  (free) 0
 400. 400. 400. 400. 400. 400. 400. 400. 300. 200.
 150. 100.  60.  40.  60. 100. 150. 200. 300. 400.
 400. 400. 400. 400. 400. 400. 400.
internal  1.0  (free) 0
 400. 400. 400. 400. 400. 400. 400. 400. 300. 200.
 150. 100.  60.  40.  60. 100. 150. 200. 300. 400.
 400. 400. 400. 400. 400. 400. 400.
internal  1.0  (free) 0
 100.  50.  50.  50.  50.
internal  1.0  (free) 0
  20.   0.   0.   0.   0.
220.
open/close  ibound.1  1  (free)  0
constant  1
constant  1
constant  1
constant  1
constant  0.3
constant  0.3
constant  0.3
constant  0.3
constant  0.3
constant  0.3
```

Name File   [file: coarse.pnm]

```
main          78  demo.mdf
wel           12  demo.wel
riv           14  demo.riv
rch           18  demo.rch
budget        50  demo.bud
head(binary)  60  demo.bhd
endpoint      75  coarse.ept
```

IBOUND Array for Layer 1 [file: ibound.1]

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

MODPATH Response Files

Submodel [file: fine.rsp]

```
@[MODPATH Version 3.00 (V3, Release 1, 9-94)]
* ENTER THE NAME FILE:
@RESPONSE: HELP LABEL = 4.2.1
fine.pnm
* DO YOU WANT TO STOP COMPUTING PATHS AFTER A SPECIFIED LENGTH OF TIME ?
@RESPONSE: HELP LABEL = 2.1.41
n
* SELECT THE OUTPUT MODE:
*     1 = ENDPOINTS
*     2 = PATHLINE
*     3 = TIME SERIES
@RESPONSE: HELP LABEL = 2.1.6
1
* HOW ARE STARTING LOCATIONS TO BE ENTERED?
*     1 = FROM AN EXISTING DATA FILE
*     2 = ARRAYS OF PARTICLES WILL BE GENERATED INTERNALLY
@RESPONSE: HELP LABEL = 2.1.9
2
* DO YOU WANT TO STORE INTERNALLY-GENERATED STARTING LOCATIONS ON DISK ?
@RESPONSE: HELP LABEL = 2.1.44
n
* IN WHICH DIRECTION SHOULD PARTICLES BE TRACKED?
*     1 = FORWARD IN THE DIRECTION OF FLOW
*     2 = BACKWARDS TOWARD RECHARGE LOCATIONS
@RESPONSE: HELP LABEL = 2.1.10
2
* HOW SHOULD PARTICLES BE TREATED WHEN THEY ENTER CELLS WITH INTERNAL SINKS ?
*     1 = PASS THROUGH WEAK SINK CELLS
*     2 = STOP AT WEAK SINK CELLS
*     3 = STOP AT WEAK SINK CELLS THAT EXCEED A SPECIFIED STRENGTH
@RESPONSE: HELP LABEL = 2.1.11
1
* DO YOU WANT TO STOP PARTICLES WHENEVER THEY ENTER ONE SPECIFIC ZONE ?
@RESPONSE: HELP LABEL = 2.1.45
n
*  STARTING LOCATIONS WILL BE GENERATED FOR 3-D SUBREGIONS.
*
* SHOULD PARTICLES BE PLACED IN CONSTANT HEAD CELLS ?
@RESPONSE: HELP LABEL = 2.1.46
y
*  ENTER DATA FOR SUBREGION  1 :
*
* DEFINE THE SUBREGION BY ENTERING THE MINIMUM AND MAXIMUM J,I,K COORDINATES.
*   ENTER:  MINIMUM J, MAXIMUM J, MINIMUM I, MAXIMUM I, MINIMUM K, MAXIMUM K
@RESPONSE: HELP LABEL = 2.1.14
4,4,4,4,4,4
* WHERE SHOULD THE PARTICLES BE LOCATED ?
*   1 = WITHIN A CELL
*   2 = ON ONE OR MORE OF THE CELL FACES
@RESPONSE: HELP LABEL = 2.1.15
2
* WHAT CELL FACES DO YOU WANT TO PLACE PARTICLES ON ?
*   (ENTER ALL THE FACE NUMBERS ON A SINGLE LINE)
@RESPONSE: HELP LABEL = 2.1.47
```

```
1 2 3 4 5 6
* ENTER A 2-D ARRAY OF PARTICLES FOR FACE 1:  NY  NZ
@RESPONSE: HELP LABEL = 2.1.17
27 27
* ENTER A 2-D ARRAY OF PARTICLES FOR FACE 2:  NY  NZ
@RESPONSE: HELP LABEL = 2.1.18
10 10
* ENTER A 2-D ARRAY OF PARTICLES FOR FACE 3:  NX  NZ
@RESPONSE: HELP LABEL = 2.1.19
21 21
* ENTER A 2-D ARRAY OF PARTICLES FOR FACE 4:  NX  NZ
@RESPONSE: HELP LABEL = 2.1.20
21 21
* ENTER A 2-D ARRAY OF PARTICLES FOR FACE 5:  NX  NY
@RESPONSE: HELP LABEL = 2.1.21
56 56
* ENTER A 2-D ARRAY OF PARTICLES FOR FACE 6:  NX  NY
@RESPONSE: HELP LABEL = 2.1.22
56 56
*   7983 PARTICLES USED OUT OF 500000 MAXIMUM
* DO YOU WANT TO SPECIFY ANOTHER SUBREGION ?
@RESPONSE: HELP LABEL = 2.1.53
n
* DO YOU WANT TO COMPUTE VOLUMETRIC BUDGETS FOR ALL CELLS ?
@RESPONSE: HELP LABEL = 3.1.4
n
*  DO YOU WANT TO CHECK DATA CELL BY CELL ?
@RESPONSE: HELP LABEL = 3.1.2
n
* SUMMARIZE FINAL STATUS OF PARTICLES IN SUMMARY.PTH FILE ?
@RESPONSE: HELP LABEL = 3.1.3
n
```

Original Model   [file: coarse.rsp]

```
@[MODPATH Version 3.00 (V3, Release 1, 9-94)]
* ENTER THE NAME FILE:
@RESPONSE: HELP LABEL = 4.2.1
coarse.pnm
* DO YOU WANT TO STOP COMPUTING PATHS AFTER A SPECIFIED LENGTH OF TIME ?
@RESPONSE: HELP LABEL = 2.1.41
n
* SELECT THE OUTPUT MODE:
*      1 = ENDPOINTS
*      2 = PATHLINE
*      3 = TIME SERIES
@RESPONSE: HELP LABEL = 2.1.6
1
* HOW ARE STARTING LOCATIONS TO BE ENTERED?
*      1 = FROM AN EXISTING DATA FILE
*      2 = ARRAYS OF PARTICLES WILL BE GENERATED INTERNALLY
@RESPONSE: HELP LABEL = 2.1.9
1
* ENTER NAME OF DATA FILE CONTAINING STARTING LOCATIONS:
@RESPONSE: HELP LABEL = 2.1.39
coarse.stl
* IN WHICH DIRECTION SHOULD PARTICLES BE TRACKED?
*      1 = FORWARD IN THE DIRECTION OF FLOW
*      2 = BACKWARDS TOWARD RECHARGE LOCATIONS
```

```
@RESPONSE: HELP LABEL = 2.1.10
2
* HOW SHOULD PARTICLES BE TREATED WHEN THEY ENTER CELLS WITH INTERNAL SINKS ?
*      1 = PASS THROUGH WEAK SINK CELLS
*      2 = STOP AT WEAK SINK CELLS
*      3 = STOP AT WEAK SINK CELLS THAT EXCEED A SPECIFIED STRENGTH
@RESPONSE: HELP LABEL = 2.1.11
1
* DO YOU WANT TO STOP PARTICLES WHENEVER THEY ENTER ONE SPECIFIC ZONE ?
@RESPONSE: HELP LABEL = 2.1.45
n
* DO YOU WANT TO COMPUTE VOLUMETRIC BUDGETS FOR ALL CELLS ?
@RESPONSE: HELP LABEL = 3.1.4
n
*  DO YOU WANT TO CHECK DATA CELL BY CELL ?
@RESPONSE: HELP LABEL = 3.1.2
n
* SUMMARIZE FINAL STATUS OF PARTICLES IN SUMMARY.PTH FILE ?
@RESPONSE: HELP LABEL = 3.1.3
n
```

APPENDIX 4. OUTPUT FILES FOR SAMPLE PROBLEM

MODFLOW Data Files

Name File [file: fine.fnm]

```
LIST                        70  fine.lst
BAS                         10  fine.bas
BCF                         11  fine.bcf
WEL                         72  fine.wel
OC                          22  fine.oc
SIP                         19  fine.slv
DATA(BINARY)                73  fine.bhd
DATA(BINARY)                74  fine.bud
```

Basic Package [file: fine.bas]

```
SUBMODEL OF WEAK SINK CELL IN ORIGINAL MODEL
--------------------------------------------
         7         7         7         1         4
FREE
         0         0
        10         1 (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
        10         1 (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
        10         1 (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
        10         1 (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1 -1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
        10         1 (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
        10         1 (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
        10         1 (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
     999.9
      0     301.5
      0     301.5
      0     301.5
      0     301.5
      0     301.5
      0     301.5
      0     301.5
        1.         1         1.
```

Block-Centered Flow Package   [file: fine.bcf]

```
         1         74      0.10E+31         0         1.         1         1
 1 3 3 3 3 3 3
      0     1.000
      0     57.14
      0     57.14
      0     50.00                                    HYCON  1
      0     289.9                                    BOT    1
      0     4.293                                    VCONT  1
      0     50.00                                    HYCON  2
      0     278.2                                    BOT    2
      0     4.293                                    VCONT  2
      0     289.9                                    TOP    2
      0     50.00                                    HYCON  3
      0     266.6                                    BOT    3
      0     4.293                                    VCONT  3
      0     278.2                                    TOP    3
      0     50.00                                    HYCON  4
      0     254.9                                    BOT    4
      0     4.293                                    VCONT  4
      0     266.6                                    TOP    4
      0     50.00                                    HYCON  5
      0     243.3                                    BOT    5
      0     4.293                                    VCONT  5
      0     254.9                                    TOP    5
      0     50.00                                    HYCON  6
      0     231.6                                    BOT    6
      0     4.293                                    VCONT  6
```

```
     0     243.3                                                   TOP     6
     0     50.00                                                   HYCON   7
     0     220.0                                                   BOT     7
     0     231.6                                                   TOP     7
```

Solver Package [file: fine.slv]

```
       150          5
  1.00     0.100E-04         1         0         1
```

Well Package [file: fine.wel]

```
       294         74
       294
         1          1          7     -114.01              2
         2          1          7     -114.01              2
         3          1          7     -114.01              2
         4          1          7     -114.01              2
         5          1          7     -114.01              2
         6          1          7     -114.01              2
         7          1          7     -114.01              2
         1          2          7     -114.01              2
         2          2          7     -114.01              2
         3          2          7     -114.01              2
         4          2          7     -114.01              2
         5          2          7     -114.01              2
         6          2          7     -114.01              2
         7          2          7     -114.01              2
         1          3          7     -114.01              2
                    .
                    .
                    .
         7          7          7     -114.01              2
         1          1          1      130.66              1
         2          1          1      130.66              1
         3          1          1      130.66              1
         4          1          1      130.66              1
         5          1          1      130.66              1
         6          1          1      130.66              1
         7          1          1      130.66              1
         1          2          1      130.66              1
         2          2          1      130.66              1
         3          2          1      130.66              1
         4          2          1      130.66              1
         5          2          1      130.66              1
         6          2          1      130.66              1
         7          2          1      130.66              1
         1          3          1      130.66              1
                    .
                    .
                    .
         7          7          1      130.66              1
         1          7          1      14.667              3
         2          7          1      14.667              3
         3          7          1      14.667              3
         4          7          1      14.667              3
         5          7          1      14.667              3
         6          7          1      14.667              3
         7          7          1      14.667              3
```

1	7	2	14.667	3
2	7	2	14.667	3
3	7	2	14.667	3
4	7	2	14.667	3
5	7	2	14.667	3
6	7	2	14.667	3
7	7	2	14.667	3
1	7	3	14.667	3
	.			
	.			
	.			
7	7	7	14.667	3
1	1	1	15.585	4
2	1	1	15.585	4
3	1	1	15.585	4
4	1	1	15.585	4
5	1	1	15.585	4
6	1	1	15.585	4
7	1	1	15.585	4
1	1	2	15.585	4
2	1	2	15.585	4
3	1	2	15.585	4
4	1	2	15.585	4
5	1	2	15.585	4
6	1	2	15.585	4
7	1	2	15.585	4
1	1	3	15.585	4
	.			
	.			
	.			
7	1	7	15.585	4
7	1	1	-0.36834	5
7	2	1	-0.36834	5
7	3	1	-0.36834	5
7	4	1	-0.36834	5
7	5	1	-0.36834	5
7	6	1	-0.36834	5
7	7	1	-0.36834	5
7	1	2	-0.36834	5
7	2	2	-0.36834	5
7	3	2	-0.36834	5
7	4	2	-0.36834	5
7	5	2	-0.36834	5
7	6	2	-0.36834	5
7	7	2	-0.36834	5
7	1	3	-0.36834	5
	.			
	.			
	.			
7	7	7	-0.36834	5
1	1	1	14.694	6
1	1	2	14.694	6
1	1	3	14.694	6
1	1	4	14.694	6
1	1	5	14.694	6
1	1	6	14.694	6
1	1	7	14.694	6
1	2	1	14.694	6
1	2	2	14.694	6

```
        1          2          3       14.694             6
        1          2          4       14.694             6
        1          2          5       14.694             6
        1          2          6       14.694             6
        1          2          7       14.694             6
        1          3          1       14.694             6
                   .
                   .
                   .
        1          7          7    14.7            6
```

Output Control Option   [file: fine.oc]

```
       15          0         73          0
        0          1          1          1
        1          0          1          0
```

MODPATH Data Files

Name File [file: fine.pnm]

```
MAIN                      71  fine.mdf
WEL                       72  fine.wel
HEAD(BINARY)              73  fine.bhd
BUDGET                    74  fine.bud
ENDPOINT                  75  fine.ept
```

Main Data File [file: fine.mdf]

```
  7  7  7  6    1  0  0     999.9     0.10E+31  0

  1  3  3  3  3  3  3
  1  1  1  1  1  1  0
constant     57.14
constant     57.14
internal  1.0  (free)
    11.65        11.65        11.65        11.65        11.65        11.65        11.65
internal  1.0  (free)
   0.0000E+00   0.0000E+00   0.0000E+00   0.0000E+00   0.0000E+00   0.0000E+00   0.0000E+00
   289.9
internal    1          (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
internal    1          (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
internal    1          (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
internal    1          (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1 -1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
internal    1          (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
internal   1         (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
internal   1         (7I3)
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
  1  1  1  1  1  1  1
constant   0.3000                                    PORAQ  1
constant   0.3000                                    PORCB  1
constant   0.3000                                    PORAQ  2
constant   0.3000                                    PORCB  2
constant   0.3000                                    PORAQ  3
constant   0.3000                                    PORCB  3
constant   0.3000                                    PORAQ  4
constant   0.3000                                    PORCB  4
constant   0.3000                                    PORAQ  5
constant   0.3000                                    PORCB  5
constant   0.3000                                    PORAQ  6
constant   0.3000                                    PORCB  6
constant   0.3000                                    PORAQ  7
```