MODPATH-LGR – Documentation of a Computer Program for Particle Tracking in Shared-Node Locally Refined Grids by using MODFLOW-LGR

Techniques and Methods 6-A38

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

COVER
 Example of particle pathlines in a hypothetical flow system that is simulated by a parent model and two child models. The particles are backtracked from a well in the parent model and from wells in the child models to the source areas (the river, the trench, the pit, and detention basin).

MODPATH-LGR – Documentation of a Computer Program for Particle Tracking in Shared-Node Locally Refined Grids by using MODFLOW-LGR

By Jesse E. Dickinson, R.T. Hanson, Steffen W. Mehl, and Mary C. Hill

Techniques and Methods 6-A38

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

U.S. Department of the Interior
KEN SALAZAR, Secretary

U.S. Geological Survey
Marcia K. McNutt, Director

U.S. Geological Survey, Reston, Virginia: 2011

Preface

The computer program described in this report, MODPATH-LGR, is designed to allow simulation of particle tracking in locally refined grids. The locally refined grids are simulated by using MODFLOW-LGR, which is based on MODFLOW-2005, the three-dimensional groundwater-flow model published by the U.S. Geological Survey. The documentation includes brief descriptions of the methods used and detailed descriptions of the required input files and how the output files are typically used.

The code for this model is available for downloading from the World Wide Web from a U.S. Geological Survey software repository. The repository is accessible from the U.S. Geological Survey Water Resources Information Web page at http://water.usgs.gov/software/ground_water.html.

The performance of the MODPATH-LGR program has been tested in a variety of applications. Future applications, however, might reveal errors that were not detected in the test simulations. Users are requested to notify the U.S. Geological Survey of any errors found in this document or the computer program by using the email address available on the Web site. Updates might occasionally be made to this document and to the MODPATH-LGR program, and users should check the Web site periodically.

Contents

Figures

Conversion Factors and Abbreviations

SI to Inch/Pound

Multiply	By	To obtain
Length		
meter (m)	3.281	foot (ft)
Area		
square meter (m^2)	10.76	square foot (ft^2)
Volume		
cubic meter (m^3)	35.31	cubic foot (ft^3)
cubic meter (m^3)	0.0008107	acre-foot (acre-ft)
Flow rate		
cubic meter per second (m^3/s)	70.07	acre-foot per day (acre-ft/d)
meter per second (m/s)	3.281	foot per second (ft/s)
cubic meter per second (m^3/s)	35.31	cubic foot per second (ft^3/s)
Hydraulic conductivity		
meter per day (m/d)	3.281	foot per day (ft/d)
Hydraulic gradient		
meter per kilometer (m/km)	5.27983	foot per mile (ft/mi)
Transmissivity*		
meter squared per day (m^2/d)	10.76	foot squared per day (ft^2/d)

This page left blank intentionally.

MODPATH-LGR – Documentation of a Computer Program for Particle Tracking in Shared-Node Locally Refined Grids by using MODFLOW-LGR

By Jesse E. Dickinson, R.T. Hanson, Steffen W. Mehl, and Mary C. Hill

Abstract

MODPATH-LGR is a particle tracking post-processing program for computing three-dimensional flow paths for steady-state and transient groundwater flow in models with local-grid refinement (sometimes called embedded models). Locally refined grids consist of a larger regional-scale parent model and one or more smaller embedded local-scale child models. This program uses output produced by MODFLOW-LGR, in which child grids in parent grids are refined by using the Shared-Node Local Grid Refinement (LGR) package of the three-dimensional finite-difference groundwater-flow model MODFLOW published by the U.S. Geological Survey (USGS). The particle tracking scheme is based on MOD-PATH, which uses an analytical expression for particle movement within each finite-difference cell. Particles are tracked among cells until discharged through model boundaries, weak sinks, or other designated discharge zones. MODPATH-LGR tracks particles between the parent and child models by using flows calculated at the interface boundaries. Flows calculated by MODFLOW-LGR are modified to account for fractional finite-difference cells along the shared-node interface between parent and child models. Required program input includes standard MODPATH input files for the parent and child models and a MODPATH-LGR control file. This documentation describes how particles are transferred between parent and child models and demonstrates program operation with two hypothetical steady-state xsimulations.

Introduction

Simulation of advective transport by using particle tracking is used widely to investigate groundwater flow through complex systems. For example, particle tracking is used to identify capture areas associated with pumping wells (Franke and others 1998, Mehl and others, 2006a; Clark and others, 2007) and springs and gaining rivers (Sanford and others, 2003). Particle tracking also is used to identify subsurface effects of artificial recharge (Reichard and others, 2003). Particle tracking can be used to determine susceptibility to groundwater contamination, improve understanding of groundwater-recharge processes, and assess vulnerabilities associated with given amounts of pumping. Particle tracking also can serve an important role in model calibration, prediction, and analysis of prediction uncertainty (Anderman and others, 1996; Anderman and Hill, 2001; Sanford and others, 2003, 2004; and Reynolds and Marimuthu, 2007).

Locally refined grids are composed of a parent (regional scale) model and one or more child (local scale) models (fig. 1). Simulation of groundwater flow through locally refined grids is often needed to understand detailed dynamics caused by heterogeneity, surface-water/groundwater interactions along streams, or to delineate contributing areas and capture areas associated with pumping wells. For example, local refinement often is needed to simulate the response to imposed hydraulic stresses such as pumping within critical subregions while maintaining the defensible boundary conditions typical of regional models (Graham and Smart, 1980; von Rosenberg, 1982; Székely, 1998; Leake and others, 1998; Mehl and Hill, 2002, 2004, 2005, 2007; Dickinson and others, 2007). Locally refined grids provide important opportunities for inverse modeling and uncertainty analysis (Mehl and Hill, 2003; Keating and others, 2003).

Rigorous coupling of the parent and child models is needed to obtain accurate flow solutions, and is critical when tracking particles among the models, investigating the importance of regional-model characteristics to particle tracking in a child model, and quantifying the contribution of uncertainties on particle tracking in either of the models. Rigorous coupling (Mehl and Hill, 2005, 2007) requires conservation of mass and energy along the boundary between parent and child models, and can be obtained by multiple numerical iterations to achieve convergence of the flows and heads simulated by parent and child models along the boundary. One-way coupling (Leake and Claar, 1999) (no iterations for convergence of simulated heads and flows along the boundary) is less accurate than iteratively-coupled locally refined grids but is sometimes

useful because one-way tracking is computationally fast (Mehl and Hill, 2004, 2005; Mehl and others, 2006b).

The computer program MODPATH-LGR tracks particles in locally refined models with multiple areas of refinement simulated by using MODFLOW-LGR (Mehl and Hill, 2005, 2007). MODPATH-LGR overcomes two difficulties associated with particle tracking in locally refined models. First, particle-transport times and locations are automatically recorded for the particles at a parent-child boundary as the particles leave one model and enter the next model, greatly reducing the effort required to track particles manually among adjacent models. Second, MODPATH-LGR makes necessary modifications of flux calculations between the adjacent models when using the shared-node method of local-grid refinement.

Purpose and Scope

This report documents the computer program MOD-PATH-LGR, a particle tracking postprocessor for single models simulated with MODFLOW-2005 (Harbaugh, 2005) or locally refined models with multiple areas of refinement simulated with MODFLOW-LGR by using the shared-node method (Mehl and Hill, 2005, 2007). MODPATH-LGR is based on MODPATH (Pollock, 1994), a USGS particle-tracking program for MODFLOW (McDonald and Harbaugh, 1988). This report is designed to be used in conjunction with the MODPATH documentation (Pollock, 1994). This document describes the input required to track particles among locally refined models, and many users also will need to refer to the

EXPLANATION

Full parent cell   Shared parent cell on model boundary   Full child cell   Child cell on model boundary   Shared child cell on model boundary
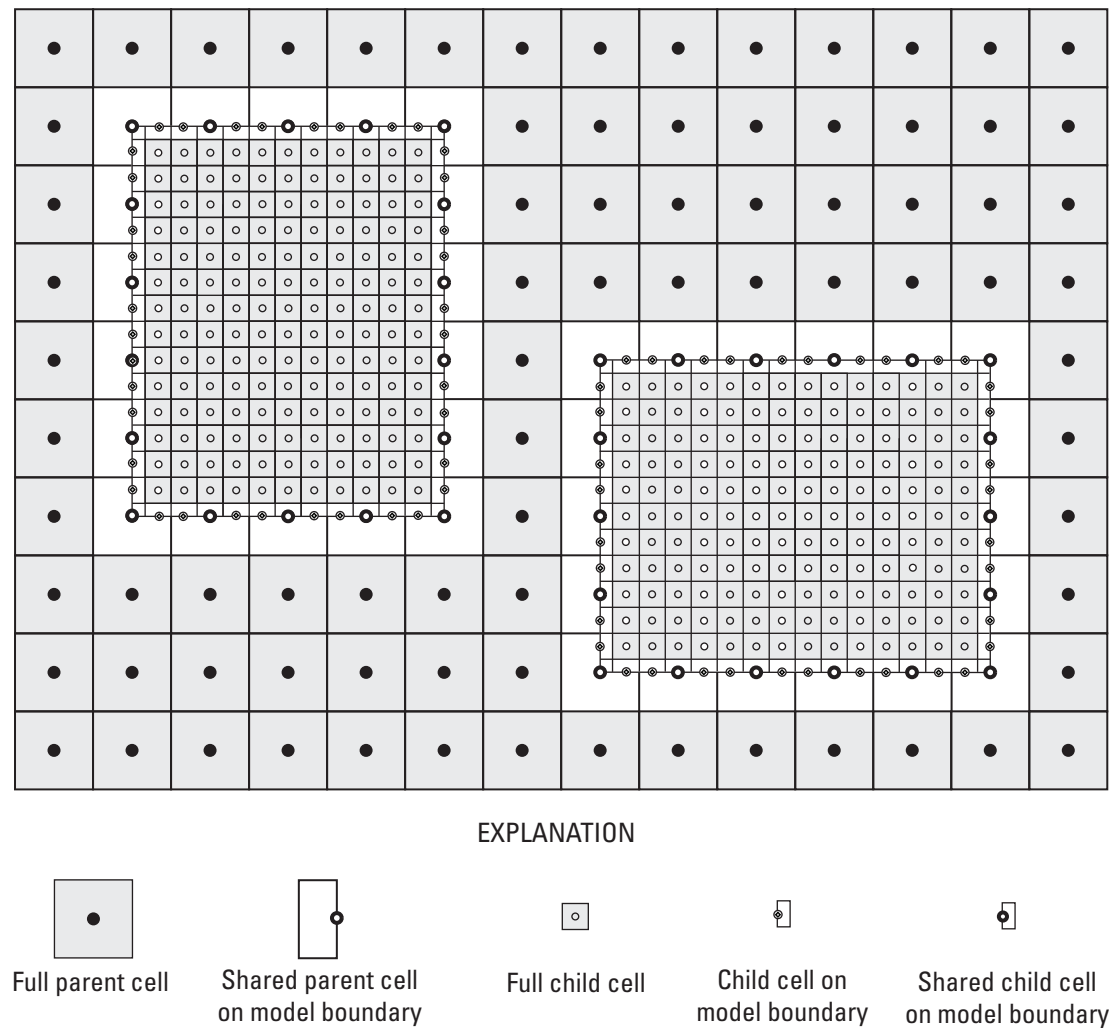
Figure 1. Relation of parent (regional-scale) and child (local-scale) model grids simulated by MODFLOW-LGR (modified from Mehl and Hill, 2007). Filled circles represent nodes in the parent-model grid, small hollow circles represent nodes in the child-model grid, and large hollow circles with heavy outlines represent nodes that are shared by the parent and child-model grids.

instructions provided for MODPATH to track particles in each model (Pollock, 1994).

The name MODPATH-LGR refers to the computer program documented in this report for particle tracking for single models and locally refined models. The name MODPATH refers to the computer program for single models described by Pollock (1994).

The report lists the changes made to the MODPATH computer program to obtain MODPATH-LGR, provides a quick guide for running MODPATH-LGR, and describes the methods developed to approximate the movement of particles between the coupled parent and child models. Two example problems are used to demonstrate the use of MODPATH-LGR. Subsequent sections review the limitations and current (2010) and potential uses of particle tracking with coupled models. Finally, this report provides input instructions for MODPATH-LGR and a summary of the changes to MODFLOW-LGR required by MODPATH-LGR.

Overview of Changes to MODPATH to Obtain MODPATH-LGR

The following new features have been added to MODPATH version 4 (Pollock, 1994) to obtain MODPATH-LGR.

1. Particle tracking in and between a parent model and multiple child models (fig. 1) simulated by MODFLOW-LGR (Mehl and Hill, 2005, 2007).

2. When run for a single uncoupled model, MODPATH-LGR is executed by using a response file. When run for coupled models, MODPATH-LGR is executed by using a control file.

3. The user defines a unique particle identification (PARTID) number for each particle in the starting locations file(s). The PARTID is used to distinguish individual particles as the particles are passed between parent and child models. The user can define optional particle identification (PARTAS) text for each particle to describe any attribute of the particle.

4. Multiple release times are allowed for backward tracking. Previous versions of MODPATH supported multiple releases only for forward tracking.

5. New output files are produced that include global pathline, endpoint, and time-series files. "Global" means that each file contains particle information from all models and uses the coordinate system of the parent model. Files also are produced for the parent model and each child model.

6. The following items are included in each global output file in addition to the kinds of items included in the equivalent MODPATH files:

   a. Particle locations in a global coordinate system, including a global z (vertical) coordinate;

   b. The global particle identification numbers described in item 3;

   c. Model identification numbers for each particle which indicate the model that contains the particle at any recorded time (IDMODEL) and the model of origin for the particle (IDMODEL_START).

   d. Optional user specified text (PARTAS) to indicate attributes for each particle.

7. An option to project the particle past boundary conditions, strong sinks, or other designated discharge zones. This option is included in MODPATH version 5. The particle projection allows for greater options in analyses of model sensitivity and prediction uncertainty.

Overview of Steps for Running MODPATH-LGR

This section describes the steps required in tracking particles through single or locally refined models by using MODPATH-LGR (fig. 2).

Step 1. Simulate groundwater flow through a single model or multiple locally refined models

The first step is to simulate flow through a single model or multiple locally refined models by using MODFLOW-2005. MODFLOW-LGR version 1.2.01 (based on MODFLOW-2005) or later must be used for particle tracking through locally refined models. MODFLOW-LGR version 1.2.01 or later provides additional flux information at the boundaries between coupled models required by MODPATH-LGR. MODPATH-LGR uses the fluxes along the boundaries to transfer particles between the parent and child models. The cell-by-cell flows are required for each model.

Step 2. Create MODPATH-LGR input files

The second step is to create input files for MODPATH-LGR. MODPATH-LGR requires only the input files used by MODPATH for simulations with one model (Pollock, 1994). MODPATH-LGR requires the input files used by MODPATH for each model and a MODPATH-LGR control file for locally refined models. MODPATH-LGR requires additional information for each particle in the starting locations file for both

types of simulations, in addition to the information required by MODPATH. Input files are described in detail in appendix A.

MODPATH-LGR control file

The MODPATH-LGR control file contains:

1. The LGR control file name required by MODFLOW-LGR.

2. A response file name for each model that is identical to the response file used by MODPATH (Pollock, 1994).

3. File names for the global output of particle coordinate data for all models in the coordinate system of the parent model.

Starting locations for particles

Particle tracking begins with identifying the starting (or, for backward tracking, the ending) location of particles. For example, in backward tracking, locations around a localized stress of interest, such as a well used for an aquifer test, may be used. Starting or ending locations may be specified in the parent or any child model, or any combination of parent and child models at the beginning of the particle-tracking simulation (fig. 1).

Starting locations and release times for particles that are tracked through locally refined models are read directly from files. Starting locations and release times for particles can be read from a file or generated automatically for rectangular blocks of cells for simulations with only a single model. The particle generation routine that was originally included in MODPATH also is included in MODPATH-LGR. The routine specifies starting locations for regularly-spaced particles over user-defined three-dimensional subregions of the model grid. Additional details on the particle generation method are described by Pollock (1994).

A separate starting locations file is required for each model that contains starting particles, and each record of the file contains the information for a single particle. The starting particle locations file is created manually or by the particle generation method through a prior execution of MODPATH (Pollock, 1994) for each individual model. The format of the starting locations file is nearly identical to the format used by MODPATH, with the exception of a release time for backward tracking analyses. Two issues exist related to using this procedure of specifying starting locations.

1. Particle locations specified in the parent-model domain that are overlapped by a child model are ignored. Particle locations specified in a child model must be listed in the starting locations file for that model.

2. The starting particle locations require one additional item of information and a second optional item in addi-

tion to the particle-location data specified for an individual model. This additional information is the particle ID (variable IDPART) and an optional character variable (variable PARTAS), which are described in appendix A.

Step 3. Execute MODPATH-LGR

MODPATH-LGR is executed by providing a response file name or interactive input when run for a single model. MODPATH-LGR is executed by using a MODPATH-LGR control file and a response file for each model when run for multiple models. The response files are generally created by saving the user responses to the interactive input feature from preliminary runs of MODPATH or MODPATH-LGR for the individual parent and child models. MODPATH-LGR cannot be executed for multiple models by using the interactive input feature that is included in MODPATH (Pollock, 1994).

Step 4. Review Model Results

MODPATH-LGR produces identical outputs as MODPATH for runs of a single model (Pollock, 1994). For runs that use locally refined models, MODPATH-LGR produces outputs for each model and global output files that include particle coordinate information for all models in the coordinate system of the parent model. The computer programs MODPATH-PLOT (Pollock, 1994) and Model Viewer (Hsieh and Winston, 2002) can generate graphical output for viewing the particle coordinate output files for individual models.

Tracking Particles through Locally Refined Models

MODPATH-LGR tracks particles through locally refined models by using sequences of particle tracking simulations for each coupled model (fig. 3). A sequence contains a particle tracking simulation for each model that has particles, and several sequences may be used to track the particles through the domain of the coupled models. In each sequence, the program first tracks particles in the parent model, and then tracks particles in the child models, following the order of the model list in the control file. A particle tracking simulation will not be executed during the sequence if a model does not contain particles. For example, a particle that starts in the parent model may enter a child model during the first sequence. The particle can be tracked through this child model and reenter the parent model in the same sequence. Next, the particle could continue the path through the parent model, enter a second child model and be transported through that child model to where the particle reenters the parent model. A third sequence would be required for the particle tracking

to continue through the parent model. MODPATH-LGR runs sequences until the end of the simulation time (specified in the response files for each model) or until all particles have discharged at boundary-condition cells, automatic termination zones, or weak or strong sinks. The criteria for discharging particles are identical to how particle tracking is terminated in MODPATH (Pollock, 1994).

MODPATH-LGR uses several steps during each sequence for tracking particles between coupled models, and for translating the times and locations for particles crossing the parent-child boundary (fig. 3). After simulating the coupled models by using MODFLOW-LGR, MODPATH-LGR reads the flow output file to obtain the fluxes of the parent model at the shared cells along the boundary with the child models. After a flux adjustment is applied for the shared parent cells, MODPATH-LGR first tracks particles for the parent model. Particles are tracked forward or backward until the particles discharge or arrive at transfer cells along the parent-child boundaries (fig. 4). MODPATH-LGR compiles the locations and times of particles at the parent-child boundaries and internally creates new starting locations for each child model. Next, particles are tracked in each child model. Once particles have exited the child model(s), MODPATH-LGR internally compiles the locations and times of particles at the parent-child boundaries and creates new starting locations for the parent model. Particles are tracked again in the parent model by using the new starting locations. This sequence of passing particles between models is continued until all particles are discharged from all models.

Particles can transfer between coupled models only when the particle is in the active domain of the next coupled model (fig. 4). This requirement allows a particle at the parent-child boundary to be accepted and to continue tracking from the new starting location in the next coupled model. However, the originating model needs to track the particle far enough into the receiving model for the particle to be accepted by the receiving model. A particle that starts in the parent model (position 1, fig. 4) transfers from the parent to a child model at a face of shared parent cell that is in the domain of the child model (position 2, fig. 4). A particle transfers from a child model to the parent model at a face of a child cell that is in the domain of the parent model (position 3, fig. 4).

Particles transfer between parent and child models at the face of the cells designated as parent-transfer cells and child-transfer cells (cells with hatched fill in fig. 4). The transfer happens at the first face of the parent- or child-transfer cell that the particle meets as the particle exits a model. For example, a particle that travels from the parent model (from position 1, fig. 4) into the child model is stopped by MODPATH-LGR at the face of the parent-transfer cell (at position 2, fig. 4). The particle is now in the active domain of child model and the particle trajectory is restarted in the child model (position 2 to 3, fig. 4). The particle reaches the face of a child transfer cell at position 3 (fig. 4) and is stopped by MODPATH-LGR. The particle is started in the parent model at point 3 and moves to point 4.

A flux adjustment to the flow output file for the parent model is made by MODPATH-LGR to track a particle

```
                    ┌─────────────┐
                    │    Start    │
                    └──────┬──────┘
                           │
                           ▼
┌────────────────────────────────────────────────────────────┐
│  Run MODFLOW-LGR to simulate flow through locally refined    │
│  models consisting of a regional parent model and one or     │
│  more local child models. Save flows for each grid.          │
└────────────────────────┬─────────────────────────────────────┘
                           │
                           ▼
┌────────────────────────────────────────────────────────────┐
│  Run MODPATH or MODPATH-LGR for each model to create         │
│  response files. Some models may not have starting particles.│
└────────────────────────┬─────────────────────────────────────┘
                           │
                           ▼
┌────────────────────────────────────────────────────────────┐
│  Run MODPATH-LGR for particle tracking in all models using   │
│  a control file.                                             │
└────────────────────────┬─────────────────────────────────────┘
                           │
                           ▼
                    ┌─────────────┐
                    │    Stop     │
                    └─────────────┘
```
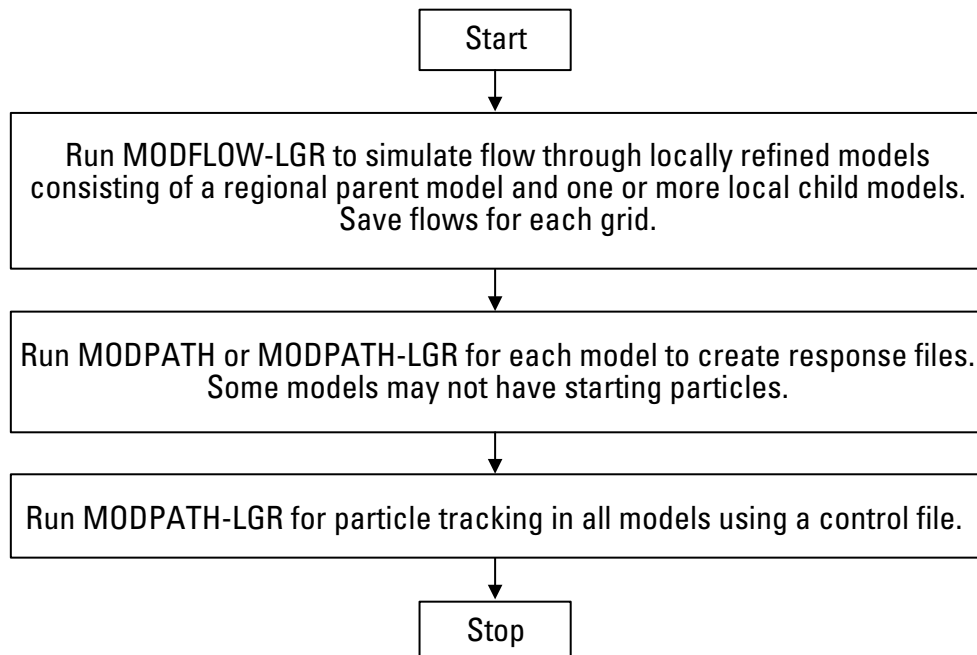
Figure 2. Flowchart showing the users' steps for particle tracking within locally refined models.

to position 2 (fig. 4) to the face of shared parent cells in the domain of the child model. The flux adjustment is required because the shared-node method of MODPATH-LGR creates inactive parts of the shared parent cells that overlap with the adjacent model, and defines the flux through the inactive face as equal to zero. The flux adjustment provides flow information through the face that allows the particle to move to the cell faces that are within the domain of the adjacent model. Further details of the flux adjustment are provided in appendix D.

Example Problems

The first example problem represents a four-layer aquifer system with steady-state flow conditions and a single pumping
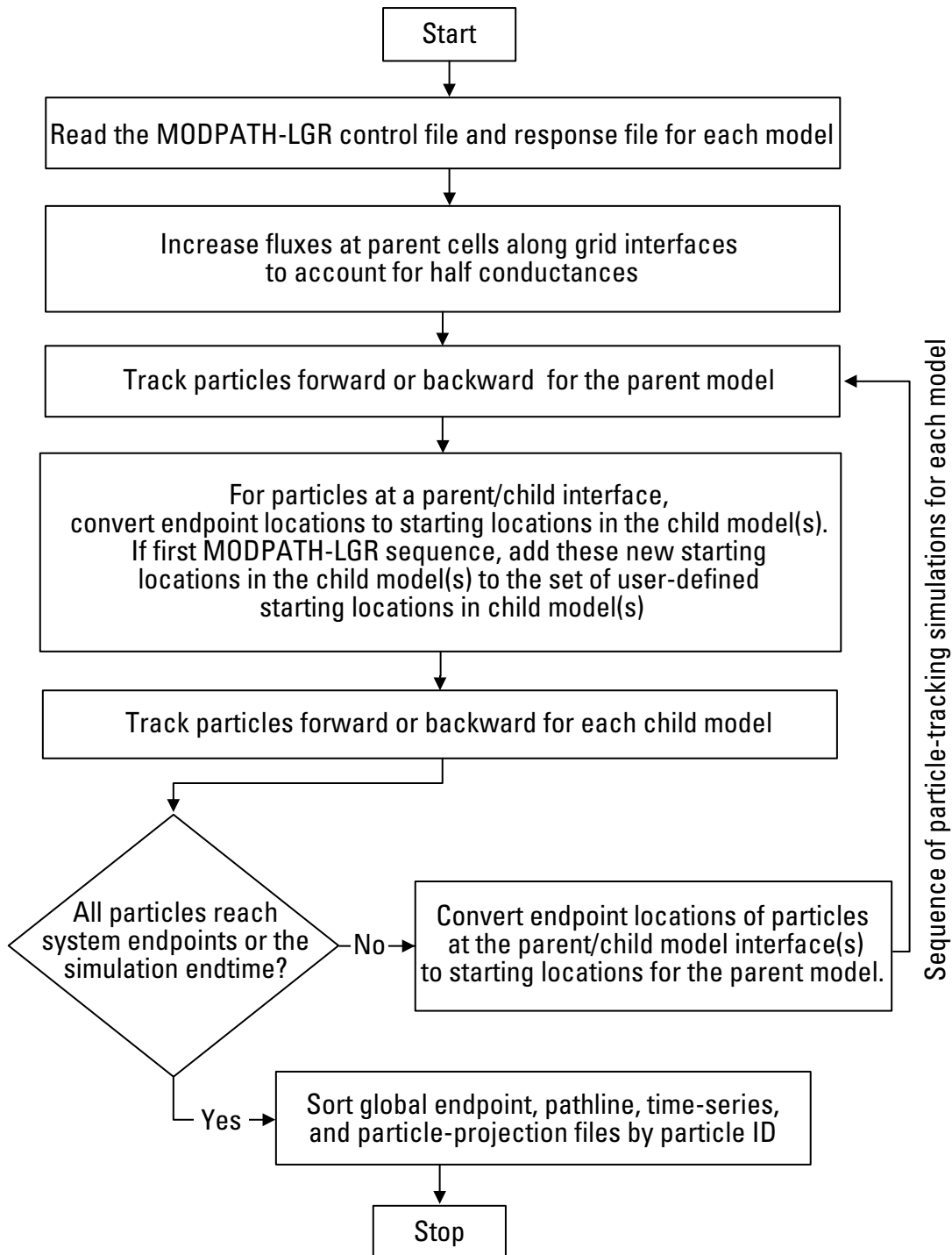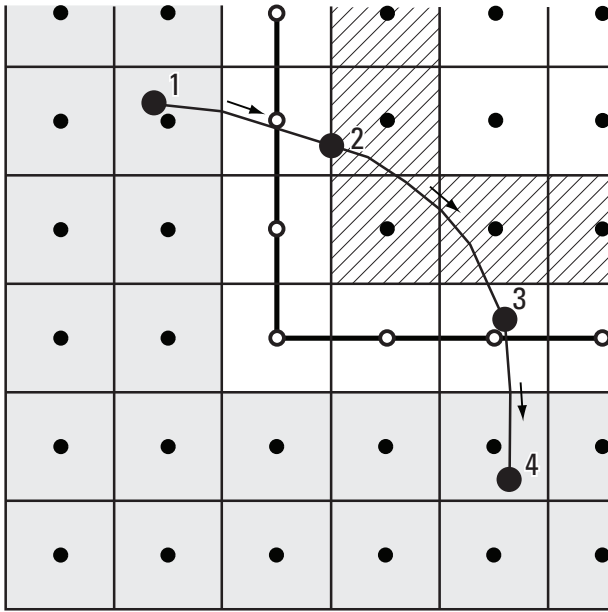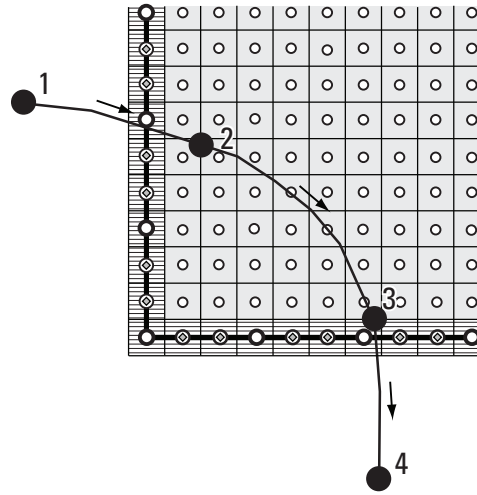
Figure 3. Flowchart showing the steps by MODPATH-LGR for particle tracking within locally refined models.
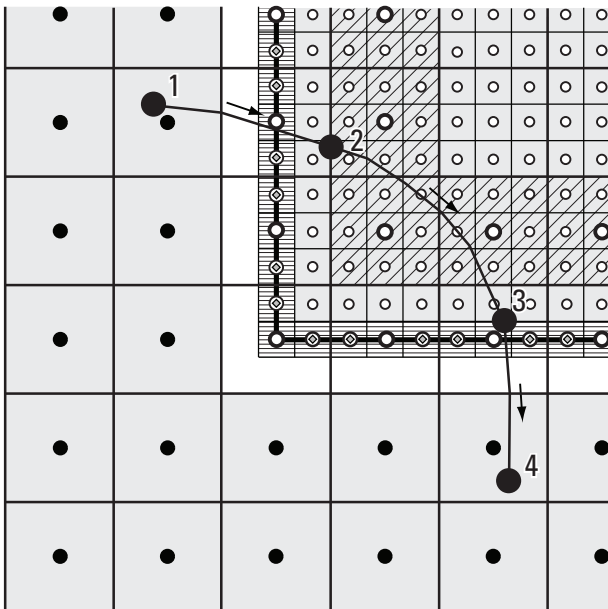
A. Parent-model grid and cell designations.

B. Child-model grid and cell designations.

C. Overlapping parent- and child-model grids and cell designations.

EXPLANATION

● Active parent cell

● Inactive parent cell

○ Active shared parent cell

⬚ Inactive parent cell designated by integer BDYFLG as a parent-transfer cell for MODPATH-LGR

○ Active child cell

◈ Active child cell designated by integer BDYFLG as a child-transfer cell for MODPATH-LGR and by IBFLG for MODFLOW-LGR

◉ Active shared child cell designated by integer BDYFLG as a child-transfer cell for MODPATH-LGR and by IBFLG for MODFLOW-LGR

●1 Particle positions 1–4 and pathline

Figure 4. Example of a particle pathline between parent and child models and cell designations for MODFLOW-LGR and MODPATH-LGR. Panels *A*, *B*, and *C* show only the parent, child, and overlapping model grids, respectively, and the cell designations for each model. The particle travels from (1) an arbitrary location in the parent model to (2) the face of a parent-model cell designated as a particle transfer zone for MODPATH-LGR. The particle leaves the parent model and is started in the child model. The particle travels through the child model to (3) the face of a child-model cell designated as a particle transfer zone, where the particle leaves the child model and is started again in the parent model. The particle travels through the parent model to an arbitrary location (4). Arrows indicate the direction of travel of the particle along the pathline. IBFLG is a negative integer used to define the interface of the child grid with the parent grid.

well in the child model. The system is based on the two-dimensional problem used by Mehl and Hill (2002, 2005) to investigate the accuracy of the shared-node method of grid refinement for heterogeneous systems. The second hypothetical problem contains the same heterogeneity structure and similar boundary conditions as the first problem, but includes additional sources and strong and weak sinks.

Example Problem 1

Example problem 1 demonstrates the use of MODPATH-LGR for tracking particles among simple steady-state parent (regional-scale) and child (local-scale) models. Selected input and output files for example problem 1 are presented in appendix C. Example problem 1 represents a four-layer aquifer system with a single pumping well in the child model (fig. 5). The heterogeneous distribution of the isotropic hydraulic conductivity is based on a laboratory experiment described in Garcia (1995) and Mapa and others (1994). Each of the four layers has identical hydraulic properties. The system has constant-head boundaries of 10.0 meter (m) on the west side and 1.0 m on the east side, and no-flow boundaries at the north and south sides. The well is screened in layers one and two and withdraws at a rate of 2.00×10^{-3} cubic meters per second (m^3/s).

The area of refinement surrounds the well (fig. 5) and extends from the top of layer 1 to the center of layer 3 in the four-layer parent-model grid. The child model has eight layers and refines parent layers 1 and 2 into three layers, and the upper two thirds of parent layer 3 into two layers. All parent-model layers have a uniform thickness of 100 m and all child-model layers have a uniform thickness of 33.3 m; thus the vertical refinement ratio by the child model is consistently 3:1. MODFLOW-LGR allows the vertical refinement to be varied, but this feature is not used in this example. The parent grid has 50 rows and 108 columns with cell dimensions of 9.25 m and 9.0 m in the row and column directions, respectively. The child grid has 100 rows and 154 columns with cell dimensions of 1.028 m and 1.00 m in the row and column directions, respectively, which results in a horizontal refinement ratio of 9:1 in both horizontal directions.

Evaluating the Accuracy in Example Problem 1

The accuracy of the particle tracking routine for multiple models is evaluated by using example problem 1 by comparing particle source locations (simulated as endpoints) to source locations of a "true" solution from a globally refined model. The source locations are simulated by backtracking 324 particles from the pumping well to their endpoints (fig. 5). Example problem 1 also is used to show that higher levels of grid refinement in the child model reduce the error in the source location in a parent model. The globally refined model has the same grid resolution for all cells as the child model, with a 9:1 refinement.

The error reduction for increased levels of grid refinement is demonstrated by using five separate simulations. Each simulation uses a parent model with the same grid resolution, and the four grid refinement ratios are 3:1, 5:1, 7:1, and 9:1. The 1:1 grid refinement ratio (no refinement) is simulated by using only a parent model; whereas, the four refinements use a child model. The backtracked particles start at the faces of the cells containing the well. However, the locations in the x and y directions of the backtracked particles differ for each grid resolution because the cell size, and location of the cell face, differs as well.

Errors in the source locations are presented as the mean of the L_1 norm calculated as

$$ L_1 = \frac{\sum \left| a_i^{"true"} - a_i \right|}{n} \, , $$

where n is the number of source locations compared, $a_i^{"true"}$ is the source location for particle i from the globally refined model, and a_i is the source location for particle i from the model being evaluated.
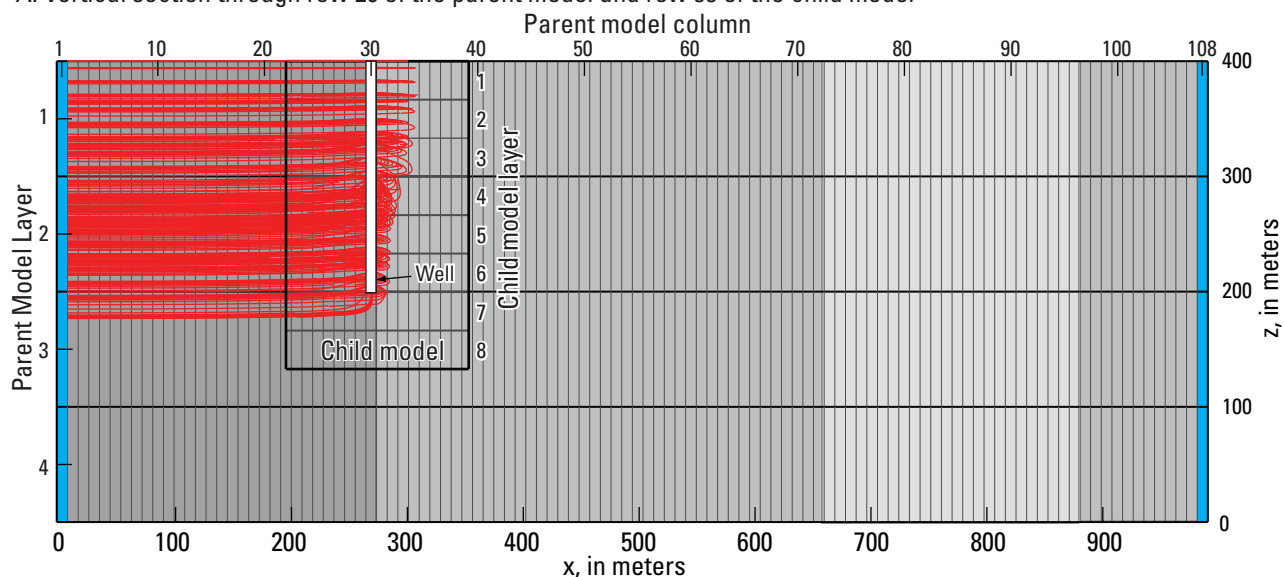
Errors are calculated for source location along the y and z axes. The position along the x axis is not used because all particles stop at the boundary condition at $\times=9.25$ m at the left side of the model (fig. 5).

The mean L_1 norm error of the endpoint locations for position along the y- and z-axes (fig. 6) indicate that the accuracy of the simulated source locations increases with higher levels of horizontal grid refinement. The error is lowest in the coupled models with 7:1 and 9:1 resolutions because the child model represents the steep head gradients near the pumping well more accurately than the coupled models with coarser grids. The flows among cells in the child model, therefore, are more accurate. Fluxes among cells of the child model are of higher resolution, which provides MODPATH-LGR with additional information to calculate curved flow paths that more accurately reflect convergent flow toward the well.
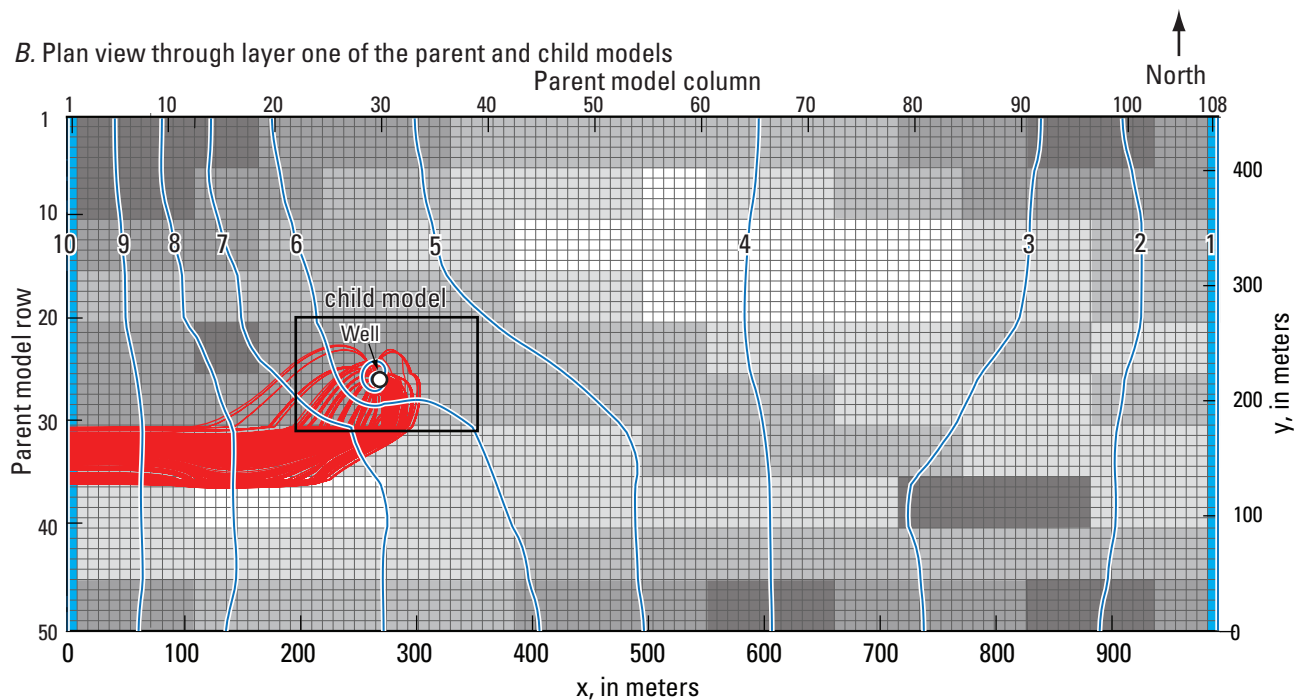
Example Problem 2

Example problem 2 represents a steady-state four-layer aquifer system with a single pumping well in each of two embedded child models (fig. 7). Selected input and output files for example problem 2 are presented in appendix C. The system is modified from example problem 1 by adding additional sources and a second child model with the same amount of grid refinement at a second well. The sources are the detention basin, the pit, and the trench (fig. 7). These additional sources and sinks add complexity to the flow system that results in tracked particles through each of the three models. The system has constant head boundaries of 1.0 m on the right (east) side of all layers, constant head boundaries of 10.0 m along the left (west) side for layers 2–4, and a river along the west side of layer 1 with a stage of 11.0 m. No-flow boundaries
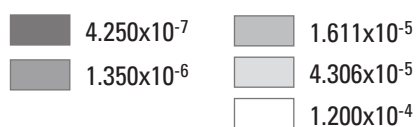
Figure 5. Vertical section (*A*) and plan view (*B*) showing steady-state head contours, heterogeneous transmissivity (same for all layers), and area of local grid refinement of example problem 1 (modified from Mehl and Hill, 2002, 2005). The particle pathlines are projected onto the plane of view through the model grid. In *B*, the heads are shown for both the parent and child models from layer 1. Particles are backtracked from the well in the child model to the source area in the parent model at x = 9.25.
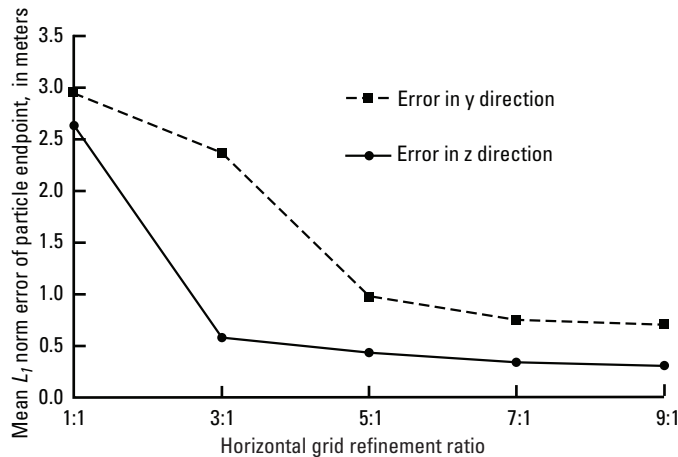
Figure 6. Errors in particle endpoint coordinates in example problem 1 (modified from Mehl and Hill, 2002, 2005) in the y and z axes for five levels of refinement by the child model in comparison to a globally refined model. The globally refined model has the same grid cell size as the child model with a 9:1 refinement.

are along the north and south sides (fig. 7). The elevated river stage results in a recharge rate of .0057 m^3/s through a 1-m thick river bed that has a conductance of 85.6 m^2/s. Additional recharge at the detention basin, trench, and pit is at the rates of 0.142 m^3/s, 1.70×10^{-3} m^3/s, and 2.88×10^{-4} m^3/s, respectively. The two pumping wells penetrate layers one and two of both child models. Each well (fig. 7) pumps at a rate of 2.20×10^{-2} m^3/s in layer 1 and a rate of 2.00×10^{-3} m^3/s in layer 2.

Particles are tracked backward from observation well 1 in the parent model, well 2 in child model 1, and well 3 in child model 2. The backward tracked particle pathlines (fig. 7) for the 1,018 particles result from the inflows and outflows simulated and related head distribution. These backward tracked particles indicate that the water in observation well 1 is from recharge applied at the surface of the detention basin in the upper layer (layer 1). Conversely, the distribution of source types at child model 1 at pumping well 2 is 6 percent from the pit, 1 percent from the trench, 17 percent from recharge at the detention basin, and 76 percent from the river infiltration. Similarly, the distribution of source types at child model 2 at pumping well 3 is 0 percent from the pit, 5 percent from the trench, 92 percent from recharge at the detention basin, and 3 percent from river infiltration. This illustrates the ability of MODPATH-LGR to identify multiple sources of water in observation wells in different coupled models.

Uses and Limitations

Utility of Tracking Particles through Locally Refined Grids

Particle tracking in locally refined models can be used to investigate the effects of well locations, stresses around

pumping wells, and heterogeneity on tracked particles. For example, particle tracking through locally refined models is critical to define travel times between sources and sinks, and the contributing areas around wells and travel times when the flow system is affected by local well hydraulics, heterogeneity, or other local features that cannot be adequately represented at the scale of the parent model (Phillips and others, 2007).

Contributing areas related to streamflow infiltration and exfiltration and spring discharge can be assessed similarly to well-capture zones in regional flow systems. The use of a locally refined model improves the simulation of weak sources and sinks in this application. This technique has been applied to regional capture of streamflow (Leake and others, 2010) but also could be applied to the simulation of other localized sources or sinks, such as spring flow in the regional flow system of Death Valley (Faunt and others, 2004).

Particle tracking, in combination with local-grid refinement by using these programs, provides the capabilities to represent essential features of the flow system in more detail, and provides a computationally efficient way to link particles between parent and child models. Particle tracking through locally refined models can be used to calibrate groundwater models, perform sensitivity and uncertainty analyses, and other types of model analysis with universal modeling programs such as UCODE-2005 (Poeter and others, 2005), PEST (Doherty, 2005), or OSTRICH (Matott, 2005).

Limitations

Particle tracking applied for locally refined models has several notable limitations, in addition to the limitations for particle tracking in a single model (Pollock, 1994). The limitations are related to the methods for transferring particles among the models, and the effects of different discretizations by the coupled models.

The method of transferring particles across the boundary between the parent and child models results in approximate locations where particles begin tracking through a child model. The locations are approximate because the particles are moved to those locations by using the adjusted fluxes for shared-parent cells. The fluxes parallel to the boundary also are reduced to account for the partial conductances between shared parent-model cells along the boundary. The fluxes that move the particles happen through the face of the shared parent-model cell in the active flow domain of a child model— these fluxes are not calculated when using shared-nodes. MODPATH-LGR approximates the fluxes for the two situations as follows: (1) the flux across the face of the shared parent-model cell is set equal to the sum of the flow across the overlapping faces of the child-model cells; (2) flows perpendicular to the boundary are increased by a factor related to the area of the shared parent-model cell face (fig. 8). This procedure produces flows that only approximate a mass balance calculation for the parent cells along the interface. The flows approximated by these methods may not adequately

represent the flow fields in areas of steep hydraulic gradients, or where flow directions change rapidly. The approximation in these situations may be a major source of error in the particle locations and travel times. These situations may be avoided by placing the boundary between locally refined models in areas of gradual changes in hydraulic gradients. Errors are expected to be small as long as the hydraulic-conductivity field is represented at the same scale in the adjoining models for a few parent-cell widths into the child model. This practice is suggested by Mehl and Hill (2005), and the models considered in this report represent heterogeneity in the parent and child models at the same scale. The alternative ghost-node method for coupling models (Dickinson and others, 2007) can more accurately simulate this transfer because the grid cells have adjoining cell faces instead of cell centers. The ghost-node method eliminates the need to adjust the

fluxes for the shared parent-model cells. Future versions of MODPATH-LGR may have the option of tracking particles through locally refined models that are coupled by using the ghost-node method.

Numerous transfers of particles between the coupled models can increase the error in the location and travel time of the particle. Numerous transfers can happen if a particle pathline is tracked in the shared cells along a trajectory that is parallel to the boundary between the models. The transfers can result from minor differences in the simulated flow fields by the parent and child models that "push" the particle back and forth. Differences in the flow fields arise from different spatial and temporal discretizations of the model grids, heterogeneities, and model features. These errors may be reduced by representing the model heterogeneities and features at similar scales along the model boundaries.
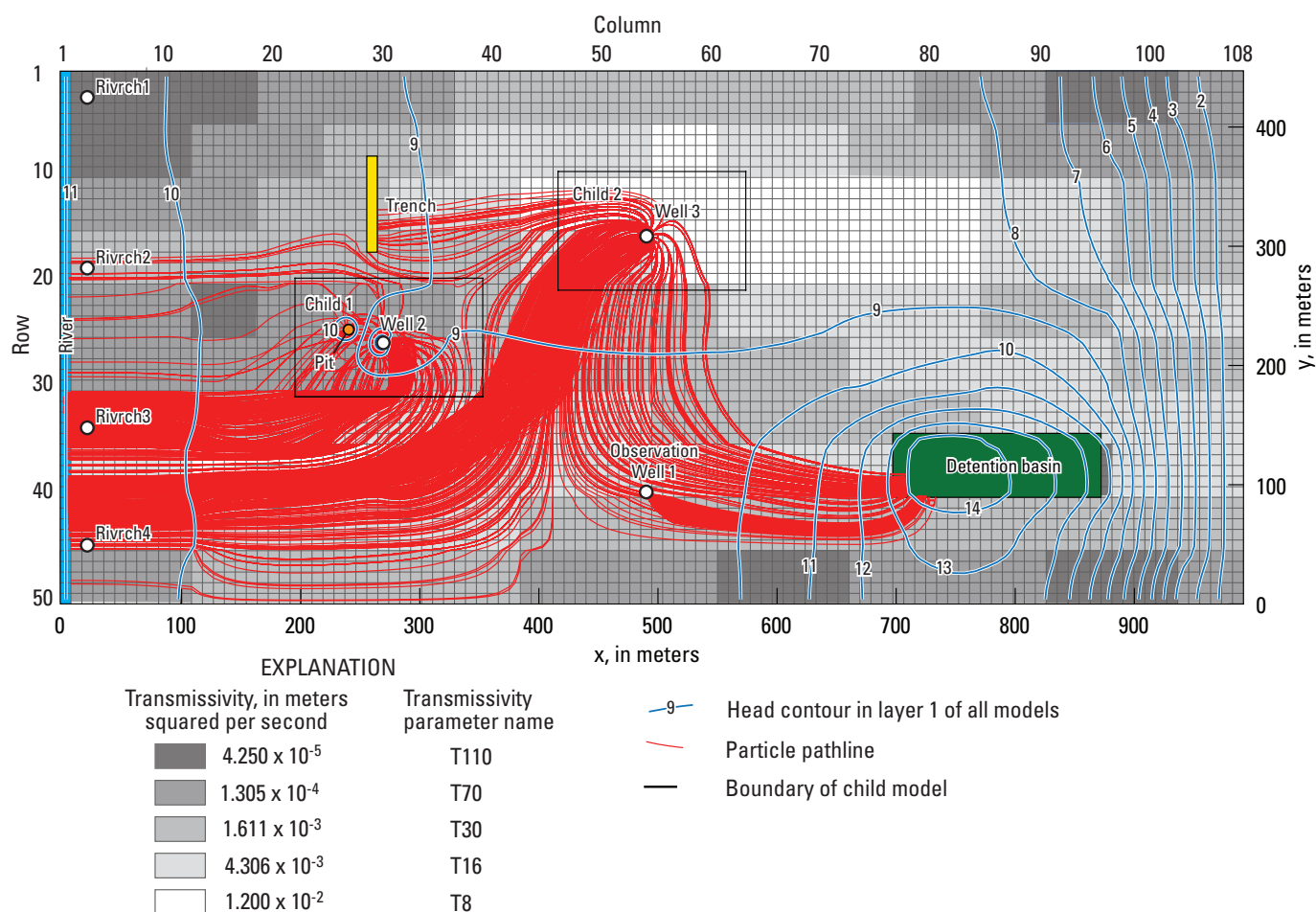
Figure 7. Map showing steady-state head contours, heterogeneity pattern, and area of local-grid refinement for each layer of example problem 2 (modified from Mehl and Hill, 2002 by adding child model 2, wells 1 and 3, the river, the trench, the pit, and the detention basin). Layers 1, 2, and 3 of the parent model are refined by the child models. Particles are backtracked from the wells in the child models to the source areas (the river, the trench, the pit, and detention basin).

Acknowledgments

References Cited

Anderman, E.R., and Hill, M.C., 2001, MODFLOW-2000, the U.S. Geological Survey modular ground-water model —Documentation of the advective-transport observation (ADV2) package, version 2: U.S. Geological Survey Open-File Report 01–54, 69 p.

Anderman, E.R., Hill, M.C., and Poeter, E.P., 1996, Two-dimensional advective transport in ground-water flow parameter estimation: Ground Water, v. 34, no. 6, p. 1,001–1,009.

Clark, B.R., Landon, M.K., Kauffman, L.J., and Hornbreger, G.Z., 2007, Simulations of ground-water flow transport, age, and particle tracking near York, Nebraska, for a study of transport of anthropogenic and natural contaminants (TANC) to public supply wells: U.S. Geological Survey Scientific Investigations Report 2007–5068, 55 p.

Dickinson, J.E., James, S.C., Mehl, S.W., Hill, M.C., Leake, S.A., Zyvoloski, G.A., Faunt, C.C., and Eddebbarh, A.A., 2007, New ghost-node method for linking different models and initial investigations of heterogeneity and nonmatching grids: Advances in Water Resources, v. 30, no. 8, p. 1,722–1,736.

Doherty, J., 2005, PEST—Software for model-independent parameter estimation: Watermark Numerical Computing, Brisbane, Australia, 336 p.

Faunt, C.C., Blainey, J.B., Hill, M.C., D'Agnese, F.A., and O'Brien, G.A., 2004, Transient numerical model, in Belcher, W. ed., Death Valley regional ground-water flow system, Nevada and California—Hydrogeologic framework and transient ground-water flow model: U.S Geological Survey Scientific Investigations Report 2004–5205, chap. F, p. 257–362.

Franke, O.L, Reilly, T.E., Pollock, D.W., and LaBaugh, J.W., 1998, Estimating areas contributing recharge to wells, lessons from previous studies: U.S. Geological Survey Circular 1174, 14 p.

Garcia, J.E., 1995, An experimental investigation of upscaling of water flow and solute transport in saturated porous media: Boulder, Colorado, University of Colorado, M.S. thesis, 135 p.

Graham, M.F., and Smart, G.T., 1980, Reservoir simulator employing a fine-grid model nested in a coarse-grid model—Paper SPE 9372 presented at the 55th annual Society of Petroleum Engineers Fall Technical Conference, Sept. 21-24, 1980, Dallas, Tex.,: Society of Petroleum Engineers.

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

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

Keating, E.H., Vesselinov, V.V., Kwicklis, E., and Lu, Z., 2003, Coupling basin- and site-scale inverse models of the Española Aquifer, Ground Water, v. 41, no. 2, p. 200–211.

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.

Leake, S.A., Lawson, P.W., Lilly, M.R., and Claar, D.V., 1998, Assignment of boundary conditions in embedded ground water flow models: Ground Water, v. 36, no. 4, p. 621–625.

Leake, S.A., Reeves, H.W., and Dickinson, J.E., 2010, A new capture fraction method to map how pumpage affects surface water flow: Ground Water, v. 48, no. 5, p. 690–700.

Mapa, R., Illangasekare, T.H., and Garcia, J.E., 1994, Upscaling of water flow and solute transport in saturated porous media: theory, computation, and experiments: Progress report submitted to the U.S. Army Waterways Experiment Station, 184 p.

Matott, L.S., 2005, OSTRICH—An optimization software tool; documentation and user's guide, version 1.6: State University of New York at Buffalo, 114 p.

McDonald, M.G., and Harbaugh, A.W., 1988, A modular three-dimensional finite difference ground-water flow model: U.S. Geological Survey Techniques in Water Resources Investigations, Book 6, chap. A1, 548 p.

Mehl, S.W., Faunt, C.C., Lacznaik, R., Li, Z., and Hill, M.C., 2006a, Examination of ground-water pumping effects using regional-scale, site-scale, and locally refined numerical models—Paper presented at MODFLOW and MORE 2006: Managing ground water systems, May 21-24, 2006, Golden, Col., International Ground Water Modeling Center, p 529–533.

Mehl, S.W., and Hill, M.C., 2002, Development and evaluation of a local grid refinement method for block-centered finite-difference groundwater models using shared nodes: Advances in Water Resources, v. 25, no. 5, p. 497–511.

Mehl, S.W., and Hill, M.C., 2003, Locally refined block-centered finite-difference groundwater models—Evaluation of parameter sensitivity and the consequences for inverse modeling, *in* Kovar, K., and Hrkal, Z., eds, Calibration and reliability in groundwater modeling—A few steps closer to reality: IAHS Publication 277, p. 227–232.

Mehl, S.W., and Hill, M.C., 2004, Three-dimensional local grid refinement for block-centered finite-difference ground-water models using iteratively coupled shared nodes: Advances in Water Resources, v. 27, p. 899–912.

Mehl, S.W., and Hill, M.C., 2005, MODFLOW-2005, The U.S. Geological Survey modular ground-water model – Documentation of Shared Node Local Grid Refinement (LGR) and the Boundary Flow and Head (BFH) package: U.S. Geological Survey Techniques and Methods 6–A12, 68 p.

Mehl, S.W., and Hill, M.C., 2007, MODFLOW-2005, the U.S. Geological Survey modular ground-water model: documentation of the multiple-refined-areas capability of Local Grid Refinement (LGR) and the Boundary Flow and Head (BFH) package: U.S. Geological Survey Techniques and Methods 6-A21, 13 p.

Mehl, S.W., Hill, M.C., and Leake, S.A., 2006b, Comparison of local grid refinement methods for MODFLOW: Ground Water, v. 44, no. 6, p. 792–796.

Phillips, S.P., Green, C.T., Burow, K.R., Shelton, J.L, and Rewis, D.L., 2007, Simulation of multiscale ground-water flow in part of the northeastern San Joaquin Valley, California: U.S. Geological Survey Scientific Investigations Report 2007–5009, 43 p.

Poeter, E.P., Hill, M.C., Banta, E.R., Mehl, S., and Christensen, S., 2005, UCODE_2005 and six other computer codes for universal sensitivity analysis, calibration, and uncertainty evaluation: U.S. Geological Survey Techniques and Methods 6-A11, 283 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, 249 p.

Reichard, E.G., Land, M., Crawford, S.M., Johnson, T., Everett, R.R., Kulshan, T.V., Ponti, D.J., Halford, K.L., Johnson, T.A., Paybins, K.S., and Nishikawa, T., 2003, Geohydrology, geochemistry, and ground-water simulation-optimization of the Central and West Coast basins, Los Angeles County, California: U.S. Geological Survey Water-Resources Investigations Report 03–4065, 184 p.

Reynolds, D.A., and Marimuthu, S., 2007, Deuterium composition and flow path analysis as additional calibration targets to calibrate groundwater flow simulation in a coastal wetlands system: Hydrogeology Journal, v. 15, no. 3, p 515–535.

Sanford, W.E., Plummer, L.N., McAda, D.P., Bexfield, L.M., and Anderholm, S.K., 2003, Use of environmental tracers to estimate parameters for a predevelopment ground-water flow model of the Middle Rio Grande basin, New Mexico: U.S. Geological Survey Water Resources Investigations Report 03–4286, 102 p.

Sanford, W.E., Plummer, L.N., McAda, D.P., Bexfield, L.M., and Anderholm, S.K., 2004, Hydrochemical tracers in the middle Rio Grande Basin, USA—2. Calibration of a groundwater-flow model: Hydrogeology Journal, v. 12, no. 4, p. 389–407.

Székely, F., 1998, Windowed spatial zooming in finite-difference ground water flow models: Ground Water, v. 36, no. 5, 718–721.

von Rosenberg, D.U., 1982, Local mesh refinement for finite difference methods, paper SPE 10974 presented at the 57th Society of Petroleum Engineers Annual Fall Technical Conference, New Orleans, La., Sept. 26–29, 1982: Society of Petroleum Engineers.

This page left blank intentionally.

Appendixes A–D

APPENDIX A. Input Instructions

This section gives instructions for creating MODPATH-LGR input files, which include:

1. A new MODPATH-LGR control file that contains information about the parent and all child models. This file is short and is generally created by the user.
2. MODPATH-LGR response files for the regional grid and each of the child grids. The response files are created by using MODPATH-LGR or a graphical user interface, and have identical appearance to the response files read by and produced by MODPATH (Pollock, 1994).
3. MODPATH-LGR name files for each model. These files are prepared by the user or a graphical user interface, and have an identical appearance to the name files read by MODPATH (Pollock, 1994).
4. MODPATH-LGR main data files for the regional grid and for each of the child-model grids. These files are prepared by the user or a graphical user interface, and have an identical appearance to the main data files used by MODPATH (Pollock, 1994).
5. One or more files defining particle starting locations. One file is needed for each model in which particle paths begin. These files are nearly identical to the particle starting locations files produced by MODPATH (Pollock, 1994). The differences are two identifiers for each particle that are in the last two columns to the right: (a) an integer particle identifier (variable IDPART) is needed for each particle, and (b) an optional character identifier (variable PARTAS) for each particle for the user to describe any attribute of the particle. These identifiers are added manually.

Input instructions here are detailed for new input variables and are abbreviated for variables described by Pollock (1994).

MODPATH-LGR Control File Input Instructions

When executed, MODPATH-LGR prompts for the name. If the user supplies a file name that is a MODPATH-LGR control file, particle tracking begins for two or more models. The names of response files for each model are listed in the MODPATH-LGR control file. If the file name is a response file for a single model, MODPATH-LGR tracks particles through only that model. A MODPATH-LGR control file is distinguished by the name of a MODFLOW-LGR control file on the first line. A response file for a single model is distinguished by the string @[MODPATH-LGR or @[MODPATH on the first line of the file. The contents of a response file for a single model are described by Pollock (1994) and are discussed briefly in this report. The contents of the MODPATH-LGR control file are as follows:

   FOR EACH SIMULATION
1. FNLGR
2. RSPPARENT
3. FNBUD
      Repeat items 4 and 5 for each child model
4. RSPCHILD
5. BDYFLG
      Repeat item 6 for each global output file
6. FNGLOB   IUGLOB   FNAME

Explanation of Variables in the MODPATH-LGR Control File

FNLGR— is the MODFLOW-LGR control file name. This file name is read as a character string and can be up to 80 characters in length.

RSPPARENT—is the file name of the response file for the parent model. This file name is read as a character string and can be up to 80 characters in length. This file is typically generated by running MODPATH-LGR for only the parent model, and is identical to a response file read by and produced by MODPATH, which is described by Pollock, 1994.

FNBUD—is the file name of the budget file in the parent-model name file. This file name is read as a character string and can be up to 80 characters in length. This file is read to adjust the fluxes through the faces of the parent-model cells that are interior to the domain of the child model.

RSPCHILD—is the file name of the response file for a child model. This file name is read as a character string and can be up to 80 characters in length. This file is typically generated by running MODPATH-LGR for only a child model, and is identical to a response file used by MODPATH (Pollock, 1994).

BDYFLG—is an integer used to identify the transfer cells in each model (fig. 4). These zones are used to identify the cells where the particle is stopped and restarted in the next model. BDYFLG must be assigned to the transfer cells in the IBOUND array listed in MODPATH-LGR main data files. BDYFLG may (but is not required) be included in the IBOUND array that is read by the parent and child models. BDYFLG must be a different value for each model. The parent-transfer cells are in the child model(s) and one cell interior to the shared parent cells. The child-transfer cells are the shared cells along the outside boundary of the model. If MODPATH-LGR and MODFLOW-LGR read different IBOUND arrays for the same child model, BDYFLG for the child model can be a different value for IBFLG. IBFLG is a variable read by MODFLOW-LGR that identifies the cells along the child boundary.

FNGLOB—is the file type keyword for output of global particle results. FNGLOB may be entered in any combination of upper-case or lowercase. The type of output must be included in the MODPATH-LGR name file for each model, except for the endpoint output that is created automatically for all simulations. This file name is read as a character string and can be up to 80 characters in length. The following file type keywords in bold are allowed:

> ***ENDPOINT*** for a global endpoint file
>
> ***PATHLINE*** for a global pathline file
>
> ***TIME-SERIES*** for a global time-series file
>
> ***ADVOBS*** for a global particle-projection file

IUGLOB—is an integer greater than zero that corresponds to the unit number where global endpoint, pathline, time-series, or particle-projection outputs are saved.

FNAME—is the name of the file, which is a character value. Pathnames may be specified as part of FNAME.

Response Files for the Parent and Child Models

Response files are required when tracking particles through coupled models, and typically end with the file extension .rsp for all models listed in the MODPATH-LGR control file. The response files contain responses to the interactive input process that is generated when running MODPATH or MODPATH-LGR for only one model. MODPATH-LGR obtains many user options from the response files such as the type of output, options for tracking particles through weak sinks, and forward or backward tracking (Pollock, 1994). In MODPATH, the response file is used to run in batch mode. The response file allows MODPATH-LGR to run in a similar batch mode for each model.

When running MODPATH-LGR for parent and child models, interactive input is not allowed. Generally, the individual response files are generated in separate runs of MODPATH or MODPATH-LGR for each model, and saved for later use to track particles in multiple models. MODPATH-LGR has the capability to generate particle starting locations in the same manner as MODPATH (Pollock, 1994) only for single-model runs. The response files for coupled models must indicate that the particle starting locations are read from an existing data file.

MODPATH-LGR Name Files for Each Model

MODPATH name files described by Pollock (1994) must be specified for each model in the response file for that model. The name file contains the names of most of the input and output files used in a MODFLOW-LGR and MODPATH-LGR simulation and the file requires the same format as the name file for MODPATH. The starting locations file, and the endpoint, pathline, time-series, and particle projection output files are specified in the name file. The name file for each model must contain a file name for the same type output as the global output for all the models. Integers 75–99 are reserved for MODPATH-LGR inputs and cannot be used for specifying unit numbers for any files in the MODPATH-LGR name file.

MODPATH-LGR Main Data Files for Each Model

MODPATH-LGR main data files in the format described by Pollock (1994) must be specified in the name file for each model. The format of the file is identical to a main data file used by MODPATH (Pollock, 1994).

Starting Locations Files for Any Model in Which Particles Start

A starting locations file for particle-tracking simulations in locally refined models is required for each model in which particles start. Starting locations for particle-tracking simulations in a single model can be generated automatically or read from a file. These options are selected in the response file, and the names of the particle locations file is specified in the MODPATH-LGR name files for each model. Particles that start in the region that is simulated by a child model need to be in a starting locations file for that child model. Particles that start in the region that is simulated by a child model but specified in the starting locations file for the parent model are NOT automatically tracked in the child model and will be ignored.

The starting locations files used by MODPATH-LGR are similar to those files read by MODPATH, except that one required and one optional item of information are added for each particle. For additional information about the starting locations file see Pollock (1994). All indices and coordinates are relative to the model of the particle origination. The origin of the J, I, and K indices is the cell at the northwestern corner of the uppermost layer. The origin of the X, Y, and Z coordinates is at the southwestern corner at the bottom of the lowest layer. The two additional items of information, IDPART and PARTAS, would subsequently need to be appended to the input line for each particle by the user if the starting locations are generated by using the particle generation routine.

These instructions are modified from instructions presented by Pollock (1994, p. A-22 to A-23). The following record is repeated for each starting particle.

1.   J   I   K   X   Y   Z   JCODE   ICODE   KCODE   TRELEAS   IDPART   PARTAS

J—is the column index of the cell containing the particle.

I—is the row index of the cell containing the particle.

K—is the layer index of the cell containing the particle. If K=0, K is set equal to the top most active model layer at the specified row and column location.

X—is the x-coordinate of the particle in the cell. The origin of the x-coordinate is specified by JCODE.

Y—is the y-coordinate of the particle. The origin of the y-coordinate is specified by ICODE.

Z—is the z-coordinate of the particle. The origin of the z-coordinate is specified by KCODE.

JCODE—indicates whether X is relative to the origin of the individual model or relative to a single model cell containing the particle. JCODE cannot be omitted as in MODPATH.

   JCODE=0; X is a local coordinate within the range 0 to 1 in a cell in column J. The value of the x-coordinate of the particle relative to the origin of the individual model is computed by MODPATH-LGR.

   JCODE=1; X is a coordinate relative to the origin of the individual model, and the particle is in a cell in column J. If the value of X lies outside the range of cells in column J, the particle is discarded.

   JCODE=2; X is a coordinate relative to the origin of the individual model. The column number J, that contains X, is computed by MODPATH from the specified value of X relative to the origin of the individual model and grid spacing data. The value of J read as input is ignored.

ICODE—indicates whether Y is relative to the origin of the individual model or relative to a single model cell containing the particle. ICODE cannot be omitted as in MODPATH.

   ICODE=0; Y is a local coordinate within the range 0 to 1 in a cell in row I. The global value of the y-coordinate of the particle is computed by MODPATH-LGR.

   ICODE=1; Y is a coordinate relative to the origin of the individual model, and the particle is in a cell in row I. If the value of Y lies outside the range of cells in row I, the particle is discarded.

   ICODE=2; Y is a coordinate relative to the origin of the individual model. The row number, I, that contains Y is computed by MODPATH-LGR from the specified value of Y relative to the origin of the individual model and grid spacing data. The value of I read as input is ignored.

KCODE—indicates whether Z is a relative to the origin of the individual model or relative to a single model cell containing the particle. KCODE cannot be omitted as in MODPATH.

KCODE=0; Z is a local coordinate within the range 0 to 1 in a cell in layer K. The value of the z-coordinate of the partical relative to the origin of the individual model is computed by MODPATH-LGR. If the layer includes an underlying quasi-3D global value of the z-coordinate of the particle is computed by MODPATH-LGR. If the layer includes an underlying quasi-3D confining layer, starting locations in the confining layer are specified by using values between -1 (bottom of the confining layer) to 0 (top of confining layer and bottom of model layer).

KCODE=1; Z is a coordinate relative to the origin of the individual model, and particle is in a cell in layer K. If the value of Z lies outside the range of cells in layer K, the particle is discarded. This option is only available for true rectangular grids (IGRID=1). If KCODE=1 and IGRID=0, the particle is discarded.

KCODE=2; Z is coordinate relative to the origin of the individual model. The layer number, K, that contains Z is computed by MODPATH from the specified value of Z relative to the origin of the individual model and grid spacing data. The value of K read as input is ignored. This option is only available for true rectangular grids (IGRID=1). If KCODE=2 and IGRID=0, the particle is discarded.

TRELEAS—is the release time for the particle. The release time is measured relative to the reference value of simulation time defined by the user (Pollock, 1994, chap. 3). Particles released at the reference simulation time have a release time equal to 0. The release time has the same time units as used in the MODFLOW simulation.

IDPART—is an identifier that must be unique to each particle, regardless of which model (and particle location file) initially contains the particle.

PARTAS—is an optional 40 character string provided for users to label particles with additional information (for example, well number). Any number, letter, or special character can be used in any order. Blanks are not allowed.

APPENDIX B. MODPATH-LGR Output

MODPATH-LGR generates global endpoint, pathline, and time-series output files for all grids in addition to the output options available in MODPATH. The global output files contain information about the tracking times and particle positions for all particles relative to the regional coordinate system of the parent model. The output files for individual models contain information relative to the local coordinate system of the individual model.

Endpoint File

An endpoint output file contains the particle endpoints (the initial and final locations) and the tracking times. An endpoint file is generated for all MODPATH-LGR runs. If the endpoint output data file is not specified in the MODPATH-LGR name file, an endpoint file named <endpoint> or <endpoint.bin> is automatically generated. The automatic endpoint file corresponds to the single model for simulations with a single model. The automatic endpoint file corresponds to the last model in which particles were tracked for runs with coupled models. Otherwise, an endpoint file is produced for each model that indicates an endpoint data file in the MODPATH-LGR name file. All models should specify the endpoint output type and the ENDPOINT keyword and file name must be included in the MODPATH-LGR control file to produce a global endpoint file. If a model does not specify the endpoint output, the final positions and travel times that stop in the model will not be recorded in the global output file.

Endpoint File for Individual Models

MODPATH-LGR produces an endpoint file for a single model if the simulation involves one model, and an endpoint for each model for simulations involving multiple models. The format and contents of the endpoint file are identical to the format and contents generated by MODPATH, with the exception of several additional data items at the end of each record. Standard text, compact text, and binary output structures from MODPATH are preserved. MODPATH-LGR produces a standard text endpoint file by default. Compact and binary formats are specified in the MODPATH-LGR main data file (Pollock, 1994).

Additional endpoint files are automatically produced for each model for simulations involving multiple models and for each MODPATH-LGR sequence. These files can be used to examine particle locations and tracking times instead of using the global endpoint file. The naming convention of the files is:

PARENT_ENDPOINT_X.DAT
CHILD_Y _ ENDPOINT_X.DAT

X is the sequence number and Y is the identification number of the child model that contains the particle endpoint. Y equals 1 for the first child model listed in the MODPATH-LGR control file, and Y increases by one for each subsequent child model.

The first line of the endpoint file is a header record indicating the MODPATH-LGR version number and the reference simulation time value used in the MODPATH-LGR run. The header record has the form:

@ [MODPATH-LGR Version 1.00 (Release 1, 1-11) (TREF= 0.000000E+00)]

Standard Text Endpoint File for Individual Models

MODPATH-LGR produces a standard text endpoint file by default. The standard text format includes the data items produced by MODPATH and 3 additional terms in columns 20, 21, and 22. The header record and any comment lines are followed by particle data records that contain the following data items in the order listed:

1. Zone code for the model cell containing the final location of the particle

2. J (column) index for the model cell containing the final location

3. I (row) index for the model cell containing the final location

4. K (layer) index for the model cell containing the final location

5. Local coordinate in the x-direction (J index direction) for the final location

6. Local coordinate in the y-direction (I index direction) for the final location

7. Local coordinate in the z-direction (K index direction) for the final location

8. Relative coordinate in the z-direction in the cell (0 to 1 within the model layer, -1 to 0 within an underlying confining layer)

9. Total tracking time

10. Local coordinate in the x-direction for starting location

11. Local coordinate in the y-direction for starting location

12. Relative coordinate in the z-direction in the cell (0 to 1 within the model layer, -1 to 0 within an underlying confining layer) for starting location

13. J index for model cell containing starting location

14. I index for model cell containing starting location

15. K index for model cell containing starting location

16. Zone code for model cell containing starting location

17. Cumulative MODFLOW time step number corresponding to the time of release

18. Particle termination code, IPCODE

19. Release time

20. Local coordinate in the z-direction (K index direction) for the starting location

21. Particle identification number IDPART assigned in the starting locations file

22. Particle attribute PARTAS specified by the user in the starting locations file

Compact Text Endpoint File for Individual Models

MODPATH-LGR produces a compact text endpoint file whenever the keyword COMPACT is specified in record 2 of the MODPATH-LGR main data file. The compact text format includes the data items produced by MODPATH and two additional terms in columns 15 and 16. The compact text format uses grid cell locations as a node number instead of the column J, row I, and layer K indices. The node number N is computed as:

N=(K-1)*NCOL*NROW+(I-1)*NCOL+J,

where NCOL is the number of columns, and NROW is the number of rows.

The compact text format omits the regional coordinate in the z-direction of the final particle location. The header record and any comment lines are followed by particle data records that contain the following data items in the order listed:

1. Zone code for the cell of the model containing the final location of the particle

2. Node number for the cell of the model containing the final location

3. Local coordinate in the x-direction (J index direction) for the final location

4. Local coordinate in the y-direction (I index direction) for the final location

5. Relative coordinate for the z-direction in the grid cell

6. Total tracking time

7. Local coordinate in the x-direction for starting location

8. Local coordinate in the y-direction for starting location

9. Relative coordinate in the z-direction in the cell for starting location

10. Node number for the model cell containing starting location

11. Zone code for model cell containing starting location

12. Cumulative MODFLOW time step number corresponding to the time of release

13. Particle termination code, IPCODE

14. Release time

15. Particle identification number IDPART assigned in the starting locations file

16. Particle attribute PARTAS specified by the user in the starting locations file

Binary Endpoint File for Individual Models

MODPATH-LGR produces a binary endpoint file whenever the keyword BINARY is specified in record 2 of the MODPATH-LGR main data file. The binary format includes the data items produced by MODPATH and two additional terms in columns 15 and 16.

The header record includes an 80-character text string describing the MODPATH version number followed by the reference simulation time used in the MODPATH run. The following segment of code shows how to read the header record in a FORTRAN program:

```
CHARACTER*80 TEXT
READ(50) TEXT,TREF
```

The header record is followed by the particle data records. The items in the data records in the binary endpoint file are identical to the endpoint file for individual models that use the compact text option.

Global Endpoint File for Coupled Models

The contents and format of the global endpoint file are identical to the endpoint file that is generated for individual models, with the exception that all regional particle coordinates are in the system of the parent model, and two additional data items per record. The format options for the global endpoint file are standard text or binary format, which are specified in the MODPATH-LGR main data file for each model. The abbreviated output of the compact format available in MODPATH is not available for the global endpoint file because the compact format does not include regional vertical (z direction) coordinates that are likely to be of interest. The standard format is used for the global endpoint file if the compact format option is specified in the main data files. Each model should use the same output format in the MODPATH-LGR main data files.

The first line of the global endpoint file is a header record indicating the MODPATH-LGR version number and the reference simulation time value used in the MODPATH-LGR run. The header record has the form:
@ [MODPATH-LGR Version 1.00 (Release 1, 1-11) (TREF= 0.000000E+00)]

Standard Text Global Endpoint File

MODPATH-LGR produces a standard text global endpoint file by default. Binary format is specified in the MODPATH-LGR main data file (Pollock, 1994). The header record and any comment lines are followed by particle data records that contain the following data items in the order listed:

1. Zone code for the cell containing the final location of the particle

2. J (column) index for the model cell containing the final location

3. I (row) index for the model cell containing the final location

4. K (layer) index for the model cell containing the final location

5. Regional coordinate in the x-direction (J index direction) for the final location

6. Regional coordinate in the y-direction (I index direction) for the final location

7. Regional coordinate in the z-direction (K index direction) for the final location

8. Relative coordinate in the z-direction in the grid cell (0 to 1 within the model layer, -1 to 0 within an underlying confining layer)

9. Total tracking time

10. Regional coordinate in the x-direction for starting location

11. Regional coordinate in the y-direction for starting location

12. Coordinate in the z-direction in the cell for starting location

13. J index for model cell containing starting location

14. I index for model cell containing starting location

15. K index for model cell containing starting location

16. Zone code for model cell containing starting location

17. Cumulative MODFLOW time step number corresponding to the time of release

18. Particle termination code, IPCODE

19. Release time

20. Regional coordinate in the z-direction (K index direction) for the starting location

21. Identification number IDMODEL_START of the model where the particle started. For particles that start in the parent model, IDMODEL_START equals one. IDMODEL_START equals two if the particle starts in the first child model listed in the MODPATH-LGR control file. IDMODEL_START increases by one for each additional child model listed in the MODPATH-LGR control file.

22. Identification number IDMODEL of the model having the final particle endpoint. The parent model is number 1 and each subsequent child model specified in the MODPATH-LGR control file is numbered in ascending order beginning with number 2.

23. Particle identification number IDPART assigned in the starting locations file

24. Particle attribute PARTAS specified by the user in the starting locations file

Binary Global Endpoint File

MODPATH-LGR produces a binary endpoint file whenever the keyword BINARY is specified in record 2 of the MODPATH-LGR main data file. The binary version of the endpoint file contains a header record that includes an 80-character text string describing the MODPATH version number followed by the reference simulation time used in the MODPATH run. The following segment of code shows how to read the header record in a FORTRAN program:

```
CHARACTER*80 TEXT
READ(50) TEXT,TREF
```

The header record is followed by the particle data records. The particle data records are stored by the node number N for the cell of the model containing the final location instead of the column J, row I, and layer K indices. The node number N is computed as:

$$N = (K-1)*NCOL*NROW + (I-1)*NCOL + J,$$

where NCOL is the number of columns, and NROW is the number of rows.

The items in the data records in the binary global endpoint file are identical to the data records in the global endpoint file when the standard text option is used.

Pathline File

Coordinates and times along the path of each particle are saved to a pathline file if pathline output mode is specified in the response files. A pathline file is only generated for the models that indicate a pathline data file in the MODPATH-LGR name file. All models should specify the pathline output type and the PATHLINE keyword and file name must be included in the MODPATH-LGR control file to produce a global pathline file. If a model does not specify the pathline output, the particle positions will not be recorded in the global output file. The content of the pathline file for individual models is described in detail in the MODPATH documentation (Pollock, 1994) and is omitted here.

Pathline Files for Individual Models

MODPATH-LGR produces a pathline file for a single model if the simulation involves one model, and a pathline file for each model for simulations involving multiple models. The format and contents of the pathline file are identical to the format and contents generated by MODPATH, with the exception of several additional data items at the end of each record. Standard text, compact text, and binary output structures from MODPATH are preserved. MODPATH-LGR produces a standard text pathline file by default. Compact and binary formats are specified in the MODPATH-LGR main data file (Pollock, 1994).

Additional pathline files are automatically produced for each model for each MODPATH-LGR sequence for simulations involving multiple models. These files can be used to examine particle locations and tracking times instead of using the global pathline file. The naming convention of the files is as follows:

PARENT_PATHLINE_X.DAT
CHILD _Y_PATHLINE_X.DAT

X is the sequence number and Y is the identification number of the child model that contains the particles. Y equals 1 for the first child model listed in the MODPATH-LGR control file, and Y increases by one for each subsequent child model.

The first line of the pathline file is a header record indicating the MODPATH-LGR version number and the reference simulation time value used in the MODPATH-LGR run. The header record has the form:
@ [MODPATH-LGR Version 1.00 (Release 1, 9-08) (TREF= 0.000000E+00)]

Standard Text Pathline File for Individual Models

MODPATH-LGR produces a standard text pathline file by default. The standard text format includes the data items produced by MODPATH and two additional terms in columns 11 and 12. The header record and any comment lines are followed by particle data records that contain the following data items in the order listed:

1. Particle index number for the set of particles beginning in the same model. The particle index number equals 1 for the first particle listed in the starting locations file, and increases by a unit for each subsequent particle. For simulations with multiple models, the particle index number resets to 1 for the starting particle for each model.

2. Local coordinate in the x-direction

3. Local coordinate in the y-direction

4. Relative coordinate in the z-direction in the grid cell (0 to 1 within the model layer, -1 to 0 within an underlying confining layer)

5. Local coordinate in the z-direction

6. Cumulative tracking time

7. J index of model cell containing the particle

8. I index of model cell containing the particle

9. K index of model cell containing the particle

10. Cumulative MODFLOW time step number

11. Particle identification number IDPART assigned in the starting locations file

12. Particle attribute PARTAS described by the user in the starting locations file

Compact Text Pathline File for Individual Models

MODPATH-LGR produces a compact text pathline file whenever the keyword COMPACT is specified in record 2 of the MODPATH-LGR main data file. The compact text format includes the data items produced by MODPATH and two additional terms in columns 9 and 10. The compact text format uses grid cell locations as a node number instead of the column J, row I, and layer K indices. The node number N is computed as:

N=(K-1)*NCOL*NROW+(I-1)*NCOL+J,

where NCOL is the number of columns, and NROW is the number of rows.

The header record and any comment lines are followed by particle data records that contain the following data items in the order listed:

1. Particle index number for the set of particles beginning in the same model. The particle index number equals 1 for the first particle listed in the starting locations file, and increases by a unit for each subsequent particle. The particle index number resets to 1 for the starting particle for each model for simulations with multiple models.

2. Local coordinate in the x-direction

3. Local coordinate in the y-direction

4. Relative coordinate in the z-direction in the grid cell (0 to 1 within the model layer, -1 to 0 within an underlying confining layer)

5. Local coordinate in the z-direction

6. Cumulative tracking time

7. Node number for the model cell containing the particle

8. Cumulative MODFLOW time step number

9. Particle identification number IDPART assigned in the starting locations file

10. Particle attribute PARTAS described by the user in the starting locations file.

Binary Pathline File for Individual Models

MODPATH-LGR produces a binary pathline file whenever the keyword BINARY is specified in record 2 of the MODPATH-LGR main data file. The binary format includes the data items produced by MODPATH and two additional terms in columns 9 and 10.

The header record includes an 80-character text string describing the MODPATH version number followed by the reference simulation time used in the MODPATH run. The following segment of code shows how to read the header record in a FORTRAN program:

```
CHARACTER*80 TEXT
READ(50) TEXT,TREF
```

The header record is followed by the particle data records. The items in the data records in the binary pathline file are identical to data records in the pathline file that uses the compact text option.

Global Pathline File for Coupled Models

The contents and format of the global pathline file are identical to the pathline file that is generated for individual models, with the exception that all regional particle coordinates are in the system of the parent model, and 2 additional data items per record. The format options for the global pathline file are standard text or binary format, and are indicated in the main data file for each model. The abbreviated output of the compact format available in MODPATH is not available for the global pathline

file. The standard format is used for the global pathline file if the compact format option is specified in the main data files. Each model should use the same output format in the MODPATH-LGR main data file. The file contains a header and comment lines followed by a sequence of particle location records listed in ascending IDPART number.

Standard Text Global Pathline File

MODPATH-LGR produces a standard text global pathline file by default. The header record and any comment lines are followed by particle data records that contain the following data items in the order listed:

1. Particle index number for the set of particles beginning in the same model. The particle index number equals 1 for the first particle listed in the starting locations file, and increases by a unit for each subsequent particle. The particle index number resets to 1 for the starting particle for each model for simulations with multiple models.

2. Regional coordinate in the x-direction

3. Regional coordinate in the y-direction

4. Relative coordinate in the z-direction in the grid cell (0 to 1 within the model layer, -1 to 0 within an underlying confining layer)

5. Regional coordinate in the z-direction

6. Cumulative tracking time

7. J index of model cell containing the particle

8. I index of model cell containing the particle

9. K index of model cell containing the particle

10. Cumulative MODFLOW time step number

11. Identification number IDMODEL_START of the model where the particle started. The parent model is number 1 and each subsequent child model specified in the MODPATH-LGR control file is numbered in ascending order beginning with number 2.

12. Identification number IDMODEL of the model that contains the particle. The parent model is number 1 and each subsequent child model specified in the MODPATH-LGR control file is numbered in ascending order beginning with number 2.

13. Particle identification number IDPART assigned in the starting locations file

14. Particle attribute described by the user in the starting locations file

Binary Global Pathline File

MODPATH-LGR produces a binary global pathline file whenever the keyword BINARY is specified in record 2 of the MODPATH-LGR main data file. The binary version of the global pathline file contains a header record that includes an 80-character text string describing the MODPATH version number followed by the reference simulation time used in the MODPATH run. The following segment of code shows how to read the header record in a FORTRAN program:

```
CHARACTER*80 TEXT
READ(50) TEXT,TREF
```

The header record is followed by the particle location data records. The items in the data records in the binary global pathline file are identical to data records in the pathline file that uses the standard text option.

Time-series File

Coordinates at specific times along the path of each particle are saved to a time-series file if time-series output mode is specified in the response files. A time-series file is only generated for the models that indicate a pathline data file in the MODPATH-LGR name file. All models should specify the time-series output type and the TIME-SERIES keyword and file name

must be included in the MODPATH-LGR control file to produce a global time-series file. If a model does not specify the time-series output, the particle positions will not be recorded in the global output file. The content of the time-series file for individual models is described in detail in the MODPATH documentation (Pollock, 1994) and is omitted here.

Time-series File for Individual Models

MODPATH-LGR produces a time-series file for a single model if the simulation involves one model, and a time-series file for each model for simulations involving multiple models. The format and contents of the time-series file are identical to the format and contents generated by MODPATH, with the exception of several additional data items at the end of each record. Standard text, compact text, and binary output structures from MODPATH are preserved. MODPATH-LGR produces a standard text time-series file by default. Compact and binary formats are specified in the MODPATH-LGR main data file (Pollock, 1994).

Additional time-series files are automatically produced for each model for each MODPATH-LGR sequence for simulations involving multiple models. These files can be used to examine particle locations and tracking times instead of using the global time-series file. The naming convention of the files is as follows:

PARENT_TIMESERIES_X.DAT
CHILD_X_TIMESERIES_Y.DAT

X is the sequence number and Y is the identification number of the child model that contains the particles. Y equals 1 for the first child model listed in the MODPATH-LGR control file, and Y increases by one for each subsequent child model.The first line of the time-series file is a header record indicating the MODPATH-LGR version number and the reference simulation time value used in the MODPATH-LGR run. The header record has the form:
@ [MODPATH-LGR Version 1.00 (Release 1, 9-08) (TREF= 0.000000E+00)]

Standard Text Time-series File for Individual Models

MODPATH-LGR produces a standard text time-series file by default. The standard text format includes the data items produced by MODPATH (items 1-11) and 2 additional terms in columns 12 and 13. The header record and any comment lines are followed by particle data records that contain the following data items in the order listed:

1 Time step index number

2. Particle index number for the set of particles beginning in the same model. The particle index number equals 1 for the first particle listed in the starting locations file, and increases by a unit for each subsequent particle. The particle index number resets to 1 for the starting particle for each model for simulations with multiple models.

3. J index of model cell containing the particle

4. I index of model cell containing the particle

5. K index of model cell containing the particle

6. Local coordinate in x-direction

7. Local coordinate in y-direction

8. Local coordinate in z-direction

9. Relative coordinate in the z-direction in the grid cell (0 to 1 within the model layer, -1 to 0 within an underlying confining layer)

10. Cumulative tracking time

11. Cumulative MODFLOW time step number

12. Particle identification number IDPART assigned in the starting locations file

13. Particle attribute PARTAS described by the user in the starting locations file

Compact Text Time-series File for Individual Models

MODPATH-LGR produces a compact text time-series file whenever the keyword COMPACT is specified in record 2 of the MODPATH-LGR main data file. The compact text format includes the data items produced by MODPATH (items 1-9) and 2 additional terms in columns 10 and 11. The compact text format presents the grid cell locations as a node number instead of the column J, row I, and layer K indices. The node number N is computed as:

$$N=(K-1)*NCOL*NROW+(I-1)*NCOL+J,$$

where NCOL is the number of columns, and NROW is the number of rows.

The header record and any comment lines are followed by particle data records that contain the following data items in the order listed:

1. Time step index number

2. Particle index number for the set of particles beginning in the same model. The particle index number equals 1 for the first particle listed in the starting locations file, and increases by a unit for each subsequent particle. The particle index number resets to 1 for the starting particle for each model for simulations with multiple models.

3. Node number of model cell containing particle

4. Local coordinate in x-direction

5. Local coordinate in y-direction

6. Local coordinate in z-direction

7. Relative coordinate in the z-direction in the grid cell (0 to 1 within the model layer, -1 to 0 within an underlying confining layer)

8. Cumulative tracking time

9. Cumulative MODFLOW time step number

10. Particle identification number IDPART assigned in the starting locations file

11. Particle attribute PARTAS described by the user in the starting locations file

Binary Time-series File for Individual Models

MODPATH-LGR produces a binary time-series file whenever the keyword BINARY is specified in record 2 of the MODPATH-LGR main data file. The binary format includes the data items produced by MODPATH (items 1–9) and 2 additional terms in columns 10 and 11.

The header record includes an 80-character text string describing the MODPATH version number followed by the reference simulation time used in the MODPATH run. The following segment of code shows how to read the header record in a FORTRAN program:

```
CHARACTER*80 TEXT
READ(50) TEXT,TREF
```

The header record is followed by the particle data records. The items in the data records in the binary time-series file are identical to data records in the time-series file that uses the compact text option.

Global Time-series File for Coupled Models

The contents and format of the global time-series file are identical to the time-series file that is generated for individual models, with the exception that all regional particle coordinates are in the system of the parent model, and two additional data items per record. The format options for the global time-series file are standard text or binary format, and are indicated in the main data file for each model. The abbreviated output of the compact format available in MODPATH is not available for the global time-series file. The standard format is used for the global time-series file if the compact format option is specified in the main data files. Each model should use the same output format in the MODPATH-LGR main data file. The file contains a header and comment lines followed by a sequence of particle location records listed in ascending IDPART number.

Standard Text Global Time-series File

MODPATH-LGR produces a standard text global time-series file by default. Each record includes the following items:

1 Time step index number

2. Particle index number for the set of particles beginning in the same model. The particle index number equals 1 for the first particle listed in the starting locations file, and increases by a unit for each subsequent particle. The particle index number resets to 1 for the starting particle for each model for simulations with multiple models.

3. J index of model cell containing the particle

4. I index of model cell containing the particle

5. K index of model cell containing the particle

6. Regional coordinate in x-direction

7. Regional coordinate in y-direction

8. Regional coordinate in z-direction

9. Relative coordinate in the z-direction in the grid cell (0 to 1 within the model layer, -1 to 0 within an underlying confining layer)

10. Cumulative tracking time

11. Cumulative MODFLOW time step number

12. Identification number of the model IDMODEL_START where the particle started. The parent model is number 1 and each subsequent child model specified in the MODPATH-LGR control file is numbered in ascending order beginning with number 2.

13. Identification number of the model IDMODEL that contains the particle at the tracking time. The parent model is number 1 and each subsequent child model specified in the MODPATH-LGR control file is numbered in ascending order beginning with number 2.

14. Particle identification number IDPART assigned in the starting locations file

15. Particle attribute PARTAS described by the user in the starting locations file

Binary Global Time-series File

MODPATH-LGR produces a binary global time-series file whenever the keyword BINARY is specified in record 2 of the MODPATH-LGR main data file. The binary version of the global time-series file contains a header record that includes an 80-character text string describing the MODPATH version number followed by the reference simulation time used in the MODPATH run. The following segment of code shows how to read the header record in a FORTRAN program:

```
CHARACTER*80 TEXT
READ(50) TEXT,TREF
```

The header record is followed by the particle location data records. The items in the data records in the binary global time-series file are identical to data records in the time-series file that uses the standard text option.

Tracking Particles Outside of the Grid

MODPATH-LGR includes the option to track particles outside of the model grids. This option is described in the release notes for MODPATH version 5.0 (Pollock, 1994). This option creates an output file that is contains particle locations at specific times. The time-series output option must be specified in the response file to use the particle projection option for individual models. The MODPATH-LGR name file also must contain the TIME-SERIES and ADVOBS keywords and file names for time-series and particle-projection output files. All models should specify the time-series output type and the TIME-SERIES and ADVOBS keywords and output file names must be included in the MODPATH-LGR control file to produce a global particle projections file.

Particle-projection File for Individual Models

MODPATH-LGR produces a particle-projection file for a single model if the simulation involves one model, and a particle-projection file for each model for simulations involving multiple models. The format and contents of the particle-projection file are identical to the format and contents generated by MODPATH, with the exception of two additional data items at the end of each record. Standard text and binary output structures are available. The binary format is used for the global particle-projection file if the compact format option is specified in the main data files. MODPATH-LGR produces a standard text particle-projection file by default. The binary format is specified in the MODPATH-LGR main data file (Pollock, 1994).

Additional particle-projection files are automatically produced for each model for each MODPATH-LGR sequence for simulations involving multiple models. These files can be used to examine particle locations and tracking times instead of using the global particle-projection file. The naming convention of the files is as follows:

PARENT_ADVOBS_X.DAT
CHILD_X_ADVOBS_Y.DAT

X is the sequence number and Y is the identification number of the child model that contains the particles. Y equals 1 for the first child model listed in the MODPATH-LGR control file, and Y increases by one for each subsequent child model.

Standard Text Particle-projection File

The standard text format includes the data items produced by MODPATH (items 1–9) and 2 additional terms in columns 10 and 11. Each line in the file contains the following items:

1. Particle index number for the set of particles beginning in the same model. The particle index number equals 1 for the first particle listed in the starting locations file, and increases by a unit for each subsequent particle. The particle index number resets to 1 for the starting particle for each model for simulations with multiple models.

2. Discharge code – 1 is not discharged, 0 is discharged

3. Local coordinate in x-direction

4. Local coordinate in y-direction

5. Local coordinate in z-direction

6. Cumulative tracking time

7. X-direction velocity on exit (0.0 if the particle has not exited)

8. Y-direction velocity on exit (0.0 if the particle has not exited)

9. Z-direction velocity on exit (0.0 if the particle has not exited)

10. Particle identification number IDPART assigned in the starting locations file

11. Particle attribute PARTAS described by the user in the starting locations file

Binary Particle-projections File

MODPATH-LGR produces a binary particle-projections file whenever the keyword BINARY is specified in record 2 of the MODPATH-LGR main data file. The binary format includes the data items produced by MODPATH (items 1–9) and 2 additional

terms in columns 10 and 11. The items in the data records in the binary particle-projections file are identical to data records in the particle-projections file that uses the standard text option.

Global Particle-projection File for Coupled Models

The contents and format of the global particle projection file are identical to the particle projection file for individual models (items 1–11), with the exception that all regional particle coordinates are in the system of the parent model, and two additional data items per record in columns 12–13. The format options for the global particle projection file are standard text or binary format, and are indicated in the main data file for each model. Each model should use the same output format in the MODPATH-LGR main data file.

Standard Text Global Particle-projection File

MODPATH-LGR produces a standard text global particle-projection file by default. Each record includes the following items:

1. Particle index number for the set of particles beginning in the same model. The particle index number equals 1 for the first particle listed in the starting locations file, and increases by a unit for each subsequent particle. The particle index number resets to 1 for the starting particle for each model for simulations with multiple models.

2. Discharge code – 1 is not discharged, 0 is discharged

3. Regional coordinate in x-direction

4. Regional coordinate in y-direction

5. Regional coordinate in z-direction

6. Cumulative tracking time

7. X-direction velocity on exit (0.0 if the particle has not exited)

8. Y-direction velocity on exit (0.0 if the particle has not exited)

9. Z-direction velocity on exit (0.0 if the particle has not exited)

10. Identification number of the model IDMODEL_START where the particle started. The parent model is number 1 and each subsequent child model specified in the MODPATH-LGR control file is numbered in ascending order beginning with number 2.

11. Identification number of the model IDMODEL that contains the particle at the tracking time. The parent model is number 1 and each subsequent child model specified in the MODPATH-LGR control file is numbered in ascending order beginning with number 2.

12. Particle identification number IDPART assigned in the starting locations file

13. Particle attribute PARTAS described by the user in the starting locations file

Binary Global Particle-projection File

MODPATH-LGR produces a binary global particle-projection file whenever the keyword BINARY is specified in record 2 of the MODPATH-LGR main data file. The items in the data records in the binary global particle-projection file are identical to data records in the global particle-projection file that uses the standard text option.

APPENDIX C. Selected Input and Output Files for the Example Problems

Example Problem 1

The sample data inputs are listed in this report for the steady-state example problem 1 (fig. 5) having a 9:1 horizontal refinement ratio.

MODPATH-LGR Control File

```
MODFLOW.LGR                     #LGR control file
PARENT.RSP                      #Response file for parent
FLOWP                           #Budget file for parent
CHILD.RSP                       #Response file for child
59                              #BDYFLG for child1
ENDPOINT GLOBAL_ENDPOINT_9_1.dat #Global endpoint file
PATHLINE GLOBAL_PATHLINE_9_1.dat #Global pathline file
```

MODFLOW-LGR Control File

```
LGR                 #LGR Keyword
2                   #NGRIDS
PARENT.nam          #NAME file
PARENTONLY          #GRIDSTATUS
00 00               #Unit #s for complimentary and coupling B.C.
CHILD.nam           #NAME file
CHILDONLY           #GRIDSTATUS
1 -59 00 00         #ISHFLG,IBFLG, UNIT #s for complimentary and coupling B.C.
40  -1              #MXLGRITER,IOUTLGR
0.40000 0.40000     #RELAXH,RELAXF
1.0E-5  1.0E-5      #HCLOSELGR,FCLOSELGR
1 20 22             #Beginning layer, row, column
3 31 39             #Ending layer, row, column
9                   #Horizontal refinement ratio
3 3 3               #Vertical refinement ratio by parent layer
```

Parent Model MODPATH-LGR Response File

(Create by using MODPATH-LGR run for the parent model alone)
```
@[MODPATH Version 4.00 (V4, Release 3, 7-2003)]
* ENTER THE NAME FILE:
@RESPONSE:
MODPATH_PARENT.DAT
* DO YOU WANT TO STOP COMPUTING PATHS AFTER A SPECIFIED LENGTH OF TIME ?
@RESPONSE:
n
* SELECT THE OUTPUT MODE:
*     1 = ENDPOINTS
*     2 = PATHLINE
*     3 = TIME SERIES
@RESPONSE:
2
* DO YOU WANT TO COMPUTE LOCATIONS AT SPECIFIC POINTS IN TIME ?
@RESPONSE:
N
```

```
* HOW ARE STARTING LOCATIONS TO BE ENTERED?
*     1 = FROM AN EXISTING DATA FILE
*     2 = ARRAYS OF PARTICLES WILL BE GENERATED INTERNALLY
@RESPONSE:
1
* IN WHICH DIRECTION SHOULD PARTICLES BE TRACKED?
*     1 = FORWARD IN THE DIRECTION OF FLOW
*     2 = BACKWARDS TOWARD RECHARGE LOCATIONS
@RESPONSE:
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:
1
* DO YOU WANT TO STOP PARTICLES WHENEVER THEY ENTER ONE SPECIFIC ZONE ?
@RESPONSE:
Y
* ENTER THE ZONE NUMBER (MUST BE > 1):
@RESPONSE:
59
* DO YOU WANT TO CHANGE ANY OF THE ZONE CODES IN THE IBOUND ARRAY ?
@RESPONSE:
N
* DO YOU WANT TO COMPUTE VOLUMETRIC BUDGETS FOR ALL CELLS ?
@RESPONSE:
N
* DO YOU WANT TO CHECK DATA CELL BY CELL ?
@RESPONSE:
N
* SUMMARIZE FINAL STATUS OF PARTICLES IN SUMMARY.PTH FILE ?
@RESPONSE:
Y
```

Parent Model MODPATH-LGR Name File

```
MAIN          10 PARENT_MAIN.DAT
DIS           15 PARENT.DIS
BUDGET        17 FLOWP2
HEAD(BINARY)  18 HEDP
PATHLINE      20 PARENT_PATHLINE.DAT
ENDPOINT      21 PARENT_ENDPOINT.DAT
```

Parent Model MODPATH-LGR Main Data File

```
0 -9.990000e+002 -9.990000e+002 0 0 0 #MAXSIZ HNOFLO HDRY NPART IRCHTP IEVTTP
                                      #Options
0 0 0 0                               #LAYCON
OPEN/CLOSE PARENT1.IBD    1 (108i3)  -9 #IBOUND layer 1
OPEN/CLOSE PARENT2.IBD    1 (108i3)  -9 #IBOUND layer 2
OPEN/CLOSE PARENT3.IBD    1 (108i3)  -9 #IBOUND layer 3
OPEN/CLOSE PARENT4.IBD    1 (108i3)  -9 #IBOUND layer 4
CONSTANT 1.00E-2                      #Porosity layer1
CONSTANT 1.00E-2                      #Porosity layer2
CONSTANT 1.00E-2                      #Porosity layer3
CONSTANT 1.00E-2                      #Porosity layer2
```

Child Model MODPATH-LGR Response File

(Create by using MODPATH-LGR run for each child model alone)

```
@[MODPATH Version 4.00 (V4, Release 3, 7-2003)]
* ENTER THE NAME FILE:
@RESPONSE:
MODPATH_CHILD.DAT
* DO YOU WANT TO STOP COMPUTING PATHS AFTER A SPECIFIED LENGTH OF TIME ?
@RESPONSE:
N
* SELECT THE OUTPUT MODE:
*     1 = ENDPOINTS
*     2 = PATHLINE
*     3 = TIME SERIES
@RESPONSE:
2
* DO YOU WANT TO COMPUTE LOCATIONS AT SPECIFIC POINTS IN TIME ?
@RESPONSE:
N
* HOW ARE STARTING LOCATIONS TO BE ENTERED?
*     1 = FROM AN EXISTING DATA FILE
*     2 = ARRAYS OF PARTICLES WILL BE GENERATED INTERNALLY
@RESPONSE:
1
* IN WHICH DIRECTION SHOULD PARTICLES BE TRACKED?
*     1 = FORWARD IN THE DIRECTION OF FLOW
*     2 = BACKWARDS TOWARD RECHARGE LOCATIONS
@RESPONSE:
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:
1
* DO YOU WANT TO STOP PARTICLES WHENEVER THEY ENTER ONE SPECIFIC ZONE ?
@RESPONSE:
Y
* ENTER THE ZONE NUMBER (MUST BE > 1):
@RESPONSE:
59
* DO YOU WANT TO CHANGE ANY OF THE ZONE CODES IN THE IBOUND ARRAY ?
@RESPONSE:
N
* DO YOU WANT TO COMPUTE VOLUMETRIC BUDGETS FOR ALL CELLS ?
@RESPONSE:
N
*  DO YOU WANT TO CHECK DATA CELL BY CELL ?
@RESPONSE:
N
* SUMMARIZE FINAL STATUS OF PARTICLES IN SUMMARY.PTH FILE ?
@RESPONSE:
Y
```

Child Model MODPATH-LGR Name File

```
MAIN         10 CHILD_MAIN.DAT
DIS          15 CHILD.DIS
BUDGET       17 FLOWC
HEAD(BINARY) 18 HEADC
PATHLINE     20 CHILD_PATHLINE.DATchild_pathline.dat
ENDPOINT     21 CHILD_ENDPOINT.DAT
LOCATIONS    22 CHILD_START.DAT
```

Child Model MODPATH-LGR Main Data File

```
0 -9.990000e+002 -9.990000e+002 0 0 0 #MAXSIZ HNOFLO HDRY NPART IRCHTP IEVTTP
                                                 #Options
 0 0 0 0 0 0 0 0                                 #LAYCON
OPEN/CLOSE CHILD.IBD    1 (108i3)    -9          #IBOUND layer   1
OPEN/CLOSE CHILD.IBD    1 (108i3)    -9          #IBOUND layer   2
OPEN/CLOSE CHILD.IBD    1 (108i3)    -9          #IBOUND layer   3
OPEN/CLOSE CHILD.IBD    1 (108i3)    -9          #IBOUND layer   4
OPEN/CLOSE CHILD.IBD    1 (108i3)    -9          #IBOUND layer   5
OPEN/CLOSE CHILD.IBD    1 (108i3)    -9          #IBOUND layer   6
OPEN/CLOSE CHILD.IBD    1 (108i3)    -9          #IBOUND layer   7
OPEN/CLOSE CHILD.IBD    1 (108i3)    -9          #IBOUND layer   8
CONSTANT 1.00E-2                                 #Porosity layer1
CONSTANT 1.00E-2                                 #Porosity layer2
CONSTANT 1.00E-2                                 #Porosity layer3
CONSTANT 1.00E-2                                 #Porosity layer4
CONSTANT 1.00E-2                                 #Porosity layer5
CONSTANT 1.00E-2                                 #Porosity layer6
CONSTANT 1.00E-2                                 #Porosity layer7
CONSTANT 1.00E-2                                 #Porosity layer8
```

Child Model Starting Locations File

The first four lines of the starting locations file are shown.

```
73 55 1 0.100E-03 0.167E+00 0.167E+00 0 0 0 0.00E+00 1 child_well
73 55 1 0.100E-03 0.167E+00 0.500E+00 0 0 0 0.00E+00 2 child_well
73 55 1 0.100E-03 0.167E+00 0.833E+00 0 0 0 0.00E+00 3 child_well
73 55 1 0.100E-03 0.500E+00 0.167E+00 0 0 0 0.00E+00 4 child_well
```

Example Problem 2

The sample data inputs are listed in this report for the steady-state hypothetical example problem 2 having a 9:1 horizontal refinement ratio, as shown in figure 7.

MODPATH-LGR Control File

```
MODFLOW.LGR                       #LGR control file
PARENT\PARENT.RSP                 #Response file for parent
PARENT\FLOWP                      #Budget file for parent
CHILD1\CHILD1.RSP                 #Response file for child1
59                                #BDFLG for child1
CHILD2\CHILD2.RSP                 #Response file for child2
60                                #BDFLG for child2
ENDPOINT 65 GLOBAL_ENDPOINT.DAT   #Global endpoint file
PATHLINE 67 GLOBAL_PATHLINE.DAT   #Global pathline file
```

MODFLOW-LGR Control File

```
LGR                    #LGR Keyword
3                      #NGRIDS
PARENT.NAM             #NAME file
PARENTONLY             #GRIDSTATUS
70 71                  #Unit #'s for complimentary and coupling B.C.
CHILD1.NAM             #NAME file
CHILDONLY              #GRIDSTATUS
1 -59 80 81            #ISHFLG,IBFLG, UNIT #'s for complimentary and coupling B.C.
50  -1                 #MXLGRITER,IOUTLGR
0.40000 0.40000        #RELAXH,RELAXF
1.0E-5  1.0E-5         #HCLOSELGR,FCLOSELGR
1 20 22                #Beginning layer, row, column
3 31 39                #Ending layer, row, column
9                      #Horizontal refinement ratio
3 3 3                  #Vertical refinement ratio by parent layer
CHILD2.NAM             #NAME file
CHILDONLY              #GRIDSTATUS
1 -60 92 93            #ISHFLG,IBFLG, UNIT #'s for complimentary and coupling B.C.
50  -1                 #MXLGRITER,IOUTLGR
0.40000 0.40000        #RELAXH,RELAXF
1.0E-5  1.0E-5         #HCLOSELGR,FCLOSELGR
1 10 46                #Beginning layer, row, column
3 21 63                #Ending layer, row, column
9                      #Horizontal refinement ratio
3 3 3                  #Vertical refinement ratio by parent layer
```

Parent Model MODPATH-LGR Response File

(Create by using MODPATH-LGR run for the parent model alone)

```
@[MODPATH Version 4.00 (V4, Release 3, 7-2003)]
* ENTER THE NAME FILE:
@RESPONSE:
MODPATH_PARENT.DAT
* DO YOU WANT TO STOP COMPUTING PATHS AFTER A SPECIFIED LENGTH OF TIME ?
@RESPONSE:
n
* SELECT THE OUTPUT MODE:
*     1 = ENDPOINTS
*     2 = PATHLINE
*     3 = TIME SERIES
@RESPONSE:
```

```
2
* DO YOU WANT TO COMPUTE LOCATIONS AT SPECIFIC POINTS IN TIME ?
@RESPONSE:
N
* HOW ARE STARTING LOCATIONS TO BE ENTERED?
*     1 = FROM AN EXISTING DATA FILE
*     2 = ARRAYS OF PARTICLES WILL BE GENERATED INTERNALLY
@RESPONSE:
1
* IN WHICH DIRECTION SHOULD PARTICLES BE TRACKED?
*     1 = FORWARD IN THE DIRECTION OF FLOW
*     2 = BACKWARDS TOWARD RECHARGE LOCATIONS
@RESPONSE:
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:
1
* DO YOU WANT TO STOP PARTICLES WHENEVER THEY ENTER ONE SPECIFIC ZONE ?
@RESPONSE:
N
* DO YOU WANT TO COMPUTE VOLUMETRIC BUDGETS FOR ALL CELLS ?
@RESPONSE:
N
*  DO YOU WANT TO CHECK DATA CELL BY CELL ?
@RESPONSE:
N
* SUMMARIZE FINAL STATUS OF PARTICLES IN SUMMARY.PTH FILE ?
@RESPONSE:
Y
```

Parent Model MODPATH-LGR Name File

```
MAIN            10  PARENT\PARENT_MAIN.DAT
WEL             12  PARENT\PARENT.WEL
DIS             15  PARENT\PARENT.DIS
RCH             18  PARENT\PARENT.RCH
RIV             19  PARENT\PARENT.RIV
BUDGET          17  FLOWP2
HEAD(BINARY)    18  PARENT\HEDP
PATHLINE        20  PARENT\PARENT_PATHLINE.DAT
ENDPOINT        21  PARENT\PARENT_ENDPOINT.DAT
LOCATIONS       22  PARENT\PARENT_START.DAT
```

Parent MODPATH-LGR Main Data File

```
0 -9.99e+002 -1.00e+003 0 1 1          #MAXSIZ HNOFLOW HDRY NPART IRCHTP IEVTTP
                                       #Options
 0 0 0 0                               #LAYCON
OPEN/CLOSE PARENT1_MP.IBD    1 (108i3) -9 #IBOUND layer 1
OPEN/CLOSE PARENT2_MP.IBD    1 (108i3) -9 #IBOUND layer 2
OPEN/CLOSE PARENT4_MP.IBD    1 (108i3) -9 #IBOUND layer 3
OPEN/CLOSE PARENT4_MP.IBD    1 (108i3) -9 #IBOUND layer 4
OPEN/CLOSE PARENT.POR        1 (FREE)  -9 #IBOUND layer 1
```

```
OPEN/CLOSE PARENT.POR        1 (FREE)  -9 #IBOUND layer 2
OPEN/CLOSE PARENT.POR        1 (FREE)  -9 #IBOUND layer 3
OPEN/CLOSE PARENT.POR        1 (FREE)  -9 #IBOUND layer 4
0                                         #Tbegin
1 1 1 10                                  #BeginPeriod BeginStep End Period EndStep
```

Parent Model Starting Locations File

The first four lines of the starting locations file are shown.
```
54 40 1 0.100E+00 0.400E+00 0.500E-02 0 0 0 0.000E+00 1001 'well1'
54 40 1 0.100E+00 0.400E+00 0.150E-01 0 0 0 0.000E+00 1002 'well1'
54 40 1 0.100E+00 0.400E+00 0.250E-01 0 0 0 0.000E+00 1003 'well1'
54 40 1 0.100E+00 0.400E+00 0.350E-01 0 0 0 0.000E+00 1004 'well1'
```

Child Model 1 MODPATH-LGR Response File

(Create by using MODPATH-LGR run for the child model alone)
```
@[MODPATH Version 4.00 (V4, Release 3, 7-2003)]
* ENTER THE NAME FILE:
@RESPONSE:
MODPATH_CHILD1.DAT
* DO YOU WANT TO STOP COMPUTING PATHS AFTER A SPECIFIED LENGTH OF TIME ?
@RESPONSE:
N
* SELECT THE OUTPUT MODE:
*     1 = ENDPOINTS
*     2 = PATHLINE
*     3 = TIME SERIES
@RESPONSE:
2
* DO YOU WANT TO COMPUTE LOCATIONS AT SPECIFIC POINTS IN TIME ?
@RESPONSE:
N
* HOW ARE STARTING LOCATIONS TO BE ENTERED?
*     1 = FROM AN EXISTING DATA FILE
*     2 = ARRAYS OF PARTICLES WILL BE GENERATED INTERNALLY
@RESPONSE:
1
* IN WHICH DIRECTION SHOULD PARTICLES BE TRACKED?
*     1 = FORWARD IN THE DIRECTION OF FLOW
*     2 = BACKWARDS TOWARD RECHARGE LOCATIONS
@RESPONSE:
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:
1
* DO YOU WANT TO STOP PARTICLES WHENEVER THEY ENTER ONE SPECIFIC ZONE ?
@RESPONSE:
N
* DO YOU WANT TO COMPUTE VOLUMETRIC BUDGETS FOR ALL CELLS ?
@RESPONSE:
N
```

```
*  DO YOU WANT TO CHECK DATA CELL BY CELL ?
@RESPONSE:
N
* SUMMARIZE FINAL STATUS OF PARTICLES IN SUMMARY.PTH FILE ?
@RESPONSE:
Y
```

Child Model 1 MODPATH-LGR Name File

```
MAIN       10      CHILD1\CHILD1_MAIN.DAT
DIS        15      CHILD1\CHILD1.DIS
BUDGET     17      CHILD1\FLOWC1
HEAD(BINARY)  18  CHILD1\HEADC1
PATHLINE      20  CHILD1\CHILD1_PATHLINE.DAT
ENDPOINT      21  CHILD1\CHILD1_ENDPOINT.DAT
LOCATIONS     22  CHILD1\CHILD1_START.DAT
```

Child Model 1 MODPATH-LGR Main Data File

```
0 -9.99e+002 -1.00e+003 0 1 1 #MAXSIZ HNOFLOW HDRY NPART IRCHTP IEVTTP
                                      #Options
 0 0 0 0 0 0 0 0                      #LAYCON
OPEN/CLOSE CHILD1.IBD    1 (154i3)   -9 #IBOUND layer 1
OPEN/CLOSE CHILD1.IBD    1 (154i3)   -9 #IBOUND layer 2
OPEN/CLOSE CHILD1.IBD    1 (154i3)   -9 #IBOUND layer 3
OPEN/CLOSE CHILD1.IBD    1 (154i3)   -9 #IBOUND layer 4
OPEN/CLOSE CHILD1.IBD    1 (154i3)   -9 #IBOUND layer 5
OPEN/CLOSE CHILD1.IBD    1 (154i3)   -9 #IBOUND layer 6
OPEN/CLOSE CHILD1.IBD    1 (154i3)   -9 #IBOUND layer 7
CONSTANT -59                          #IBOUND layer 8
OPEN/CLOSE CHILD1.POR    1 (FREE)    -9 #Porosity layer 1
OPEN/CLOSE CHILD1.POR    1 (FREE)    -9 #Porosity layer 2
OPEN/CLOSE CHILD1.POR    1 (FREE)    -9 #Porosity layer 3
OPEN/CLOSE CHILD1.POR    1 (FREE)    -9 #Porosity layer 4
OPEN/CLOSE CHILD1.POR    1 (FREE)    -9 #Porosity layer 5
OPEN/CLOSE CHILD1.POR    1 (FREE)    -9 #Porosity layer 6
OPEN/CLOSE CHILD1.POR    1 (FREE)    -9 #Porosity layer 7
OPEN/CLOSE CHILD1.POR    1 (FREE)    -9 #Porosity layer 8
0                                        #Tbegin
1 1 1 10                             #BeginPeriod BeginStep End Period EndStep
```

Child Model 1 Starting Locations File

The first four lines of the starting locations file are shown.
```
73 55 1 0.100E-03 0.125E+00 0.125E+00 0 0 0 0.000E+00 2001 'well2'
73 55 1 0.100E-03 0.125E+00 0.375E+00 0 0 0 0.000E+00 2002 'well2'
73 55 1 0.100E-03 0.125E+00 0.625E+00 0 0 0 0.000E+00 2003 'well2'
73 55 1 0.100E-03 0.125E+00 0.875E+00 0 0 0 0.000E+00 2004 'well2'
```

Child Model 2 MODPATH-LGR Response File

(Create by using MODPATH-LGR run for the child model alone)

```
@[MODPATH Version 4.00 (V4, Release 3, 7-2003)]
* ENTER THE NAME FILE:
@RESPONSE:
MODPATH_CHILD2.DAT
* DO YOU WANT TO STOP COMPUTING PATHS AFTER A SPECIFIED LENGTH OF TIME ?
@RESPONSE:
N
* SELECT THE OUTPUT MODE:
*     1 = ENDPOINTS
*     2 = PATHLINE
*     3 = TIME SERIES
@RESPONSE:
2
* DO YOU WANT TO COMPUTE LOCATIONS AT SPECIFIC POINTS IN TIME ?
@RESPONSE:
N
* HOW ARE STARTING LOCATIONS TO BE ENTERED?
*     1 = FROM AN EXISTING DATA FILE
*     2 = ARRAYS OF PARTICLES WILL BE GENERATED INTERNALLY
@RESPONSE:
1
* IN WHICH DIRECTION SHOULD PARTICLES BE TRACKED?
*     1 = FORWARD IN THE DIRECTION OF FLOW
*     2 = BACKWARDS TOWARD RECHARGE LOCATIONS
@RESPONSE:
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:
1
* DO YOU WANT TO STOP PARTICLES WHENEVER THEY ENTER ONE SPECIFIC ZONE ?
@RESPONSE:
N
* DO YOU WANT TO COMPUTE VOLUMETRIC BUDGETS FOR ALL CELLS ?
@RESPONSE:
N
*  DO YOU WANT TO CHECK DATA CELL BY CELL ?
@RESPONSE:
N
* SUMMARIZE FINAL STATUS OF PARTICLES IN SUMMARY.PTH FILE ?
@RESPONSE:
Y
```

Child Model 2 MODPATH-LGR Name File

```
MAIN           10  CHILD2\CHILD2_MAIN.DAT
DIS            15  CHILD2\CHILD2.DIS
BUDGET         17  CHILD2\FLOWC2
HEAD(BINARY)   18  CHILD2\HEADC2
PATHLINE       20  CHILD2\CHILD2_PATHLINE.DAT
ENDPOINT       21  CHILD2\CHILD2_ENDPOINT.DAT
LOCATIONS      23  CHILD2\CHILD2_START.DAT
```

Child Model 2 MODPATH-LGR Main Data File

```
0 -9.99e+002 -1.0e+003 0 1 1           #MAXSIZ HNOFLOW HDRY NPART IRCHTP IEVTTP
                                       #Options
 0 0 0 0 0 0 0 0                       #LAYCON
OPEN/CLOSE CHILD2.IBD    1 (154i3)  -9 #IBOUND layer   1
OPEN/CLOSE CHILD2.IBD    1 (154i3)  -9 #IBOUND layer   2
OPEN/CLOSE CHILD2.IBD    1 (154i3)  -9 #IBOUND layer   3
OPEN/CLOSE CHILD2.IBD    1 (154i3)  -9 #IBOUND layer   4
OPEN/CLOSE CHILD2.IBD    1 (154i3)  -9 #IBOUND layer   5
OPEN/CLOSE CHILD2.IBD    1 (154i3)  -9 #IBOUND layer   6
OPEN/CLOSE CHILD2.IBD    1 (154i3)  -9 #IBOUND layer   7
CONSTANT -60                           #IBOUND layer   8
OPEN/CLOSE CHILD2.POR    1 (FREE)   -9 #Porosity layer   1
OPEN/CLOSE CHILD2.POR    1 (FREE)   -9 #Porosity layer   2
OPEN/CLOSE CHILD2.POR    1 (FREE)   -9 #Porosity layer   3
OPEN/CLOSE CHILD2.POR    1 (FREE)   -9 #Porosity layer   4
OPEN/CLOSE CHILD2.POR    1 (FREE)   -9 #Porosity layer   5
OPEN/CLOSE CHILD2.POR    1 (FREE)   -9 #Porosity layer   6
OPEN/CLOSE CHILD2.POR    1 (FREE)   -9 #Porosity layer   7
OPEN/CLOSE CHILD2.POR    1 (FREE)   -9 #Porosity layer   8
0                                      #Tbegin
1 1 1 10                        #BeginPeriod BeginStep End Period EndStep
```

Child Model 2 Starting Locations File

The first four lines of the starting locations file are shown.
```
73 55 1 0.100E-03 0.125E+00 0.125E+00 0 0 0 0.000E+00 3001 'well3'
73 55 1 0.100E-03 0.125E+00 0.375E+00 0 0 0 0.000E+00 3002 'well3'
73 55 1 0.100E-03 0.125E+00 0.625E+00 0 0 0 0.000E+00 3003 'well3'
73 55 1 0.100E-03 0.125E+00 0.875E+00 0 0 0 0.000E+00 3004 'well3'
```

APPENDIX D. Estimation of Flow Boundary Conditions at Parent Interface Cells

The flux boundary conditions at the parent-child model interface in the shared-node version of MODFLOW-LGR are not compatible with fluxes required by the original version of MODPATH (Pollock, 1994). MODFLOW-LGR version 1.2.01 and later versions include the flux values at the parent interface cells that are adjusted by MODPATH-LGR. The nodes and cells along the interface overlap in the shared-node method of MODPATH-LGR, and the conductance terms between these cells for flows parallel to the model interface are halved to account for this overlap. Flows in each grid parallel to the interface would be counted twice if this conductance correction was not made. Further discussion of halved conductance values is presented by Mehl and Hill (2002, 2005).

The amount of the flux adjustment for parent-model cells at the model interfaces depends on the number of local-model cells that are overlapped by the parent-model cell (fig. 8) that results from three different configurations of interfaces between parent- and child-model grids: side, edge, and corner. The parent-model flows that are parallel to the interface boundary (y direction) happen through faces with areas reduced by ½ at side interfaces (fig. 8B). These flows are doubled to account for the half conductance that results from the halved area of the cell face in that direction. The parent-model flows that are parallel to the interface boundary (y direction) happen through faces with areas reduced by ¾ at edge and corner interfaces. These flows are increased by $^4/_3$ (fig 8C, 8D). The parent-model flow term Q_{x_2}, which is perpendicular to the interface boundary (x direction), is obtained from the child model at side interfaces (fig. 8B). Q_{x_2} is equal to the sum of all the child boundary flows from the adjacent shared child cells.

The corrected fluxes are written to a new cell-by-cell budget file that is read by MODPATH-LGR. Fluxes of the child model are not adjusted because the particles do not move into the shared child-model cells. The flux values and the file format produced by MODFLOW-LGR are identical to previous versions for all other cells in the parent and child models.
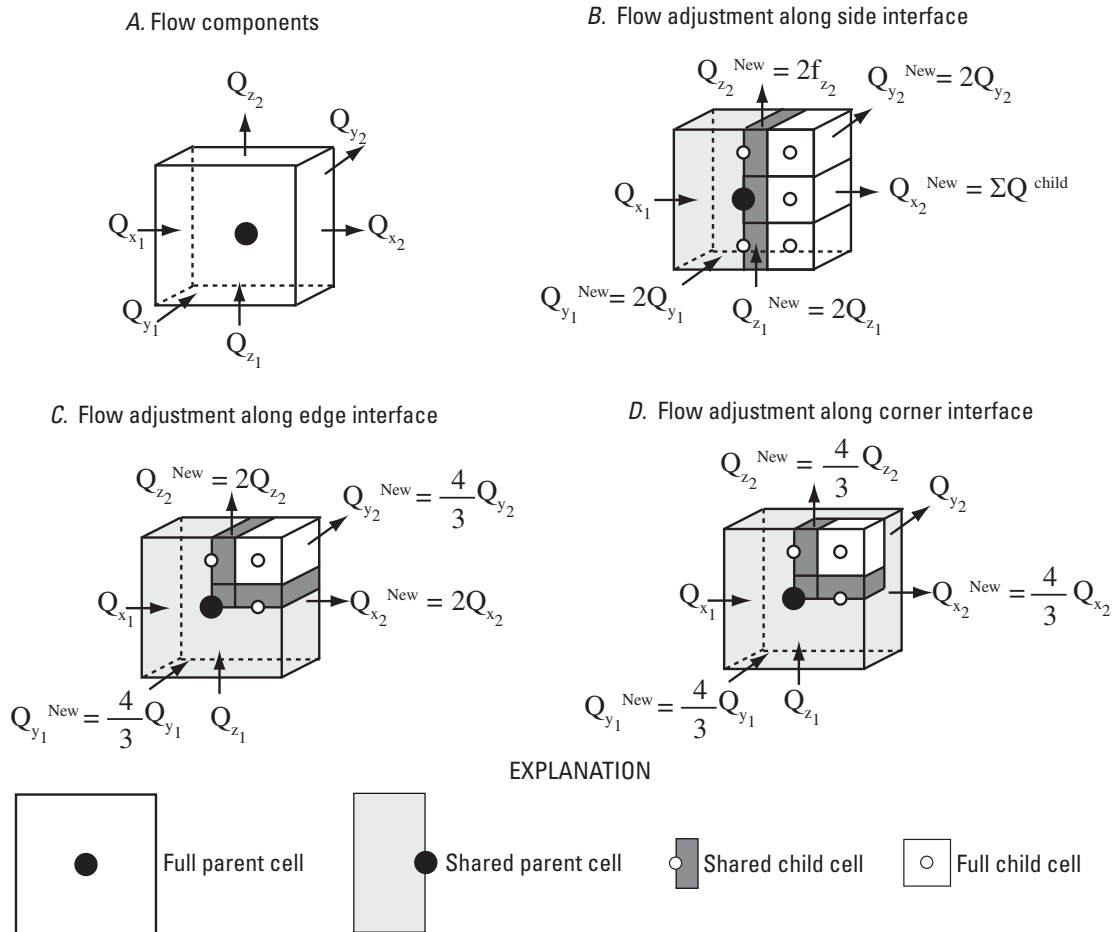

Figure 8. Schematic showing flow adjustments at the faces of the shared parent cells for *(A)* full parent cells, *(B)* parent cells having side interfaces, *(C)* edge interfaces, and *(D)* corner interfaces.

USGS

Dickinson and others—**MODPATH-LGR – Documentation of a Computer Program for Particle Tracking in Shared-Node Locally Refined Grids**—Techniques and Methods 6-A38