USGS
science for a changing world

Groundwater Resources Program

Prepared in collaboration with AMEC

MODFLOW–USG Version 1: An Unstructured Grid Version of MODFLOW for Simulating Groundwater Flow and Tightly Coupled Processes Using a Control Volume Finite-Difference Formulation

Chapter 45 of
Section A, Groundwater
Book 6, Modeling Techniques

Techniques and Methods 6–A45

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

Cover. The cover image is a three-dimensional rendering of the simulated results from the quadtree example problem described in this report. The cell shading and top elevations correspond to the simulated water table elevation.

MODFLOW–USG Version 1: An Unstructured Grid Version of MODFLOW for Simulating Groundwater Flow and Tightly Coupled Processes Using a Control Volume Finite-Difference Formulation

By Sorab Panday, Christian D Langevin, Richard G. Niswonger, Motomu Ibaraki, and Joseph D. Hughes

Chapter 45 of
Section A, Groundwater
Book 6, Modeling Techniques

Prepared in collaboration with AMEC

Groundwater Resources Program

Techniques and Methods 6–A45

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

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

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

U.S. Geological Survey, Reston, Virginia: 2013

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

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

To order this and other USGS information products, visit http://store.usgs.gov

Suggested citation:
Panday, Sorab, Langevin, C.D., Niswonger, R.G., Ibaraki, Motomu, and Hughes, J.D., 2013, MODFLOW–USG version 1: An unstructured grid version of MODFLOW for simulating groundwater flow and tightly coupled processes using a control volume finite-difference formulation: U.S. Geological Survey Techniques and Methods, book 6, chap. A45, 66 p.

Preface

This report describes an unstructured grid version of MODFLOW, called MODFLOW-USG. MODFLOW-USG is based on the U.S. Geological Survey (USGS) MODFLOW-2005 groundwater flow model. MODFLOW-USG simulates groundwater flow using a generalized control volume finite-difference approach, which allows grids other than the orthogonal structured grids required by previous MODFLOW versions to be used for groundwater flow simulations. The performance of this computer program has been tested using models of simplified systems; however, future applications of the programs may reveal errors that were not detected in the test simulations. Users are requested to notify the USGS if errors are found in the documentation report or in the computer program.

Although the computer program has been written and used by the USGS, no warranty, expressed or implied, is made by the USGS or the U.S. Government as to the accuracy and functionality of the program and related program material. Nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the USGS in connection therewith. MODFLOW-USG, MODFLOW-2005, and other groundwater programs are available online from the USGS at the following address:

http://water.usgs.gov/software/ground_water.html

Input instructions for MODFLOW-USG and additional information for using the program are included with the distribution. The distribution can be downloaded from the link shown above.

Acknowledgments

The authors thank Anthony (Tony) Daus for championing this work within AMEC and for his encouragement and support throughout the development effort. The authors also thank the numerous staff at AMEC for their support including beta-testing the code and reviewing the documentation. The authors acknowledge Richard Winston, Daniel Feinstein, and Paul Barlow for providing thoughtful and constructive technical reviews of this report.

Contents

Figures

Tables

Conversion Factors and Abbreviations

SI to Inch/Pound

Multiply	By	To obtain
meter (m)	3.281	foot (ft)
square meter (m^2)	0.0002471	acre
meter per day (m/d)	3.281	foot per day (ft/d)
cubic meter per day (m^3/d)	35.31	cubic foot per day (ft^3/d)

Abbreviations

CFP	Conduit Flow Process
CSR	compressed sparse row
CVFD	control volume finite difference
CVFE	control volume finite element
DIS	Discretization (input file)
DISU	Unstructured Discretization (input file)
GUI	graphical user interface
GWF	Groundwater Flow (Process)
HUF	Hydrogeologic-Unit Flow (Package)
LGR	Local Grid Refinement
MNW	Multi-Node Well (Package)
PCG	preconditioned conjugate gradient (Solver)
TMR	telescopic mesh refinement
UPW	UPstream Weighting (Package)
USGS	U.S. Geological Survey
UZF	Unsaturated Zone Flow (Package)

MODFLOW–USG Version 1: An Unstructured Grid Version of MODFLOW for Simulating Groundwater Flow and Tightly Coupled Processes Using a Control Volume Finite-Difference Formulation

Sorab Panday,[1] Christian D. Langevin,[2] Richard G. Niswonger,[2] Motomu Ibaraki,[3] and Joseph D. Hughes[2]

Abstract

A new version of MODFLOW, called MODFLOW–USG (for UnStructured Grid), was developed to support a wide variety of structured and unstructured grid types, including nested grids and grids based on prismatic triangles, rectangles, hexagons, and other cell shapes. Flexibility in grid design can be used to focus resolution along rivers and around wells, for example, or to subdiscretize individual layers to better represent hydrostratigraphic units. MODFLOW–USG is based on an underlying control volume finite difference (CVFD) formulation in which a cell can be connected to an arbitrary number of adjacent cells. To improve accuracy of the CVFD formulation for irregular grid-cell geometries or nested grids, a generalized Ghost Node Correction (GNC) Package was developed, which uses interpolated heads in the flow calculation between adjacent connected cells. MODFLOW–USG includes a Groundwater Flow (GWF) Process, based on the GWF Process in MODFLOW–2005, as well as a new Connected Linear Network (CLN) Process to simulate the effects of multi-node wells, karst conduits, and tile drains, for example. The CLN Process is tightly coupled with the GWF Process in that the equations from both processes are formulated into one matrix equation and solved simultaneously. This robustness results from using an unstructured grid with unstructured matrix storage and solution schemes. MODFLOW–USG also contains an optional Newton-Raphson formulation, based on the formulation in MODFLOW–NWT, for improving solution convergence and avoiding problems with the drying and rewetting of cells. Because the existing MODFLOW solvers were developed for structured and symmetric matrices, they were replaced with a new Sparse Matrix Solver (SMS) Package developed specifically for MODFLOW–USG. The SMS Package provides several methods for resolving nonlinearities and multiple symmetric and asymmetric linear solution schemes to solve the matrix arising from the flow equations and the Newton-Raphson formulation, respectively.

Introduction

The U.S. Geological Survey (USGS) develops and supports the MODFLOW computer program for simulation of three-dimensional, steady-state and transient groundwater flow. The standard MODFLOW releases (McDonald and Harbaugh, 1988; Harbaugh and McDonald, 1996; Harbaugh and others, 2000; Harbaugh, 2005) are all based on a rectangular finite-difference grid. There are two notable restrictions with a standard finite-difference grid. The first is that irregularly shaped domain boundaries cannot be easily fitted with a rectangular grid. Although there are options for inactivating parts of the grid outside the domain of interest, the domain is still bounded by rectangular grid cells that may not follow irregular boundaries; as a result, information about the entire grid, including inactive cells, is read and processed. The second limitation of a rectangular finite-difference grid is that it is difficult to refine the grid resolution in areas of interest. Column and row widths can be variably spaced in order to focus grid resolution, but the added resolution must be carried out to the edges of the grid.

There have been a number of efforts to relieve the restrictions of the rectilinear finite-difference grid required by MODFLOW. These efforts have primarily focused on implementing curvilinear grids and nested grid methods.

[1]AMEC.

[2]U.S. Geological Survey.

[3]School of Earth Sciences, Ohio State University.

Curvilinear grids have been implemented for MODFLOW-based codes by Romero and Silver (2006), for example, but the approach has not been widely used. Use of nested grids with MODFLOW, however, has been a common approach for adding targeted resolution to areas of interest. The simplest nested grid approach is to use heads or fluxes from a regional model as boundary conditions for a higher resolution child model. This one-way coupling is commonly called telescopic mesh refinement (TMR). Leake and Claar (1999) developed the MODTMR computer program to facilitate the design of child grid boundary conditions from the output of a regional parent model. Mehl and Hill (2002, 2004, and 2005) improved upon the TMR approach through the development of the Local Grid Refinement (LGR) capability for MODFLOW–2005. MODFLOW–LGR iteratively solves the groundwater flow equations for parent and child grids until a converged solution is obtained for all grids. Schaars and Kamps (2001) also prototyped a LGR approach for MODFLOW, but used a single matrix solution as an alternative to iteration between grids.

This report describes implementation of a generalized control volume finite-difference (CVFD) formulation (sometimes referred to as an integrated finite-difference approach) into the MODFLOW–2005 framework. The formulation is similar to the CVFD formulation implemented in TOUGH2 (Pruess and others, 1999). The formulation is based on an unstructured grid approach, which allows users to design flexible grids that conform to aquifer boundaries and can be refined in areas of interest. This new program is called MODFLOW–USG, to denote a version of MODFLOW that supports UnStructured Grids. The approach implemented in MODFLOW–USG provides an alternative to other MODFLOW approaches for fitting irregularly shaped boundaries and adding targeted resolution to areas of interest. In contrast to the nested grid approaches of TMR and LGR in which individual parent and child grids are linked or coupled in an iterative manner, MODFLOW–USG simulates groundwater flow on all simple and nested grid connections using a fully implicit solution. A MODFLOW–USG grid can be a typical MODFLOW rectangular finite-difference grid, a combination of an arbitrary number of nested rectangular grids, or a grid composed of triangles, hexagons, irregular shapes, or combinations of these. For nested rectangular grids, the MODFLOW–USG approach is similar to the approach developed by Schaars and Kamps (2001). A key advantage of the MODFLOW–USG design is that groundwater flow over the entire grid is solved using a single matrix solution. For many complex groundwater problems, this approach will require less iteration for convergence than refinement approaches that iterate between grids. This approach also makes it easier to support packages that may move water between different grid nesting levels, such as the Stream Flow Routing (SFR) Package, for example.

The finite-difference formulation implemented in the Groundwater Flow (GWF) Process of MODFLOW generates a set of matrix equations that have a fixed pattern of nonzero entries. For example, a three-dimensional structured rectangular grid generates a 7-point connection pattern that includes a cell and its back and front neighbors in the three principal directions. An unstructured approach, however, generates nonzero matrix connections that are not based on a fixed pattern. An unstructured approach allows for an arbitrary number of connections between cells. In addition, the matrix can be expanded to include other flow processes. For example, this report also describes the Connected Linear Network (CLN) Process, which solves for flow through a connected linear network that may represent karst solution conduits, underground excavations (tunnels), agricultural tile drainage, or wells. For simulations that combine the CLN and GWF Processes, the unstructured matrix approach simultaneously solves for flow within the linear network, within the aquifer, and between the linear network and the aquifer in a single matrix. Furthermore, the linear sparse matrix solvers implemented in MODFLOW–USG can solve matrix equations that have an asymmetric conductance matrix. The option to handle asymmetric matrices is used here to incorporate the Newton-Raphson formulation developed by Niswonger and others (2011), as well as a fully implicit Ghost Node Correction (GNC) Package.

Purpose and Scope

The purpose of this report is to document the MODFLOW–USG computer program by (1) describing the underlying CVFD method and how the method is implemented for unstructured grids; (2) describing the design of the program and implementation details; (3) providing guidance on designing and running MODFLOW–USG models, and (4) demonstrating MODFLOW–USG for several different test cases. A description of the input instructions for MODFLOW–USG is not included herein; however, input instructions, format of the binary model output, and other information that may evolve with the program is included in the software distribution available on the Internet.

For this report, the reader is assumed to be familiar with MODFLOW–2005 and its underlying concepts. Those not familiar with MODFLOW–2005 are referred to Harbaugh (2005). Users may also need to refer to the relevant documentation for some of the other packages and processes incorporated into MODFLOW–USG. For example, MODFLOW–USG contains the Lake Package, and thus, users will need to refer to the Lake Package documentation (Merritt and Konikow, 2000) for a detailed description. The list of symbols and notations used in this report is compiled in appendix 1.

MODFLOW–USG Overview

Most components of MODFLOW–USG are functionally similar to MODFLOW–2005. Both programs use stress periods and time steps for temporal discretization. Like MODFLOW–2005, MODFLOW–USG runs from the command line and reads a name file containing a list of active packages, processes, and input and output data files. MODFLOW–USG

is capable of simulating an existing MODFLOW–2005 dataset of supported packages provided the original MODFLOW–2005 solver package is replaced with the Sparse Matrix Solver (SMS) Package described herein. The ability to use cell shapes other than rectangles is another important difference between MODFLOW and MODFLOW–USG. Although cells can be variably shaped in the horizontal direction, MODFLOW–USG requires that cells are prismatic in the vertical direction. Cells can also be grouped into layers for easier processing, and sublayering can be used to further divide cells. To facilitate this new flexibility in grid design, MODFLOW–USG identifies cells by node number when used with an unstructured grid dataset, instead of by layer, row, and column, as is done in MODFLOW. There is an obvious difference between indexing cells on the basis of node numbers and indexing cells on the basis of layer, row, and column; however, the concept of applying hydrologic stresses, using boundary packages, to individual cells remains consistent with MODFLOW concepts, even if the model cells have a nonrectangular shape. Another substantial difference is the way in which the connectivity between cells is represented. In MODFLOW, there is no need to specify connection information, because each cell is logically connected to the six surrounding cells in the principal directions, and connected cells are easily determined from the layer, row, and column indices. With the unstructured-grid option in MODFLOW–USG, users must define this connectivity. Users provide connectivity information to MODFLOW–USG in the form of two arrays; the first array contains the number of connections for each cell, and the second array contains a list of the connected node numbers for each cell. Application of graphical user interfaces (GUIs) greatly simplifies input for unstructured grids by internally generating this connectivity information.

A Newton-Raphson formulation was recently developed for MODFLOW–2005 by Niswonger and others (2011). This formulation eliminates inactivation of dry cells and the abrupt reactivation of rewetted cells, which can cause model convergence problems. MODFLOW–USG contains this Newton-Raphson formulation to help resolve nonlinearities associated with wetting and drying of model grid cells as well as those nonlinearities introduced by some boundary packages and processes.

The matrix solvers distributed with MODFLOW–2005 were specifically developed for a structured grid in which each cell is connected to the six adjacent cells, and the coefficients in the matrix are symmetric about the main diagonal. These matrix solvers cannot be used with an unstructured grid, and are therefore not included in MODFLOW–USG. MODFLOW–USG contains several flexible matrix solvers that can be used with an unstructured grid. These solvers are packaged into the SMS Package and include an asymmetric sparse matrix solver called χMD (Ibaraki, 2005) and an unstructured preconditioned conjugate gradient (PCG) solver developed by White and Hughes (2011) for symmetric equations. Solution of the groundwater flow equation is managed in MODFLOW–USG by the SMS Package. The SMS Package manages the outer (nonlinear) iteration loop by use of either Picard iteration or the Newton-Raphson formulation described by Niswonger and others (2011) and implements under-relaxation and residual control measures as required. The SMS Package also invokes the selected linear matrix solver. The unstructured matrix solvers managed by the SMS Package provide the foundation for many of the capabilities provided by MODFLOW–USG.

The CLN Process is also included with MODFLOW–USG. One-dimensional features simulated with the CLN Process can represent wells, a network of tile drains, or any other network of tubular conduits (for example, karst conduits and underground excavations) that is present within an aquifer. This new package replicates some of the capabilities of the Conduit Flow Process (Shoemaker and others, 2007) and the Multi-Node Well (MNW) Packages (Halford and Hanson, 2002; Konikow and others, 2009), which are not included with the current version of MODFLOW–USG. The CLN nodes are implemented into the simulation in a fully implicit manner and solved in the same matrix as the groundwater flow equation to improve convergence properties. The fully implicit and tight coupling of CLN nodes to aquifer cells is possible because of the unstructured design of the matrix solvers provided with MODFLOW–USG. The formulation and implementation of the CLN Process is also detailed herein.

For accurate flux calculations, the CVFD method requires certain geometrical properties pertaining to the cell connections. For most grid types, this means that the line connecting the centers of two cells should bisect the shared edge at right angles. This CVFD requirement is violated for irregular polygon cell geometries or nested grids, thus introducing errors in simulated flows and heads. The larger the deviation is from this CVFD requirement, the larger the error. MODFLOW–USG includes a GNC Package for reducing such errors. The GNC Package is optional because no correction is needed for simple grid (as opposed to nested grid) connections of regular polygon, equilateral triangle or rectangular shaped cells. In addition, the package may not be needed for many grids even when they violate these geometric cell properties. But for certain grid types and flow patterns, the corrections may be required to ensure an accurate solution. The GNC Package was developed in an implicit manner such that the corrections are part of the matrix solution; however, options for updating the GNC terms on the right-hand-side vector are also included so that the symmetric linear solvers available with MODFLOW–USG can be used. A simple test problem using a nested grid is presented herein to demonstrate the effectiveness of the GNC Package.

Packages and Processes Supported in Version 1

With the exception of the matrix solvers, MODFLOW–USG supports most of the packages documented by Harbaugh (2005). MODFLOW–USG also supports additional packages. A notable exception is the Unsaturated Zone Flow (UZF) Package, which is not supported in the MODFLOW–USG version documented herein. A list of the packages supported in Version 1 of MODFLOW–USG is provided in table 1.

MODFLOW–2005 has three packages for representing groundwater flow between cells—Block Centered Flow (BCF), Layer Property Flow (LPF), and Hydrogeologic-Unit Flow (HUF)—and users must select one of these for a simulation. MODFLOW–USG has only one package to represent groundwater flow between adjacent cells; this package is activated by providing either a BCF Package input file or an LPF Package input file. Therefore, MODFLOW–USG only supports the capabilities of BCF and LPF; the HUF Package and its capabilities are not supported in the present version of MODFLOW–USG. The capabilities of the Upstream Weighting (UPW) Package in MODFLOW–NWT (Niswonger and others, 2011) were also implemented in MODFLOW–USG. These capabilities are included as an additional LAYTYP option in input files of the BCF and LPF Packages. MODFLOW–USG does not read a UPW Package input file as does MODFLOW–NWT. Details of BCF, LPF, HUF, and UPW packages and associated LAYTYP options are available in the MODFLOW–2005 and MODFLOW–NWT documentation (Harbaugh, 2005; Niswonger and others, 2011).

Mathematical and Numerical Formulation

The use of unstructured grids is common in the Earth sciences to accurately discretize complex geometries that occur in the subsurface. These grids can be solved numerically using finite element or finite difference approaches. Finite-element solutions for groundwater flow that provide flexibility in spatial discretization are discussed by Torak (1993a, b), Cooley (1992), DHI–WASY GmbH (2010), Huyakorn and Pinder (1983), and Therrien and others (2004). The MODFE code discussed by Torak (1993a, b) and Cooley (1992) provides two-dimensional areal and axisymmetric solutions for confined and unconfined groundwater flow on triangular finite-element meshes. The SUTRA code (Voss and Provost, 2002) represents three-dimensional groundwater flow and solute transport using hexahedral finite elements. The FEFLOW code by DHI–WASY GmbH (2010) uses a Galerkin based finite-element method with prismatic quadrilaterals or

Table 1. List of packages and processes supported in Version 1 of MODFLOW-USG documented in this report.

[UPW, Upstream Weighting Package]

Abbreviation	Process or package name	Reference
BAS	Basic	Harbaugh (2005)
BCF6	Block-centered flow, includes upstream weighting capability of MODFLOW-NWT	Harbaugh (2005), Niswonger and others (2011)
LPF	Layer-property flow includes UPW capability of MODFLOW-NWT	Harbaugh (2005), Niswonger and others (2011)
HFB	Horizontal flow barrier	Hsieh and Freckleton (1993)
CHD	Time-variant specified head option	Harbaugh (2005)
RCH	Recharge	Harbaugh (2005)
EVT	Evapotranspiration	Harbaugh (2005)
FHB	Transient flow and head boundary	Leake and Lilly (1997)
WEL	Well	Harbaugh (2005)
DRN	Drain	Harbaugh (2005)
RIV	River	Harbaugh (2005)
GHB	General-head boundary	Harbaugh (2005)
STR7	Stream	Prudic (1989)
SFR2	Streamflow routing with unsaturated flow beneath streams	Prudic and others (2004), Niswonger and Prudic (2005)
GAGE	Stream-gage monitoring	Prudic and others (2004), Niswonger and Prudic (2005), Merritt and Konikow (2000)
LAK3	Lake	Merritt and Konikow (2000)
SUB	Subsidence	Hoffmann and others (2003)
SMS	Sparse Matrix Solver (includes options for Newton-Raphson formulation, linear solvers, and under-relaxation formulas)	This report
GNC	Ghost-node correction	This report
CLN	Connected linear network	This report

triangles. Huyakorn and Pinder (1983) introduced the use of influence coefficients with the Galerkin finite-element method to avoid time-consuming elemental integration. This was expanded for nested elements with an unstructured connectivity by Panday and others (1993). The HydroGeoSphere code of Therrien and others (2004) uses the control volume finite element (CVFE) method to provide gridding flexibility and avoid some of the disadvantages of upstream weighting in finite-element discretizations (Forsyth, 1991).

The multiphase flow code TOUGH2 (Pruess and others, 1999) provides gridding flexibility by using the CVFD methodology with unstructured grids. The MODFLOW–SURFACT (Panday and Huyakorn, 2008) and MODHMS (Panday and Huyakorn, 2004) codes that solve for subsurface and integrated hydrologic problems, respectively, include an orthogonal curvilinear grid capability which, along with the MODFLOW option of deactivating cells outside of the domain, provides flexibility in handling irregular domain geometries and refined grid requirements within a structured connectivity framework.

Coupled Processes and the Generalized CVFD Formulation

MODFLOW–USG provides a framework for tightly coupling multiple hydrologic processes. The tight coupling, in contrast to a sequential or iterative coupling approach, occurs through the formulation of a global conductance matrix that includes the cells for all processes. The framework allows individual MODFLOW–USG processes to add to the global conductance matrix in order to represent fluxes between cells within a process as well as with cells of other processes. The global conductance matrix can be symmetric or asymmetric and is unstructured, indicating that an individual cell may have an arbitrary number of connections with other cells. The CVFD formulation accommodates this unstructured framework of tightly coupling flow processes as well as of allowing flexibility in cell geometry and connectivity within processes.

Following is the general form of a CVFD balance equation for cell n:

$$\sum_{m \in \eta_n} C_{nm}(h_m - h_n) + HCOF_n(h_n) = RHS_n , \qquad (1)$$

where

C_{nm} is the inter-cell conductance between cells n and m,

h_n and h_m are the hydraulic heads at cells n and m,

$HCOF_n$ is the sum of all terms that are coefficients of h_n in the balance equation for cell n, and

RHS_n is the right-hand-side value of the balance equation.

Note that the summation of the first term is over all cells m that are an element of η_n, the set of cells that are connected to cell n. C_{nm} is a constant in some cases (for flow between two cells in a confined aquifer, for example) but is often dependent on the values of h_n and h_m (for flow between two cells in an unconfined aquifer, for example). Further, note that the $HCOF_n$ terms result from changes in storage, and boundary fluxes that are dependent on the value of h_n. Also, RHS_n contains terms related to storage and (or) boundary conditions. The first term of equation 1 expresses the volumetric flow, Q_{nm}, between two connected cells, n and m, as

$$Q_{nm} = C_{nm}\left(h_m - h_n\right) . \qquad (2)$$

Equation 1 is expressed in matrix form as

$$\mathbf{Ah=b}, \qquad (3)$$

where in MODFLOW–USG,

A is the global conductance matrix,

h is the vector of hydraulic heads, and

b is the right-hand-side vector.

The diagonal terms of **A** (where $n = m$) correspond to the *HCOF* vector minus the sum of the off-diagonal conductances. For confined cases, equation 3 is linear and can be solved for **h**, the distribution of heads, at any given time step or stress period. For unconfined cases, equation 3 is nonlinear whereby one or more of the coefficients in the conductance matrix are functions of hydraulic head. In that case, an iterative Picard solution approach repeatedly solves equation 3 until a specified level of convergence is met. For each Picard iteration, the global conductance matrix is reformulated using heads from the previous iteration. An optional Newton-Raphson approach in MODFLOW–USG can be used to accelerate and improve the convergence of unconfined groundwater simulations and other nonlinear problems.

This report documents the tight coupling of the GWF and CLN Processes within MODFLOW–USG; however, the program supports addition of new processes that can be coupled with GWF, CLN, or other processes that may be implemented in the future. MODFLOW–2005 and its predecessors provide a framework for adding packages and processes that interact with GWF. In the MODFLOW–2005 framework, packages interact with the GWF Process primarily as sources and sinks; for the case when the source or sink head is itself a variable, the MODFLOW–2005 framework does not support an approach for adding new cells that can be solved simultaneously with the GWF Process. Instead, the boundary variable is solved separately from, and in an iterative fashion with, the GWF Process solution. For many packages, this is not a problem; however, when hydraulic features are strongly connected to an aquifer, the iterative approach may cause oscillations in the flow solution. Consequently, the solution may not converge efficiently, and in some cases, it may not converge at all. MODFLOW–USG provides a different framework whereby the new cells can be solved simultaneously with the GWF Process. The MODFLOW–USG program also supports implementation of sources and sinks as packages, but it further extends the modular concept to the matrix level so that the packages apply to the GWF Process, the CLN process,

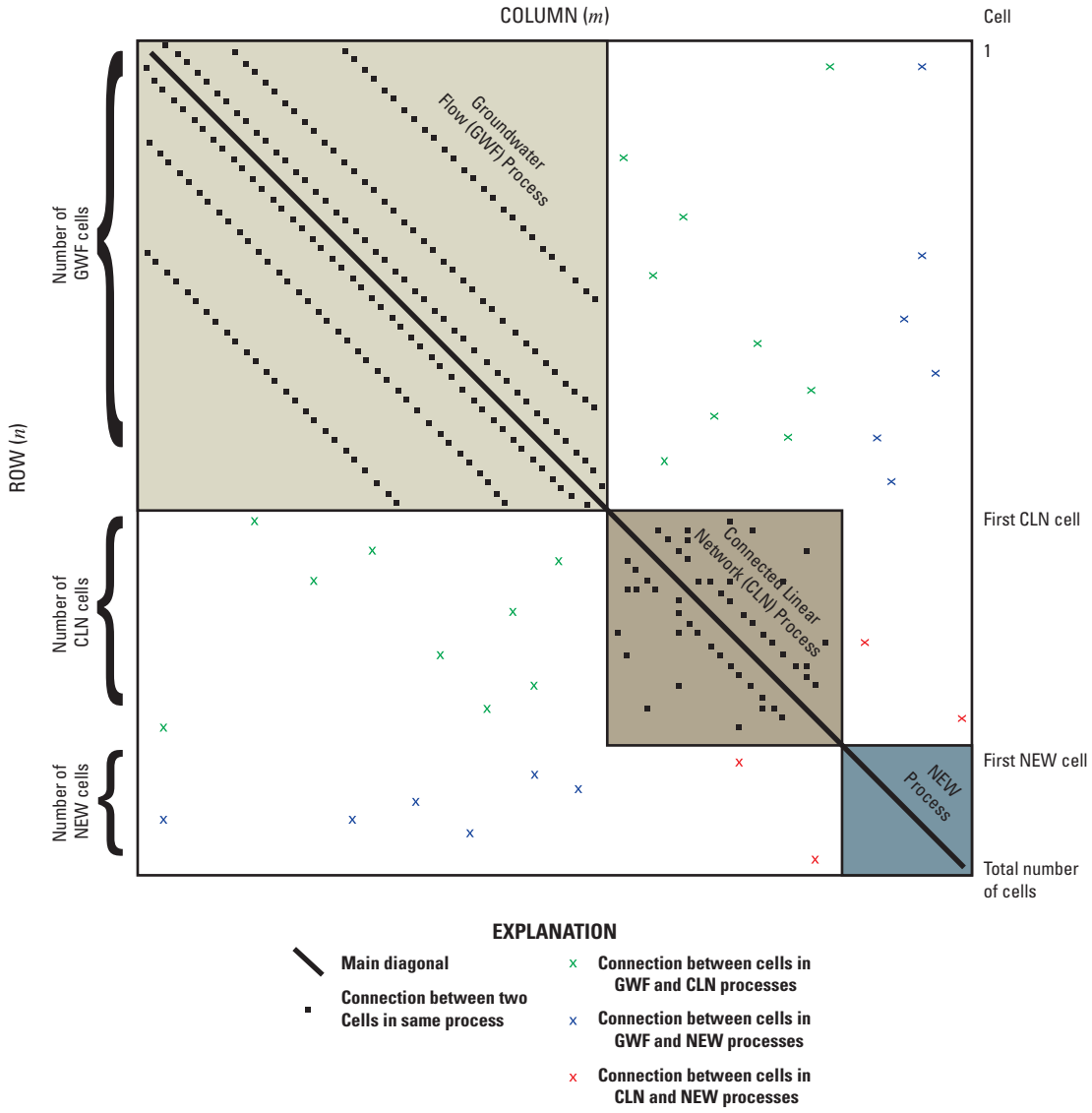

Figure 1. Conductance matrix in MODFLOW-USG illustrating the framework for tightly coupling cells with cells from other processes.

and other processes that may be added in the future. This concept is graphically illustrated in figure 1 as a schematic of a conductance matrix for a hypothetical MODFLOW–USG simulation. The conductance matrix is square, with the number of columns and number of rows equal to the total number of cells in the problem. Cells of the GWF Process correspond to rows 1 through the number of GWF cells. Rows for the CLN Process and some other new process that might be added in the future are also shown in figure 1.

A key component of the MODFLOW–USG approach is the ability of the linear sparse matrix solvers to handle the unstructured nature of the global conductance matrix. MODFLOW–USG takes advantage of this flexibility by providing a framework for connecting cells of different processes. As described herein, MODFLOW–USG also takes advantage of this flexibility within a process by allowing cells to be connected to an arbitrary number of neighbors. Thus, grids other than the structured grids required by MODFLOW–2005 and its predecessors can be used with MODFLOW–USG. Complicated networks of linear flow features can also be represented.

The Groundwater Flow (GWF) Process

The GWF Process in MODFLOW–USG is an extension of the GWF Process in MODFLOW–2005. The primary difference is that the GWF Process in MODFLOW–USG is based on an unstructured grid formulation, which allows a cell to be connected to an arbitrary number of other cells. This section describes the GWF Process as implemented in MODFLOW–USG.

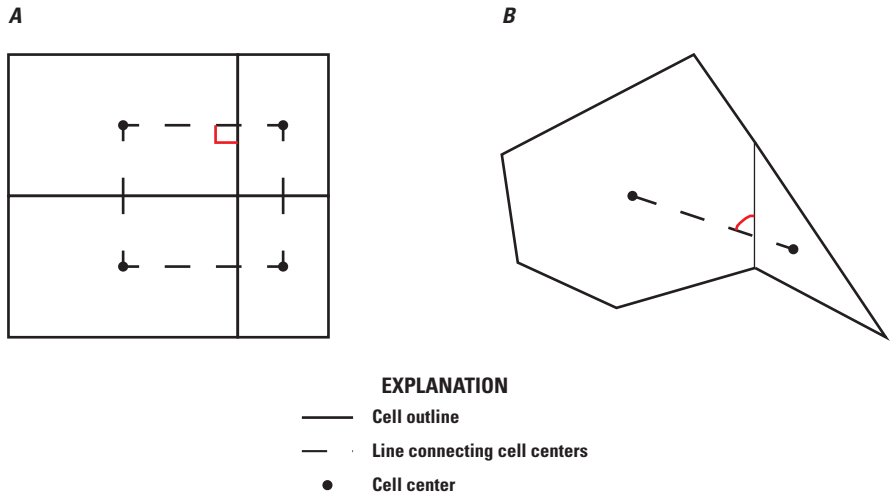

Figure 2. Examples of two different types of cell connections: *A*, a line connecting the centers of adjacent cells passes through the shared face at a right angle, and *B*, a connecting line does not intersect the shared face at a right angle.

Mathematical Model for Groundwater Flow

Narasimhan and Witherspoon (1976) provide the details on how to rewrite the groundwater flow equation into a form that provides the foundation for the CVFD method implemented in MODFLOW–USG. Dehotin and others (2011) provide a similar derivation for the case of two-dimensional groundwater flow. The equations are briefly summarized here. The three-dimensional transient groundwater flow equation can be written as

$$\nabla \cdot \left(K \nabla h \right) = S_s \frac{\partial h}{\partial t} + W \,, \tag{4}$$

where

K is hydraulic conductivity [L/T],
h is hydraulic head [L],
S_s is specific storage [1/L],
t is time [T], and
W is a volumetric source or sink per unit volume [1/T].

By integrating over a small control volume, V [L^3/T], equation 4 can be rewritten as

$$\int_V \left(K \nabla h \right) dV = \frac{\partial}{\partial t} \int_V \left(S_s h \right) dV + \int_V W dV \,. \tag{5}$$

Application of the divergence theorem, also known as Green's Theorem, can be used to convert the advective term on the left-hand side of equation 5 from a volume integral into a surface integral:

$$\int_S \left(K \nabla h \right) \cdot \mathbf{n} dS = S_s V \frac{\partial h}{\partial t} + WV \,, \tag{6}$$

where

S is the surface of the control volume, and
$\mathbf{n}$ is an outward-pointing unit normal (perpendicular) vector on the volume surface.

Equation 6 expresses the concept that within a control volume, the sum of inflows and outflows across the surface must balance any changes in storage and any fluxes from internal sources or sinks.

Unstructured Grid Discretization

Application of a CVFD approximation to equation 6 requires imposition of spatial and temporal discretization. In MODFLOW, space is discretized in three dimensions using a rectangular finite-difference grid (fig. 2–1 of Harbaugh, 2005). The grid is created from layers, rows, and columns of cells ordered in a Cartesian coordinate system with each cell being connected to the two adjacent cells along each coordinate direction. In three dimensions, this results in a 7-point structured connectivity for the discretized set of equations. This means that a single model cell is connected, at most, to six surrounding model cells. Because the number of connections remains constant in space (except along boundaries), the grid used by MODFLOW is called a structured grid. A regular grid of hexagons is also structured, because the number of connections is the same for all cells.

The term "unstructured grid" simply means that the number of connections may be variable for each cell. For example, with mesh-centered triangular finite-elements, a node may be common to several elements, and this connectivity may vary for each node. This variability results in an unstructured system of equations. Similarly, in CVFD schemes the connectivity of a cell depends on the number of shared faces, which may vary for each cell.

For accurate solutions, the standard CVFD formulation requires that a line drawn between the centers of two connected cells should intersect the shared face at a right angle (fig. 2). Furthermore, the intersection point should coincide with an

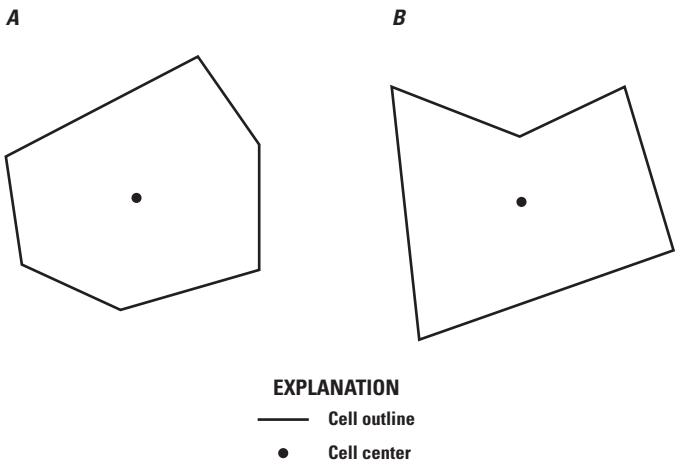

Figure 3. Examples of *A*, convex and *B*, concave polygons. A convex polygon is one in which all interior angles are less than or equal to 180 degrees. A concave polygon has at least one interior angle that is greater than 180 degrees. MODFLOW-USG cells should have a convex shape.

appropriate mean position on the shared face (Narasimhan and Witherspoon, 1976). For grids based on a Cartesian coordinate system, this mean position will be the center of the shared face; therefore, a line connecting cell centers should bisect the shared face at a right angle. For cylindrical grids, the mean position on the shared face does not coincide with the midpoint, but rather, the logarithmic mean of the radii (Narasimhan and Witherspoon, 1976). Although this CVFD requirement is met for a simple grid of regular polygons, equilateral triangles, and rectangles, it is violated for nested grids and may be violated for grids with nonregular polygon-shaped cells. The requirement is also violated for cells with a concave shape unless cylindrical or spherical coordinates are used; thus, convex shapes should be used for most grid types (fig. 3). The smaller the deviation from this CVFD requirement, the smaller the loss of accuracy in the groundwater flow solution. In addition, the errors generally decrease as resolution increases, but they are difficult to quantify. The GNC Package, which is described herein, can be used for some grid types to improve accuracy when the CVFD requirement is violated. The possibility of violations of, and corrections to, the CVFD requirement are noted in discussions of grid types or cell geometries throughout this report.

The unstructured grid formulation for MODFLOW–USG is developed in a similar manner to the CVFD methodology implemented in the TOUGH2 code (Pruess and others, 1999). In TOUGH2, the domain is defined by a list of finite volumes and a list of flow connections between them. The geometric information for spatial discretization is provided in the form of a list of volumes, interface areas, and nodal distances, and there is no reference whatsoever to a global system of coordinates for a particular flow problem. MODFLOW–USG also does not require information about cell shapes or how cells are positioned in space. Instead, MODFLOW–USG only requires information about connection and cell properties. This means that users can construct a wide variety of different grid types, even ones that can substantially violate the CVFD requirement, as may be created by using common finite-element and finite-volume mesh generation software. With this flexibility, it is incumbent upon the user to ensure that the grid used to discretize the domain is appropriate for the problem geometry and flow system, and that violation of the CVFD requirement does not introduce large errors in the flow solution, or that the appropriate correction is provided with the GNC Package—something that may not be straightforward for connections of irregular grid shapes. Otherwise, errors in the simulated results may be large, even for converged solutions with small mass-balance errors.

Figure 4 shows examples of different types of structured and unstructured grids that may be defined for the GWF Process in MODFLOW–USG. The top part of figure 4 shows structured model grids in which the number of connections is the same for all cells (except for along boundaries). For structured rectangular grids (fig. 4*A*), the CVFD methodology is identical to a conventional finite-difference formulation (for example, Peaceman, 1977, and Moridis and Pruess, 1992). For the unstructured grids shown in the bottom part of figure 4, the number of connections for each cell is variable throughout the grid. Unstructured grids are useful when the scale of interest or the magnitude of the hydraulic gradient varies throughout the domain. Unwarped structured grids are appealing because they do not violate the standard CVFD requirement. The radial grid (fig. 4*J*) also meets the CVFD requirement. The warped triangular (fig. 4*F*) and quadrilateral (fig. 4*G*) grids and the unstructured grids in figure 4 can be used with MODFLOW–USG, but it may be necessary to use the GNC Package to improve the accuracy of the flow solution.

MODFLOW–USG requires that top and bottom cell faces are horizontal and that side faces are vertical; therefore, cells are prismatic in the vertical direction. The vertices defining the top cell face must have the same x and y coordinates as the vertices defining the bottom face of the cell. Cell tops and bottoms are horizontal and flat so that the transition between unconfined and confined conditions is handled the same way as it is handled in MODFLOW and MODFLOW–NWT. For convertible layers, when the water table is above the cell top, cell transmissivity is a function of cell thickness, whereas when the water table is below the cell top, transmissivity is a function of the cell saturated thickness. Conversion between unconfined and confined storage properties is also dependent on the cell head in relation to the cell top, as it is in MODFLOW.

MODFLOW–USG uses the concept of layers to facilitate easier pre- and post-processing, and the approach is flexible in that the number of cells can differ between layers. Alternatively, a three-dimensional, multi-layer grid can be input as a single layer to MODFLOW–USG, in which case, additional pre- and post-processing may be required to create the model input and analyze the results. If the layer concept is used for a simulation, cells need to be labeled consecutively within a grid from the top layer downward. Therefore, the lowest-numbered cells must be

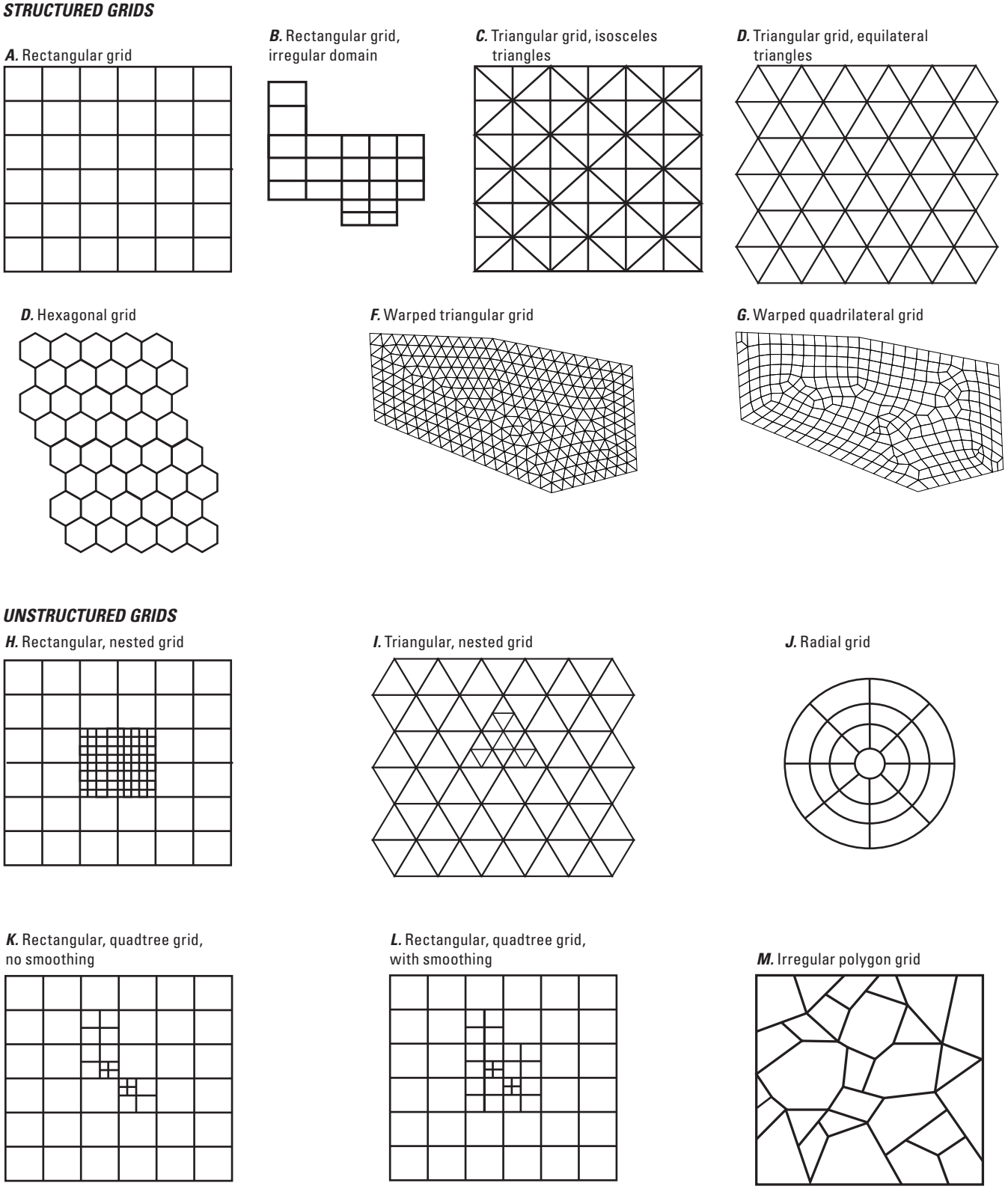

Figure 4. Examples of different types of structured and unstructured grids.

at the top of the grid, and cell numbers must increase downward. This numbering structure is used internally by MODFLOW–USG to identify the downward direction for cell connections. In MODFLOW–USG layers can also be subdiscretized in the vertical dimension. This capability can be useful for adding vertical resolution near partially penetrating wells, for example.

Several different layering schemes are shown in figure 5 for a hypothetical aquifer system in which an upper aquifer is separated from a lower aquifer by a confining unit. In the simplest scheme, the grid configuration is the same for both aquifers and the confining unit (fig. 5*A*). In plan view, the grid can be unstructured, but the same horizontal grid is used for all layers. MODFLOW–USG will also accept a grid in which a different configuration is used for each layer. This approach can be useful, for example, if the upper aquifer contained a discontinuous confining unit as shown in figure 5*B*. In this configuration, the upper aquifer is represented as three layers. The cells marked 18 to 24 correspond to a confining unit and are assigned to layer 2. The cells marked 25 to 31 correspond to areas in the upper aquifer that are beneath the discontinuous confining unit. Figure 5*C* illustrates the vertical subdiscretization concept in MODFLOW–USG. In this configuration, additional vertical resolution was added within the upper and lower aquifer. When vertical subdiscretization is used, cells within a layer do not have to be numbered from top to bottom. Instead, MODFLOW–USG requires an additional input array that indicates whether a connection between two cells is vertical or not. Also, a larger node number should reside below a smaller node number to identify the downward direction.

CVFD and the Finite-Difference Approximation

For the structured grid used by MODFLOW, the finite-difference expansion of the groundwater flow equation (Harbaugh, 2005, eq. 6–1) is:

$$\begin{aligned} &CV_{i,j,k-1/2}h_{i,j,k-1} + CR_{i-1/2,j,k}h_{i-1,j,k} + CC_{i,j-1/2,k}h_{i,j-1,k} + \\ &(-CV_{i,j,k-1/2} - CR_{i-1/2,j,k} - CC_{i,j-1/2,k} - CV_{i,j,k+1/2} - \\ &CR_{i+1/2,j,k} - CC_{i,j+1/2,k} + HCOF_{i,j,k})h_{i,j,k} + CV_{i,j,k+1/2}h_{i,j,k+1} \\ &+CR_{i+1/2,j,k}h_{i+1,j,k} + CC_{i,j+1/2,k}h_{i,j+1,k} = RHS_{i,j,k} \end{aligned}, \quad (7)$$

where

CV, CR, and CC are the inter-cell conductances for the layer, row, and column directions, respectively, and

$i, j,$ and k represent the row, column and layer location of a cell.

The *HCOF* and *RHS* accumulators include the combined stresses and storage terms. Stress terms that are a function of the head in cell i,j,k are added to *HCOF*, whereas constant stress terms are subtracted from the *RHS* array on the right side of the flow equation. Equation 7 indicates that cell i,j,k, is connected to its neighbors (cells i,j,k-1, i,j,k+1, i-1,j,k , i+1,j,k , i,j-1,k and i,j+1,k) yielding a structured 7-point connectivity for a three-dimensional Cartesian system.

Equation 4 can be rearranged and written in a general form for cell n connected to each of its cell m neighbors as

$$HCOF_n(h_n) + \sum_{m\in\eta_n} C_{nm}(h_m - h_n) = RHS_n . \quad (8)$$

This is the same form as the global CVFD equation presented previously in equation 1; therefore, the CVFD methodology can be readily implemented into the MODFLOW framework by a generalization of the rectangular flow connection terms. The time-dependent storage term included in the $HCOF_n$ and RHS_n terms may be expressed for nonrectangular cells directly in terms of the cell volume, as in equation 2–26 of the MODFLOW–2005 documentation (Harbaugh, 2005), as

$$HCOF_n = \frac{-SS_n V_n}{\Delta t} \quad (9)$$

and

$$RHS_n = \frac{-SS_n V_n h_n^{t-1}}{\Delta t} , \quad (10)$$

where

t-1 is the previous time step,

Δt is the time step size,

SS_n is the specific storage of the cell defined as the volume of water that can be injected per unit volume of aquifer material per unit change in head, and

V_n is the volume of cell n.

The terms $HCOF_n$ and RHS_n may be varied further to accommodate storage term conversion as detailed in equations 5–35 through 5–38 of the MODFLOW–2005 documentation (Harbaugh, 2005), or may be written in a continuous form for the Newton-Raphson formulation as implemented in the UPW Package (Niswonger and others, 2011).

Inter-Cell Conductance

The inter-cell conductance terms in the column, row and layer directions (the coefficients CC, CR and CV of MODFLOW–2005 as expressed by equation 2–9 in the MODFLOW–2005 documentation (Harbaugh, 2005) for coefficient CR) may be generalized for the CVFD formulation to provide an inter-cell conductance term C_{nm} between a cell n and any of its neighbors m as:

$$C_{nm} = \frac{a_{nm}K_{nm}}{\left[L_{nm} + L_{mn}\right]} , \quad (11)$$

where

a_{nm} (equal to a_{mn}) is the perpendicular saturated flow area between cells n and m,

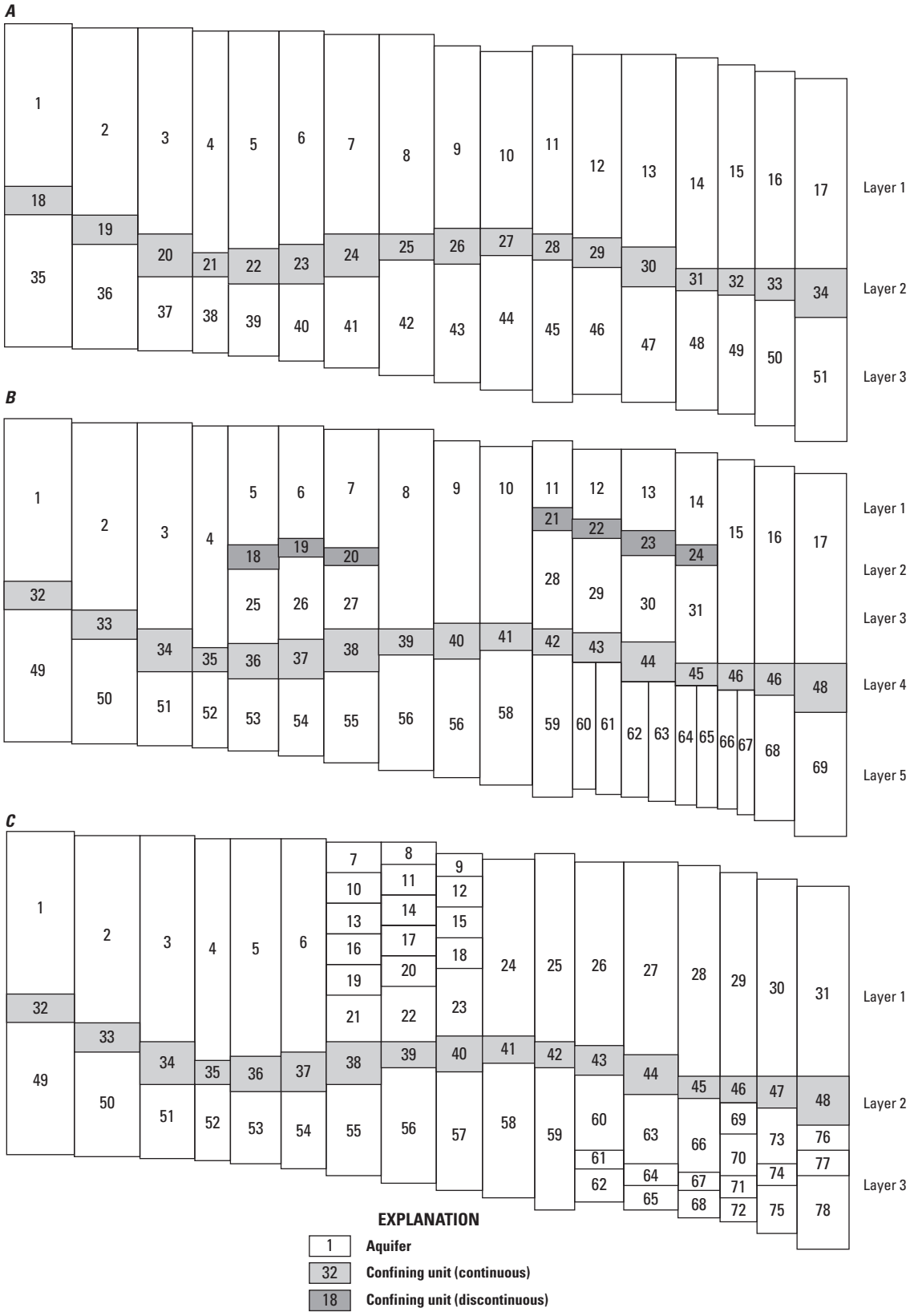

Figure 5. Types of layering schemes that can be used with MODFLOW-USG: *A*, the grid configuration is the same for all layers; *B*, the grid configuration is different for different layers, and *C*, one or more layers contain vertical subdiscretization.

K_{nm} is the inter-cell hydraulic conductivity between cells n and m, and

L_{nm} and L_{mn} are the perpendicular distances between the shared n-m interface and a point within cells n and m respectively.

For structured grids, equations 8, 9, 10, and 11 reduce to the finite-difference equations of MODFLOW, where some of these terms are internally computed using rectangular geometric considerations.

For connected cells that meet the standard CVFD requirement, the point used to determine the connection lengths corresponds to the cell centers. In most cases, the center is defined as the cell centroid, but depending on the cell shape, it may be advantageous to use other center definitions. For example, the circumcenter can be used for triangular cells as it defines the intersection point of the cell bisectors (Lal, 1998; Lal and others, 2005). Where the standard CVFD requirement is not met, L_{nm} and L_{mn} should be calculated using geometric properties of the connected cells. For example, in figure 6, the line connecting the centers of cells n and m (shown as filled black circles) does not bisect the shared face at a right angle. One way to calculate L_{nm} and L_{mn} is to locate the midpoint of the shared face (P_{nm}), and extend a perpendicular line outward from this point in both directions. L_{nm} and L_{mn} are defined as the distances between the midpoint of the shared face (P_{nm}) and points on the perpendicular line (shown as an open circles) closest to the cell centers. These points may vary within a cell for each connection, and may be considered locations for ghost nodes (as discussed later). Additional details on these calculations for groundwater models are provided, for example, by de Marsily (1986) and Dickinson and others (2007) for nested grids and by Dehotin and others (2011) for irregular polygons. The procedure for determining L_{nm} and L_{mn} from geometrical considerations also applies in three dimensions whereby a line normal to the shared face is extended outward in both directions from the center of the shared face. Figure 7 shows the face areas and connection lengths in three dimensions for various cell types.

When unstructured grids are used with MODFLOW–USG, users must enter the individual terms of the conductance equation 11. Users must provide the area of the shared cell face for every cell connection. For horizontal connections between unconfined cells as in figure 7*A*, where the saturated area is a function of the water table height, users enter the area of the shared face based on the average cell thicknesses (top minus bottom) of the two interacting cells n and m. For vertically nested grids as in figure 7*B*, the user should select the minimum cell thickness of the two interacting cells, n and m to compute the area of the face between them. During the simulation, MODFLOW–USG adjusts the cell connection area to a saturated flow area using the simulated water table height.

The numerator of the inter-cell conductance term, $a_{nm}K_{nm}$, in equation 11 depends on the selected options for inter-cell conductance provided in the BCF or the LPF Package inputs, as discussed in the MODFLOW–2005 documentation

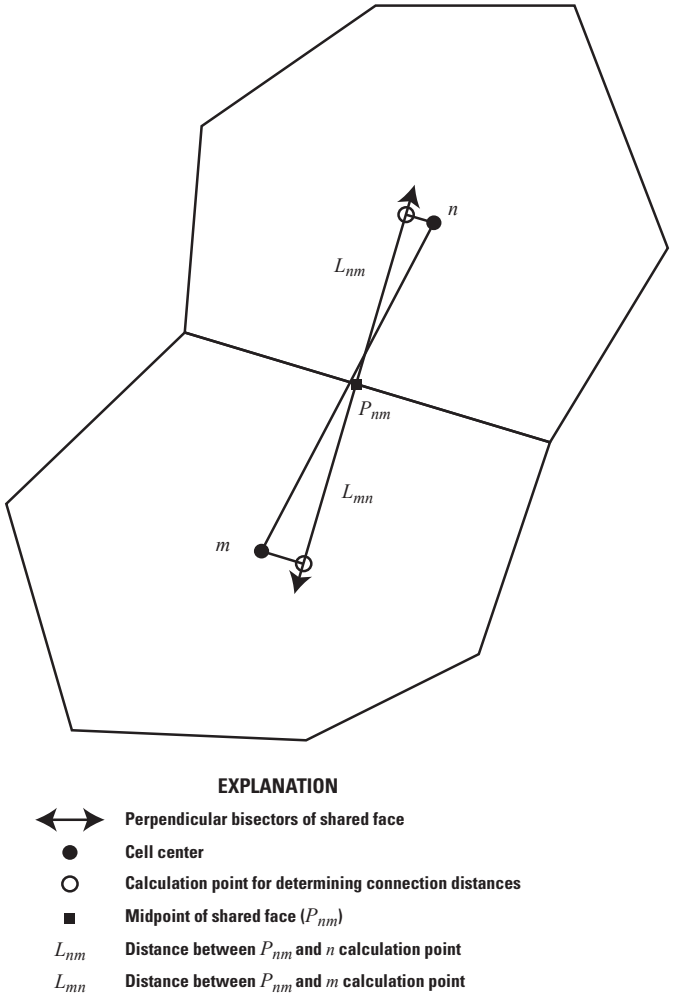

Figure 6. Connection between cells n and m showing the connection distances to use in the conductance formulation.

(Harbaugh, 2005). Specifically, the horizontal inter-cell conductance is determined by the LAYAVG option selected for the simulation. Thus, for the BCF Package, if LAYAVG = 0, a weighted harmonic mean transmissivity of cells n and m is selected to represent $a_{nm}K_{nm}$; if LAYAVG = 1, an arithmetic mean transmissivity is used, and if LAYAVG = 2, the logarithmic-mean transmissivity method is used as detailed in chapter 5 of Harbaugh (2005). The LTYPE option of the LPF Package (or the LAYCON option of the BCF Package) is used to determine if the cell is treated as confined, unconfined, or convertible, as also discussed in Harbaugh (2005). Accordingly, the total or saturated cell thicknesses are used in the transmissivity computation.

For the upstream weighting formulation of Niswonger and others (2011), the numerator of the inter-cell conductance term, $a_{nm}K_{nm}$, is calculated as the inter-cell conductivity, K_{nm}, times the saturated area of the upstream cell, a_{nm}. The inter-cell hydraulic conductivity for this case is computed from the hydraulic conductivity values of the individual cells using averaging options provided by the LAYAVG flag just

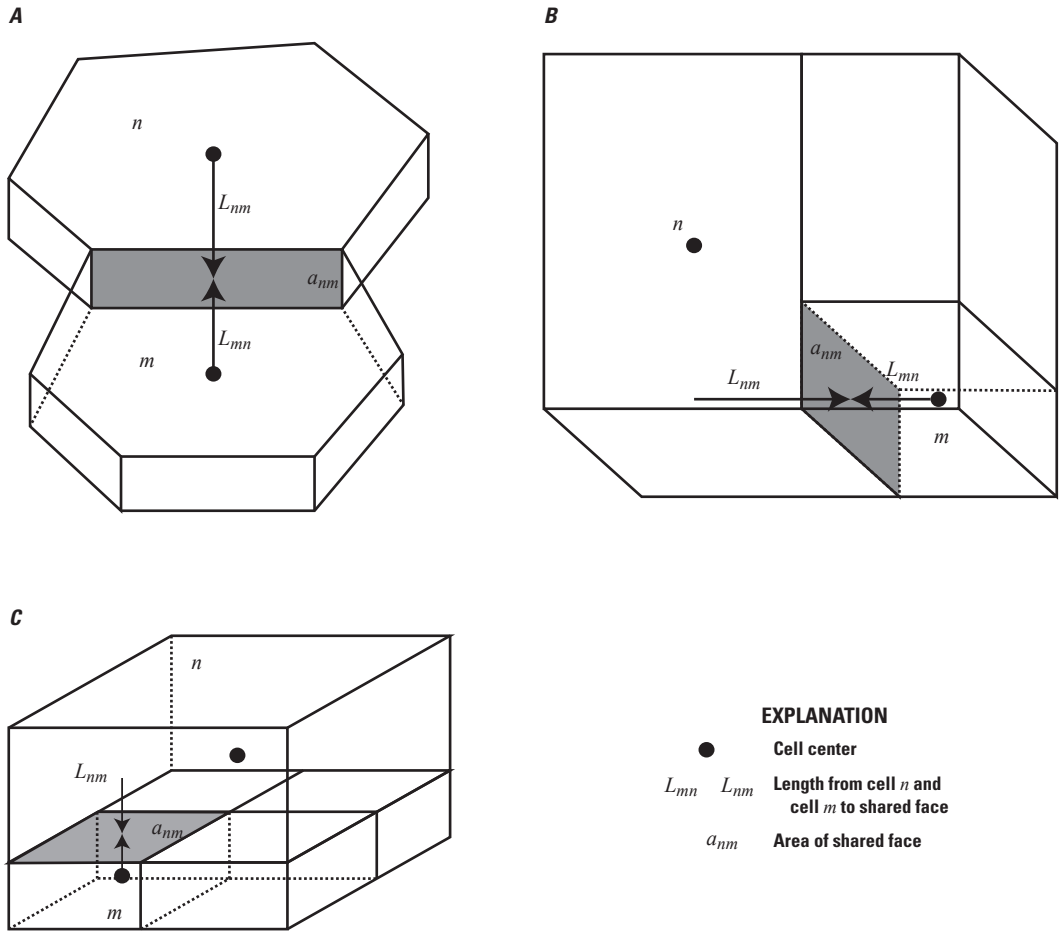

Figure 7. Geometrical properties for *A*, a horizontal connection between two irregular shaped cells, *B*, a horizontal connection for a nested grid, and *C*, a vertical connection. The flow area for the connection is shown in gray.

discussed. As described in the MODFLOW–NWT documentation (Niswonger and others, 2011), the saturated thickness of a cell is calculated as a smooth function of the cell head whereby abrupt transitions from dry to wet and from unconfined to confined have continuous slopes as required by the Newton-Raphson methodology. MODFLOW–USG uses a similar approach, whereby the saturated fraction of a cell, denoted by $f(h)$, is calculated from the following equation

$$
\begin{aligned}
&f(h) = 0 \quad for \quad S_F < 0 \\
&f(h) = \frac{A_\Omega}{2\Omega} S_F^{\,2} \quad for \quad 0 \le S_F < \Omega \\
&f(h) = A_\Omega S_F + \frac{1}{2}(1 - A_\Omega) \quad for \quad \Omega \le S_F < (1-\Omega), \\
&f(h) = 1 - \frac{A_\Omega}{2\Omega}(1 - S_F)^2 \quad for \quad (1-\Omega) \le S_F < 1 \\
&f(h) = 1 \quad for \quad S_F > 1
\end{aligned}
\tag{12}
$$

where

$$S_F = \frac{h - BOT}{TOP - BOT}, \tag{13}$$

TOP is the top elevation of the cell,
BOT is the bottom elevation of the cell,
Ω is a small distance over which the slope discontinuity is smoothed, and

$$A_\Omega = \frac{1}{1-\Omega}. \tag{14}$$

As described by Niswonger and others (2011), the value for Ω should be set to a small value. In MODFLOW–USG, this value is fixed at 1×10^{-6}.

Horizontal Flow Barriers

The Horizontal Flow Barrier (HFB) Package of MODFLOW–2005 also is incorporated in MODFLOW–USG to allow representation of thin, low-permeability features located between two cells. The formulation assumes that the

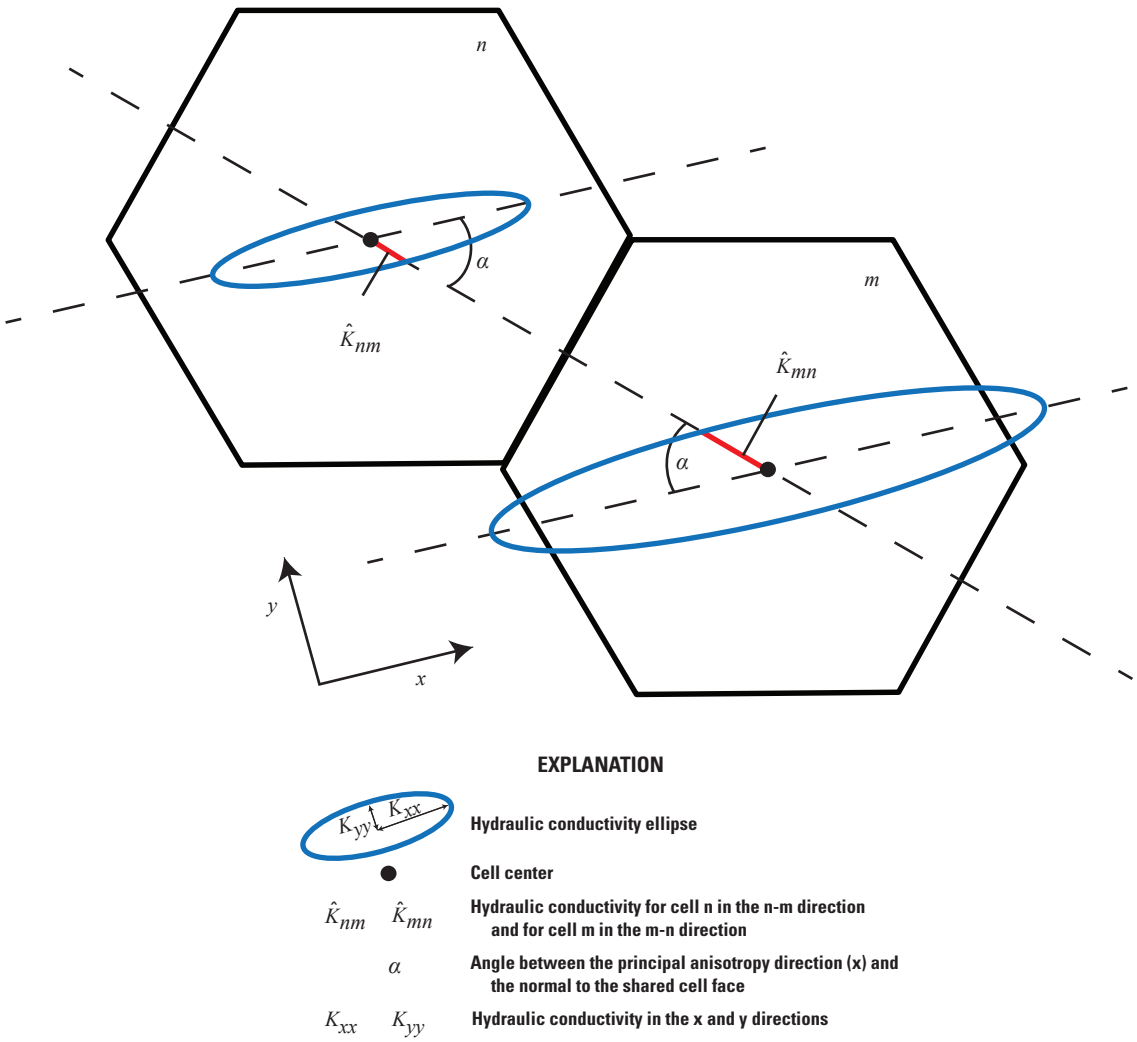

Figure 8. Geometrical characteristics for representation of horizontal anisotropy. In the present formulation, the orientation of the hydraulic conductivity ellipse is defined by the user and is constant throughout a layer.

width of the barrier is negligible compared to the distance between the cell centers. The formulation also assumes that the barrier covers the entire area of the connection between the two cells. The formulation and implementation of the HFB Package is detailed in the MODFLOW–2005 documentation (Harbaugh, 2005) for the MODFLOW–2005 flow averaging options and in Niswonger and others (2011) for the upstream weighting formulation. The current implementation in MODFLOW–USG is the same as these, except that the barrier can be aligned along any face of an unstructured grid.

Anisotropy

For structured grids, MODFLOW–USG approximates the effects of horizontal anisotropy using the same approach as MODFLOW; the principal axes of the grid must be aligned with the principal hydraulic conductivity axes. If the LPF Package input structure is used, a single value for horizontal anisotropy can be specified for each layer using the CHANI input variable; alternatively, a layer can be assigned one value of horizontal anisotropy for each cell using the HANI input variable. Following the MODFLOW–2005 implementation, only a single value can be specified for horizontal anisotropy for each layer when using the BCF Package input.

Two options are available for specifying horizontal anisotropy for unstructured grids. One option is for the user to pre-calculate the saturated inter-cell conductance for every connection in the grid and provide these conductances as input to MODFLOW–USG. Effective conductance values can be generated during pre-processing by computing the effective hydraulic conductivity or conductance (or by using any other inter-cell averaging scheme of the user's choice, which may include horizontal flow barriers), with each cell hydraulic conductivity value computed from geometric considerations as shown in figure 8 to include anisotropic effects. The second option for specifying horizontal anisotropy requires that the user impose x- and y-coordinate axes, which are aligned with

the principal anisotropy directions. With this option, the user must specify the angle in radians, α, between the principal anisotropy direction and the normal to the face, as shown in figure 8, for every connection. MODFLOW–USG uses these angles to calculate a hydraulic conductivity value in the *n-m* direction for cell *n* ($\hat{K}_{nm}$) and for cell *m* ($\hat{K}_{mn}$) as

$$\begin{aligned} \hat{K}_{nm} &= K_{xx,n}\cos\alpha + K_{xx,n} \times CHANI \sin\alpha \\ \hat{K}_{mn} &= K_{xx,m}\cos\alpha + K_{xx,m} \times CHANI \sin\alpha \end{aligned}, \quad (15)$$

where

K_{xx} is the hydraulic conductivity value along the x-direction, and

$CHANI$ is the *x:y* anisotropy factor.

These hydraulic conductivities in the *n-m* direction are then used with the selected averaging scheme (harmonic mean weighting, for example) to calculate the K_{nm} value used in equation 11 for computing inter-cell conductance.

For cases where lines connecting adjacent cell centers do not correspond with principal hydraulic conductivity axes, which can happen with nonrectangular cells such as the hexagons shown in figure 8, MODFLOW–USG will incur errors in the simulated flow system because the cross component terms of anisotropy are neglected. Errors caused by neglecting the cross component terms of anisotropy increase with the angle between the face and the anisotropy axis and are largest at a 45-degree angle. For this reason, the capability to specify horizontal anisotropy should be used with caution and should be used only for slight anisotropy ratios close to 1:1.

Vertical Flow Calculations for Partially Dewatered Cells

The MODFLOW–2005 program has three options (NOCVCORRECTION, CONSTANTCV, and NOVFC) for the LPF Package, in addition to the default option, that affect the vertical flow calculation for cells that are underlain by partially dewatered cells. The differences between the options are subtle, but important, because they can affect the simulated results as well as the stability of the solution. As these options are available in the LPF Package for MODFLOW–USG, they are described here in the context of the unstructured flow formulation. The equations developed in this section are based on figure 9, which shows two partially dewatered cells with cell *m* underlying cell *n*. For the equations presented here, flow is defined as being positive out of cell *n*.

For the default LPF Package conditions in which none of the options are activated, flow between cells *n* and *m* is calculated as

$$Q_{nm} = \frac{a_{nm}}{\frac{1/2(h_n - BOT_n)}{VK_n}}(h_n - TOP_m), \quad (16)$$

where

VK_n is the vertical hydraulic conductivity of cell *n*.

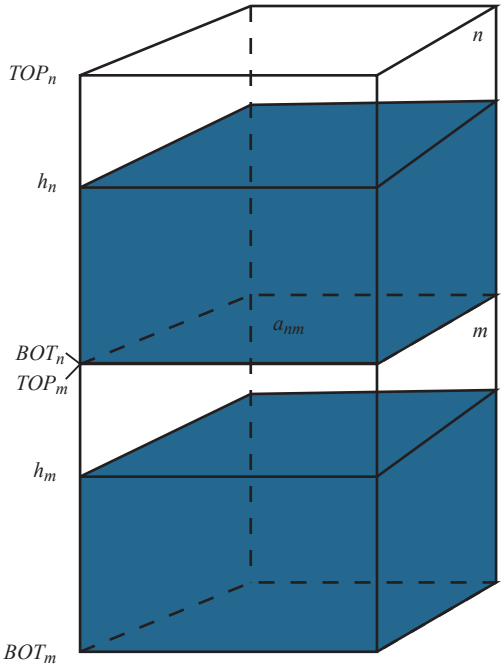

Figure 9. Partially dewatered cells.

MODFLOW–2005 cannot formulate this equation directly, because it would introduce asymmetry into the conductance matrix. The asymmetry is caused by the replacement of h_m in the general flow formulation (equation 8) with TOP_m. To handle this condition, correction terms are added to cells *n* and *m* as described by Harbaugh (2005). Although this condition could be treated implicitly in MODFLOW–USG by formulating an asymmetric coefficient matrix, the correction approach is also used in MODFLOW–USG in order to simplify assembly of the Newton Raphson formulation. This default condition for the LPF Package can be unstable for some simulations as the saturated thickness of cell *n* becomes thin. When this occurs, the vertical conductance can become very large as the denominator in equation 16 approaches zero.

The first LPF Package option is called NOCVCORRECTION. The NOCVCORRECTION option uses the following equation for flow between cells *n* and *m*:

$$Q_{nm} = \frac{a_{nm}}{\frac{1/2(h_n - BOT_n)}{VK_n} + \frac{1/2(TOP_m - BOT_m)}{VK_m}}(h_n - TOP_m). \quad (17)$$

The difference between this option and the default LPF package conditions of equation 16 is that the entire conductance of cell *m* is included in determining the effective conductance between cells. Note that correction terms are required when the NOCVCORRECTION option is used since the flow is not governed by the head gradients between the nodes but by the term (h_n - TOP_m), when cell *m* is partially dewatered. This option is more stable than the default LPF Package condition because the vertical conductance does not change abruptly when cell *m* becomes partially dewatered, and because the vertical conductance does not become very large as the saturated thickness in cell *n* becomes small.

The second LPF Package option, called CONSTANTCV, uses the full thickness of cells *n* and *m* in the vertical conductance calculation. Accordingly, the flow between cells *n* and *m* is

$$Q_{nm} = \frac{a_{nm}}{\frac{\frac{1}{2}(TOP_n - BOT_n)}{VK_n} + \frac{\frac{1}{2}(TOP_m - BOT_m)}{VK_m}}(h_n - TOP_m) \quad ,(18)$$

With the CONSTANTCV option, the vertical conductance is calculated once at the beginning of the simulation and is then held constant throughout the simulation since it does not depend on the head solution. Note that correction terms are also required when the CONSTANTCV option is used since the flow is not governed by the head gradients between the nodes but by the term (h_n - TOP_m), when cell *m* is partially dewatered. The CONSTANTCV option is more stable than the default LPF Package condition and the NOCVCORRECTION option because conductance is not a function of head as that nonlinearity is removed entirely. The CONSTANTCV option for the LPF Package is the formulation that is implemented in the BCF Package.

The last option governing vertical flow to partially dewatered cells in the LPF Package is NOVFC, which stands for "no vertical flow correction". With this option, the vertical conductance is calculated using the NOCVCORRECTION approach, but the vertical flow correction is not applied. This results in the following equation for flow between cells *n* and *m*

$$Q_{nm} = \frac{a_{nm}}{\frac{\frac{1}{2}(h_n - BOT_n)}{VK_n} + \frac{\frac{1}{2}(TOP_m - BOT_m)}{VK_m}}(h_n - h_m) \quad , \quad (19)$$

The NOVFC option is the most stable of these LPF Package options, because no vertical flow correction is applied. NOVFC should be used for most simulations unless there is indication that perched conditions exist and that their accurate representation is important to the simulation. The NOVFC option is always applied for the UPW Package of MODFLOW–NWT. To obtain results with the upstream weighting option of MODFLOW–USG that compare with MODFLOW–NWT, the NOVFC and CONSTANTCV options should be specified.

Connected Linear Network (CLN) Process

The CLN Process was developed for MODFLOW–USG to provide the framework for incorporating one-dimensional connected features into a structured or unstructured three-dimensional GWF Process grid. A one-dimensional CLN feature is any hydrogeologic or hydrologic water conveyance feature that has a cross-sectional dimension which is much smaller than the longitudinal flow dimension and the size of the encompassing GWF cell. Flow is computed in the longitudinal direction of the network of connected one-dimensional features using specified cross-sectional properties; flow between CLN cells and GWF cells is computed across the wetted perimeter of the one-dimensional CLN feature. The CLN Process thus provides a mechanism for including features with small cross-sectional areas, relative to GWF cell sizes, without having to build this level of detail into the grid used for the GWF domain. An example problem is included in this report to demonstrate use of the CLN Process.

The CLN Process domain is solved simultaneously with the GWF Process. This means that the total number of cells in a MODFLOW–USG simulation is equal to the combined total of GWF cells and CLN cells. The CLN flow process solves for flow of water within a network of linear features as well as for the interaction of the features with the porous medium. There are then, two types of flow calculations that occur with the addition of the CLN domain—flow within the CLN domain, and flow between CLN and GWF cells. The CLN Process does not inactivate dry cells as is done in MODFLOW–2005 for GWF cells. Instead, a head value is always calculated for active CLN cells. For unconfined cases, however, as the saturated thickness approaches zero, the conductance values with connected cells are reduced and the cell responds as if it were dry.

For flow within the CLN domain, the CLN Process implements solution of one nonlinear equation per CLN cell. The CLN cells may represent wells, pipes, fractures, canals, rivers, streams or other linear features within a simulation domain that need to be represented by flow connections that are separate from those of the aquifer. The formulation is general enough whereby different types of features can flow into each other. For instance, pipes can flow from or to open-channel features. Extension of the code to include other features or geometries is a straightforward exercise in implementing functional forms for cross-sectional areas, and volumes as functions of flow depths at the CLN cells within the appropriate subroutines. The current version of MODFLOW–USG only incorporates the laminar flow formulation for cylindrical conduit geometry types. Other geometry types and flow formulations can be added in the future to incorporate kinematic- or diffusion-wave equations for open channel flow, for example. The current implementation could also be extended to support other pipe or section geometries, tabular input of volume and cross-sectional flow area as a function of head, and turbulent flow conditions. These geometry types and flow conditions can be included by providing functional forms of the longitudinal flow conductance terms

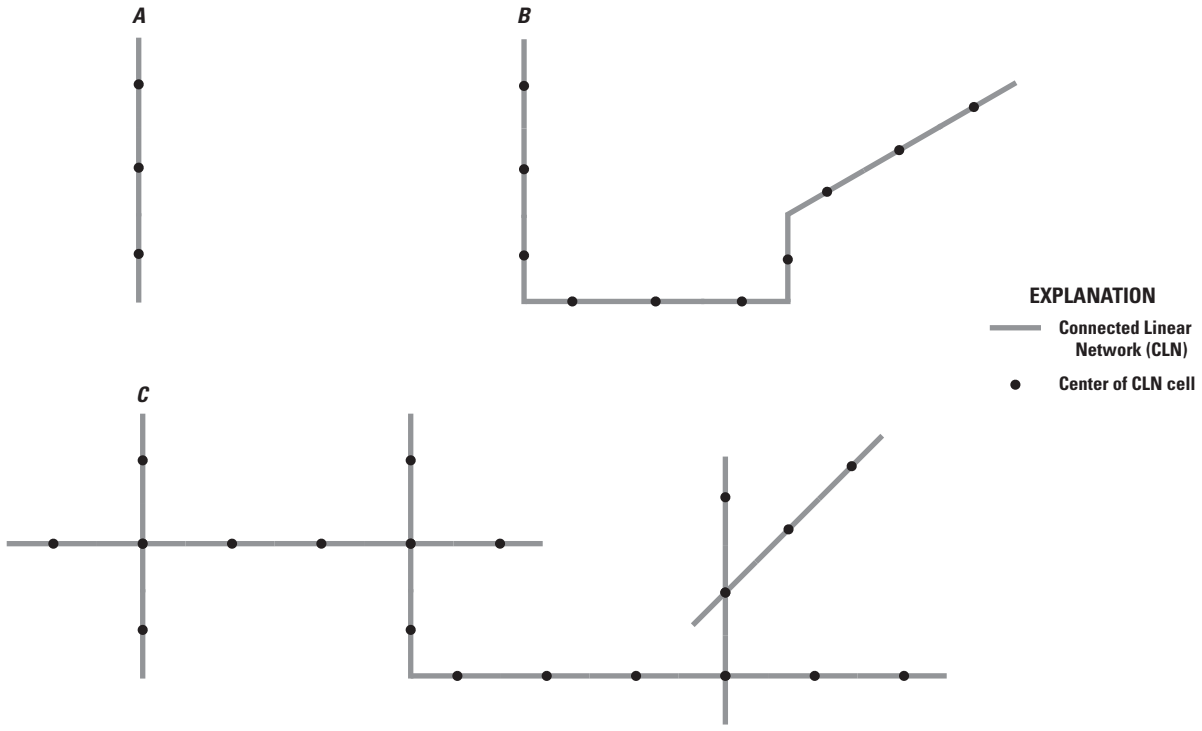

Figure 10. Several different connected linear network geometries: *A*, a single linear feature discretized with three Connected Linear Network (CLN) cells, *B*, a multi-dimensional CLN segment, and *C*, a network of CLN segments.

within the appropriate subroutines. The dynamic wave equation for flow in open channels requires solution to two equations per channel segment (one for continuity and a separate one for momentum) and therefore is not readily accommodated by the present form of the CLN Process.

For flow between CLN cells and connected GWF cells, the CLN Process implements solution of one nonlinear flow equation per CLN-to-GWF connection. Various options are provided to compute the effective leakance between them, including input of the effective leakance value, computation from skin conductance and thickness, or use of a Thiem solution to provide the radial influences and skin resistance considerations for flow between CLN and GWF cells. The formulation is general enough whereby extension of the code to include other connection geometries or special considerations is straightforward as long as functional forms for the respective wetted surface areas with flow depths for the CLN-to-GWF connection can be provided. Furthermore, the CLN-to-GWF transfer equations are generalized to accommodate unstructured grid cells.

The CLN Process uses the concept of a CLN segment, which is one or more CLN cells connected end to end. A CLN segment may be used to represent a well, for example. CLN segments can also be connected to one another to form a network, as shown in figure 10. A CLN network may be useful for representing tile drains, radial collector wells, or other connected linear features. A CLN cell can be vertical, horizontal, or tilted. Cell lengths may be dimensioned such that several CLN cells (from the same or different CLN segments) may be connected to one GWF cell, or one CLN cell may be connected to several GWF cells depending on the scale of conceptualization of flow within the CLN and GWF domains. In addition, there is no need for specific upstream to downstream sequential ordering of cells as may be necessary for other packages or codes.

The CLN Process with the cylindrical conduit geometry type can be used to simulate flow through a linear network of karst conduits similar to what can be done with the Conduit Flow Process (CFP) (Shoemaker and others, 2007) of MODFLOW–2005. Areal flow functionality of CFP (CFP modes 2 and 3) is presently not available in MODFLOW–USG. An advantage of the MODFLOW–USG implementation is that the conduit network is solved simultaneously with the groundwater flow equation, but the present version of MODFLOW–USG supports only laminar flow, whereas CFP can represent turbulent effects within the conduit network.

The CLN Process, when applied with cylindrical conduits and the Thiem solution for CLN-to-GWF connection, provides some of the functionalities of the Multi-Node Well (MNW and MNW2) (Halford and Hanson, 2002; Konikow and others, 2009) Packages of MODFLOW–2005. A single cylindrical CLN cell connected to multiple GWF cells may be pumped to simulate multi-node well conditions, in which the CLN cell extracts water from the GWF cells connected to it as part of the solution to the coupled CLN and GWF flow equations. A CLN cell can be pumped by use of the Well (WEL) Package

by assigning a source or sink to that CLN cell. If flow through a narrow conduit is important to the production of a multi-layer well (for instance, the well-bore resistance governs how much water is produced from each of multiple aquifers) the well may also be represented by a segment consisting of multiple CLN cells. In this case, a CLN cell can be added to each GWF cell or layer, and laminar flow would be simulated within the conduit. This may be especially important when the conduit radius is small and resistance in the well bore affects flow through the conduit. This approach would also be useful for simulating borehole flow within wells that are selectively screened to multiple aquifers.

Formulation of Flow within the CLN Domain

The continuity equation for a CLN cell is a function of flows from connected CLN cells and flows from connected GWF cells and may be written in difference form as

$$\frac{V_n f_{vn}}{\Delta t} = \sum_{m \in \eta_n} \frac{a_{cnm} K_{cnm} f_{unm} (h_n - h_m)}{0.5(l_{cn} + l_{cm})} + \sum_{p \in \eta_n} \Gamma_{cpn} \,, \qquad (20)$$

where

f_{vn} is the fraction of the volume of CLN cell n that is saturated (or the variable part of the volume term which is a function of the head of cell n),

f_{unm} is the fraction of the upstream CLN cell volume that is saturated (= f_{vn}, if n is upstream and = f_{vm} if m is upstream),

a_{cnm} is the cross-sectional area of the connection between the CLN cells (=min (a_{cn} , a_{cm})),

a_{cn} is the cross-sectional area of CLN cell n,

a_{cm} is the cross-sectional area of CLN cell m,

K_{cnm} is the saturated linear conductivity of the connection between the CLN cells n and m,

l_{cn} and l_{cm} are the length of CLN cells n and m, respectively, and

Γ_{cpn} is the volumetric flow from a connected GWF cell p to CLN cell n.

The left-hand side of equation 20 is the storage term of the CLN cell, the first term on the right-hand side is the flow between cell n and connected CLN cells, m, and the second term on the right-hand side is the interaction flow to all connected GWF cells, p. The summation in the first flow term on the right-hand side is over all m CLN cells connected to CLN cell n, whereas the summation in the second term on the right-hand side (the interaction term), is over all p GWF cells connected to CLN cell n.

The saturated linear conductivity of the connection between CLN cells is expressed as the length-weighted harmonic mean of the linear conductivity of each of the connected CLN cells giving:

$$K_{cnm} = (l_{cn} + l_{cm})(K_{cn} K_{cm}) / (K_{cn} l_{cn} + K_{cm} l_{cm}) \,, \qquad (21)$$

where K_{cn} and K_{cm} are the saturated linear conductivity values for CLN cells n and m respectively.

The fraction of the total volume of a CLN cell that is saturated, f_{vn}, depends on whether the 1-dimensional cell is vertical or horizontal. For a vertical CLN cell, the saturated fraction, f_{vn}, is obtained as

$$\begin{aligned} f_{vn} &= 0 \qquad for \quad (h_n - BOT_n) < 0 \\ f_{vn} &= \frac{(h_n - BOT_n)}{l_n} \qquad for \quad 0 < (h_n - BOT_n) < 1 \,, \\ f_{vn} &= 1 \qquad for \quad (h_n - BOT_n) > 1 \end{aligned} \qquad (22)$$

where BOT is the bottom elevation of the cell. If the cell is angled, the assumption is made that $l_n >> r_n$ where r_n is the hydraulic radius of the CLN cell (that is, the length of the CLN segment is much larger than its cross-sectional dimension), and therefore, the term l_n in equation 22 is replaced by $l_n \sin(\theta)$, where θ is the angle that the CLN cell makes with the horizontal. The second part of equation 22 contains a singularity when the angle θ is zero; therefore, if $l_n \sin(\theta)$ is less than twice the hydraulic radius of the CLN cell ($2r_n$), a horizontal conduit is assumed, governed by a different set of equations as shown next. In practice, equation 12 is used to determine f_{vn} in order to smoothen slope discontinuities as required by the Newton-Raphson method.

For a horizontal CLN segment, the saturated fraction, f_{vn}, is obtained as

$$f_{vn} = \frac{a_{wn}}{a_{cn}} \,, \qquad (23)$$

where a_{wn} is the wetted cross-sectional area of CLN cell n, obtained from geometric considerations as a function of the flow depth of the CLN cell.

Formulation of Flow between the CLN and GWF Domains

The interaction flow between CLN and GWF cells, Γ_{cpn}, in equation 20 may be expressed as

$$\Gamma_{cpn} = \alpha_{cpn} f_{upn} (h_p - h_n) \,, \qquad (24)$$

where

α_{cpn} is the saturated conductance between the CLN cell n and GWF cell p,

h_p is the head in the GWF cell, and

f_{upn} is the wetted fraction of the upstream perimeter, which is a function of the flow depth.

The term f_{upn} is the same as the saturated fraction, f_{vn}, for vertical and angled CLN cells. Hence, for vertical and angled CLN cells,

$$f_{upn} = f_{vn}, \tag{25}$$

where f_{vn} is computed from equation 22 using the upstream head of h_p and h_n. For horizontal CLN cells, the wetted fraction is the ratio of wetted perimeters of the CLN cells, obtained as

$$f_{upn} = P_{wu} / P_t, \tag{26}$$

where

P_{wu} is the wetted perimeter computed using the upstream head of h_p and h_n, and

P_t is the total perimeter of the CLN cell.

Several options are provided for computing the conductance term, α_{cpn}, between the CLN cell and the GWF cell. The first option uses a CFP formulation option for water exchange between the GWF and CLN cells in which the conductance values, α_{cpn}, are input to the model. The second option also uses a CFP formulation option in which skin thickness and skin hydraulic conductivity are input to the model and the conductance is computed using CLN geometry as

$$\alpha_{cpn} = \frac{K_s P_t l}{b_s}, \tag{27}$$

where

K_s is the hydraulic conductivity of the skin surrounding the CLN-to-GWF interface, and

b_s is the thickness of the skin surrounding the CLN-to-GWF interface.

The third option for computing the conductance term α_{cpn} follows Bennett and others (1982) and provides for head loss between the GWF cell and the CLN cell by use of the Thiem equation to compute the flow, as adapted in the MNW Package (Halford and Hanson, 2002; Konikow and others, 2009). For this case, the conductance is approximated for a vertical CLN cell connection with a GWF cell as

$$\alpha_{cpn} = \left[\frac{\ln\left(r_{oz} / r_n\right) + S_f}{2\pi l K_{xx}\sqrt{1/\Re_{rz}}}\right]^{-1} + \left[C\Gamma_{cpn}{}^{P-1}\right]^{-1}, \tag{28}$$

where

r_{oz} (representing the radius of influence in the Thiem equation) is the effective external radius of a GWF cell for a connected vertical CLN cell,

r_n is the effective radius of CLN cell n,

S_f is the skin factor,

$\Re_{rz}$ is the x:y anisotropy ratio (K_{xx} / K_{yy}),

C is the nonlinear well-loss coefficient; and

P is the exponent for the nonlinear well loss term.

For a horizontal CLN cell connection, the conductance is approximated as

$$\alpha_{cpn} = \left[\frac{\ln\left(r_{oh} / r_n\right) + S_f}{2\pi l K_{xx}\sqrt{1/\Re_{rh}}}\right]^{-1} + \left[C\Gamma_{cpn}{}^{P-1}\right]^{-1}, \tag{29}$$

where

r_{oh} is the effective external radius of a GWF cell for a connected horizontal CLN cell, and

$\Re_{rh}$ is the horizontal to vertical anisotropy ratio (K_{xx} / K_{zz}).

The last term in equations 28 and 29 is nonlinear and may cause numerical problems; moreover, this term is sometimes unnecessary. As a result, the term has been eliminated from the current CLN Process (that is, $C = 0$ in equations 28 and 29). The vertical conduit assumption is used for a slanted conduit, thereby incorporating the horizontal radius of influence and anisotropy approximations of equation 28.

The effective external radius of the GWF cell with a vertical CLN cell connection is computed as

$$r_{oz} = 0.28\left[\frac{2\sum_{m\in\eta_{nh}}(L_{nm})^2}{N_m}\right]^{1/2}, \tag{30}$$

where η_{nh} is the set of horizontally connected GWF cells, N_m is the number of adjacent horizontal connections to GWF cell, n. Equation 30 reduces to the isotropic equation for the effective external radius for a rectangular coordinate system of a finite-difference connectivity, as defined by Peaceman (1978). The horizontal isotropic approximation of equation 30 is considered adequate for unstructured grids, as opposed to evaluating components along the horizontal principal coordinate directions for horizontally anisotropic conditions.

The effective external radius of a GWF cell with a horizontal CLN cell connection is computed as

$$r_{oh} = 0.28\left[\frac{\sqrt{\Delta H^2\sqrt{1/\Re_{rh}} + \Delta Z^2\sqrt{\Re_{rh}}}}{\sqrt[4]{1/\Re_{rh}} + \sqrt[4]{\Re_{rh}}}\right], \tag{31}$$

where

ΔZ is the thickness of the GWF cell, and

ΔH is the horizontal cell dimension normal to the line of the CLN cell, approximated as

$$\Delta H = \left[\frac{2\sum_{N_m\in\eta_n}(L_{nm})}{N_m}\right]. \tag{32}$$

Equations 31 and 32 reduce to the anisotropic equation for the effective external radius, for a rectangular coordinate system of a finite-difference connectivity, as defined by Peaceman (1983). The approximations of equations 31 and 32 are deemed adequate, as opposed to evaluating components along the horizontal coordinate directions normal to the line of the conduit cell, because the basic equations of Peaceman (1983) are themselves approximations for horizontal conduits. Because suitable equations for estimating cell-to-well conductance of horizontal wells are not well defined, it is not warranted to make the connection to horizontal conduits more complex. Furthermore, because of incomplete knowledge of conduit geometry or of the skin factor or well efficiency, and regardless of the option selected to assemble α_{cpn}, it is likely that an estimated value of this parameter will be modified during model calibration. Equations 28 through 32 are generalizations of the equations for rectangular finite-difference cells presented by Halford and Hanson (2002) and Konikow and others (2009).

Confined Option for CLN Flow

An option for confined flow between CLN cells is also provided and is similar to the confined option available for each layer of the GWF Process grid. Each CLN cell can be designated as confined or convertible. With this option, the term f_{umm} in equation 20 is held constant at a value of 1, allowing for flow from an upstream cell to occur under saturated conditions, regardless of the saturation condition of the CLN cell. The option can be useful for different cases of flow in horizontal or vertical conduits. This option applies only to CLN–CLN flow and not to flow between CLN and GWF domains.

Flow to Convertible Cells

Special considerations are required for flow to convertible cells that can dry and rewet. When flow is between two cells that are wet, the head difference between them creates the driving force for flow. When the downstream cell is dry, however, a reference elevation can be used instead of the downstream head to express flow between the nodes. This "flow-to-dry-cell" option is the default condition for the BCF and LPF Packages of MODFLOW–2005 for downward flow to a partially saturated cell.

The "flow-to-dry-cell" option is also available for flow between two CLN cells or for flow between a CLN cell and a GWF cell if the cells are convertible. For this case, the respective flow can be expressed as

$$\begin{aligned} Q_{nm} &= C^0_{nm} f(h_{ups})(h_m - h_n) \quad for \quad h_n > e_n \\ Q_{nm} &= C^0_{nm} f(h_{ups})(h_m - e_n) \quad for \quad h_n < e_n \end{aligned}, \tag{33}$$

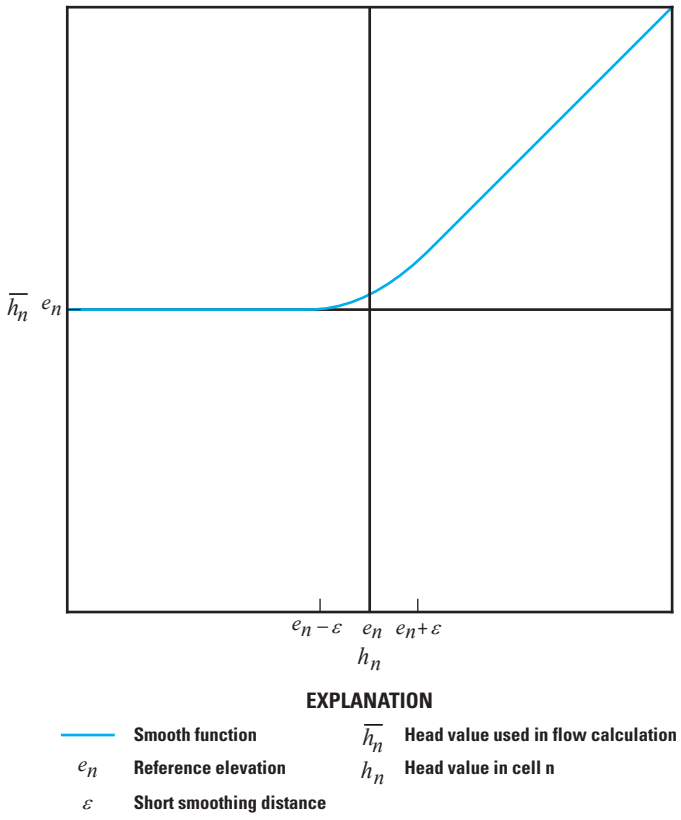

Figure 11. Smooth function used to express the head value used in the flow calculation to the head value of the downstream cell as it transitions from saturated to partially saturated conditions.

where

C^0_{nm} is the constant portion (fully saturated) of the conductance term between cells *n* and *m*,

$f(h_{ups})$ is the upstream-weighted nonlinear term defined in equation 12, and

e_n is a reference elevation for the downstream node, *n*.

For the remainder of this section, the upstream node is assumed to be *m* and the downstream node is assumed to be *n*. In the GWF Process, when downward flow is from cell *m* to a partially saturated cell *n*, the top elevation of cell *n* is the reference elevation, e_n. Equation 33 has a slope discontinuity at $h_n = e_n$. For a smooth transition, which is required by the Newton-Raphson method, equation 33 can be recast as

$$Q_{nm} = C^0_{nm} f(h_{ups})(h_m - \overline{h_n}), \tag{34}$$

where $\overline{h_n}$ is a function that transitions between head at the downstream node, h_n, and the reference elevation, e_n in equation 34, in a smooth manner as shown on figure 11. The function is given as:

$$\overline{h_n} = h_n \quad for \quad (h_n - e_n) > \varepsilon,$$
$$\overline{h_n} = \frac{(h_n - e_n)^2}{4\varepsilon} + \frac{(h_n - e_n)}{2} + \frac{\varepsilon}{4} + e_n \quad for \quad \varepsilon > (h_n - e_n) > -\varepsilon, \qquad (35)$$
$$\overline{h_n} = e_n \quad for \quad (h_n - e_n) < -\varepsilon,$$

where ε is the distance over which the function is smoothed. Equation 35 satisfies the continuity of the function and of the derivatives at $(h_n - e_n) = \pm\varepsilon$. Thus,

$$\overline{h_n} = e_n \quad for \quad (h_n - e_n) = -\varepsilon, \qquad (36)$$

$$\frac{d\overline{h_n}}{dh} = 0 \quad for \quad (h_n - e_n) = -\varepsilon, \qquad (37)$$

and

$$\frac{d\overline{h_n}}{dh} = 1 \quad for \quad (h_n - e_n) = \varepsilon. \qquad (38)$$

Note that ε is a small number (10^{-4} in MODFLOW–USG).

Just as was done for vertical flow correction in the GWF Process of MODFLOW–2005, flow to a convertible cell is expressed using Picard linearization in MODFLOW–USG as a correction on the right-hand-side vector of the matrix equations, with assembly of the conductivity term into the left-hand-side coefficient matrix in a symmetric manner, as is done for the traditional Darcy flow equation. Thus, equation 34 can be expanded as

$$Q_{nm} = C^0_{nm} f(h_{ups})(h_m - h_n) + C^0_{nm} f(h_{ups})(h_n - \overline{h_n}), \qquad (39)$$

where the first term on the right-hand-side of equation 39 is assembled into the coefficient matrix and the second term is assembled into the right-hand-side vector.

For flow between two CLN cells, the reference elevation, e_n, is taken as the bottom elevation of the upstream CLN cell; for flow between a CLN cell and a GWF cell, the reference elevation, e_n, is taken as the higher of the two bottom elevations:

$$e_n = \begin{cases} BOT_{CLN} \; for \quad BOT_{CLN} > BOT_{GWF} \\ BOT_{GWF} \; for \quad BOT_{GWF} > BOT_{CLN} \end{cases}, \qquad (40)$$

where

BOT_{CLN} is the bottom elevation of the CLN cell, and
BOT_{GWF} is the bottom elevation of the GWF cell.

Cylindrical Conduit Geometry Type

The CLN Process is designed in a modular fashion to support different geometry types. In the present MODFLOW–USG version, the CLN Process supports laminar flow in cylindrical conduits. Other one-dimensional features could also be represented by the process if functional forms can be provided for the following:

- The cross-sectional area, perimeter, hydraulic radius, and saturated hydraulic conductivity, as a function of its input geometric characteristics, and
- The saturated cross-sectional area and wetted perimeter as a function of the potentiometric head. This latter condition is only required for nonlinear (unconfined) conditions and is not required for confined (linear) conditions.

For a cylindrical conduit represented as a CLN cell, the input required is the radius of the cylindrical conduit, R_{cond}. The cross-sectional area, a_c, is then computed for a cylindrical conduit as

$$a_c = \pi R^2_{cond}. \qquad (41)$$

The total perimeter, P_t, for a cylindrical conduit is the circumference, given as

$$P_t = 2\pi R_{cond}. \qquad (42)$$

The effective radius, r_n, of a cylindrical conduit is the same as its radius and, therefore,

$$r_n = R_{cond}. \qquad (43)$$

The saturated hydraulic conductivity for flow in a cylindrical conduit may be obtained by assuming laminar flow. Following Sudicky and others (1995), the term K_c in equation 21 can be expressed for laminar flow by the Hagen-Poiseuille equation as

$$K_c = \frac{\rho g}{8\mu} R^2_{cond}, \qquad (44)$$

where

ρ is the density of water,
g is the gravitational constant, and
μ is the dynamic viscosity of water.

In addition, the wetted cross-sectional area of a cylindrical conduit is required for computing the saturated fraction of equation 23 for a horizontal unconfined CLN cell. This area is computed for a given potentiometric head as (Panday and Huyakorn, 2004):

$$
\begin{aligned}
a_w &= 0 \qquad\qquad for \quad d < 0 \\
a_w &= R_{cond}^2 \cos^{-1}\left(\frac{R_{cond}-d}{R_{cond}}\right) - \\
&\quad (R_{cond}-d)\sqrt{R_{cond}^2-(R_{cond}-d)^2} \qquad for \quad 0 < d < R_{cond} \\
a_w &= R_{cond}^2\left[\pi - \cos^{-1}\left(\frac{d-R_{cond}}{R_{cond}}\right)\right] + \\
&\quad (d-R_{cond})\sqrt{R_{cond}^2-(R_{cond}-d)^2} \qquad for \quad R_{cond} < d < 2R_{cond}
\end{aligned}
\tag{45}
$$

where d is the depth of flow in the horizontal conduit, obtained as

$$
d = h - BOT\ . \tag{46}
$$

Finally, the wetted perimeter is required to compute the surface area for interaction between the CLN and GWF domains, as noted in equation 26. The wetted perimeter for a circular conduit section is obtained as

$$
\begin{aligned}
P_w &= 0 \qquad\qquad for \quad d < 0 \\
P_w &= 2R_{cond}\cos^{-1}\left(\frac{R_{cond}-d}{R_{cond}}\right) \qquad for \quad 0 < d < R_{cond} \\
P_w &= 2R_{cond}\left[\pi - \cos^{-1}\left(\frac{d-R_{cond}}{R_{cond}}\right)\right] \qquad for \quad R_{cond} < d < 2R_{cond} \\
P_w &= 2\pi \quad R_{cond} \qquad\qquad for \quad d > 2R_{cond}
\end{aligned}
\tag{47}
$$

Similar equations can also be written for different section types.

For fully saturated flow in a CLN cell, the saturated fraction $f_v = 1$. For this case, equation 20 expresses linear steady-state flow through the CLN domain and equations 22, 23, 45, and 46 are not required. Furthermore, for the saturated case, equation 26 is unity and equation 47 is not required.

Newton-Raphson Linearization of the Upstream Weighting Formulation

The equations generated by the upstream weighting formulation of Niswonger and others (2011) provide a monotonic, continuous function to resolve the nonlinearities of unconfined flow in an aquifer associated with drying and rewetting of cells. The flow terms of the CLN equations (specifically, the second term in equation 20 for flow along the conduits and equation 24 for the matrix-conduit interaction) are similarly expressed by an upstream weighting formulation for monotonic and continuous wetting and drying of the conduit cells. Newton-Raphson linearization of the upstream-formulated terms may enhance convergence of the nonlinearities associated with unconfined water table fluctuations and with wetting and drying of cells. A detailed derivation of the Newton-Raphson terms is provided for the finite-difference formulation by Niswonger and others (2011). The following derivation generalizes the finite-difference expansion for unstructured connections.

The Newton-Raphson method can be applied to a system of non-linear equations, like those resulting from discretization of the unconfined groundwater flow equation. For a system of equations, the following matrix equation expresses the Newton-Raphson method:

$$
J_R^{k-1}\left(h^k - h^{k-1}\right) = -R^{k-1}\ , \tag{48}
$$

where

J_R^{k-1} is the Jacobian matrix,

h^k is a vector of the unknown heads being estimated for the current k Newton-Raphson iteration,

h^{k-1} is a vector of the heads from the previous Newton-Raphson iteration, and

R^{k-1} is the residual vector, which contains the mass-balance error, or flow residual, for each cell.

The k-1 superscript indicates that J_R and R are evaluated relative to h at the k-1 iteration. This matrix equation is of the form **Ax=b**, and can therefore be solved using the linear solvers described herein. With the matrix and vectors expanded, this equation is written as shown below

$$
\begin{pmatrix}
\frac{\partial R_1}{\partial h_1} & \frac{\partial R_1}{\partial h_2} & \cdots & \frac{\partial R_1}{\partial h_{NODES}} \\
\frac{\partial R_2}{\partial h_1} & \frac{\partial R_2}{\partial h_2} & \cdots & \frac{\partial R_2}{\partial h_{NODES}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial R_{NODES}}{\partial h_1} & \frac{\partial R_{NODES}}{\partial h_2} & \cdots & \frac{\partial R_{NODES}}{\partial h_{NODES}}
\end{pmatrix}
\begin{pmatrix}
h_1^k - h_1^{k-1} \\
h_2^k - h_2^{k-1} \\
\vdots \\
h_{NODES}^k - h_{NODES}^{k-1}
\end{pmatrix}
=
\begin{pmatrix}
R_1 \\ R_2 \\ \vdots \\ R_{NODES}
\end{pmatrix}
\tag{49}
$$

where $R_1, R_2, R_3 \ldots R_{NODES}$ are the residuals for flow to each node. If this form were used in MODFLOW–USG, the linear solver would produce an estimate for $(h^k - h^{k-1})$, which is an upgrade vector, instead of directly solving for h^k. To maintain consistency with MODFLOW and MODFLOW–NWT, equation 48 is recast as follows so that h^k is solved directly:

$$J_R^{k-1} h^k = -R^{k-1} + J_R^{k-1} h^{k-1} . \quad (50)$$

This equation is also of the form, Ax=b, because all of the terms on the right side are known. The balance equation for cell n can be written for MODFLOW–USG as

$$HCOF_n(h_n) - RHS_n + \sum_{m \in \eta_n} Q_{nm} = R_n . \quad (51)$$

where R_n is the residual of the balance equation for cell n. Applying the Newton formulation (equation 50) to the balance equation 51 yields the following equation for cell n:

$$\frac{\partial R_n}{\partial h_n} h_n^k + \sum_{m \in \eta_n} \frac{\partial R_n}{\partial h_m} h_m^k = -R_n + \frac{\partial R_n}{\partial h_n} h_n^{k-1} + \sum_{m \in \eta_n} \frac{\partial R_n}{\partial h_m} h_m^{k-1} . \quad (52)$$

The derivative terms in equation 52 can be obtained by differentiating the balance equation 51 to yield

$$\frac{\partial R_n}{\partial h_n} = HCOF_n - \frac{\partial (RHS_n)}{\partial h_n} + \sum_{m \in \eta_n} \frac{\partial Q_{nm}}{\partial h_n} , \quad (53)$$

and

$$\frac{\partial R_n}{\partial h_m} = \frac{\partial Q_{nm}}{\partial h_m} . \quad (54)$$

The flow term Q_{nm} between cells n and neighbor m may be expressed in general form for the upstream formulation as

$$Q_{nm} = C_{nm}^0 f(h_{ups})(h_m - h_n) , \quad (55)$$

where $f(h_{ups})$ is the upstream-weighted nonlinear term defined in equation 12.

The derivatives with respect to h_n and h_m are obtained from equation 55 as

$$\frac{\partial Q_{nm}}{\partial h_n} = -C_{nm}^0 f(h_{ups}) + C_{nm}^0 \frac{\partial f(h_{ups})}{\partial h_n}(h_m - h_n) , \quad (56)$$

and

$$\frac{\partial Q_{nm}}{\partial h_m} = C_{nm}^0 f(h_{ups}) + C_{nm}^0 \frac{\partial f(h_{ups})}{\partial h_m}(h_m - h_n) . \quad (57)$$

Note for the upstream weighting formulation that

$$\frac{\partial f(h_{ups})}{\partial h_n} = \begin{cases} \frac{\partial f(h_n)}{\partial h_n} & \text{if } n \text{ is upstream} \\ 0 & \text{if } m \text{ is upstream,} \end{cases} \quad (58)$$

and

$$\frac{\partial f(h_{ups})}{\partial h_m} = \begin{cases} \frac{\partial f(h_m)}{\partial h_m} & \text{if } m \text{ is upstream} \\ 0 & \text{if } n \text{ is upstream .} \end{cases} \quad (59)$$

When equations 53 through 59 are substituted into 52 and R is replaced using equation 51, the balance equation with Newton-Raphson terms becomes equation 60, shown below. As per equations 58 and 59, the first term within the summation on the right side of equation 60 is zero when m is upstream, and the second term is zero when n is upstream. When the Newton-Raphson terms are excluded, this equation reduces to the standard balance equation. Thus, if the flow term is of the form of equation 55, the Newton-Raphson procedure can be implemented within the MODFLOW–2005 solution framework by adding the last terms of equations 56 and 57 to the left-hand-side coefficient matrix generated by Picard discretization of the flow equation—the first term on the right-hand side of equations 56 and 57 being the flow conductance term that is already assembled by the Picard procedure of the MODFLOW–2005 formulate routines. In addition, the summation term on the right side of equation 60 is added to the right-hand-side vector to complete the Newton procedure of linearizing a flow term expressed in the form of equation 55.

$$\begin{aligned} &\left(HCOF_n - \frac{\partial (RHS_n)}{\partial h_n} + \sum_{m \in \eta_n} \left[-C_{nm}^0 f(h_{ups}) + C_{nm}^0 \frac{\partial f(h_{ups})}{\partial h_n}(h_m^{k-1} - h_n^{k-1}) \right] \right) h_n^k + \\ &\sum_{m \in \eta_n} \left[\left(C_{nm}^0 f(h_{ups}) + C_{nm}^0 \frac{\partial f(h_{ups})}{\partial h_m}(h_m^{k-1} - h_n^{k-1}) \right) h_m^k \right] = RHS_n - \frac{\partial (RHS_n)}{\partial h_n} h_n^{k-1} + \\ &\sum_{m \in \eta_n} \left[\left(C_{nm}^0 \frac{\partial f(h_{ups})}{\partial h_n}(h_m^{k-1} - h_n^{k-1}) \right) h_n^{k-1} + \left(C_{nm}^0 \frac{\partial f(h_{ups})}{\partial h_m}(h_m^{k-1} - h_n^{k-1}) \right) h_m^{k-1} \right] \end{aligned} \quad (60)$$

The Newton-Raphson expansion of equation 34 for flow to a convertible cell can also be formulated in this manner. The derivative terms for this case are obtained as

$$\frac{\partial Q_{nm}}{\partial h_n} = -C^0_{nm} f(h_m) \frac{\partial \overline{h}_n}{\partial h_n} \tag{61}$$

and

$$\frac{\partial Q_{nm}}{\partial h_m} = C^0_{nm} f(h_m) + C^0_{nm} \frac{\partial f(h_m)}{\partial h_m}(h_m - \overline{h}_n) , \tag{62}$$

where n is the downstream node. Similar equations can be obtained for m being a downstream node. Because the $C^0_{nm} f(h_{ups})$ term is already appropriately entered into the solution matrix for Picard linearization of the equations, the Newton-Raphson expansion further requires that the pivot term of row n and off-diagonal term of row m be multiplied by $\frac{\partial \overline{h}_n}{\partial h_n}$ and that the second term of equation 62 be added to the pivot term of row m and off-diagonal term of row n to form the Jacobian matrix. As per equation 60, the right-hand-side vector is expanded as

$$RHS = -C^0_{nm} f(h_m)\left(\frac{\partial \overline{h}_n}{\partial h_n} h_n - \overline{h}_n\right) + C^0_{nm} \frac{\partial f(h_m)}{\partial h_m}(h_m - \overline{h}_n) h_m \tag{63}$$

Considering that the second term of equation 39 is already entered into the right-hand-side vector during Picard considerations, application of the Newton-Raphson update necessitates removing h_n and adding $\frac{\partial \overline{h}_n}{\partial h_n} h_n$ along with the upstream derivative terms to give the update to the right-hand-side vector for this "flow-to-dry-cell" option as

$$RHS_{update} = -C^0_{nm} f(h_m)\left(\frac{\partial \overline{h}_n}{\partial h_n} h_n - h_n\right) + C^0_{nm} \frac{\partial f(h_m)}{\partial h_m}(h_m - \overline{h}_n) h_m \tag{64}$$

Ghost Node Correction (GNC) Package

The CVFD formulation is a second-order approximation when the line connecting two cells is perpendicular to, and coincides with, the midpoint of the shared face (Dehotin and others, 2011). This condition is satisfied for a simple grid composed of combinations of equilateral triangles, rectangles, and other regular higher-order polygons (regular pentagons, regular hexagons, and so forth) with coinciding face lengths between two cells. However, the CVFD formulation is a lower order approximation when a line between two connected nodes does not bisect the shared face at a right angle. The consequence is an error in the simulated heads and flows (Edwards, 1996). This is the case, for example, for rectangular nested grids in which the face length of a parent grid is divided among the various nested child grids. Use of a ghost node, as shown in figure 12, can correct for this flux error and maintain local mass conservation. The term "ghost node" was introduced by Dickenson and others (2007) to indicate the fictitious node at a location at which the variable of interest (in this case, groundwater head) should be evaluated, and is used for computation of flow between parent and child grids.

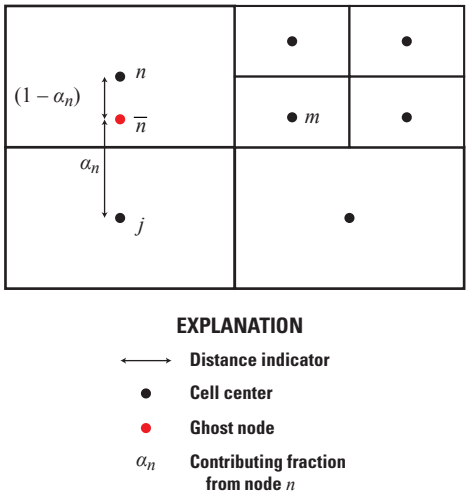

Figure 12. Ghost node conceptualization for nested grids.

The GNC Package is an optional addition that provides higher-order correction terms to a MODFLOW–USG simulation; therefore, the formulation provides an adjustment to the already assembled coefficients of the CVFD equations to include the GNC adjustment. Ghost nodes are not directly added to the system of equations, and consequently, head values are never explicitly calculated at the ghost node locations. Instead, the concept of a ghost node is implicitly built into the system of equations through interpolation factors with surrounding cells. The GNC Package is compatible with the GWF and CLN Processes and can also be used to correct fluxes between GWF and CLN domains.

Correction Formulation

Using the nested rectangular grids of figure 12 as an example, the higher-order representation of flow between cells n and m in equation 8 is obtained using a modified flow term as

$$Q_{nm} = C_{nm}(h_{m,\bar{n}})(h_m - h_{\bar{n}}) , \quad (65)$$

where

$C_{nm}(h_{m,\bar{n}})$ is the conductance between cells *n* and *m* (which can be nonlinear for unconfined cases), and

$h_{\bar{n}}$ is the head at the ghost node location, $\bar{n}$.

The value of $h_{\bar{n}}$ is obtained for the case of two-dimensional rectangular nested grids of figure 12 as a linear combination of the heads at cells *n* and *j* as

$$h_{\bar{n}} = \alpha_n h_n + (1-\alpha_n)h_j , \quad (66)$$

where the coefficient α_n is the contributing fraction from cell *n*, and $1-\alpha_n$ is the contributing fraction from cell *j*, towards the head at node $\bar{n}$. The contributing fractions may be computed by use of linear interpolation, or by use of a Darcy-weighted interpolation as detailed by Mehl and Hill (2005). Other methods of evaluating the correction term have been presented by Dehotin and others (2011) and Edwards (1996).

In multiple dimensions, equation 66 may be expanded to provide the head at the ghost node location, $\bar{n}$, as a linear combination of the head values at cell *n* and of all the adjacent contributing cells j as

$$h_{\bar{n}} = \alpha_n h_n + \sum_{j\in\eta_n} \alpha_j h_j , \quad (67)$$

where

α_n is the contributing fraction of cell *n*,

$\sum_{j\in\eta_n}$ is the summation over all *j* contributing cells adjacent to cell n , which are also contributing to the interpolated head value at the ghost node location, and

α_j is the contributing fraction of each additional contributing cell j .

The sum of all the contributing fractions is unity. Moreover, a cell cannot contribute negatively, and thus, all $\alpha_j \geq 0$. Therefore,

$$\sum_{j\in\eta_n} \alpha_j = 1 - \alpha_n . \quad (68)$$

Substituting equation 67 into equation 65 gives the GNC-corrected flux term as

$$Q_{nm} = C_{nm}(h_m - \alpha_n h_n - \sum_{j\in\eta_n} \alpha_j h_j) . \quad (69)$$

When expressed implicitly, equation 69 provides extra matrix connections to the *j* adjacent contributing cells, for flow between cells *n* and *m*, as the coefficient matrix will include nonzero terms in columns *j* of rows *n* and *m* resulting from the correction. Because the original finite-difference flow term (equation 2) is assembled into the matrix by the CVFD flow assembly routines, the adjustment that needs to be applied for flow between cells n and m subtracts equation 2 from equation 69 to give the correction term as

$$\Delta Q_{nm}^{GNC} = C_{nm}\left[(1-\alpha_n)h_n - \sum_{j\in\eta_n}\left(\alpha_j h_j\right)\right] = C_{nm}\left[\left(\sum_{j\in\eta_n}\alpha_j\right)h_n - \sum_{j\in\eta_n}\left(\alpha_j h_j\right)\right] . \quad (70)$$

where ΔQ_{nm}^{GNC} is the correction term that adjusts the regular CVFD assembled equations to account for the head adjustment of the ghost location, and the second equality results from equation 68. All of the GNC equations just discussed reduce to the original CVFD formulation when $\alpha_n = 1$ and all the $\alpha_j = 0$.

Conductance Options for Unconfined Flow with Ghost Nodes

The conductance between cells *n* and *m* does not vary with water level for confined cases. For unconfined cases, however, the conductance is a function of the water level at cells *n* and (or) *m*. Two options are provided in MODFLOW–USG to compute this nonlinear conductance term. For the first option, $C_{nm}(h_{m,\bar{n}})$ can be expressed only as a function of heads at cells *n* and (or) *m* (that is, the adjustment to the ghost node location for conductance computation is ignored). For the second option, the contributions to $\bar{n}$ from h_n , as well as all connecting h_j cells, is included in the conductance computation. Option 1 is invoked when the input variable I2Kn is specified as 0, whereas option 2 is invoked when I2Kn is specified as 1. These options are included in the GNC Package because limited testing has shown that the more accurate and computationally demanding option (Option 2) is not required for many problems.

The fully implicit CVFD formulation implements $C_{nm}(h_{m,n})$ in the off-diagonal terms and $-C_{nm}(h_{m,n})$ in the diagonal terms of the left-hand-side matrix for rows *n* and *m*, for the flow term of equation 55. With use of option 2 (flag I2Kn=1) to express the ghost node unconfined conductance, the term $\pm C_{nm}(h_{m,n})$ needs to be replaced by the term $\pm C_{nm}(h_{m,\bar{n}})$ instead. This is done using the head values at cell *n* and contributing cells *j* from the previous iteration to compute the conductance. For option 1 (flag I2Kn = 0), this replacement is not required.

Options for Including GNC Terms in Matrix Equations

The GNC term may be applied to the left-hand side matrix fully implicitly, or in an iterative manner on the right-hand-side vector. For a fully implicit implementation, the coefficient matrix is adjusted for rows n and m as follows.

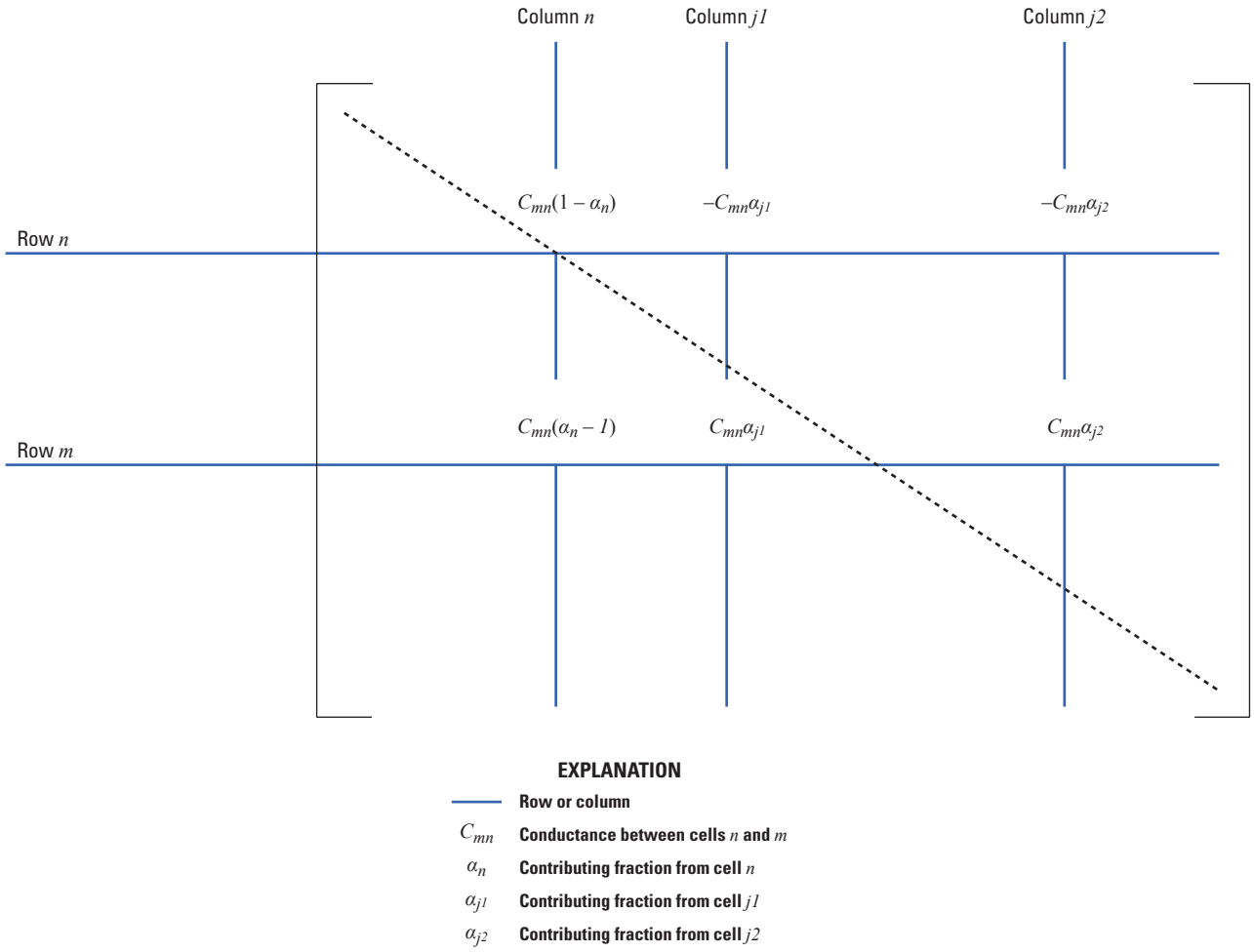

Figure 13. Implicit Ghost Node Correction (GNC) adjustment to finite-difference coefficient matrix.

For row n, the term $C_{nm}[1-\alpha_n]$ is added to column n of the matrix, and the terms $-C_{nm}\alpha_j$ are added to each column j of the matrix within row n. The term in column n of the matrix, $C_{nm}[1-\alpha_n]$, is the negative of the sum of the terms in columns j of the matrix, $-C_{nm}\alpha_j$, because of equation 68; therefore, the components of column n may be added directly when filling the terms in column j using the right-most equality of equation 70. In addition, because $Q_{nm}=-Q_{mn}$, what is added to row n is subtracted from row m for the respective columns of the matrix. This implicit implementation of the GNC terms is demonstrated in figure 13. An implicit GNC implementation makes the matrix asymmetric, because columns of the contributing cells j are filled in rows m and n but nothing is filled in columns m and n of rows j during application of the GNC term. Furthermore, the implicit GNC terms may make the matrix less diagonally dominant. In addition, if the selected j contributing cells are not part of the principal direction finite-difference connections, the matrix will need to be expanded to include the other cells within the connectivity framework. Finally, because the original CVFD-assembled matrix has been altered as a result of the implicit GNC terms, the original matrix should be reset before performing back-calculations for mass-balance computations.

For an iterative update on the right-hand-side vector, equation 70 is computed from a previous iteration's estimate of heads and is added to row n of the vector and subtracted from row m of the vector. Equation 70 also needs to be applied during mass-balance computation of the cell-by-cell flow terms to adjust for the GNC.

Ghost node and contributing cell representations for some nonrectangular grid geometries are illustrated in figure 14. The code provides an option to implement the ghost node correction implicitly or explicitly in an iterative fashion. The explicit formulation combined with Picard linearization allows for use of a symmetric matrix solver.

Newton-Raphson Linearization of GNC Terms for Upstream Weighting Formulation

If Newton-Raphson linearization is used with the upstream weighting formulation of Niswonger and others (2011), it also needs to be adjusted for the ghost node

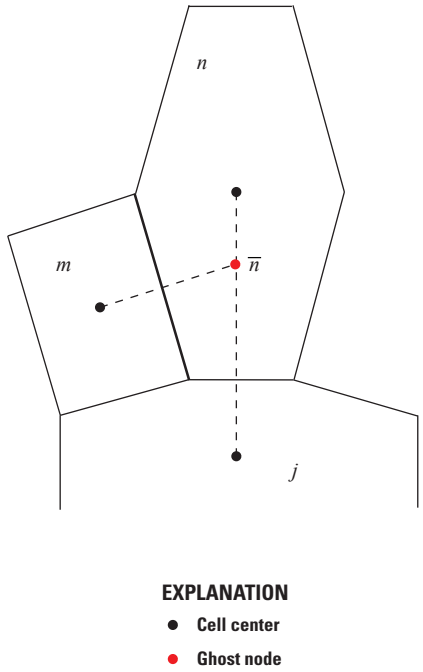

Figure 14. Ghost node conceptualization for cells with irregular geometries.

correction just discussed if the GNC Package is used. The Newton-Raphson expansion for Q_{nm} with use of ghost nodes that includes connections to adjacent cells j is given as

$$\frac{\partial Q_{nm}}{\partial h_n}\Delta h_n + \sum_{j\in\eta_n}\frac{\partial Q_{nm}}{\partial h_j}\Delta h_j + \frac{\partial Q_{nm}}{\partial h_m}\Delta h_m = -Q_{nm}\,. \tag{71}$$

Substituting the expression, $\Delta h = h^k - h^{k-1}$, (where h^k and h^{k-1} are the current and previous iteration numbers respectively) into equation 71 and rearranging gives

$$\begin{aligned}&\frac{\partial Q_{nm}}{\partial h_n}h_n^k + \sum_{j\in\eta_n}\frac{\partial Q_{nm}}{\partial h_j}h_j^k + \frac{\partial Q_{nm}}{\partial h_m}h_m^k = -Q_{nm} + \\ &\frac{\partial Q_{nm}}{\partial h_n}h_n^{k-1} + \sum_{j\in\eta_n}\frac{\partial Q_{nm}}{\partial h_j}h_j^{k-1} + \frac{\partial Q_{nm}}{\partial h_m}h_m^{k-1}\end{aligned}\,. \tag{72}$$

For the flow term of equation 65 with conductance expressed using the upstream formulation, the derivative terms in equation 72 are expressed as

$$\begin{aligned}&\frac{\partial Q_{nm}}{\partial h_n} = -\alpha_n C^0_{nm} f(h_{ups}) + \\ &C^0_{nm}\frac{\partial f(h_{ups})}{\partial h_n}(h_m - \alpha_n h_n - \sum_{j\in\eta_n}\alpha_j h_j)\end{aligned}\,, \tag{73}$$

$$\frac{\partial Q_{nm}}{\partial h_j} = -\alpha_j C^0_{nm} f(h_{ups})\,, \tag{74}$$

and

$$\begin{aligned}&\frac{\partial Q_{nm}}{\partial h_m} = C^0_{nm} f(h_{ups}) + \\ &C^0_{nm}\frac{\partial f(h_{ups})}{\partial h_m}(h_m - \alpha_n h_n - \sum_{j\in\eta_n}\alpha_j h_j)\end{aligned}\,. \tag{75}$$

Note that $\frac{\partial f(h_{ups})}{\partial h_j}$ is zero for option 1 (flag I2Kn=0) because the nonlinear conductance term is independent of head at cell j for this case. For option 2 (flag I2Kn=1), this term is neglected and treated in a Picard manner in the formulation. Equations 73 through 75 are substituted into 72 to provide a Newton-Raphson linearization for the ghost node formulation. The first terms on the right-hand side of equations 73, 74, and 75 are the flow conductance terms already assembled into the coefficient matrix by Picard implementation of the ghost node correction, as discussed in the previous section. The MODFLOW–USG code then implements the second term of equations 56 and 57 for a general Newton-Raphson linearization of the flow terms of equation 8. Because, equations 73 and 75 need to be used for ghost nodes in the coefficient matrix of equation 72 instead of equations 56 and 57, the adjustment necessary for the coefficient matrix is obtained by subtracting the second term of equations 56 and 57 and adding the second term of equations 73 and 75 instead. Thus, the adjustment for column n in rows m and n is

$$\begin{aligned}&LHS\_Adjust_n = -C^0_{nm}\frac{\partial f(h_{ups})}{\partial h_n}(h_m - h_n) + \\ &C^0_{nm}\frac{\partial f(h_{ups})}{\partial h_n}(h_m - \alpha_n h_n - \sum_{j\in\eta_n}\alpha_j h_j)\end{aligned}\,, \tag{76}$$

which upon readjustment and cancellation of common terms yields

$$LHS\_Adjust_n = -C^0_{nm}\frac{\partial f(h_{ups})}{\partial h_n}\left[[1-\alpha_n]h_n - \sum_{j\in\eta_n}\alpha_j h_j\right]. \tag{77}$$

Similarly, the adjustment for column m in rows m and n is

$$LHS\_Adjust_m = -C^0_{nm}\frac{\partial f(h_{ups})}{\partial h_m}\left[[1-\alpha_n]h_n - \sum_{j\in\eta_n}\alpha_j h_j\right]. \tag{78}$$

No further adjustment is needed for column *j* because there are no additional Newton terms for that column. The right-hand-side term of equation 72 can be expanded further by substituting equations 69, 73, 74 and 75 for the flux and its derivatives to give

$$\begin{aligned} RHS_{GNC} &= -C^0_{nm} f(h_{ups})(h_m - \alpha_n h_n - \sum_{j\in\eta_n} \alpha_j h_j) \\ &+ \left[-\alpha_n C^0_{nm} f(h_{ups}) + C^0_{nm} \frac{\partial f(h_{ups})}{\partial h_n} (h_m - \alpha_n h_n - \sum_{j\in\eta_n} \alpha_j h_j) \right] h_n \\ &+ C^0_{nm} f(h_{ups}) \sum_{j\in\eta_n} \alpha_j h_j \\ &+ \left[C^0_{nm} f(h_{ups}) + C^0_{nm} \frac{\partial f(h_{ups})}{\partial h_m} (h_m - \alpha_n h_n - \sum_{j\in\eta_n} \alpha_j h_j) \right] h_m \end{aligned} \quad ,(79)$$

where RHS_{GNC} is the right-hand-side term of equation 72 resulting from the GNC expansion. Expanding equation 79 and rearranging gives

$$\begin{aligned} RHS_{GNC} &= \left[C^0_{nm} \frac{\partial f(h_{ups})}{\partial h_n} (h_m - \alpha_n h_n - \sum_{j\in\eta_n} \alpha_j h_j) \right] h_n \\ &+ \left[C^0_{nm} \frac{\partial f(h_{ups})}{\partial h_m} (h_m - \alpha_n h_n - \sum_{j\in\eta_n} \alpha_j h_j) \right] h_m \end{aligned} \quad . \quad (80)$$

Noting that equation 60 is already filled into the right-hand-side vector for the Newton formulation, the ghost node formulation needs to adjust the right-hand-side vector by subtracting the right-hand side part of equation 60 and then adding equation 80 to the already filled right-hand-side vector to give the adjustment factor *RHS_Adjust*. Therefore,

$$\begin{aligned} RHS\_Adjust &= \left[C^0_{nm} \frac{\partial f(h_{ups})}{\partial h_n} (h_m - \alpha_n h_n - \sum_{j\in\eta_n} \alpha_j h_j) \right] h_n \\ &+ \left[C^0_{nm} \frac{\partial f(h_{ups})}{\partial h_m} (h_m - \alpha_n h_n - \sum_{j\in\eta_n} \alpha_j h_j) \right] h_m \\ &- \left[C^0_{nm} \frac{\partial f(h_{ups})}{\partial h_n} (h_m - h_n) \right] h_n - \left[C^0_{nm} \frac{\partial f(h_{ups})}{\partial h_m} (h_m - h_n) \right] h_m \end{aligned} \quad . \quad (81)$$

Expanding and rearranging equation 81 gives

$$\begin{aligned} &RHS\_Adjust = \\ &-C^0_{nm} \left[[1-\alpha_n] h_n - \sum_{j\in\eta_n} \alpha_j h_j \right] \left[\frac{\partial f(h_{ups})}{\partial h_n} h_n + \frac{\partial f(h_{ups})}{\partial h_m} h_m \right] \end{aligned} \quad . \quad (82)$$

Equation 82 is used to adjust the right-hand-side vector for Newton linearization with ghost node correction.

Computation of the α_j Contributing Factors from Adjacent Cells

In nested grid literature, geometric as well as flow considerations have been applied to obtain the α_j contributing factor. Mehl and Hill (2005) discuss linear interpolation for nesting in two dimensions, as shown on figure 12, and bilinear interpolation for three-dimensional nested grids. They also discuss a Darcy interpolation scheme that takes into account the heterogeneity of the system in influencing the head value at the ghost node. Triangulation and inverse-distance methods also may be applied to estimate the α_j contributing factors. For example, figure 15 shows the geometric lengths for computation of contributing factors. In figure 15*A*, contributing factors are

$$\alpha_j = \frac{L_{nj}}{2\left(L_{nj} + L_{jn}\right)} \quad (83)$$

and

$$\alpha_n = \frac{\left(\frac{L_{nj}}{2} + L_{jn}\right)}{\left(L_{nj} + L_{jn}\right)} \quad . \quad (84)$$

In figure 15*B*, contributing factors are

$$\alpha_j = \frac{2L_{nj}}{3\left(L_{nj} + L_{jn}\right)} \quad (85)$$

and

$$\alpha_n = \frac{\left(\frac{L_{nj}}{3} + L_{jn}\right)}{\left(L_{nj} + L_{jn}\right)} \quad . \quad (86)$$

It has been noted, however, that geometric interpolation schemes are not necessarily adequate (Panday and Langevin, 2012). For such cases, flow considerations may be employed to determine the α_j contributing factors.

In using flow considerations for determining the α_j contributing factors, it is assumed that a contributing cell's influence on the ghost node head is proportional to the conductance between the contributing cell and the ghost node. Thus, the contributing factor for each cell *j* (which can also include cell *n*) to the value at $\bar{n}$ may be obtained as

$$\alpha_j = \frac{C_{j\bar{n}}}{\sum C_{j\bar{n}}} \quad , \quad (87)$$

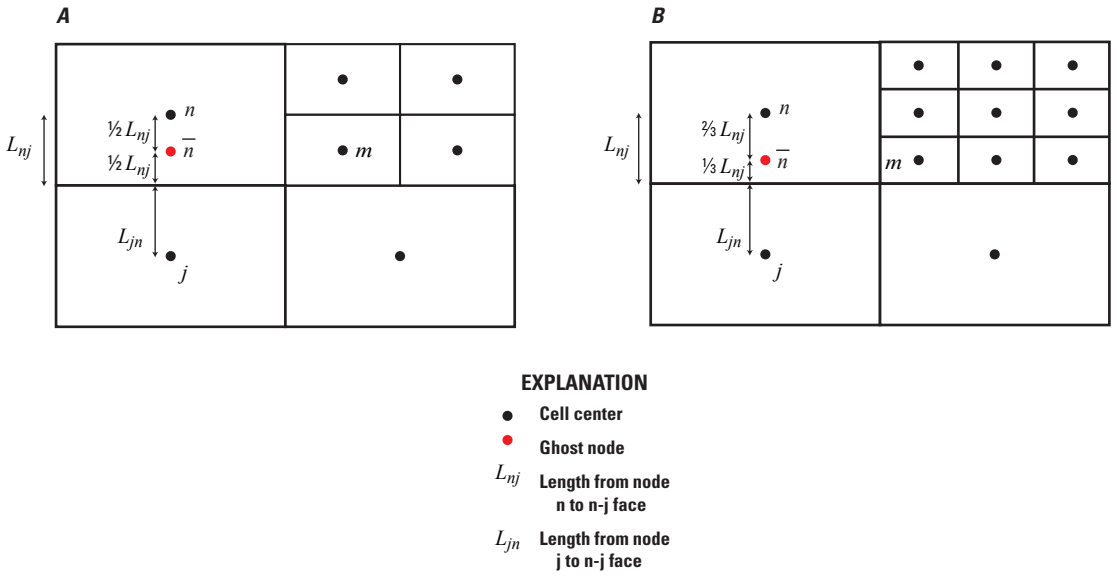

Figure 15. Length terms for the computation of contributing factors for nested grids in which *A*, a parent cell is connected to two child cells and *B*, a parent cell is connected to three child cells.

where $C_{j\bar{n}}$ is the conductance between the contributing cell j and the ghost node $\bar{n}$, and the summation is over all j cells contributing to the interpolated value at $\bar{n}$. The conductance $C_{j\bar{n}}$ is obtained as the hydraulic conductivity between cell j and the ghost node $\bar{}$, multiplied by the interacting area and divided by the separation distance. Therefore, the scheme reduces to an inverse-distance formulation for a homogeneous medium discretized with a uniform grid size. In addition, if $\bar{n}$ is located at the center of cell n, all α_j contributing factors will be zero with $\alpha_n = 1$, because the separation distance to n is zero, and thus, $C_{n\bar{n}} = \infty$.

It is further noted that the α_j contributing factors may vary with conductance for an unconfined case. For this situation, the saturated conductances may be scaled using the nodal saturated thickness value of the upstream cells between j and $\bar{n}$ to determine an unconfined conductance value to be used in equation 87. The unconfined $C_{j\bar{n}}$ values and α_j contributing factors are computed at the start of every iteration. Water levels at adjacent cells would be expected to fluctuate together, and therefore, this nonlinearity may not be substantial. As a result, a Newton-Raphson expansion of this term may not be warranted and is not done in MODFLOW–USG.

Representation of Subgrid Scale Displacements Using the GNC Formulation

The GNC concept provides a robust approach to account for subgrid-scale discrepancies between the location of a feature, such as a well, and the location of the cell center. A conceptual accounting of subgrid scale displacements of features within a simulation can reduce errors of implementing small-scale features within larger cells of regional large-scale simulations. For example, if cell m in figure 16 represents a well simulated using the CLN Process, then flow between the CLN cell m and GWF cell n is governed by the head in the GWF cell at a ghost node location $\bar{n}$, and not by the head at the GWF cell center. The contributing fraction for displacement of GWF head at location $\bar{n}$ can be computed using flow considerations of equation 87, with contributing cells n and m and the adjacent cells j because they all have an influence on the head value at the ghost node location $\bar{n}$. Geometric considerations do not work for this case, because the conduit cell cannot be readily conceptualized using distance weighting. The value of $C_{j\bar{n}}$ may be obtained for the GWF cell by drawing a control volume around the ghost node location with respect to all contributing cells. As shown in figure 16, for example, the face area of the control volume $a_{j\bar{n}}$ is the perpendicular bisector to the line joining j and $\bar{n}$, of length $L_{j\bar{n}}$, times the saturated thickness. The conductance $C_{j\bar{n}}$ is then determined as

$$C_{j\bar{n}} = K_{j\bar{n}} a_{j\bar{n}} / L_{j\bar{n}}, \tag{88}$$

where $K_{j\bar{n}}$ is the effective hydraulic conductivity between j and $\bar{n}$. The conductance of a CLN–GWF connection may be obtained from equation 27, 28 or 29, as discussed earlier.

The ghost node corrections discussed in equations 65 through 88 are applied at active cells in the domain. The concept could also be applied to reduce the subgrid scale boundary flux errors at boundary conditions simulated by the Stream (SFR and STR) or River (RIV) Packages, for example. Panday and Langevin (2012) show how to improve the flux at the boundary location by using the head at an

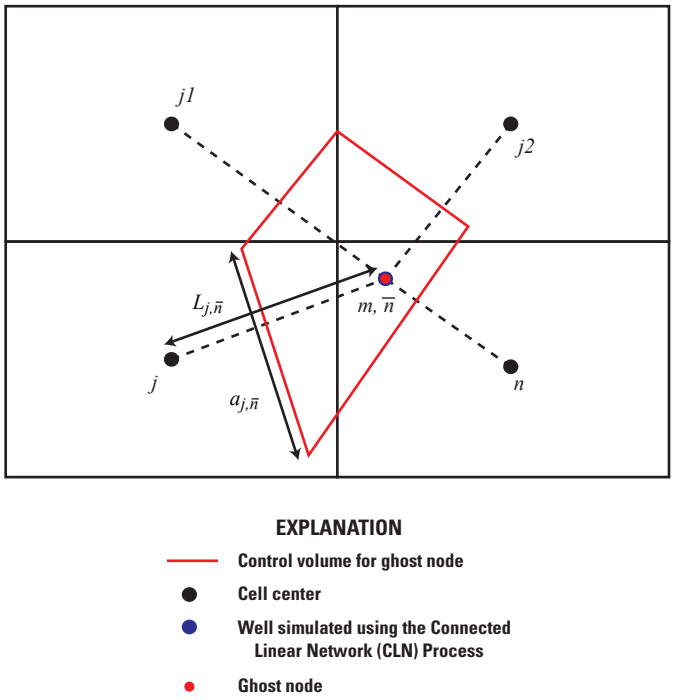

Figure 16. Control volume around ghost node for computing contributing factors with subgrid scale displacement. Note that the ghost node is collocated with a well represented by the Connected Linear Network (CLN) Process.

interpolated ghost node location that coincides with the boundary, rather than at the cell center.

Boundary Packages

Several existing MODFLOW–2005 boundary packages were modified to work with MODFLOW–USG. Boundary packages included in the MODFLOW–USG release documented here include the

- Prescribed head boundary (CHD) Package,
- Recharge (RCH) Package,
- Evapotranspiration (EVT) Package,
- Well (WEL) Package,
- General-Head Boundary (GHB) Package,
- Drain (DRN) Package,
- River (RIV) Package,
- Stream (STR7) Package,
- Stream Flow Routing (SFR7) Packages,
- Lake (LAK7) Package, and the
- Transient Flow and Head Boundary (FHB) Package.

The formulations for these boundary packages (the equations for adding terms to the RHS and HCOF accumulators, for example) are consistent with their implementation in MODFLOW–2005; however, the packages have been extended to work with node indexing for unstructured grids, in addition to working with the structured layer, row, and column indexing option.

All of the boundary conditions of MODFLOW–USG previously described may be applied to GWF cells and to CLN cells, when an unstructured grid framework is used for the GWF Process. These boundary packages are applied to GWF or CLN cells by referencing the global node number of the cell; therefore, with the unstructured grid framework, boundary package input files may contain both GWF cell numbers and CLN cell numbers. The global node number for a GWF cell corresponds to its GWF node number; however, the global node number for a CLN cell must be calculated by adding the number of GWF cells to the CLN node number. In addition, as with the GWF Process, the CLN Process uses an IBOUND array to indicate active (positive value), inactive (zero value), or prescribed head (negative value) conditions.

The CLN Process may also be used when a structured grid is used for the GWF Process. This capability allows for conduits to be tightly coupled with a structured grid model. In this case, the WEL Package may also include CLN Process cell numbers to allow for injection or withdrawal from a CLN cell. This use of the CLN Process to simulate multi-node wells is demonstrated in the third example shown later. The other boundary conditions cannot be assigned to CLN cells when a structured grid is used for the GWF Process.

Automated Flux Reduction for the Well (WEL) Package

The WEL Package includes an optional automated flux reduction capability, as implemented in MODFLOW–NWT (Niswonger and others, 2011). If the saturated thickness of the GWF or CLN cell containing the well is less than 1 percent of the cell thickness, then the well will automatically lower its pumping rate, depending on the flow to the well from the connected GWF cells. In essence, the well operates as if the head in the well was equal to the cell bottom until the water level recovers. The formulation for the automated flux reduction is provided in the MODFLOW–NWT documentation (Niswonger and others, 2011). The only difference between the MODFLOW–USG and MODFLOW–NWT implementation is that the user specifies the percent of saturated thickness for the reduction to occur in MODFLOW–NWT, whereas this value is fixed at 1 percent in MODFLOW–USG.

The Sparse Matrix Solver (SMS) Package

The system of equations formulated by MODFLOW–USG is solved using the SMS Package. The SMS Package provides several nonlinear methods to handle conditions in

which conductance is a function of head and several linear solution schemes to solve the matrix equations. The nonlinear solution methods include Picard iterations for the unconfined options of MODFLOW–2005 and Newton-Raphson linearization for the upstream weighting option of simulating unconfined conditions, as in MODFLOW–NWT. Residual reduction and under-relaxation schemes are also implemented to control evolution of the nonlinear solution. The various schemes have advantages depending on computer speed, memory, and robustness considerations. Users are encouraged to experiment with the different solution options to find the best approach for their particular problem.

The coefficient matrix generated by MODFLOW–USG is always stored in an unstructured format even if the problem is structured. For this reason, the structured linear solvers available in MODFLOW–2005 and its predecessors cannot be used in their current form with MODFLOW–USG. The PCGU solver of White and Hughes (2011) is therefore implemented to solve the unstructured symmetric system of equations arising from confined flow or from Picard iterations to unconfined flow with any and all asymmetric updates provided in the right-hand-side vector. Furthermore, the Newton-Raphson formulation and the fully implicit ghost node formulation can generate an asymmetric coefficient matrix. The χMD solver of Ibaraki (2005) provides several robust options for solving the linearized set of asymmetric unstructured matrix equations. Flexibility includes (1) options for matrix reordering including red-black ordering for a reduced system; (2) options for various levels of fill for ILU decomposition used as a preconditioning step with addition of drop-tolerance schemes for further efficiency; and (3) a choice of BiCGSTAB and ORTHOMIN acceleration schemes for asymmetric matrices or Conjugate Gradient acceleration for symmetric matrices. To guide the nonlinear iterations, the delta-bar-delta methodology is adapted from neural network literature to provide cell-by-cell under-relaxation if requested by the user. In this scheme, an under-relaxation factor is provided to every element of the head change vector. If there is an oscillation in the head change from the previous iteration history, the factor for a specific cell is reduced by a user-defined amount. If the head change is in the same direction as the previous iteration history, the factor for a specific cell is incremented by a user-defined amount. A momentum term is also included that adds a user-defined fraction of the previous head update to the current one. The scheme is efficient in finding the solution to problems that encounter oscillatory behavior in the nonlinear iterations. The Cooley (1983) scheme is provided as a less memory-intensive alternative for under-relaxation of oscillatory behavior during iterations.

The optimal settings of parameters for the matrix solver and the nonlinear backtracking and under-relaxation schemes are unique for every problem. Therefore, output summaries are provided for each of these processes in the output listing file to aid a modeler with analyzing solution behavior; these summaries are discussed with the input instructions of the SMS Package. Input instructions for the SMS Package provide some guidance on setting parameters and adjusting them as per run behavior.

The flowchart for the main SMS Package is shown on figure 17. For each nonlinear iteration, the program calls the GWF2SMS8AP subroutine after the flow equations and boundary conditions have been formulated and assembled into the coefficient matrix and right-hand-side vector. Subroutine GWF2SMS8AP controls what is done with the assembled flow equations. First, the subroutine encounters the residual reduction routines. If residual reduction is active, the latest residual is computed and if the residual reduction criterion is exceeded, the head change vector is reduced by a user-prescribed amount and the subroutine is exited and control is passed to the beginning of the iteration to reassemble the matrix and right-hand-side vector with heads computed from the reduced head change vector. The algorithms are detailed by Niswonger and others (2011). The residual reduction routines include additional user controls on backtracking including maximum backtracking count and tolerance limits on residual reduction.

After the residual reduction section, if there is no more residual reduction to be performed, the subroutine checks to see if the Newton-Raphson linearization option is active. If active, the Newton-Raphson Jacobian terms are assembled as detailed by Niswonger and others (2011). Following this, a call is made to the appropriate linear sparse matrix solver. After solving for heads, a check is made on convergence for the nonlinear system of flow equations, followed by application of under-relaxation schemes if required, before exiting subroutine GWF2SMS8AP for the next nonlinear iteration or, if converged, for computation of mass balance and output of requested results before moving on to the next time step (for a transient simulation).

Implementation and Program Design

MODFLOW–USG is similar in many respects to MODFLOW–2005. The MODFLOW–USG program was developed by starting with the MODFLOW–2005 source code and modifying the code according to the concepts described herein. As with MODFLOW–2005, MODFLOW–USG is based on the concept of processes, packages, and modules. This section describes how the unstructured grid approach is implemented into the MODFLOW–2005 framework.

Program Structure

The sequence of subroutine calls within the MODFLOW–USG code is similar to the sequence of calls within MODFLOW–2005. Differences between the two programs are in the details, whereby the “formulate” procedure in MODFLOW–USG includes the CLN Process and the GNC Package, which require the addition of call statements to the main program to include their functionality. In addition, the cell-by-cell flow term write statements are divided into separate subroutines to accommodate matrix re-adjustment for the GNC Package and to facilitate separate output of GWF and CLN domain flows.

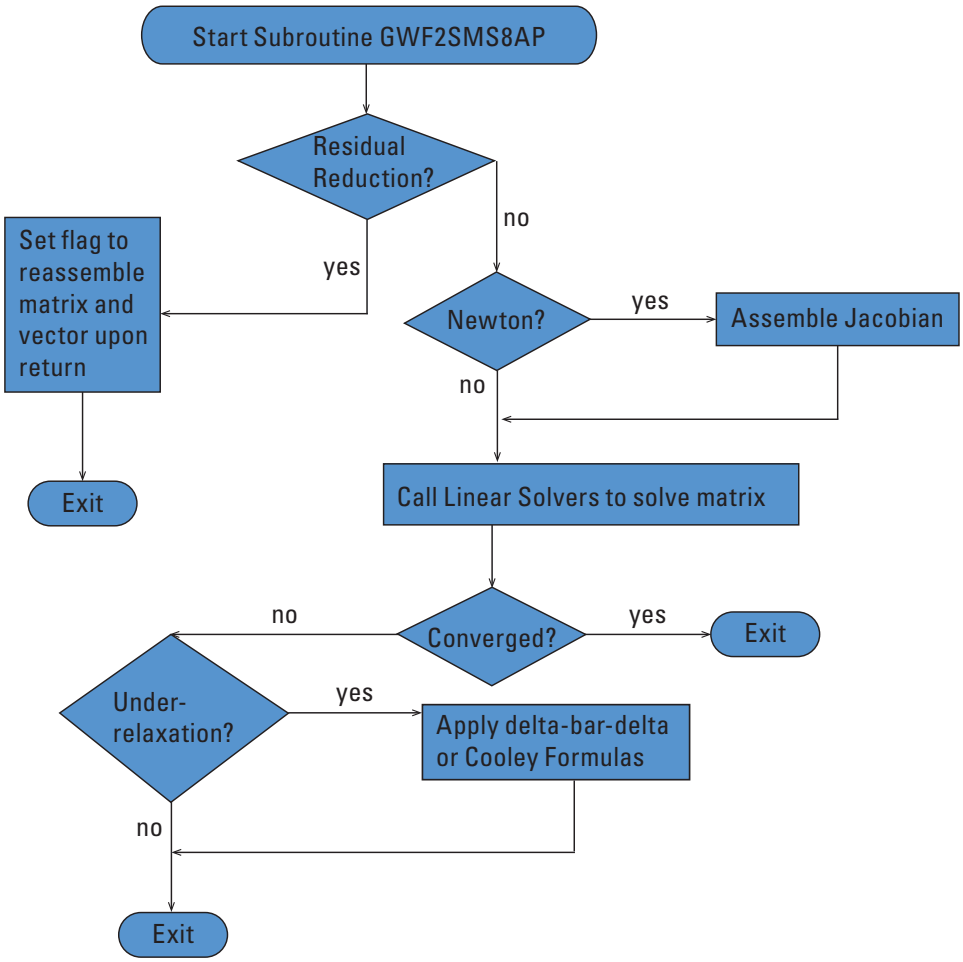

Figure 17. Generalized flowchart for the Sparse Matrix Solver (SMS) approximate subroutine (GWF2SMS8AP).

Internal Array Storage

A fundamental difference between MODFLOW–2005 and MODFLOW–USG is the way in which internal data are stored. In MODFLOW–2005, arrays that store information for every cell, such as the IBOUND array, are dimensioned as (NCOL, NROW, NLAY). In MODFLOW–USG, these types of arrays are treated as one-dimensional arrays of size (NODES), where NODES is the total number of cells. For a structured grid, the number of GWF cells is equal to the product of the number of layers, rows, and columns.

A key component of MODFLOW–2005 is the assembly of the coefficient **A** matrix, which contains the coefficients for the linear system of equations. The **A** matrix has dimensions of [NODES, NODES], but it is sparse, and most of the terms are zero. MODFLOW–2005 stores these coefficients using four 3-dimensional arrays of size (NCOL, NROW, NLAY). Off-diagonal terms are stored in the CC, CR, and CV arrays. The diagonal coefficients are constructed from HCOF and the CC, CR, and CV arrays. This efficient storage scheme can be used for the **A** matrix, because MODFLOW–2005 uses a structured grid and because the matrix is symmetric. MODFLOW–USG is based on the concept of an arbitrary number of cell connections, and so an alternative storage scheme must be used for the **A** matrix. For most problems involving unstructured grids, a compressed matrix storage scheme is used. If the matrix is not stored in a compressed format, the amount of memory required to store a matrix of size [NODES, NODES] would exceed the available memory of most computers for even modest-sized problems.

MODFLOW–USG uses the compressed sparse row (CSR) format for the **A** matrix and for other information required for each connection. With the CSR format, all nonzero coefficients in a row of the matrix are stored sequentially, beginning with the first one in the row and ending with the last. Pointers IA and JA index the location within the array, where the information for rows and columns of the matrix are stored. IA points to the starting location of a row in the matrix, and JA indexes the

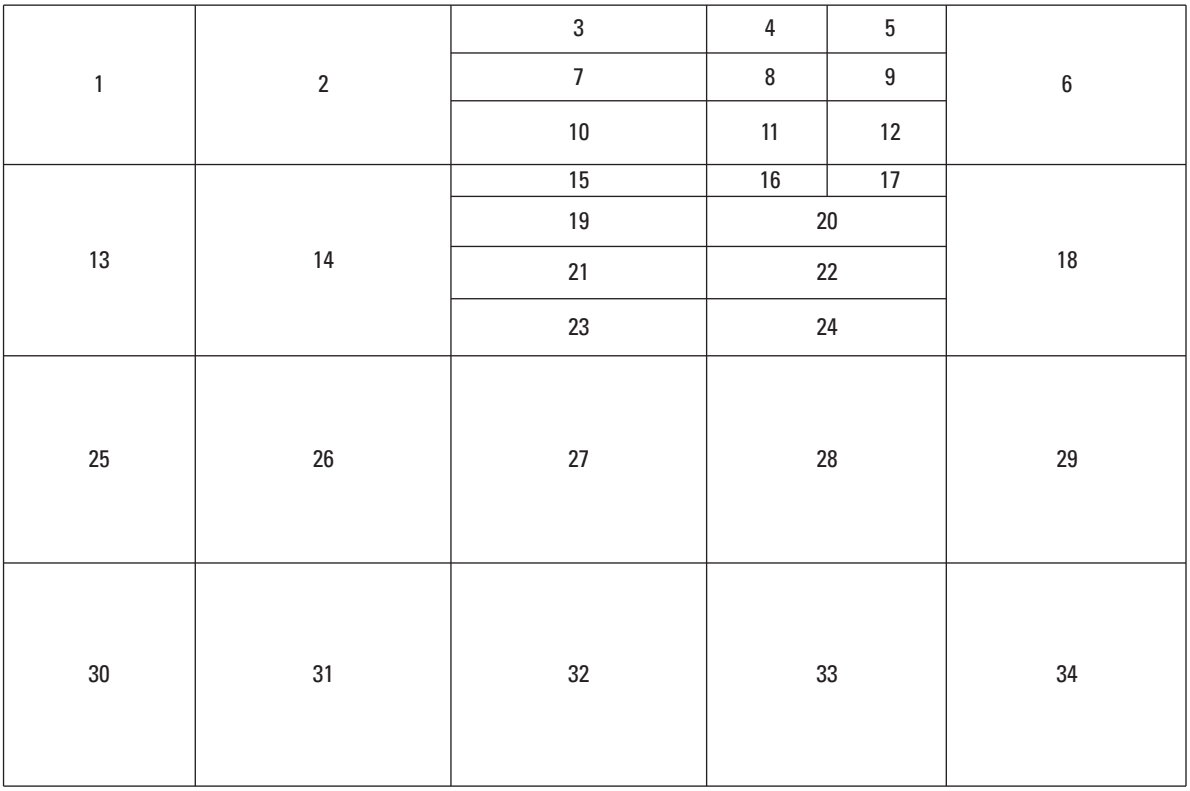

Figure 18. Example of an unstructured grid showing the arbitrarily assigned cell numbers.

nonzero columns of the rows. IA is dimensioned as (1 : NODES + 1); JA is dimensioned as (NJA), which is the total number of nonzero coefficients in the matrix. In MODFLOW–USG, NJA is equal to the number of connections per cell plus one, summed over all computational cells in the simulation.

An example of an unstructured grid is shown in figure 18. The **A** matrix for this grid would have the shape shown in figure 19, where a solid square indicates a possible nonzero coefficient value. The locations of the nonzero values for the matrix shown in figure 19 can be stored in the CSR format using the IA and JA pointers. For this example, the length of the JA index array is 162, which is the total number of solid squares in figure 19. The JA index array is constructed by traversing each row, inserting the column position of the diagonal, and then listing in consecutive order the nonzero column positions for each off-diagonal entry. For this example, JA consists of the following sequence:

-1 2 13 -2 1 3 7 10 14 -3 2 4 7 -4 3 5 8 -5 4 6 9 -6 5 9 12 18 -7 2 3 8 10 -8 4 7 9 11 -9 5 6 8 12 -10 2 7 11 15 -11 8 10 12 16 -12 6 9 11 17 -13 1 14 25 -14 2 13 15 19 21 23 26 -15 10 14 16 19 -16 11 15 17 20 -17 12 16 18 20 -18 6 17 20 22 24 29 -19 14 15 20 21 -20 16 17 18 19 22 -21 14 19 22 23 -22 18 20 21 24 -23 14 21 24 27 -24 18 22 23 28 -25 13 26 30 -26 14 25 27 31 -27 23 26 28 32 -28 24 27 29 33 -29 18 28 34 -30 25 31 -31 26 30 32 -32 27 31 33 -33 28 32 34 -34 29 33

To help illustrate the sequence, the row position is marked in the JA array with a negative value. The IA array is constructed by listing the position of each diagonal (indicated by the negative numbers) within the JA index array. One additional number is added to the end of the IA index array so that the number of connections for the last cell can be determined. For this example, the IA index array has the following 35 (NODES+1) entries:

1 4 10 14 18 22 27 32 37 42 47 52 57 61 69 74 79 84 91 96 102 107 112 117 122 126 131 136 141 145 148 152 156 160 163

Internally, MODFLOW–USG uses the IA and JA arrays to store connection information; however, users are asked to provide an IAC array in the unstructured discretization input files instead of an IA array. The IAC array is of size (NODES) and contains the number of connections for each cell plus one (the diagonal position). The IAC array is used for input instead of the IA array because it is considered to be more straightforward. For the example used here, the IAC array is as follows:

3 6 4 4 4 5 5 5 5 5 5 5 4 8 5 5 5 7 5 6 5 5 5 5 4 5 5 5 4 3 4 4 4 3

With these index arrays, it is possible to loop through each connection for each cell using the following example pseudo code:

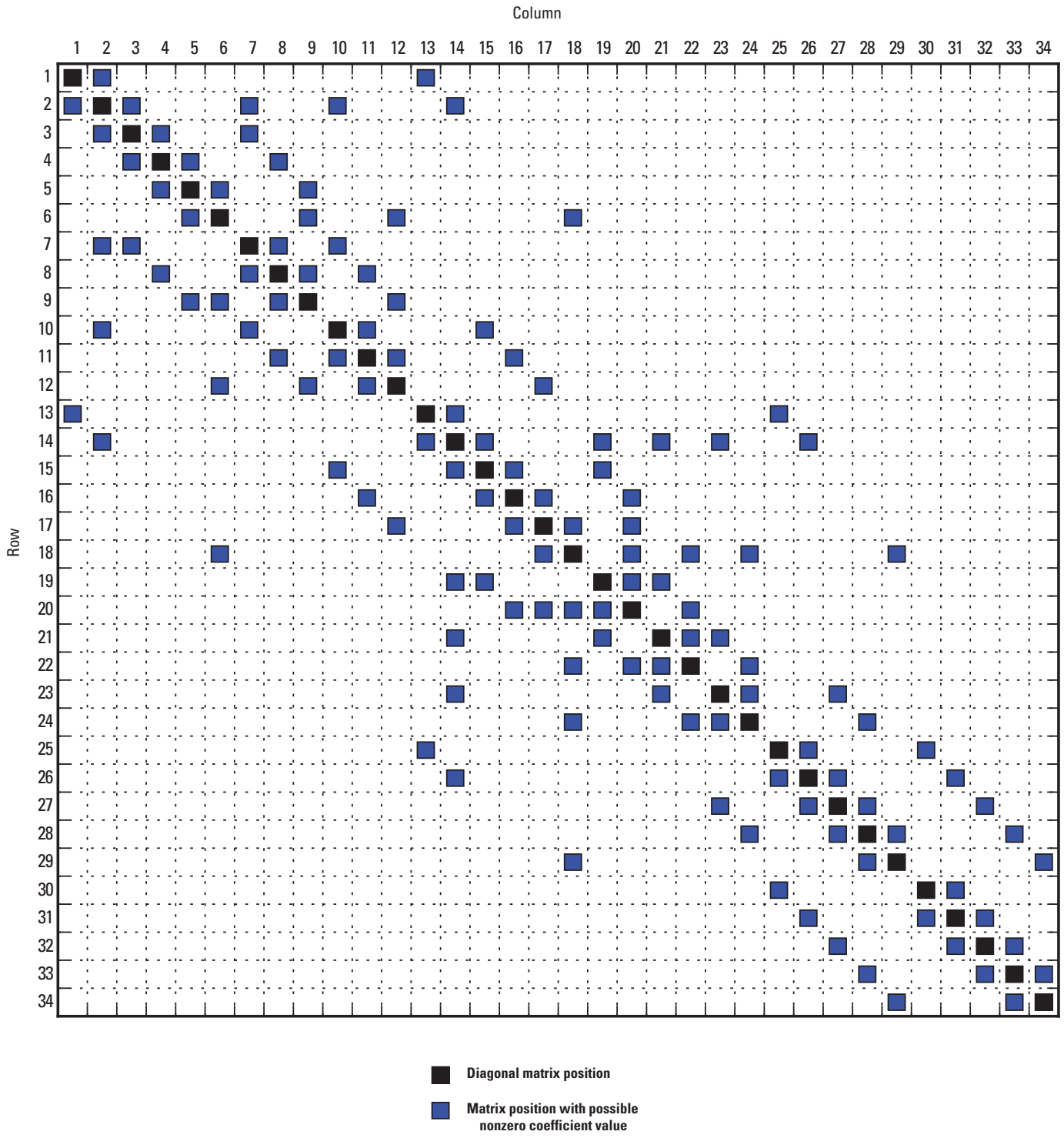

Figure 19. Matrix of size 34 by 34 corresponding to the unstructured grid in figure 17. A square indicates a connection between two cells. The diagonal is shown in black.

```
do n=1 to NODES
  do j=IA(n) to IA(n+1)-1
    if JA(j) equals n then
      print "diagonal for node: ", n
    else
      print "conn. nodes: ", n, JA(j)
    end if
  end do
end do
```

Precision of Real Variables

The MODFLOW–USG code uses mixed precision in a manner similar to MODFLOW–2005. Variables are declared as ***Real*** or ***Integer*** depending on their purpose. In addition, certain variables are declared as ***Double Precision*** where additional precision is needed. The double precision arrays include HNEW and HOLD (which store the old and new water-level values at every cell); the coefficient matrix AMAT; the right-hand-side vector, RHS; the top elevation array TOP and the bottom elevation array, BOT. Double precision variables include DELT, TOTIM and PERTIM. These variables are however converted to ***Real*** before being written to any ASCII or binary output file, in accordance with the MODFLOW–2005 paradigm.

Special Considerations for Using "Highest Active Layer" Options

Some packages have options for applying a flux to the "highest active layer." For example, the RCH and EVT Packages have an option to interact with the highest active cell in a vertical column of cells. The RCH and EVT Packages of MODFLOW–2005 apply the flux to the highest active layer in the domain when their respective options NRCHOP and NEVTOP are set equal to 3. As each layer becomes inactive in a simulation in response to water levels falling below its base, the recharge or evapotranspiration flux is applied to the next active layer beneath it.

The option of applying flux to the topmost active layer requires special consideration in MODFLOW–USG when the upstream weighting formulation is not used. Because unstructured-grid configurations can result in one cell that underlies multiple cells or vice versa, applying a flux to the uppermost active cell is complicated when the water table fluctuates through layer interfaces. For this case, net flux is applied only to the first underlying cell encountered in the connectivity list, beneath the dry cell. In this manner, the total recharge or evapotranspiration flux is conserved; however, the spatial distribution of the flux may not be maintained as layers become dry or wet during a simulation. An example of this situation is shown in figure 20, which shows cells 4, 5, 7, 8, and 10 as being dry. Because cell 11 is the lowest numbered cell beneath cell 10, the recharge applied to cells 4, 5, 7, and 8 is instantaneously applied to cell 11.

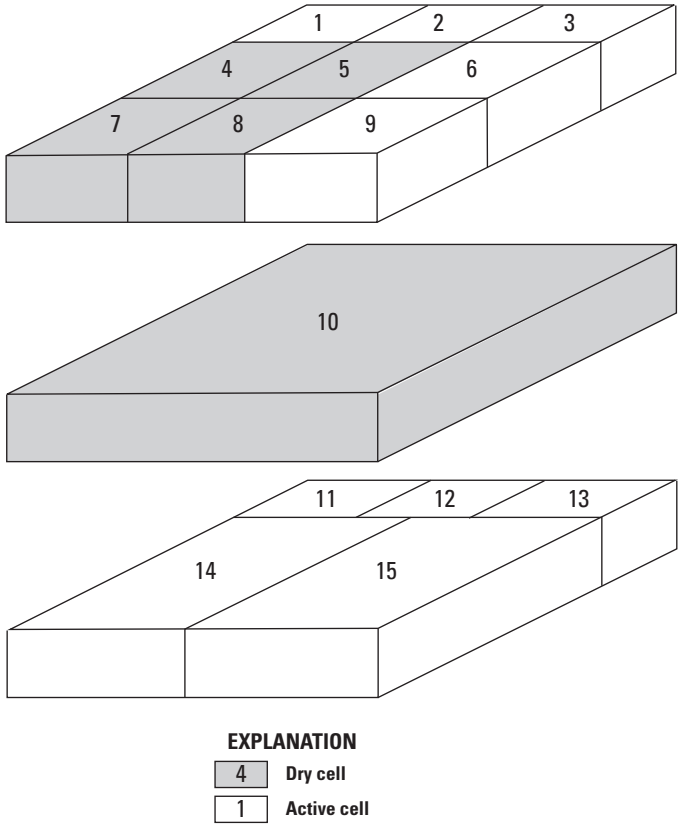

Figure 20. Three layer model with a different discretization pattern in each layer. Dry cells are shown in gray. When the NRCHOP=3 option is used in the Recharge Package and the Newton-Raphson linearization is not used, then recharge added to cells 4, 5, 7, and 8 is instantaneously added to cell 11. Cell 11 is the lowest numbered cell directly beneath cell 10.

Issues with the highest active layer option also can arise in the LAK3 and SFR2 Packages where the flux exchange occurs with the first active layer. To avoid this behavior, which may not be intended by the user, the grid should have the same topology in all layers through which the water table fluctuates. This is not a problem if the upstream weighting formulation is used, because cells are not inactivated and will still transmit a flux vertically, even if the calculated head is below the cell bottom.

Water Budget Calculations

Water budget computations are done as in MODFLOW–2005, after convergence is achieved for the flow equation computations. If the GNC Package is used in a fully implicit manner, the matrix is first reset to pre-ghost node conditions. The flux through each face of every cell is then computed by back-substitution of the solution heads into the off-diagonal flow terms of the coefficient matrix. The asymmetric part of the flow term is then adjusted for ghost nodes if the GNC Package is used (implicit or explicit), before output of the cell-by-cell flows; this is done for both GWF and CLN cells. The

storage term for every GWF and CLN cell is also computed for transient stress periods. Boundary flux terms are then computed in the same manner as in MODFLOW–2005 for every boundary condition type that is used in the simulation.

Model Output Files

The output structure of MODFLOW–USG is similar to that of MODFLOW–2005, with the same types of information being generated by the model. In fact, the output files for structured grids have an identical structure to those written by MODFLOW–2005, and therefore, any MODFLOW post-processor may be used to analyze results from a structured grid simulation of MODFLOW–USG. If the CLN domain is active, a separate set of outputs is also generated for the CLN domain.

Every MODFLOW–USG simulation generates a run-listing file that consists of the input data for the simulation; the solver and nonlinear outputs at user-requested detail; head and drawdown solutions, if requested; mass-balance information; and time-step information for the simulation. Optional ASCII and binary files can also be created by setting flags in the Output Control input file. Head and drawdown output are typically generated in binary files with extension HDS or DDN for the GWF domain cells. Separate binary files may be generated for the CLN domain cells. The HDS and DDN files contain a header and head or drawdown values for all cells in each layer as requested. For unstructured grids, the number of cells may be different for each layer. If requested, a binary file containing the cell-by-cell flow terms for the GWF and CLN cells can also be generated.

Guidance for Applying MODFLOW–USG

This section provides guidance on applying MODFLOW–USG, including suggestions for getting started and designing grids. The next report section demonstrates parts of this guidance through detailed description of several example problems.

Getting Started with MODFLOW–USG

Although MODFLOW–USG is similar to MODFLOW–2005 in several respects, many differences between the codes require a thorough understanding before MODFLOW–USG can be used correctly. Therefore, the first step in getting started with MODFLOW–USG is to read this report and understand its capabilities before attempting any simulations. The next step is to run the test problems that are distributed with the MODFLOW–USG program. These test problems were designed to familiarize users with many of the MODFLOW–USG features. Users should then consider running an existing MODFLOW–2005 model with MODFLOW–USG. This can be done by replacing the input file for the MODFLOW–2005 solver package with the input file for the SMS Package. Results from MODFLOW–USG should be compared with MODFLOW–2005 results to ensure that appropriate SMS Package input parameters have been correctly specified. Users may also want to experiment with the different SMS Package options to understand important solver differences between MODFLOW–2005 and MODFLOW–USG. A next step is to convert the input for the model into in an unstructured format. In addition to restructuring the discretization (DIS) input file, users will need to replace all references to layer, row, and column in the input packages to cell number. Results from this unstructured grid model can also be compared with the original MODFLOW–2005 model to ensure that the conversion was successful. Finally, users may want to experiment with other grid types, such as a nested grid, a quadtree grid, or a grid composed of triangular volumes.

Grid Design

MODFLOW–USG is capable of simulating groundwater flow using a wide range of different grid types. Unless a structured MODFLOW grid is used, additional effort will be required by MODFLOW users to understand and represent the grid connectivity, create input files, and post-process results. Existing MODFLOW users may want to consider using grids that are based on rectangular cells, such as LGR-style grids and quadtree grids. One might also consider using a finely discretized structured grid for the top layer to better represent surface features. Coarser structured grids could then be used for deeper layers. Applications of MODFLOW–USG to complex field problems will require graphical pre- and post-processing software or user-developed customized programs to create input files and evaluate model output. In general, the pre- and post-processing required for unstructured grids will be more complicated than the processing required for MODFLOW structured grids.

As noted earlier, the flexibility in gridding raises issues regarding the accuracy of a given solution if certain rules for gridding are not followed. The GNC Package can be used to improve accuracy for many grid types, but use of the package is not straightforward for all grid types. Unless grids are designed appropriately or proper corrections are made using the GNC Package, large errors in simulated heads and flows can result. These errors can be difficult to detect because they do not show up as errors in the simulated water budget. It is important for MODFLOW–USG users to develop an understanding of these errors, what causes them, and how to reduce them through appropriate grid design strategies and flux correction approaches.

Nesting model grids (as shown in fig. 5*B–C*) is a straightforward way of adding refinement in areas where accurate representation of hydraulic gradients is important or where refined hydraulic data are available or required. An LGR-style grid, such as the grid required by MODFLOW–LGR (Mehl and Hill, 2005), is a good way for MODFLOW users

to begin taking advantage of the benefits of the unstructured grid approach. With MODFLOW–LGR, the user starts with a parent MODFLOW model and adds higher-resolution child models. The parent and child models are separate MODFLOW models consisting of their own datasets. During program execution, MODFLOW–LGR ensures that groundwater fluxes between models are accurately calculated. A different approach is used to simulate nested grids with MODFLOW–USG, whereby multiple grids can be combined to form a single LGR-style grid; internally, the program is not aware that the grid is composed of multiple nested grids. Unlike MODFLOW–LGR, MODFLOW–USG does not have restrictions on the number of nested grids, the number of grid nesting levels, or the shapes of the nested grids. Because MODFLOW–USG requires a single grid that contains all of the cell connections, however, sophisticated pre- and post-processors will generally be required to create the input files for MODFLOW–USG.

For problems requiring accurate simulation of groundwater flow at different scales in different areas of the domain, users may want to try quadtree refinement (or octree refinement for three-dimensional grids), which is another way to develop a nested grid. Quadtree refinement is based on the notion that any cell (normally a square cell) can be divided into four equally sized cells. Quadtree grids are often smoothed, which means that a cell connects to no more than two cells in any one direction. An advantage of a smoothed quadtree grid is that it is straightforward to apply the GNC Package, and consequently, accurate simulations can be designed for problems involving different scales in different parts of the domain. For example, quadtree refinement would be useful for representing multiple extraction wells and their effects on a stream network.

Nested grids and quadtree refinement are terms for describing grids, and they may be important for preprocessors that construct these mesh types and the associated input files; however, the present version of MODFLOW–USG does not require information about the specific grid type or reference location. All of the grid information is characterized by the connectivity, connection lengths, connection flow areas, and cell areas and volumes. This information is provided to MODFLOW–USG in the unstructured discretization file.

Pinching Hydrostratigraphic Layers and Faults

The gridding flexibility in MODFLOW–USG offers advantages for representing discontinuous (pinching) hydrostratigraphic layers and faults compared to traditional MODFLOW approaches. Consider the simple hydrostratigraphic cross section shown in figure 21*A*. This type of system is difficult to model with MODFLOW, especially when there is a need to represent each aquifer layer by a single model layer. MODFLOW users have developed approaches for assigning model layer properties in areas where the hydrostratigraphic unit assigned to a model layer is absent. These approaches, which often involve specifying minimum layer thicknesses, can produce accurate results, but they often result in more cells than are necessary to characterize the problem. Figure 21*B* shows one example of an unstructured grid consisting of eight model layers that could be used to discretize the pinching layer system. In this example, ghost nodes could be used to improve the accuracy of horizontal flux calculations.

The presence of faults can also introduce complications for designing a structured grid, because a hydrostratigraphic unit on one side of the fault may not be connected with its counterpart on the other side of the fault, as shown in figure 22*A*. In MODFLOW, a straightforward method does not exist for representing this type of system if the intent is to maintain correspondence between hydrostratigraphic layers and model layers. Users often apply the Horizontal Flow Barrier (HFB) Package and adjust horizontal and vertical aquifer properties as a way to approximate groundwater flow through a faulted aquifer system. Faulted systems can be represented more easily by MODFLOW–USG because of the flexibility in assigning cell connections. The unstructured grid shown in figure 22*B* is one way to discretize the faulted system shown in figure 22*A*. Here the grid was constructed so that cell 11 is laterally connected to cells 12, 34, and 56. If necessary, the HFB Package in MODFLOW–USG also might be used here to simulate the increased horizontal resistance to flow that can occur along faults.

In the example just presented, the GNC Package could be used to improve the flux calculation between cells 11 and 56, for example. Ghost nodes could be placed in both cells at the elevation equal to the shared face midpoint. In cell 11, the ghost node head would include a contribution from cells 11 and 33, whereas the ghost node in cell 56 would include contributions from cells 56 and 34.

Selection of Solver Options

The SMS Package includes several options for solving the governing flow equations. The package controls linearization, under-relaxation/residual control, and solution to the set of linear sparse matrix equations. An understanding of the various solver options and their compatibility with the various formulations can help with building more robust and efficient models.

For confined simulations involving the GWF and/or CLN domains, the system of equations is symmetric and linear. For this case, the Newton-Raphson linearization is not required and the numerical solution is obtained in only one outer iteration, providing the linear solver converged. However, there are conditions for confined systems that require more outer iterations, because of the numerical solution method. For example, use of the GNC Package with the GNC terms applied to the right-hand-side vector requires an iterative procedure to converge to a solution. Boundary conditions such as those available with the EVT, LAK or SFR Packages also introduce nonlinearities and require multiple outer iterations.

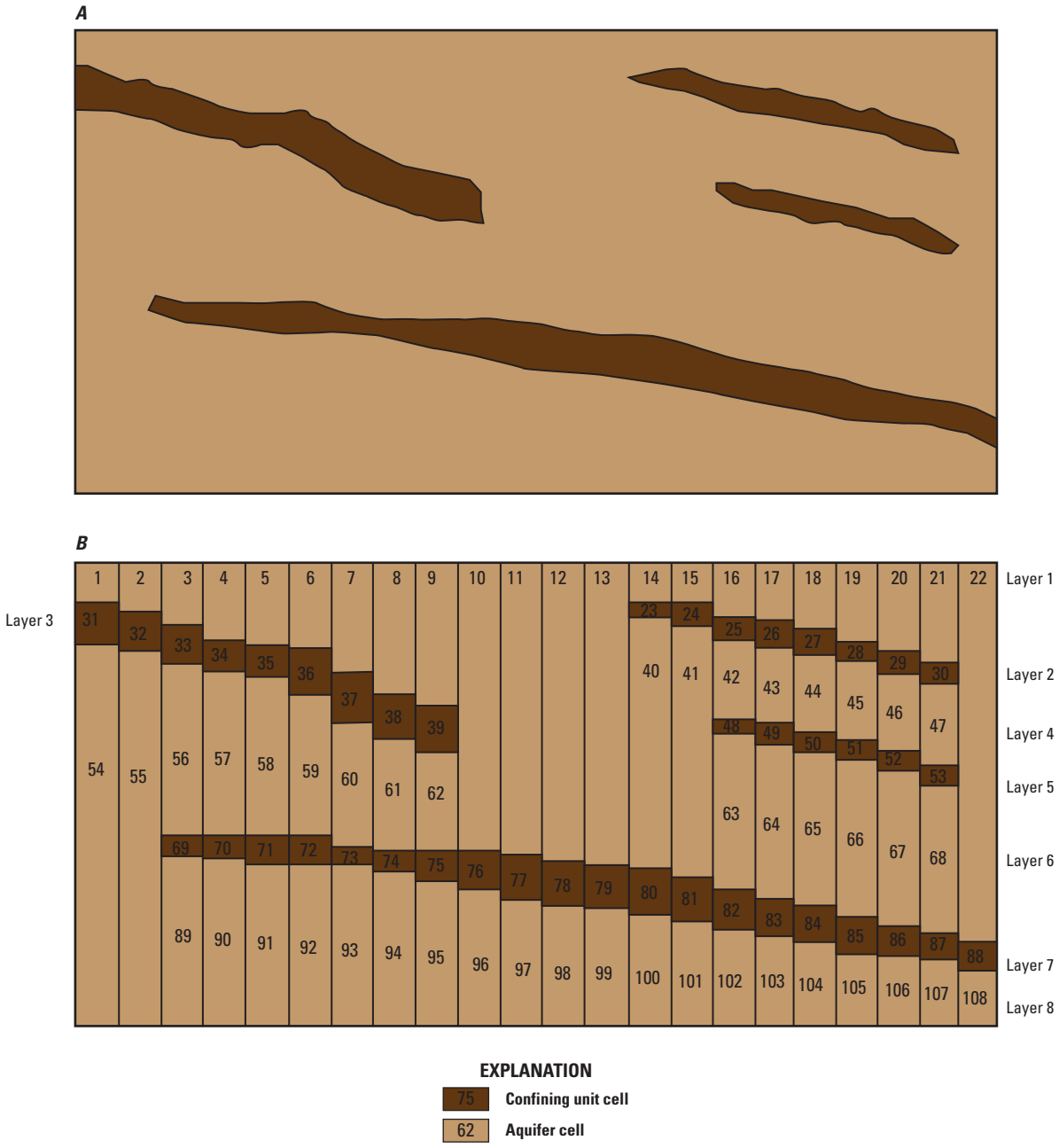

Figure 21. Cross-sectional diagrams showing *A*, pinching hydrostratigraphic layers and *B*, one example of an unstructured grid that could be used to represent the hydrostratigraphic layers.

For unconfined simulations, the system of equations used is nonlinear. For these conditions, Newton-Raphson linearization may be applied only with the upstream-weighting formulation for the GWF Process—other GWF Process flow options are not affected by Newton-Raphson linearization except for when flow is to a convertible cell above an unconfined layer. If an unconfined CLN domain is included in a simulation, the Newton-Raphson methodology should be used to improve robustness because the CLN domain flow solution uses the upstream-weighting formulation only and does not include the other flow options of the GWF Process. The system of equations becomes asymmetric when the Newton-Raphson procedure is applied. The system of equations also becomes asymmetric when the GNC Package is used and if the GNC terms are applied to the coefficient matrix instead of the right-hand-side vector.

If the system of equations is symmetric, its solution is more robust and more efficiently obtained by using symmetric matrix solution schemes even though asymmetric schemes may be used to find the solution. The PCGU solver of the SMS

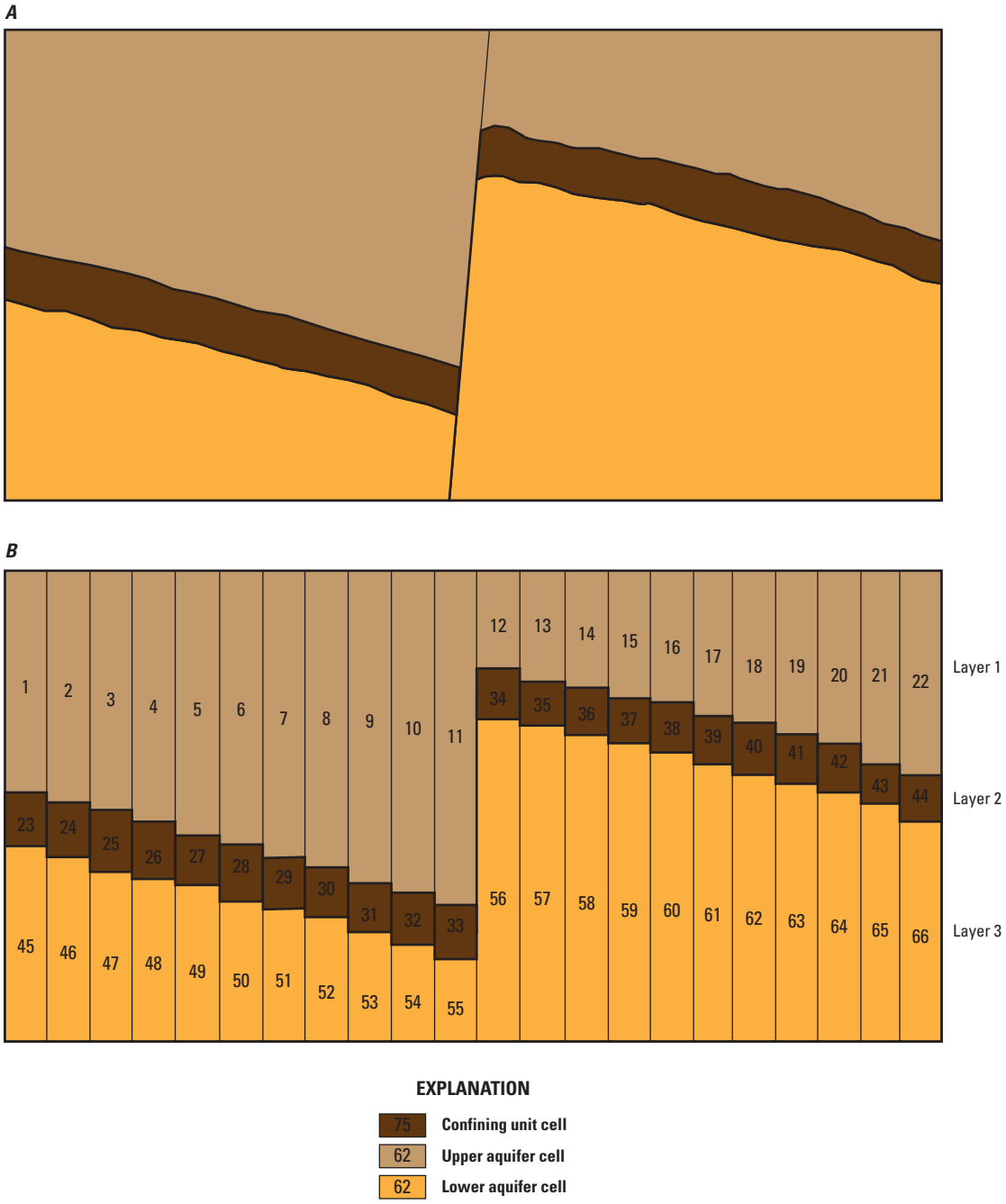

Figure 22. Cross-sectional diagrams showing *A*, offset of hydrostratigraphic layers from a fault and *B*, one example of an unstructured grid that could be used to represent the faulted system.

Package is well suited for handling the unstructured symmetric system of equations. The Conjugate Gradient option of the χMD solver also solves the unstructured symmetric system in a robust and efficient manner.

If the system of equations is asymmetric, only the asymmetric solution schemes contained in the χMD solver of the SMS Package may be used. The Orthomin and Bi-CGSTAB schemes are available for solving an asymmetric system of equations and one may be more robust than the other depending on the problem; therefore, the user is encouraged to test both during preliminary simulations to identify any problems.

If Newton-Raphson linearization is not used for a problem, the GNC terms may be applied to the right-hand-side vector to keep the matrix symmetric to enable use of the more efficient symmetric solvers. However, a right-hand-side update of the GNC terms may require more outer iterations. Again, a user is encouraged to try both combinations

(GNC term update on the right-hand-side vector with a symmetric linear solver, or left-hand-side implementation of the GNC term with an asymmetric solver) for best overall efficiency of solution.

Under-relaxation and residual control are also handled by the SMS Package. These schemes are required to control the head-change vector of nonlinear iterations using the Newton-Raphson methodology, and have been beneficial in guiding convergence for the Picard scheme as well. Residual control is typically not required for transient problems because the storage term controls the change in the residual. Under-relaxation, however, is beneficial for most unconfined problems with substantially less computational burden than residual control measures. The delta-bar-delta methodology for under-relaxation is more robust than the Cooley under-relaxation alternative, but also uses more memory. The input instructions provide guidance on selecting values for the under-relaxation and residual control parameters.

Example Problems

The following three example problems are provided to depict application of MODFLOW–USG.

Nested Grid Example

A simple example was designed to demonstrate the use of nested grids with MODFLOW–USG. The demonstration problem consists of two-dimensional, confined groundwater flow in a homogeneous aquifer. Flow is from a constant head boundary on the left having a prescribed head value of 1.0 to a constant head boundary on the right having a prescribed head value of zero. The problem is solved using a coarse grid having cell dimensions of 100 and a nested finer grid with 9 cells for every 1 coarse grid cell (fig. 23). Cells were sequentially numbered within the coarse grid first, starting in the upper left corner, and then within the finer grid, ending in the lower right corner. Cell numbering does not generally affect the performance of MODFLOW–USG; any numbering scheme could have been used for this problem. If more than one model layer were used, cells would have to be numbered sequentially starting with the top layer.

The basic package input file for this problem is shown in figure 24. The input for this model will be read using the unstructured format because a DISU filetype was provided in the Ftype list of the NAME file. The IBOUND array, which was reformatted here for presentation purposes according to the numbers of rows and columns in the coarse and nested grids, contains -1 for constant head cells and 1 for active cells. Because this is a steady-state simulation, starting heads are inconsequential except for the starting head values for the left and right constant head boundaries.

The unstructured discretization input file (fig. 25) describes the spatial discretization, which is a full description of the grid connections, and the temporal discretization, which for this problem is simply one steady-state stress period with one time step. The first record contains a number of key input variables, including the number of cells (NODES), number of layers (NLAY), and the total number of connections for the unstructured grid (NJAG). The first record also contains IVSD, which indicates how the model is discretized vertically. Because this is a one-layer model without vertical subdiscretization, an IVSD value of zero indicates there is no need to enter the IVC array in record 9. If IVSD were specified as one, then the IVC array would be required to indicate whether a connection is vertical or horizontal. The second record (LAYCBD) indicates that there are no quasi-three-dimensional confining beds in the simulation. NODELAY contains the number of cells in each layer. For this two-dimensional simulation, there is only one layer and NODELAY is equal to NODES. Records 4 and 5 indicate that the aquifer has a flat top and flat bottom with elevations equal to 0 and -100 m, respectively. Record 6 shows that the area (in plan view) for cells 1 through 40 is 10,000 m^2 (100 m)^2 and that the area for cells 41 through 121 is 1111.11 m^2 (33.333 m)^2.

Records 6 and 7 contain the IAC and JA arrays, which describe how the cells are connected to one another. These two arrays are indicative of the important differences between MODFLOW and MODFLOW–USG, and also provide the foundation for the underlying iterative solution methods in the code. The IAC array contains the number of nonzero elements in the matrix associated with a given cell and can be used to determine the number connections for each cell by subtracting 1 from each IAC value. In the JA array shown next, negative values are assigned to the cell in the diagonal; the following positive numbers are connected cells. This negative index allows easy identification of the cell connections using only the JA array, because it is obvious to see how many connections there are for each cell. MODFLOW–USG allows these negative numbers in the input file, but converts them to positive values at the start of the simulation. So for example, the first three values in the JA array are -1, 2, and 8. This means that cell 1 is connected to cells 2 and 8. From figure 23, one can also see that cell 41 is connected to cells 10, 16, 42, and 50. This can be verified in the JA array by finding the -41 cell value, and confirming that cell 41 is connected to cells 10, 16, 42, and 50. This negative value indexing is strictly a matter of convenience, and positive values could have been used instead, with each cell and its connections being identified on a separate line to facilitate readability. With MODFLOW–USG, the JA array must be symmetric, which means that if cell 1 is connected to cell 2, then cell 2 must also be connected to cell 1. Accurate specification of the cell connectivity is essential to accurately simulate groundwater flow using MODFLOW–USG.

The connection length array (CL12) is shown in Record 9. This array contains NJAG values, and its order corresponds with the order of the cells shown in the JA array. The first three values of the JA array indicate that cell 1

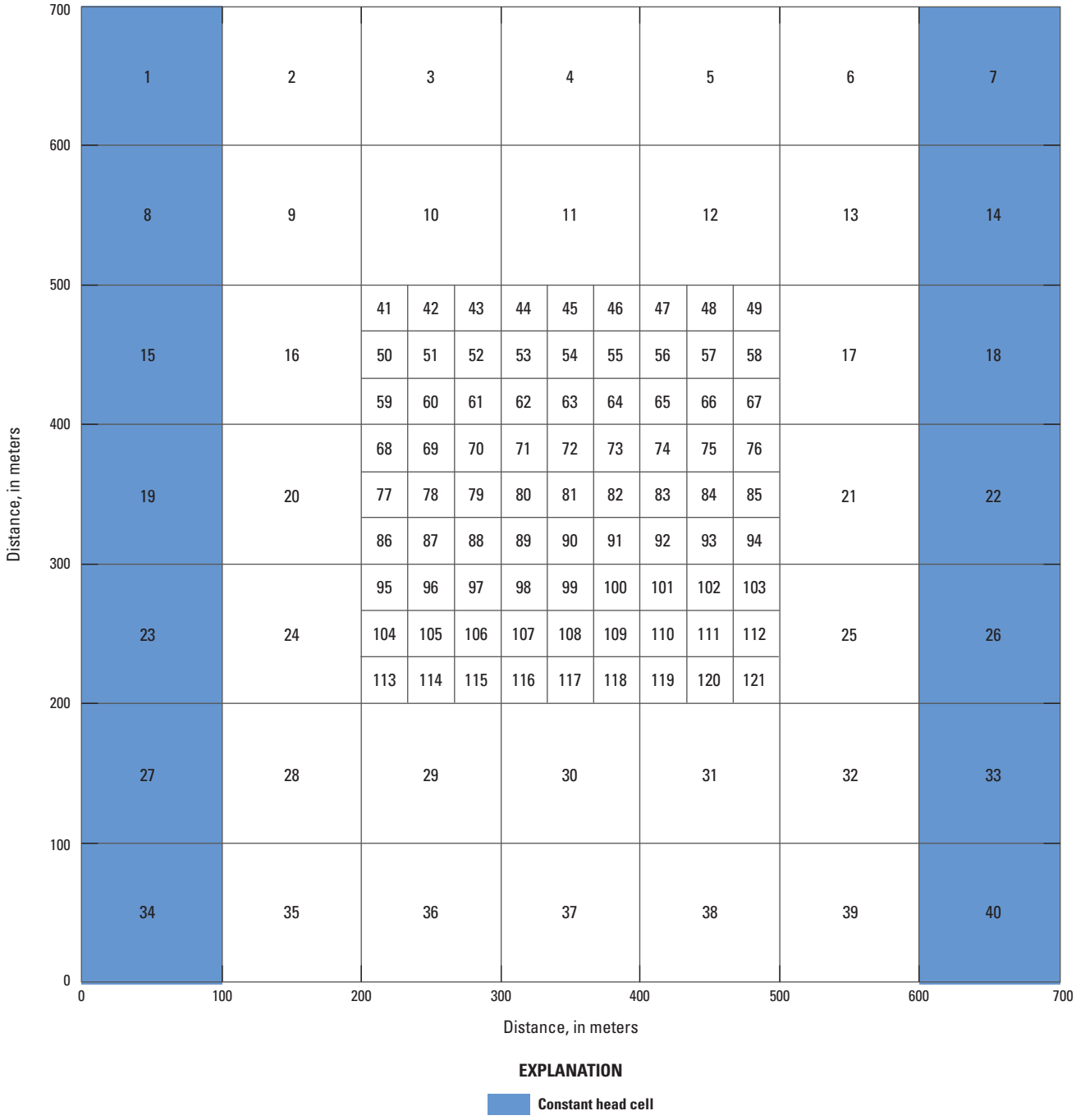

Figure 23. A MODFLOW-USG grid consisting of a finer grid nested within a coarser grid. Cell numbers are shown within each cell. Constant head boundaries are shown in blue. Grid dimensions are 700 meters in each direction.

is connected to cells 2 and 8. The connection length is the perpendicular distance from the cell center (or the ghost node location, if present) to the cell face it shares with the connected cell. The diagonal entries in this array are zero, because there is no connection length of a cell to itself. For example, the first value in the CL12 array is zero, although it could have been assigned any value because this entry has no meaning to the simulation and is not used to assemble the conductance values used in the coefficient matrix. Positions 2 and 3 of the CL12 array contain the value 50, which represents the distance from the center of cell 1 to the right face of the cell (which is connected to cell 2), and from the center of cell 1 to the lower face of cell 1 (which is connected to cell 8). These connection lengths are used by MODFLOW–USG to calculate inter-cell conductances.

```
# Basic package file for MODFLOW-USG
 FREE
 INTERNAL 1 (FREE) -1  IBOUND ARRAY
-1  1  1  1  1  1 -1
-1  1  1  1  1  1 -1
-1  1           1 -1
-1  1           1 -1
-1  1           1 -1
-1  1  1  1  1  1 -1
-1  1  1  1  1  1 -1
 1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1
-999.99
INTERNAL 1.0 (FREE) -1  STARTING HEAD ARRAY
1 0 0 0 0 0 0
1 0 0 0 0 0 0
1 0       0 0
1 0       0 0
1 0       0 0
1 0 0 0 0 0 0
1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

Figure 24. Basic (BAS6) Package input file for the nested grid problem.

Record 10 is the FAHL array, which contains the perpendicular flow area for all connections. Values in the FAHL array are either 0, 10,000, or 3,333.33. Similar to the CL12 array, a value of zero in the diagonal position is ignored by MODFLOW–USG because it is not a flow connection. A value of 10,000 corresponds to all connections in the coarse grid, whereas a value of 3333.33 corresponds to all connections for a cell in the nested grid.

The last record in the discretization input file is Record 11, which contains the length of the stress period (PERTIM), the number of time steps in the stress period (NSTP), the time step multiplier (TSMULT), and the steady-state or transient flag (SS/TR).

The input file for the Layer Property Flow (LPF) Package is shown in figure 26. This file is relatively simple and compares closely to an LPF input file for the standard version of MODFLOW. The IKCFLAG is specific to MODFLOW–USG; a value of 0 indicates that the hydraulic conductivity values specified for HK are cell hydraulic conductivities; there is also an option (IKCFLAG=1) to directly specify hydraulic conductivity for each connection, in which case the user would have to preprocess the hydraulic conductivities based on some appropriate averaging method.

The input file for the output control option of the Basic (BAS6) Package is shown in figure 27. These options result in heads and budgets being saved to binary files as well as being printed to the listing file. The output control options work the same way as they do for MODFLOW–2005, including the option for specifying output control on the basis of numeric values.

The input file for the SMS Package is shown in figure 28. The first two numbers are convergence criteria for head. The next two numbers specify the maximum number of outer iterations and the maximum number of inner iterations for the matrix solver. A NLMETH value of zero indicates that Picard iteration will be used to resolve nonlinearities in the solution; a value of one indicates that the Newton-Raphson formulation will be used. This simple model does not have convergence

```
# Discretization file for MODFLOW-USG
121 1 601 -1 1 4 2 0 0 Record 1. NODES NLAY NJAG IVSD NPER ITMUNI LENUNI IDSYMRD IPRIAJA
  0                    Record 2. LAYCBD(NLAY)
CONSTANT 121           Record 3. NODELAY(NLAY)
CONSTANT 0             Record 4. TOP(K=1)
CONSTANT -100          Record 5. BOT(K=1)
INTERNAL 1.0 (FREE) -1 Record 6. AREA(K=1)
10000   10000   10000   10000   10000   10000   10000   10000   10000   10000
10000   10000   10000   10000   10000   10000   10000   10000   10000   10000
10000   10000   10000   10000   10000   10000   10000   10000   10000   10000
10000   10000   10000   10000   10000   10000   10000   10000   10000   10000
1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11
```

Figure 25. Unstructured discretization (DISU) input file for the nested grid problem.

```
1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11
1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11
1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11
1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11
1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11
1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11
1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11 1111.11
1111.11
INTERNAL 1 (FREE) -1 Record 7. IAC(NODES)
3 4 4 4 4 4 3 4 5 7 7 7 5 4 4 7 7 4 4 7 7 4 4 7 7
4 4 5 7 7 7 5 4 3 4 4 4 4 4 3 5 5 5 5 5 5 5 5 5 5
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
INTERNAL 1 (FREE) -1 Record 8. JA(NJA)
 -1    2    8   -2    1    3    9   -3    2    4   10   -4    3    5   11   -5    4    6   12   -6
  5    7   13   -7    6   14   -8    1    9   15   -9    2    8   10   16  -10    3    9   11   41
 42   43  -11    4   10   12   44   45   46  -12    5   11   13   47   48   49  -13    6   12   14
 17  -14    7   13   18  -15    8   16   19  -16    9   15   20   41   50   59  -17   13   18   21
 49   58   67  -18   14   17   22  -19   15   20   23  -20   16   19   24   68   77   86  -21   17
 22   25   76   85   94  -22   18   21   26  -23   19   24   27  -24   20   23   28   95  104  113
-25   21   26   32  103  112  121  -26   22   25   33  -27   23   28   34  -28   24   27   29   35
-29   28   30   36  113  114  115  -30   29   31   37  116  117  118  -31   30   32   38  119  120
121  -32   25   31   33   39  -33   26   32   40  -34   27   35  -35   28   34   36  -36   29   35
 37  -37   30   36   38  -38   31   37   39  -39   32   38   40  -40   33   39  -41   10   16   42
 50  -42   10   41   43   51  -43   10   42   44   52  -44   11   43   45   53  -45   11   44   46
 54  -46   11   45   47   55  -47   12   46   48   56  -48   12   47   49   57  -49   12   17   48
 58  -50   16   41   51   59  -51   42   50   52   60  -52   43   51   53   61  -53   44   52   54
 62  -54   45   53   55   63  -55   46   54   56   64  -56   47   55   57   65  -57   48   56   58
 66  -58   17   49   57   67  -59   16   50   60   68  -60   51   59   61   69  -61   52   60   62
 70  -62   53   61   63   71  -63   54   62   64   72  -64   55   63   65   73  -65   56   64   66
 74  -66   57   65   67   75  -67   17   58   66   76  -68   20   59   69   77  -69   60   68   70
 78  -70   61   69   71   79  -71   62   70   72   80  -72   63   71   73   81  -73   64   72   74
 82  -74   65   73   75   83  -75   66   74   76   84  -76   21   67   75   85  -77   20   68   78
 86  -78   69   77   79   87  -79   70   78   80   88  -80   71   79   81   89  -81   72   80   82
 90  -82   73   81   83   91  -83   74   82   84   92  -84   75   83   85   93  -85   21   76   84
 94  -86   20   77   87   95  -87   78   86   88   96  -88   79   87   89   97  -89   80   88   90
 98  -90   81   89   91   99  -91   82   90   92  100  -92   83   91   93  101  -93   84   92   94
102  -94   21   85   93  103  -95   24   86   96  104  -96   87   95   97  105  -97   88   96   98
106  -98   89   97   99  107  -99   90   98  100  108 -100   91   99  101  109 -101   92  100  102
110 -102   93  101  103  111 -103   25   94  102  112 -104   24   95  105  113 -105   96  104  106
114 -106   97  105  107  115 -107   98  106  108  116 -108   99  107  109  117 -109  100  108  110
118 -110  101  109  111  119 -111  102  110  112  120 -112   25  103  111  121 -113   24   29  104
114 -114   29  105  113  115 -115   29  106  114  116 -116   30  107  115  117 -117   30  108  116
118 -118   30  109  117  119 -119   31  110  118  120 -120   31  111  119  121 -121   25   31  112
120
INTERNAL 1.0 (FREE) -1 Record 9. CL12(NJA)
 0 50 50  0 50 50 50  0 50 50 50  0 50 50 50 0 50 50 50  0 50 50 50  0 50 50  0 50 50 50
 0 50 50 50 50  0 50 50 50 50 50 50  0 50 50 50 50 50 50  0 50 50 50 50 50 50  0 50 50 50
50  0 50 50 50  0 50 50 50  0 50 50 50 50 50 50  0 50 50 50 50 50 50  0 50 50 50  0 50 50
50  0 50 50 50 50 50 50  0 50 50 50 50 50 50 0 50 50 50  0 50 50 50  0 50 50 50 50 50 50
 0 50 50 50 50 50 50  0 50 50 50  0 50 50 50 0 50 50 50 50  0 50 50 50 50 50 50  0 50 50
50 50 50 50  0 50 50 50 50 50 50  0 50 50 50 50  0 50 50 50  0 50 50  0 50 50 50  0 50 50
```

Figure 25. Unstructured discretization (DISU) input file for the nested grid problem.—Continued

```
50  0 50 50 50  0 50 50 50  0 50 50 50  0 50 50
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667 0 16.6667 16.6667 16.6667 16.6667
0 16.6667 16.6667 16.6667 16.6667
INTERNAL 1.0 (FREE) -1 Record 10. FAHL(NJA)
0 10000 10000  0 10000 10000 10000  0 10000 10000 10000  0 10000 10000 10000 0 10000 10000 10000 0 10000
10000 10000 0 10000 10000
0 10000 10000 10000 0 10000 10000 10000 10000
0 10000 10000 10000 3333.33 3333.33 3333.33  0 10000 10000 10000 3333.33 3333.33 3333.33
0 10000 10000 10000 3333.33 3333.33 3333.33
0 10000 10000 10000 10000  0 10000 10000 10000  0 10000 10000 10000
0 10000 10000 10000 3333.33 3333.33 3333.33  0 10000 10000 10000 3333.33 3333.33 3333.33
0 10000 10000 10000  0 10000 10000 10000
0 10000 10000 10000 3333.33 3333.33 3333.33  0 10000 10000 10000 3333.33 3333.33 3333.33
0 10000 10000 10000  0 10000 10000 10000  0 10000 10000 10000 3333.33 3333.33 3333.33
0 10000 10000 10000 3333.33 3333.33 3333.33  0 10000 10000 10000  0 10000 10000 10000
```

Figure 25. Unstructured discretization (DISU) input file for the nested grid problem.—Continued

```
0 10000 10000 10000 10000
0 10000 10000 10000 3333.33 3333.33 3333.33  0 10000 10000 10000 3333.33 3333.33 3333.33
0 10000 10000 10000 3333.33 3333.33 3333.33
0 10000 10000 10000 10000 0 10000 10000 10000  0 10000 10000  0 10000 10000 10000  0 10000 10000 10000 0
10000 10000 10000  0 10000 10000 10000
0 10000 10000 10000  0 10000 10000
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33 0 3333.33 3333.33 3333.33 3333.33
1.0 1 1.0 SS  Record 11. PERTIM, NSTP, TSMULT, TR/SS
```

Figure 25. Unstructured discretization (DISU) input file for the nested grid problem.—Continued

```
        50  -1.0e+30         0         0   Record 1. ILPFCB HDRY NPLPF IKCFLAG [OPTIONS]
 0                                         Record 2. LAYTYP(NLAY)
 0                                         Record 3. LAYAVG(NLAY)
 1                                         Record 4. CHANI(NLAY)
 1                                         Record 5. LAYVKA(NLAY)
 0                                         Record 6. LAYWET(NLAY)
 CONSTANT          1                       Record 11. HK
 CONSTANT          1                       Record 12. VKA
```

Figure 26. Layer Property Flow (LPF) Package input file for the nested grid problem.

```
# Output control package file
HEAD PRINT FORMAT 2
HEAD SAVE UNIT 30
PERIOD 1 STEP 1
    SAVE HEAD
    PRINT HEAD
    SAVE BUDGET
    PRINT BUDGET
```

Figure 27. Input file for the output control (OC) option of the Basic (BAS6) Package for the nested grid problem.

```
1e-06 1e-06 100 100 1 0 1       HCLOSE HICLOSE MXITER ITER1 IPRSMS NLMETH LINMETH
2 0 0 7 0 0.0 0 0.001000     IACL NORDER LEVEL NORTH IREDSYS RRCTOL IDROPTOL EPSRN
```

Figure 28. Sparse Matrix Solver (SMS) Package input file for the nested grid problem.

```
# Name file for MODFLOW-USG
LIST           2     ex02.list
DISU           11    ex02.disu
BAS6          13     ex02.bas
LPF           15     ex02.lpf
OC            14     ex02.oc
DATA(BINARY)  50     ex02.cbc
DATA(BINARY)  30     ex02.hds
SMS           27     ex02.sms
```

Figure 29. MODFLOW-USG name file for the nested grid problem.

```
                              MODFLOW-USG
   U.S. GEOLOGICAL SURVEY MODULAR FINITE-DIFFERENCE GROUNDWATER FLOW MODEL
                          Version 1.0.00 01/01/2013
 Using NAME file: ex02.nam
 Run start date and time (yyyy/mm/dd hh:mm:ss): 2011/06/29  7:27:27
 Solving:  Stress period:     1    Time step:     1    Ground-Water Flow Eqn.
 Run end date and time (yyyy/mm/dd hh:mm:ss): 2011/06/29  7:27:27
 Elapsed run time:  0.000 Seconds
  Normal termination of simulation
```

Figure 30. Console window showing MODFLOW-USG output information.

problems with cells converting between wet and dry, but if it did, the Newton-Raphson formulation with upstream weighting could be used to improve convergence. The LINMETH value of 1 indicates that the χMD matrix solver will be used. The second line in the SMS input file contains input parameters specific to the χMD matrix solver. The value of 2 for IACL indicates that Bi-CGSTAB will be used for the acceleration step of the solution; the conjugate gradient and orthomin acceleration methods could also have been used for this problem and the results would be approximately the same. The remaining input parameters on the second line of the SMS Package are settings that typically work well for small easily solved problems. The name file for this problem is shown in figure 29. The name of this file is provided to MODFLOW–USG at the start of the simulation. This problem should take less than a second to run on most computers. Upon completion, users should see the information in the console window shown in figure 30.

Using Darcy's Law and the input parameters for this problem, the correct groundwater flow rate through the

```
VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF TIME STEP       1 IN STRESS PERIOD       1
 ---------------------------------------------------------------------------------------
    CUMULATIVE VOLUMES      L**3       RATES FOR THIS TIME STEP      L**3/T
    ------------------                 ------------------------

          IN:                                   IN:
          ---                                   ---
            STORAGE =           0.0000              STORAGE =           0.0000
      CONSTANT HEAD =         118.2356        CONSTANT HEAD =         118.2356
           TOTAL IN =         118.2356             TOTAL IN =         118.2356
         OUT:                                  OUT:
         ----                                  ----
            STORAGE =           0.0000              STORAGE =           0.0000
      CONSTANT HEAD =         118.2356        CONSTANT HEAD =         118.2356
          TOTAL OUT =         118.2356            TOTAL OUT =         118.2356
           IN - OUT =       4.8504E-05             IN - OUT =       4.8504E-05
 PERCENT DISCREPANCY =           0.00     PERCENT DISCREPANCY =           0.00
```

Figure 31. Volumetric budget excerpt from the MODFLOW–USG listing file.

aquifer system is 116.67 m^3/d. The volumetric budget from the MODFLOW–USG listing file is shown in figure 31. The numbers on the right side of the budget table are groundwater flow rates and have dimensions of L^3/T. As shown in the listing file, the total flow through the system is 118.2356 m^3/d. This means that the groundwater flow rate calculated by MODFLOW–USG is in error by about 1.3 percent; however, the percent discrepancy shown in the listing file has a value of 0.00. This error is caused by the structure of the nested grid.

Based on the characteristics of this problem, the simulated head field should be a plane that passes through the constant head values on the left and right side of the model. For each cell, the error in the simulated head was calculated by subtracting the correct known value from the value simulated by MODFLOW–USG. These errors are shown in figure 32 and clearly identify the issue with using the standard uncorrected CVFD formulation with nested grids. The standard CVFD method requires that a line drawn between the centers of any two connected cells should bisect the shared edge at a right angle. Inspection of figure 32 shows that this is not the case along the boundary of the nested grid. For example, a line drawn between the center of cell 16 and the center of cell 41 does not cross the shared face at a right angle. The consequences, which are particularly obvious for this problem of one-dimensional flow, are errors in the simulated aquifer flow rates and simulated heads. Although the errors in simulated head are relatively small, the error is orders of magnitude larger than the head convergence criterion of 1×10^{-6} m.

The errors just described can be reduced by using the GNC Package. Figure 32 helps clarify the calculations provided by the GNC Package. The groundwater flux calculated between cell 11 and cell 44, for example, should not be calculated using the head at cell 11, but rather some interpolated head between cell 11 and cell 10. In this case, where the aquifer is confined and homogeneous, the appropriate head to use in the flow calculation would be one that is weighted two-thirds by the head at cell 11 and one-third by the head at cell 10. Figure 33 shows the GNC Package input file developed for this problem. The sixth line of this file contains the ghost node information for the flux calculation between cell 11 and cell 44. The first number (11) is the number of the cell for which head interpolation is required; the second number (44) is the number of the connected cell; the third number (10) denotes the cell used for the interpolation, and the fourth number (1/3) is the contributing (or interpolation) factor from this adjacent node (10). There is no need for ghost node information for connections such as that between 10 and 42, because the line connecting their centers does bisect the shared face at a right angle.

By adding this input file to the name file and running the simulation, the total groundwater flux through the system is calculated correctly as 116.6666 m^3/d, as shown in the volumetric water balance (fig. 34). The error in the simulated head was also recalculated and is shown in figure 35. The head errors are less than the head convergence criterion value of 1×10^{-6} m.

Quadtree Refinement

Quadtree refinement is a straightforward way to focus resolution in areas of interest. It has the appealing characteristic that resolution can be increased along lines and within polygons, as necessary, to better represent hydraulic gradients around hydraulically important features or to more accurately represent known variations in hydraulic properties or boundary conditions. Use of quadtree refinement is also an intuitive way for MODFLOW users to transition to MODFLOW–USG simulations, because the approach uses familiar rectangular cells. Quadtree refinement works well when the areas of interest are scattered throughout the model domain. An existing MODFLOW–2005 model of the Biscayne aquifer in southern Florida (Brakefield and others, in press) is

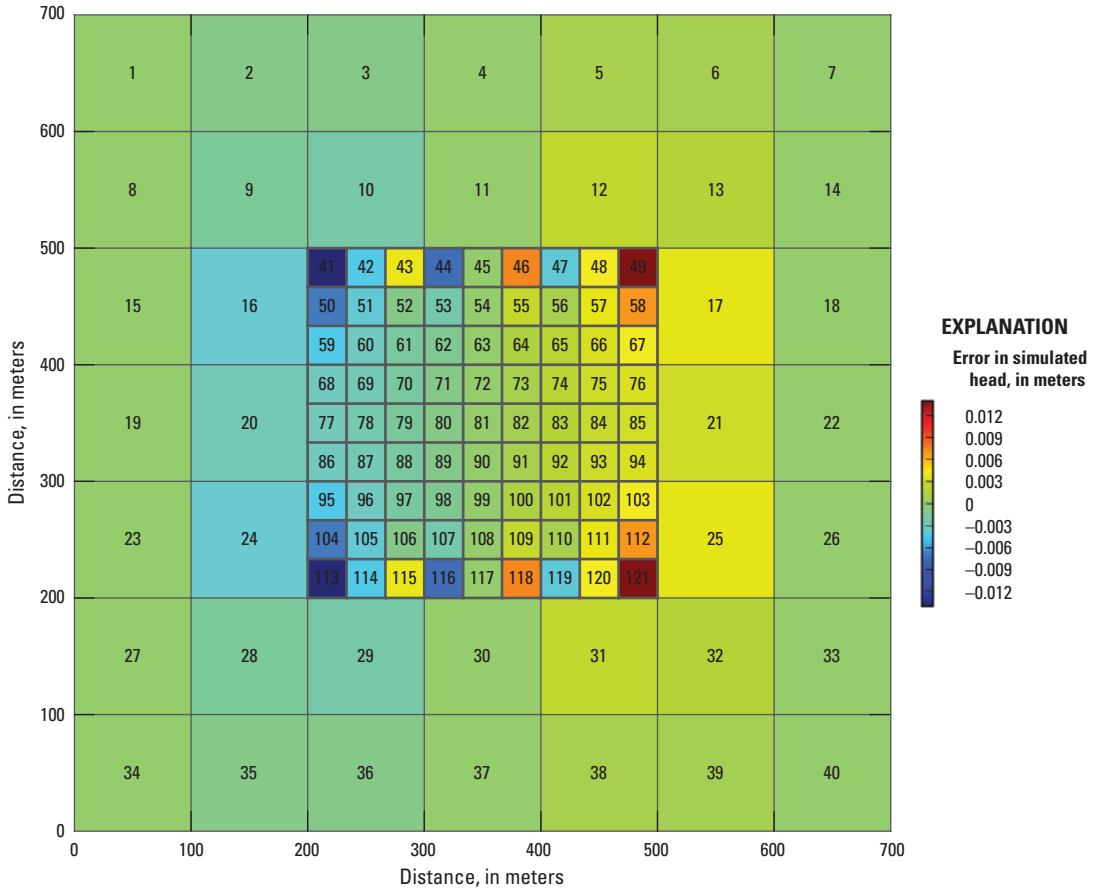

Figure 32. Errors in simulated heads using the standard finite-volume formulation without ghost node corrections.

used here to demonstrate the use of MODFLOW–USG with quadtree refinement for a large and realistic problem.

The Biscayne aquifer model of Brakefield and others (in press) uses a 50-m resolution grid consisting of 1,730 rows and 930 columns. The model has one unconfined layer, with the aquifer bottom set to the base of the Biscayne aquifer. The model has 3,248 daily stress periods, which start on January 1, 1996, and extend through November 21, 2004. The active part of the model is shown in figure 36. A time-varying constant head boundary was used to represent the Biscayne Bay marine estuary on the eastern side of the model. Brakefield and others (in press) created a spatially variable hydraulic conductivity field by kriging the results from aquifer performance tests in the study area. Specific yield was specified as 0.2 everywhere. The extensive canal network was represented using the RIV Package with spatially and temporally varying stage values. Stage, conductance, and river bottom were assigned based on measured values, lengths of the canal within each grid cell, and the hydraulic conductivity of the grid cell. Temporally varying withdrawals at municipal well fields were represented using the WEL Package. Aquifer recharge and evapotranspiration were represented using the RCH and EVT Packages, respectively.

The 50-m resolution MODFLOW model of Brakefield and others (in press) was recreated from original data, as was the MODFLOW–USG quadtree model, to ensure that intersections of hydrologic features (canals, coastline, and municipal wells) with the structured 50-m grid and the unstructured quadtree grid followed a consistent approach. In some instances, there was no way to exactly reproduce, in an automated way, characteristics of the Brakefield and others (in press) model. Some additional simplifications were also made. For example, the models created here are limited to 1,000 stress periods, and the constant head boundary used along the coastline was fixed at a value of zero for the entire simulation instead of varying by stress period. Consequently, the models presented here contain differences from the model described by Brakefield and others (in press) and, thus, are presented here only for demonstration purposes.

The MODFLOW–USG quadtree grid uses an underlying 800-m resolution structured grid that covers the same domain as the 50-m grid. This 800-m grid represents the zero level of refinement from which the locations of hydrologic features are used to further divide individual cells. Within a 1,000-m radius of each municipal withdrawal well, the grid was refined down to 4 levels, which corresponds to a 50-m cell size—the same

```
# Ghost node file for MODFLOW-USG, generated by Flopy.
         0         0        24         1         0
0         0
16 41 9  0.333333333333
10 41 9  0.333333333333
10 43 11  0.333333333333
11 44 10  0.333333333333
11 46 12  0.333333333333
12 47 11  0.333333333333
17 49 13  0.333333333333
12 49 13  0.333333333333
16 59 20  0.333333333333
17 67 21  0.333333333333
20 68 16  0.333333333333
21 76 17  0.333333333333
20 86 24  0.333333333333
21 94 25  0.333333333333
24 95 20  0.333333333333
25 103 21  0.333333333333
24 113 28  0.333333333333
29 113 28  0.333333333333
29 115 30  0.333333333333
30 116 29  0.333333333333
30 118 31  0.333333333333
31 119 30  0.333333333333
25 121 32  0.333333333333
31 121 32  0.333333333333
```

Figure 33. Ghost Node Correction (GNC) Package input file.

```
 VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF TIME STEP       1 IN STRESS PERIOD       1
 ---------------------------------------------------------------------------------------
    CUMULATIVE VOLUMES      L**3       RATES FOR THIS TIME STEP      L**3/T
    ------------------                 ------------------------

          IN:                                  IN:
          ---                                  ---
            STORAGE =           0.0000              STORAGE =           0.0000
      CONSTANT HEAD =         116.6666        CONSTANT HEAD =         116.6666
           TOTAL IN =         116.6666             TOTAL IN =         116.6666
         OUT:                                 OUT:
         ----                                 ----
            STORAGE =           0.0000              STORAGE =           0.0000
      CONSTANT HEAD =         116.6666        CONSTANT HEAD =         116.6666
          TOTAL OUT =         116.6666            TOTAL OUT =         116.6666
           IN - OUT =       1.4402E-05             IN - OUT =       1.4402E-05
PERCENT DISCREPANCY =           0.00     PERCENT DISCREPANCY =           0.00
```

Figure 34. Volumetric water balance.

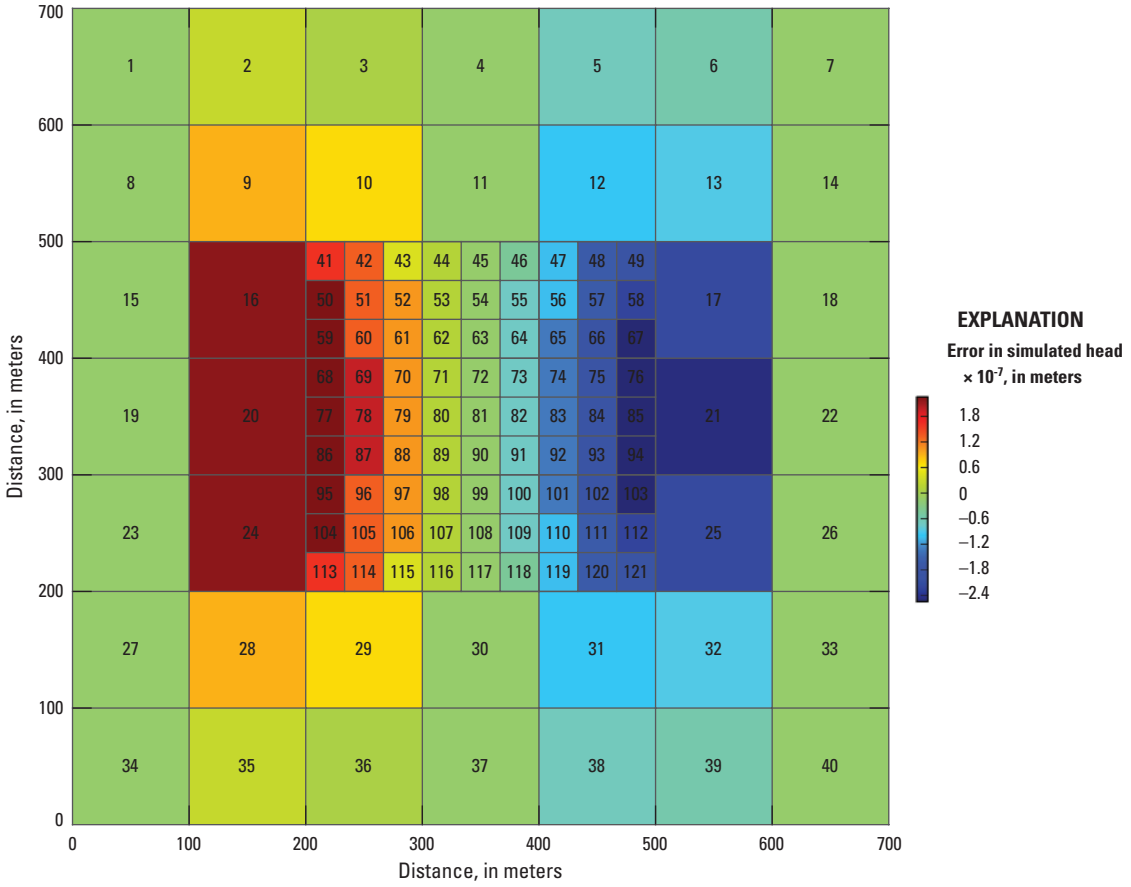

Figure 35. Errors in simulated heads from the simulation with the ghost node correction.

resolution as the MODFLOW–2005 model. Along each canal and along the coastline, the grid was also refined down to 4 levels. The grid was then smoothed, so that the level difference between any two connected cells did not exceed a value of 1; this means that a cell would be connected to no more than two cells in any single direction. The quadtree grid is shown in figure 37. Values for hydraulic conductivity, the evapotranspiration surface in the EVT Package, and aquifer bottom were calculated for each quadtree cell by averaging the values from the 50-m structured grid model.

A comparison between MODFLOW–USG and MODFLOW–2005 for the Biscayne aquifer simulations highlights some of the advantages of using the unstructured grid approach for this particular problem. Plots of simulated groundwater head for the last stress period show similar head patterns for the two different models (fig. 38). Results from MODFLOW–USG are less refined than the results from MODFLOW–2005 in some areas of the domain, but along canals, the coastline, and within the well fields, the heads simulated by MODFLOW–USG are in close agreement with MODFLOW–2005 results. Figure 38 illustrates the capability of MODFLOW–USG to focus resolution in areas with relatively large hydraulic gradients. A comparison of the cumulative water budget for the entire simulation period is shown in figure 39. In general, the individual water budget components compare well between the two models. The largest percent difference is for river leakage into the model. MODFLOW–USG simulates a cumulative leakage value that is about 11 percent greater than the value simulated by MODFLOW–2005.

Several important differences between the MODFLOW–2005 and MODFLOW–USG simulations are listed in table 2. The MODFLOW–2005 simulation uses about 15 times more cells than the MODFLOW–USG simulation and it takes about 3.5 times longer to run to completion. The MODFLOW–USG simulations were solved using the PCGU solver with head (0.0001 m) and residual (100,000 m^3/d) convergence tolerances set equal to the tolerances specified for the MODFLOW–2005 simulation. MODFLOW–USG simulations were also performed with and without ghost node corrections. For some larger cells having multiple connections in each direction, the ghost node corrections had a large effect on the simulated heads (up to 0.4 m), but for most cells, the simulated heads with the correction were within about 0.05 m of the simulated heads without the ghost node corrections.

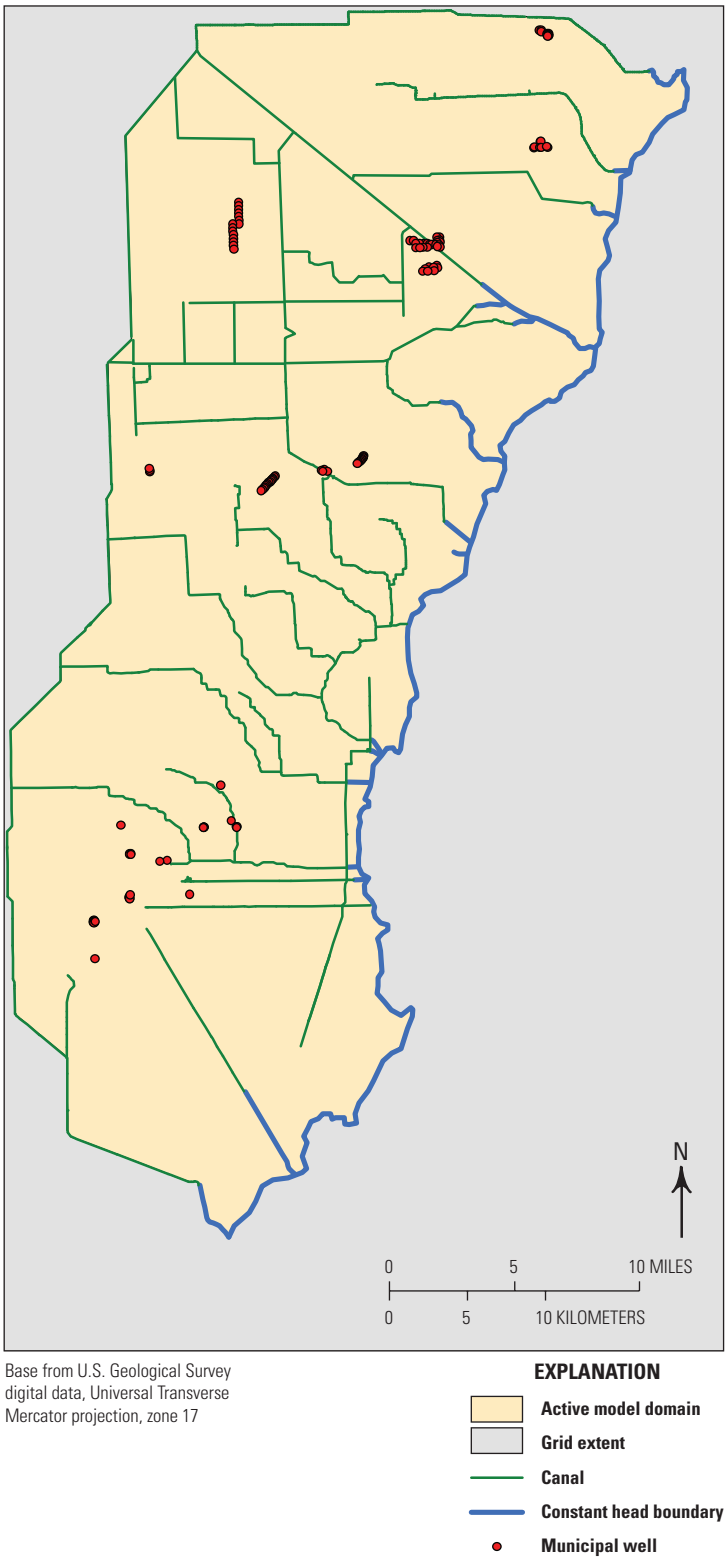

Figure 36. Hydrologic features included in the Biscayne aquifer model.

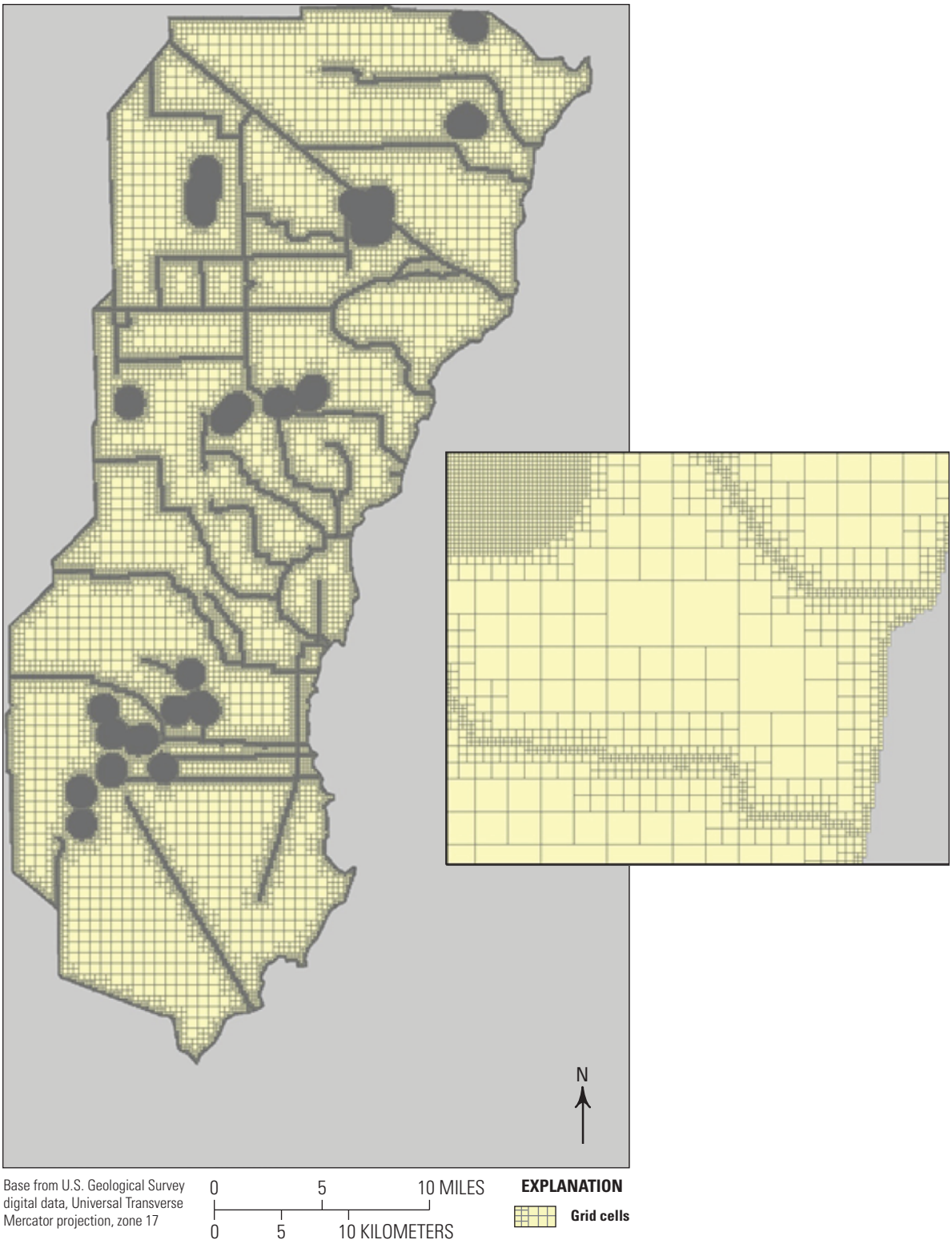

Figure 37. Quadtree grid for the Biscayne aquifer example problem. The grid is based on an 800-meter (m) structured grid. Within 1,000 m of a municipal well and along canals and the coastline, cells are refined down 4 levels to a cell size of 50 m. The quadtree grid was then smoothed so that every cell is connected to no more than two cells in any direction.

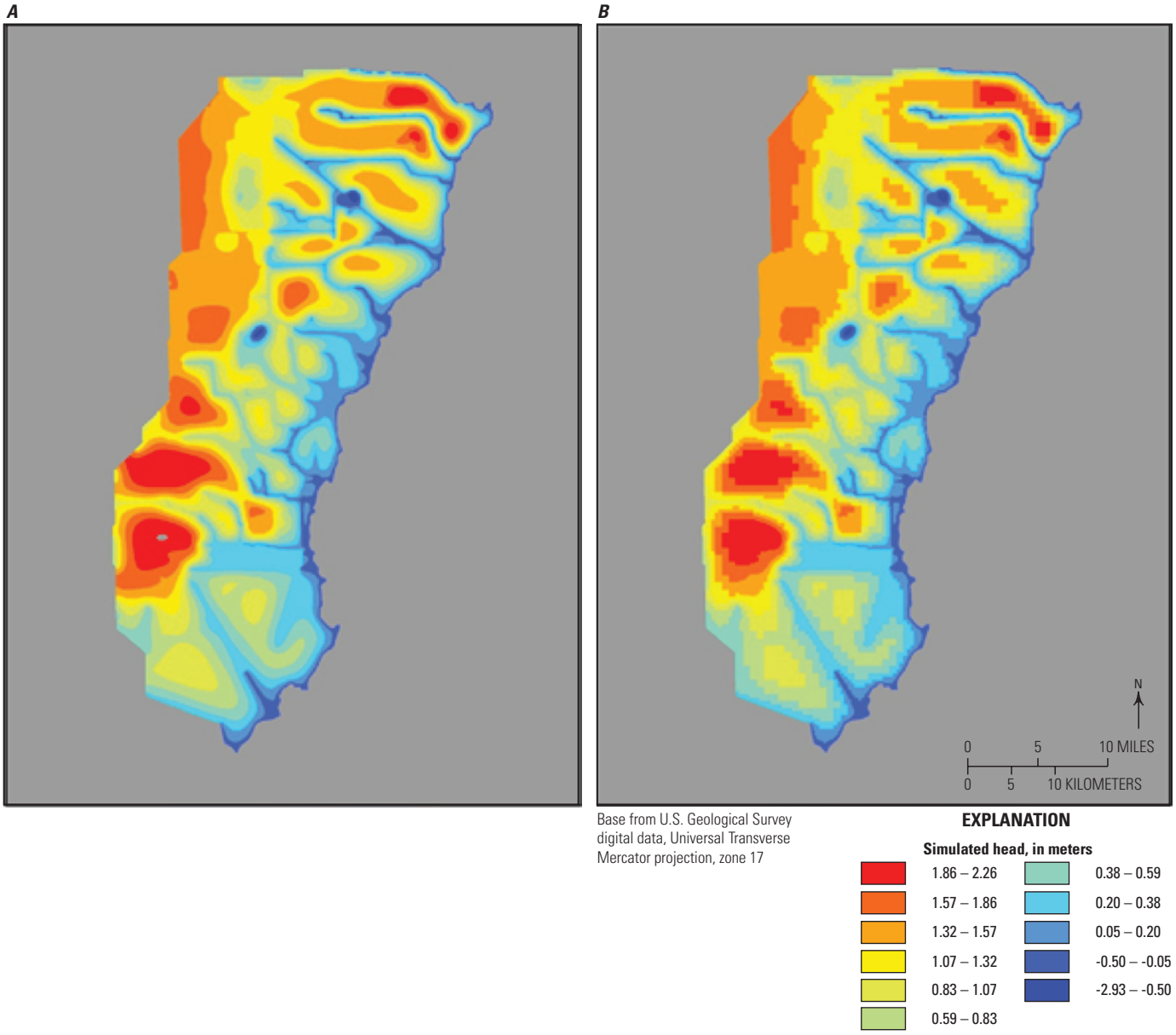

Figure 38. Simulated head values from stress period 1,000 using *A*, MODFLOW-2005 with a 50-meter regularly spaced structured grid, and *B*, MODFLOW-USG using a quadtree grid.

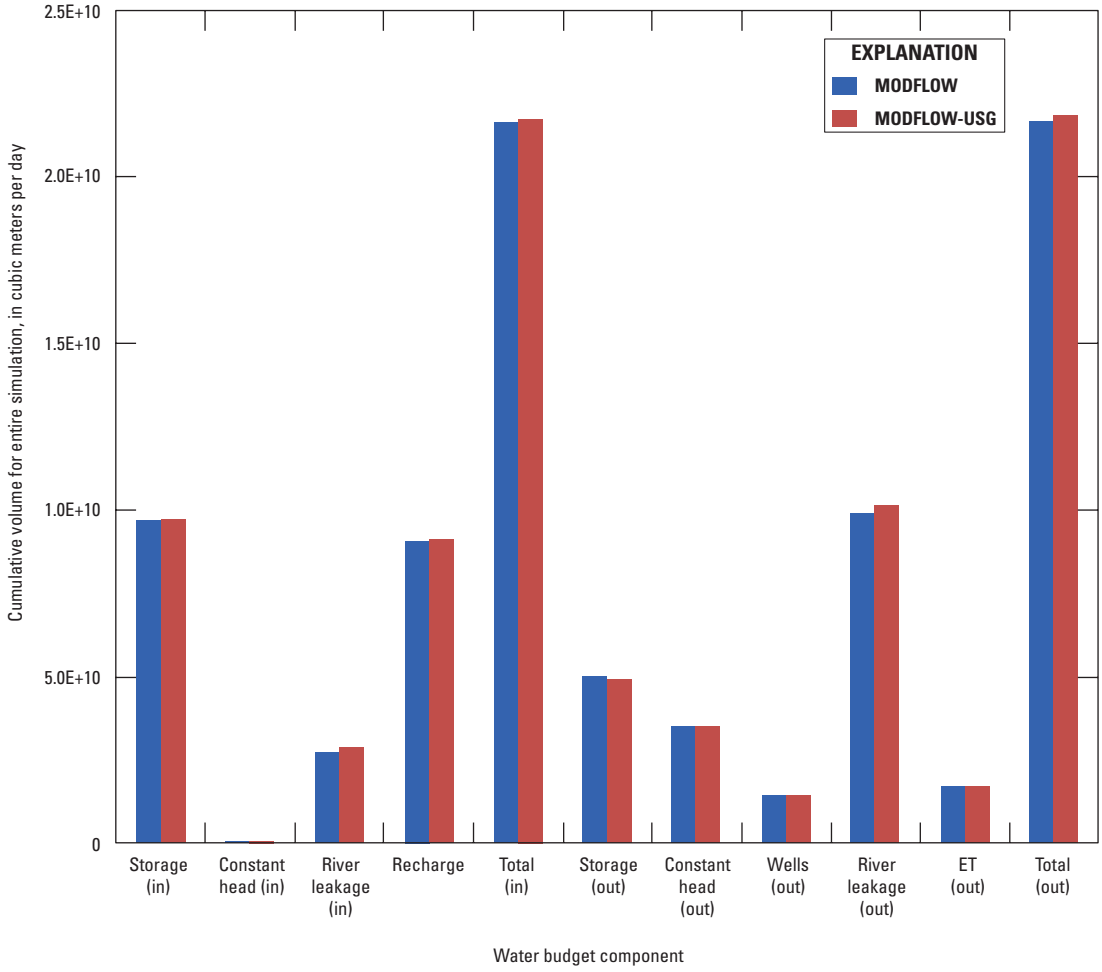

Figure 39. Cumulative water budget for the entire simulation of the Biscayne aquifer example problem.

Table 2. Summary of differences between the MODFLOW-2005 and MODFLOW-USG simulation for the Biscayne aquifer example problem.

[m, meter; s, second]

Simulation characteristic	MODFLOW-2005	MODFLOW-USG
Number of cells	1,608,900	106,257
Minimum cell size (m)	50	50
Runtime (s)	2,400	654
Computer memory (kilobytes)	202.3	13.1

```
# Basic package file for MODFLOW-USG
# Conduit flow problem
         0         1(25I3)                        -1     IBOUND Layer   1
         0         1(25I3)                        -1     IBOUND Layer   2
9.9900e+02
         0    10.000(10e12.4)                     -1     STARTING HEADS Layer   1
         0    30.000(10e12.4)                     -1     STARTING HEADS Layer   2
```

Figure 40. Basic (BAS6) Package input file for the conduit flow problem.

```
# Discretization package file for MODFLOW-USG
# Conduit flow problem
2  100  100  1  4  0
 1 0
         0   470.000(10E12.4)                      0     DELTA-X or C
         0   470.000(10E12.4)                      0     DELTA-Y or R
         0  -100.000(10e12.4)                     -1     TOP of Model
         0  -110.000(10e12.4)                     -1     BOTTOM of Layer   1
         0  -120.000(10e12.4)                     -1     BOTTOM of Confining Layer   1
         0  -130.000(10e12.4)                     -1     BOTTOM of Layer   2
  1.400000  30  1.000000  TR
```

Figure 41. Structured discretization (DIS) input file for the nested grid problem.

Conduit Flow Problem

This example problem demonstrates use of the CLN Process for simulating flow to a well pumping from two aquifers separated by an impermeable confining unit. A structured finite-difference grid was used for this problem to simplify discretization. Therefore, the BAS Package and DIS input files are the same as those used by a MODFLOW–2005 simulation. The lateral extent of the square domain is 47,000 m × 47,000 m, and the lateral edges of both aquifers are no-flow boundary conditions. Initial heads are 10 m in aquifer 1 and 30 m in aquifer 2. The BAS Package for this problem is reproduced in figure 40.

The domain is discretized into 100 rows and 100 columns using square cells, each 470 m by 470 m in extent. The elevation of the top of layer 1 is -100 m, the bottom of layer 1 is -110 m, the top of layer 2 is -120 m and the bottom of layer 2 is -130 m. The confining unit between layers 1 and 2 is impermeable and is represented using a quasi-three-dimensional approach. With this configuration, the only way a stress from one aquifer can be propagated to another is through the borehole that penetrates both aquifers. The DIS input file for this problem is reproduced in figure 41.

The hydraulic conductivity values of the upper and lower aquifers are 100 and 400 m/d, respectively. Both aquifers have a primary storage coefficient of 0.0001 and a specific yield of 0.01; this example problem is for a confined aquifer; therefore, specific yield is not used in the storage calculations even though it is specified. The confining unit is assumed to be impermeable. The BCF Package was used to parameterize the system, as reproduced in figure 42.

A vertical conduit well is located at the center of the domain and has a radius of 0.5 m. The well pumps 62,840 m^3/d and is open fully to both aquifers from top to bottom. The CLN Process was used with a circular conduit geometry type to discretize the well bore with two conduit cells, one in each layer. The WEL Package was used to pump from the bottom CLN cell. The problem could also have been solved using only one CLN conduit cell to represent the well connecting both aquifer layers, yielding the same results. The Thiem solution is applied to represent flow between the well and the aquifer layers. The CLN Process input for this problem is reproduced in figure 43. Output options for the CLN domain are set in the CLN input file to print the heads and flux through the conduit between layers 1 and 2.

The WEL Package input for this problem is reproduced in figure 44. With a structured grid input, the WEL Package has been extended from MODFLOW–2005 to accommodate pumping of conduit cells.

The input file for the output control option of the Basic (BAS6) Package is shown in figure 45. Numeric control flags are used for this problem. These options result in groundwater heads and budgets being saved to binary files but not printed to the listing file. As mentioned earlier, the output control options work the same way as they do for MODFLOW–2005.

The input file for the SMS Package is the same as shown in the first example problem. The name file for this problem is shown in figure 46. The name of this file is provided to MODFLOW–USG at the start of the simulation.

```
# BCF package file for MODFLOW-USG
# Conduit flow problem
        50 -1.000e+30         0     1.000         5         0         0         0
0         0         0         0         1
0404
         0     1.000(10E12.4)                    0     ANISOTROPY
         0  1.000e-4(10e12.4)                   -1     PRIMARY STORAGE Layer   1
         0  1.000e+2(10e12.4)                   -1     HYDRAULIC CONDUCTIVITY Layer   1
         0  0.000e00(10e12.4)                   -1     LEAKANCE Layer   1
         0  1.000e-2(10e12.4)                   -1     SECONDARY STORAGE Layer   1
         0  1.000e-4(10e12.4)                   -1     PRIMARY STORAGE Layer   2
         0  4.000e+2(10e12.4)                   -1     HYDRAULIC CONDUCTIVITY Layer   2
         0  1.000e-2(10e12.4)                   -1     SECONDARY STORAGE Layer   2
```

Figure 42. Block-Centered Flow (BCF) Package input file for the conduit flow problem.

```
# Connected Linear Network Process file for MODFLOW-USG
# Conduit flow problem
         1        -1       -35        36         0         0         2         1
         0         2                                    CLN cells per segment
         1         1         0      10.0     -110.      1.57
         2         1         0      10.0     -130.      1.57
         1         1        50        50         0         0      10.0       1.0
         2         2        50        50         0         0      10.0       1.0
         1       0.5   3.23e10                          Conduit geometry and Ksat
     0         1                              -1       ibound of CLN cells
    13     1.000(10e12.4)                    -1       initial CLN heads
    10.000       30.00
```

Figure 43. Connected Linear Network (CLN) Process input file for the conduit flow problem.

```
# WEL package file for MODFLOW-USG
# Conduit flow problem
         1        54         0      MXACTW IWELCB IWELQV
         0         0         1      Stress Period 1 - one well at CLN cell
         2   -62840.                CLN cell #2 pumps 62,840 cu-ft/d
```

Figure 44. Well (WEL) Package input file for conduit flow problem.

```
         0         0        30        31
         0         1         1         1  Stress Period 1, Time Step 1
         0         0         1         1
         0         1         1         1  Stress Period 1, Time Step 2
         0         0         1         1
… (removed records for time steps 3 to 29)
         0         1         1         1  Stress Period 1, Time Step 30
         0         0         1         1
```

Figure 45. Input file for the output control (OC) option of the Basic (BAS6) Package for the conduit flow problem.

```
LIST 7 ex3.lst
BAS6  1 ex3.bas
WEL  14 ex3.WEL
DIS  12 ex3.dis
BCF6 11 ex3.bcf
CLN 13 ex3.CLN
OC  22 ex3.oc
SMS 19 ex3.SMS
DATA(BINARY) 50 ex3.cbb
DATA(BINARY) 30 ex3.hds
DATA(BINARY) 31 ex3.ddn
DATA(BINARY) 35 ex3.cdc
DATA(BINARY) 36 ex3.cdh
```

Figure 46. MODFLOW-USG name file for the conduit flow problem.

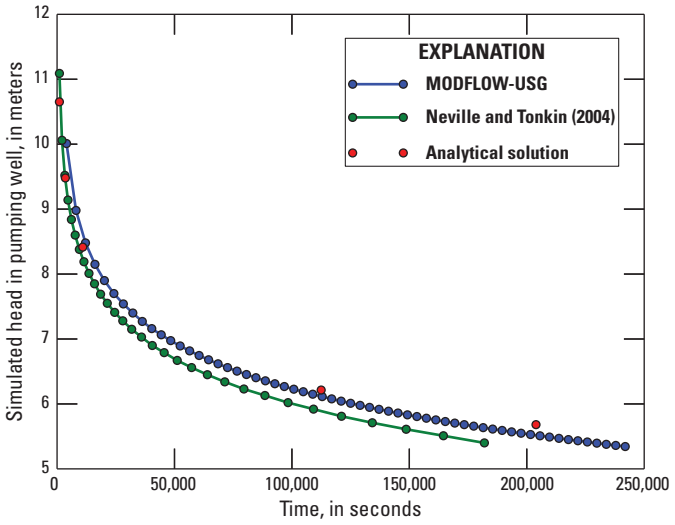

Figure 47. Simulated head in the pumping well.

After running MODFLOW–USG with this problem, the output includes an ASCII listing file, EX3.LST, which contains all the simulation information. Output binary files include the head in the aquifer reflected in file EX3.HDS, drawdown in the aquifer reflected in file EX3.DDN, and the cell-by-cell fluxes reflected in file EX3.CBB. Because a structured grid input was used, these binary files are of the same format as for MODFLOW–2005 and can be read by any standard MODFLOW postprocessor. EX3.CDC and EX3.CDH contain the flows and heads simulated by the CLN Process.

This example problem is equivalent to the one solved using the multi-aquifer well package described by Neville and Tonkin (2004). Figure 47 shows the simulated head in the pumping well and figure 48 shows the net flow and the individual aquifer flows through time for this simulation case, compared to the analytical solution and to the results of Neville and Tonkin (2004). Results from the CLN Process simulation of MODFLOW–USG are comparable to the multi-aquifer well simulation and to the analytical solution (Chris Neville and Matt Tonkin, S.S. Papadopulos & Associates, Inc., written commun., 2011). The example also demonstrates use of a conduit to simulate multi-aquifer pumping wells.

An unconfined example was also simulated with this problem setup to demonstrate the capabilities and options of the CLN Process. Specifically, and following the options provided for the GWF Process, the CLN Process also includes options for confined flow and for flow to convertible cells using the equations (33 to 40), discussed previously. Use of these options is demonstrated here.

The same simulation setup described earlier is used for this unconfined case, however, the aquifer top and bottom elevations are adjusted to produce unconfined conditions. The elevation of the top of layer 1 is 10 m, the bottom of layer 1 is 0 m, the top of layer 2 is -10 m, and the bottom of layer 2 is -20 m. The initial heads in the upper and lower aquifer are at elevations of 10 and 30 m, respectively, as per the confined case. Four cases were evaluated in terms of discretization and simulation options for the CLN Process (fig. 49). For the first case (fig. 32*A*), the CLN domain is discretized using two vertical conduit cells, one in each layer, as was done for the confined situation discussed earlier. Figure 49*A* also depicts the behavior of unconfined flow in the conduit when the CLN cell in layer 1 becomes dry. For the second case (fig. 49*B*), only one vertical CLN cell is used to discretize the well. This is conceptually equivalent to the MNW methodology and does not solve for flow within the well. The behavior of unconfined flow between the well and layer 1 is also noted in figure 49*B* to be different from that of figure 49*A* in that the flow from layer 1 to the well neglects the dry-cell condition whereby head in the well is below the bottom of layer 1. This is similar to the case of figure 49*C* where the well is discretized using two conduit geometry CLN cells but with the confined option for flow within the conduit and between CLN and GWF domains. The fourth case shown in figure 49*D* uses only one CLN cell to discretize the well but includes the "flow-to-dry-cell" option to limit flow from any layer (layer 1 in this case) when the head in the CLN cell is below the bottom of the layer.

Figure 50 shows the water levels in the pumping well, and figure 51 shows the individual aquifer discharges and the total well discharge through time for the four simulation cases. The total flow from the well is noted to decrease below the prescribed pumping level when the water level in the well drops to the bottom of the lower aquifer for all simulation cases. In addition, the results for cases A and D are similar, and those for cases B and C are similar. Furthermore, flow from layer 1 is higher for cases B and C than for cases A and D because they do not consider effects of downstream head being below the bottom of the layer; the impact on layer 2 and on the total well flow is not as large. These examples, therefore, demonstrate the implications of the selected options. The examples also demonstrate that multiple GWF cells may be connected to the same CLN cell, and it may not be necessary

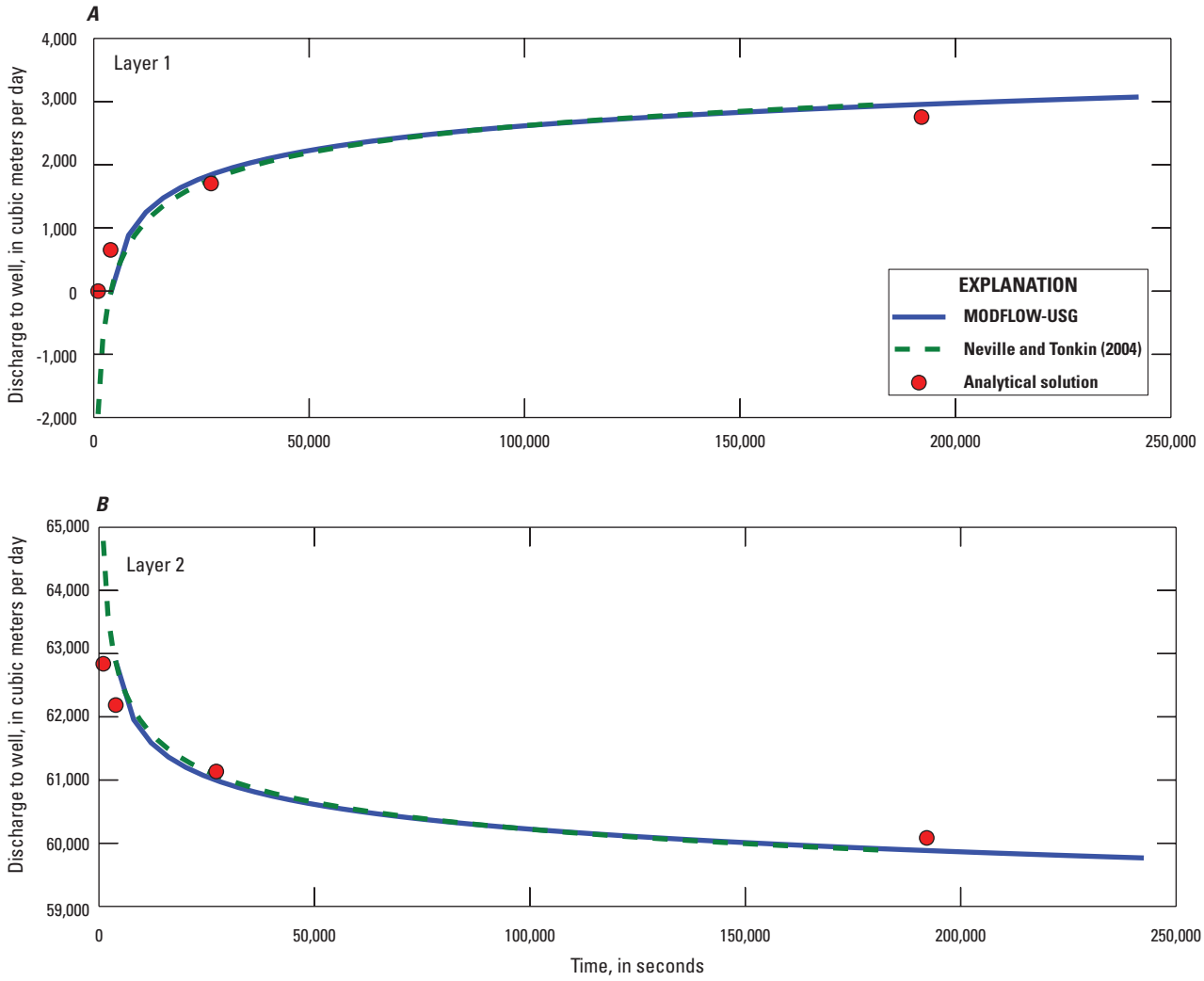

Figure 48. Simulated flow to the well from each model layer.

to subdiscretize the CLN domain to correspond with GWF layering. In that regard, the CLN domain can and should have its discretization governed by the scale of flow processes within the CLN domain, independent of the scale of flow processes of the GWF domain.

Summary

This report describes a new, unstructured grid version of MODFLOW called MODFLOW–USG. MODFLOW–USG is based on a CVFD formulation that provides new flexibility in grid design, including the capability to use nonrectangular cell shapes such as irregular polygons, hexagons, and triangles. MODFLOW–USG can be used with simple grids as well as nested grids that allow for resolution to be focused in areas of interest. A new GNC Package was developed for MODFLOW–USG to reduce errors that stem from the CVFD formulation when a line connecting adjacent cell centers does not bisect the shared face at right angles. This report also describes the CLN Process for MODFLOW–USG. The CLN Process can be used to simulate a connected set of karst conduits, and when paired with the WEL Package, can be used to simulate multi-node wells. MODFLOW–USG also includes a robust Newton-Raphson formulation for resolving the nonlinearities that occur in observed water table conditions and some boundary packages. The features and capabilities of the MODFLOW–USG program are demonstrated using three example problems.

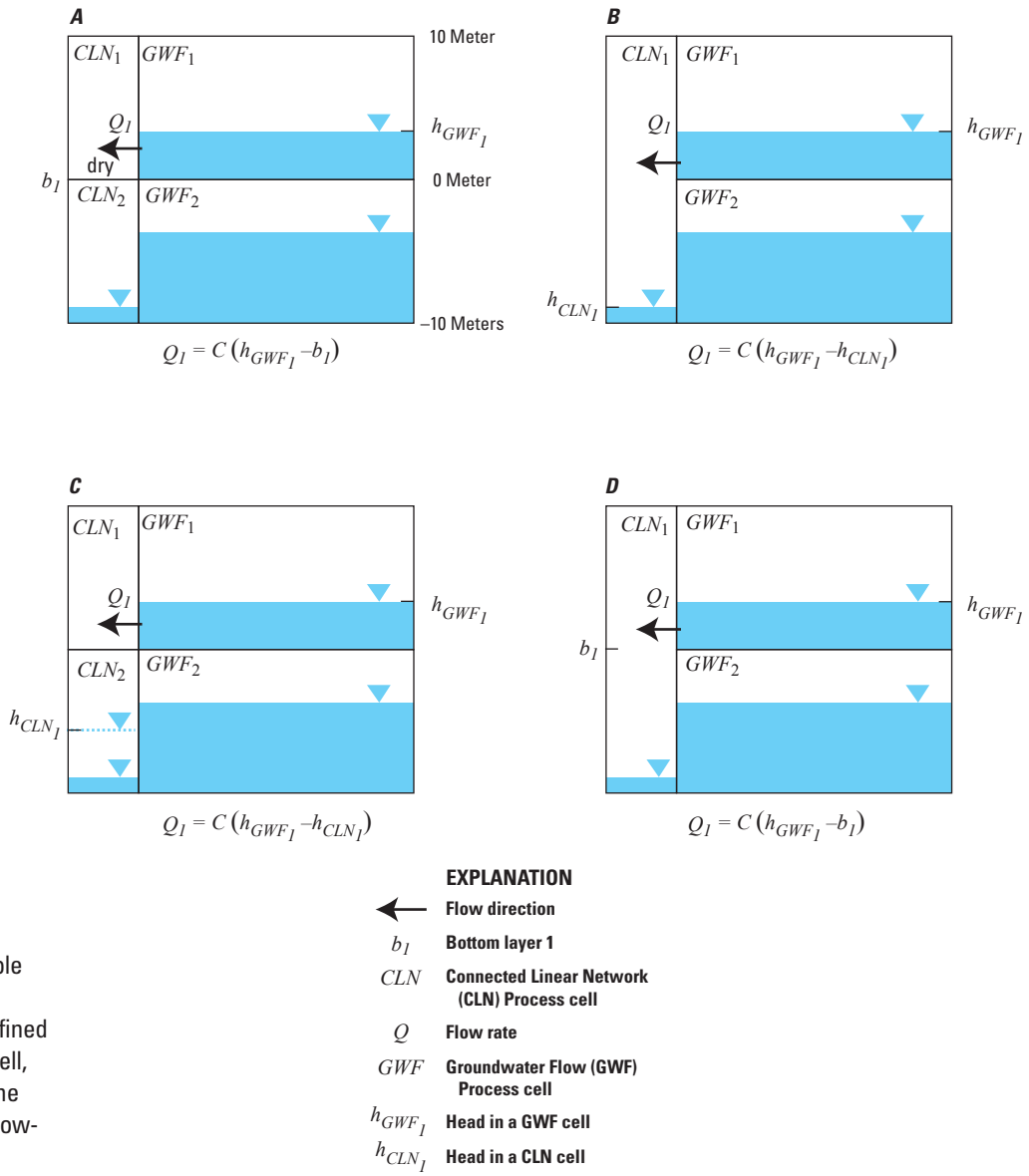

Figure 49. Unconfined flow example demonstrating discretization and simulation options for *A*, two unconfined CLN cells, *B*, one unconfined CLN cell, *C*, two confined CLN cells, and *D*, one unconfined CLN cell but with the "flow-to-dry-cell" option.

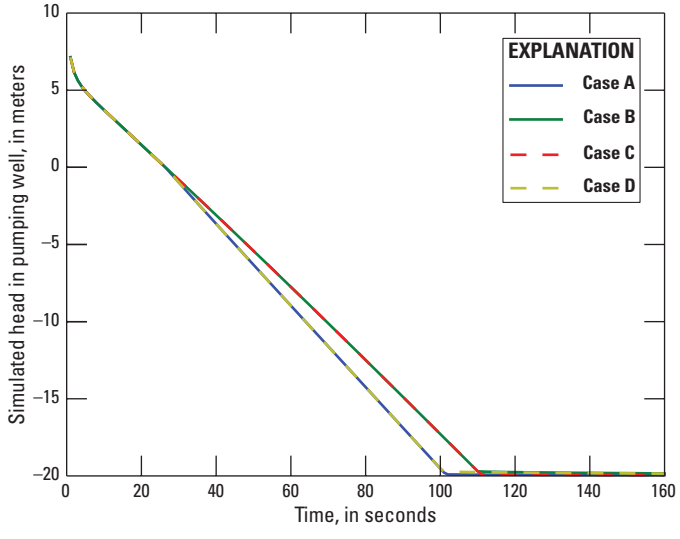

Figure 50. Simulated head in the pumping well for the various simulation cases.

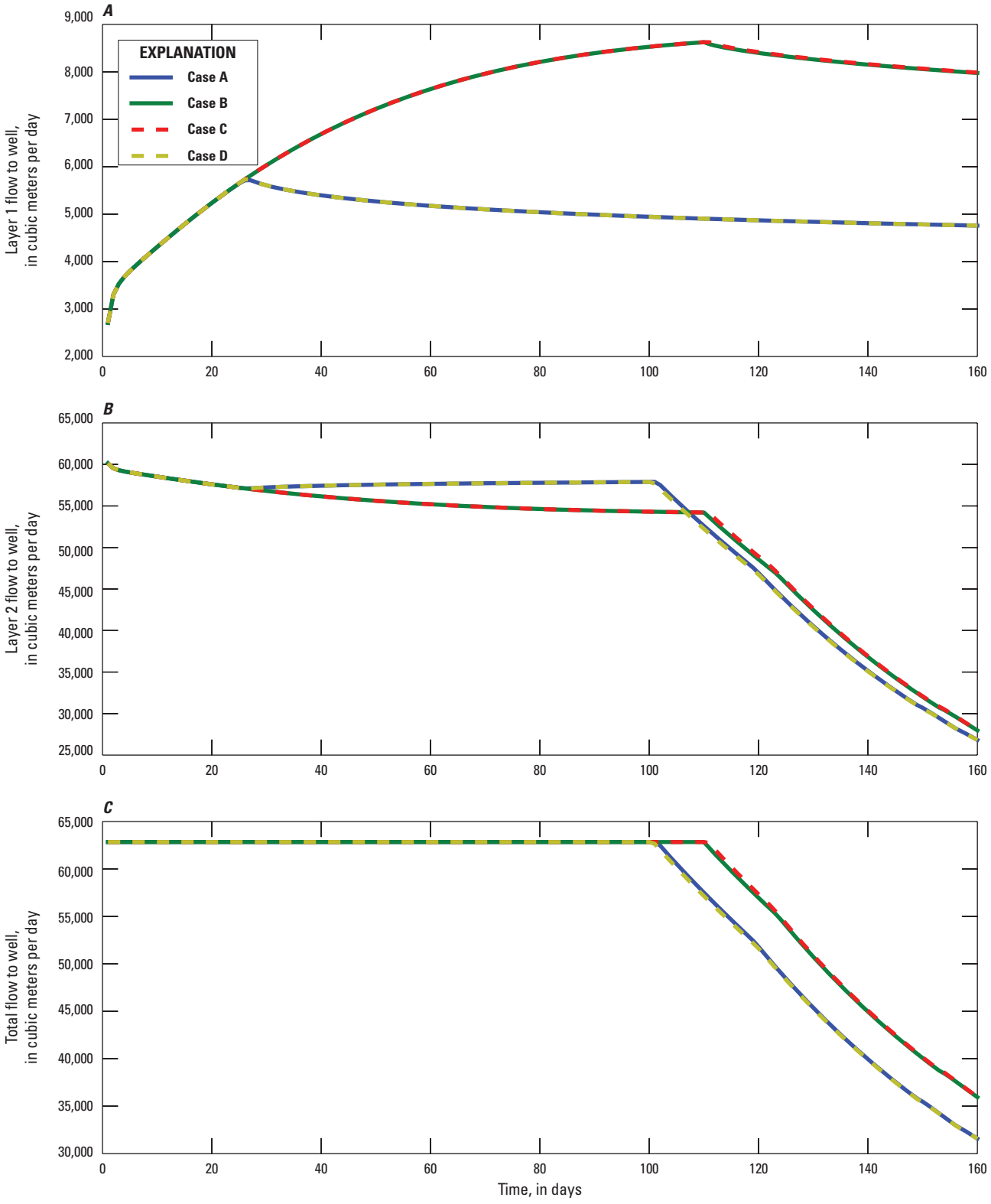

Figure 51. Simulated flow in the well for the various simulation cases.

References Cited

Bennett, G.D., Kontis, A.L., and Larson, S.P., 1982, Representation of multiaquifer well effects in three-dimensional ground-water flow simulation: Ground Water, v. 20, no. 3, p. 334–341.

Brakefield, Linzy, Langevin, C.D., Hughes, J.D., and Chartier, Kevin, in press, Estimation of capture zones and drawdown at the Northwest and West Well Fields, Miami-Dade County, Florida, using an unconstrained Monte Carlo analysis: Recent (2004) and proposed conditions: U.S. Geological Survey Open-File Report 2013–1086.

Cooley, R.L., 1983, Some new procedures for numerical solution of variably saturated flow problems: Water Resources Research, v. 19, p. 1271–1285.

Cooley, R.L., 1992, A MODular Finite-Element model (MODFE) for areal and axisymmetric ground-water-flow problems, part 2--derivation of finite-element equations and comparisons with analytical solutions: U.S. Geological Survey Techniques of Water-Resources Investigations, book 6, chap. A4.

Dehotin, J., Vazquez, R.F., Braud, I., Debionne, S., and Viallet, P., 2011, Modeling of hydrological processes using unstructured and irregular grids—2D groundwater application: Journal of Hydrologic Engineering, v. 16, no. 2, p. 108–125.

de Marsily, G., 1986, Quantitative hydrogeology—Groundwater for engineers: London, Academic Press, 1986.

DHI–WASY GmbH, 2010, FEFLOW–6; finite element subsurface flow & transport simulation system, user manual, DHI–WASY GmbH: Berlin, Germany.

Dickinson, J.E., James, S.C., Mehl, S., Hill, M.C., Leake, S.A., Zyvoloski, G.A., Faunt, C.C., Eddebbarh, A., 2007, A new ghost-node method for linking different models and initial investigations of heterogeneity and nonmatching grids: Advances in Water Resources, v. 30, p. 1722–1736.

Edwards, M., 1996, Elimination of adaptive grid interface errors in the discrete cell centered pressure equation: Journal of Computational Physics, v. 126, article no. 0143, p. 356–372.

Forsyth, P.A., 1991, A control volume finite element approach to NAPL groundwater contamination: SIAM Journal on Scientific and Statistical Computing, v. 12, no. 5, p. 1029–1057.

Halford, K.J., and Hanson, R.T., 2002, User guide for the drawdown- limited, multi-node well (MNW) package for the U.S. Geological Survey's modular three-dimensional finite-difference ground-water flow model, versions MODFLOW–96 and MODFLOW–2000: U.S. Geological Survey Open-File Report 02–293, 33 p.

Harbaugh, A.W., 2005, MODFLOW–2005, the U.S. Geological Survey modular ground-water model—The Ground-Water Flow Process: U.S. Geological Survey Techniques and Methods, book 6, chap. A16, variously paged.

Harbaugh, A.W., Banta, E.R., Hill, M.C., and McDonald, M.G., 2000, MODFLOW–2000, the U.S. Geological Survey modular ground-water model—User guide to modularization concepts and the Ground-Water Flow Process: U.S. Geological Survey Open-File Report 00–92, 121 p.

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

Höffmann, J., Leake, S.A., Galloway, D.L., and Wilson, A.M., 2003, MODFLOW–2000 ground-water model—User guide to the Subsidence and Aquifer-System Compaction (SUB) Package: U.S. Geological Survey Open-File Report 03–233, 44 p.

Hsieh, P.A., and Freckleton, J.R., 1993, Documentation of a computer program to simulate horizontal-flow barriers using the U.S. Geological Survey modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 92–477, 32 p.

Huyakorn, P.S., and Pinder, G.F., 1983, Computational methods in subsurface flow: Academic Press, Inc.

Ibaraki, M., 2005, χMD User's guide—An efficient sparse matrix solver library, version 1.30: Columbus, Ohio State University School of Earth Sciences.

Konikow, L.F., Hornberger, G.Z., Halford, K.J., and Hanson, R.T., 2009, Revised multi-node well (MNW2) package for MODFLOW ground-water flow model: U.S. Geological Survey Techniques and Methods, book 6, chap. A30, 67 p.

Lal, A.M., 1998. Weighted implicit finite-volume model for overland flow: Journal of Hydraulic Engineering, v. 124, no. 9, p. 941–950.

Lal, A.M., Van Zee, R., and Belnap, M., 2005. Case study: Model to simulate regional flow in south Florida: Journal of Hydraulic Engineering, v. 131, no. 4, p. 247–258.

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., and Lilly, M.R., 1997, Documentation of a computer program (FHB1) for assignment of transient specified-flow and specified-head boundaries in applications of the modular finite- difference ground-water flow model (MODFLOW): U.S. Geological Survey Open-File Report 97–571, 50 p.

McDonald, M.G., and Harbaugh, A.W., 1988, A modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Techniques of Water-Resources Investigations, book 6, chap. A1, 586 p.

Mehl, S., 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. and Hill, M.C., 2004, Three-dimensional local grid refinement for block-centered finite-difference groundwater models using iteratively coupled shared nodes—A new method of interpolation and analysis of errors: Advances in Water Resources, v. 27, no. 9, 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, book 6, chap. A12.

Merritt, M.L., and Konikow, L.F., 2000, Documentation of a computer program to simulate lake-aquifer interaction using the MODFLOW ground-water flow model and the MOC3D solute-transport model: U.S. Geological Survey Water-Resources Investigations Report 00–4167, 146 p.

Moridis, G., and Pruess, K., 1992, TOUGH simulations of Updegraff's set of fluid and heat flow problems: Berkeley, Calif., Lawrence Berkeley Laboratory Report LBL–32611.

Narasimhan, T.N., and Witherspoon, P.A., 1976, An integrated finite-difference method for analyzing fluid flow in porous media: Water Resources Research, v. 12, no. 1, p. 57–64.

Neville, C.J., and Tonkin M.J., 2004, Modeling multi-aquifer wells with MODFLOW: Groundwater, v. 42, no. 6, p. 910–919.

Niswonger, R.G., Panday, Sorab, and Ibaraki, Motomu, 2011, MODFLOW–NWT, A Newton formulation for MODFLOW–2005: U.S. Geological Survey Techniques and Methods, book 6, chap. A37, 44 p.

Niswonger, R.G., and Prudic, D.E., 2005, Documentation of the Streamflow-Routing (SFR2) Package to include unsaturated flow beneath streams—A modification to SFR1: U.S. Geological Survey Techniques and Methods, book 6, chap. A13, 47 p.

Panday, S., and Huyakorn, P.S., 2004, A fully coupled physically-based spatially-distributed model for evaluating surface/subsurface flow: Advances in Water Resources, v. 27, no. 4, p. 361–382.

Panday, S., and Huyakorn, P.S., 2008, MODFLOW SURFACT: A state-of-the-art use of vadose zone flow and transport equations and numerical techniques for environmental evaluations: Vadose Zone Journal, v. 7, no. 2, p. 610–631.

Panday, S., Huyakorn, P.S., Therrien, R., and Nichols, R.L., 1993, Improved three-dimensional finite element techniques for field simulation of variably saturated flow and transport: Journal of Contaminant Hydrology, v. 12, p. 3–33.

Panday, S., and Langevin, C.D., 2012. Improving subgrid scale accuracy of boundary features in regional finite-difference models: Advances in Water Resources, v. 41, p. 65–75.

Peaceman, D.W., 1977, Fundamentals of numerical reservoir simulation: Amsterdam, The Netherlands, Elsevier.

Peaceman, D.W., 1978, Interpretation of well-block pressures in numerical reservoir simulation: Society of Petroleum Engineers Journal, v. 18, no. 3, p. 183–194.

Peaceman, D.W., 1983, Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability: Society of Petroleum Engineers Journal, p. 531–543.

Prudic, D.E., 1989, Documentation of a computer program to simulate stream-aquifer relations using a modular, finite-difference, ground-water flow model: U.S. Geological Survey Open-File Report 88–729, 113 p.

Prudic, D.E., Konikow, L.F., and Banta, E.R., 2004, A new Streamflow-Routing (SFR1) Package to simulate stream-aquifer interaction with MODFLOW–2000: U.S. Geological Survey Open File Report 2004–1042, 95 p.

Pruess, K., Oldenburg, C., and Moridis, G., 1999, TOUGH2 user's guide, version 2.0: Berkeley, Calif., Ernest Orlando Lawrence Berkeley National Laboratory Report No. LBNL 43134.

Romero, D.M., and Silver S.E., 2006, Grid cell distortion and MODFLOW's integrated finite-difference numerical solution: Ground Water v. 44, p. 797–802. (Also available at http://dx.doi.org/10.1111/j.1745-6584.2005.00179.x)

Schaars, F., and Kamps, P., 2001, MODGRID: Simultaneous solving of different groundwater flow models at various scales, *in* Seo, H.S., Poeter, E.P., Zheng, C., and Poeter, O., eds., Proceedings of MODFLOW 2001 and Other Modeling Odysseys Conference, Golden, Colorado, September 11–14: Colorado School of Mines, v. 1, p. 38–44.

Shoemaker, W.B., Kuniansky, E.L., Birk, S., Bauer, S., and Swain, E.D., 2007, Documentation of a Conduit Flow Process (CFP) for MODFLOW–2005: U.S. Geological Survey Techniques and Methods, book 6, chap. A24.

Sudicky, E.A., Unger, A., and Lacombe, S., 1995, A noniterative technique for the direct implementation of well bore boundary conditions in three-dimensional heterogeneous formations: Water Resources Research, v. 31, no. 2. (Also available at http://dx.doi.org/10.1029/94WR02854)

Therrien, R., McLaren, R.G., Sudicky, E.A., and Panday, S., 2004, HydroGeoSphere, a three-dimensional numerical model describing fully-integrated subsurface and surface flow and solute transport: Waterloo, Ontario, Groundwater Simulations Group.

Torak, L.J., 1993a, A MODular Finite-Element model (MODFE) for areal and axisymmetric ground-water-flow problems, part 1—Model description and user's manual: U.S. Geological Survey Techniques of Water-Resources Investigations, book 6, chap. A3.

Torak, L.J., 1993b, A MODular Finite-Element model (MODFE) for areal and axisymmetric ground-water-flow problems, part 3—Design philosophy and programming details: U.S. Geological Survey Techniques of Water-Resources Investigations, book 6, chap. A5.

Voss, C.I., and Provost, A.M., 2002, SUTRA, a model for saturated-unsaturated variable-density ground-water flow with solute or energy transport: U.S. Geological Survey Water-Resources Investigations Report, 02–4231, 291 p.

White, J.T., and Hughes, J.D., 2011, An unstructured GPGPU preconditioned conjugate gradient solver for MODFLOW 2005, MODFLOW and More 2011, June 5–8, 2011, Golden, Colorado.

Appendix 1. Notation Used in this Report

$\mathbf{A}$ = Global conductance matrix.

$a_{nm} = a_{mn}$ = Perpendicular saturated flow area between cells *n* and *m*.

$a_{j\bar{n}}$ = Effective saturated area of face between cell *j* and ghost location $\bar{n}$.

$a_{nm}K_{nm}$ = Numerator of the inter-cell conductance term.

a_{cnm} = Cross-sectional area of the connection between the CLN cells ($= \min(a_{cn}, a_{cm})$).

a_{cn} = Cross-sectional area of CLN cell *n*.

a_{cm} = Cross-sectional area of CLN cell *m*.

a_{wn} = Wetted cross-sectional area of CLN cell *n*.

$\mathbf{b}$ = Right-hand-side vector.

b_s = Thickness of the skin surrounding the CLN-to-GWF interface.

BOT = Bottom elevation of a cell.

C = Nonlinear well-loss coefficient.

C_{nm}^0 = Constant portion (fully saturated) of the conductance term between cells *n* and *m*.

C_{nm} = Inter-cell conductance between cells *n* and *m* (is nonlinear for an unconfined simulation).

$C_{nm}(h_{m,\bar{n}})$ = Conductance between cells *n* and *m* for flow to ghost node $\bar{n}$.

$C_{j\bar{n}}$ = Conductance between the contributing cell *j* and the ghost node $\bar{n}$.

$C_{n\bar{n}}$ = Conductance between the cell *n* and the ghost node location $\bar{n}$.

CV, *CR*, *CC* = Inter-cell conductances for the layer, row and column directions, respectively of a structured MODFLOW–2005 grid.

CHANI = Horizontal anisotropy factor.

d = Depth of flow in the horizontal conduit.

e_n = Reference elevation for the downstream node, *n*, for "dry flow" option.

f_{vn} = Fraction of the volume of CLN cell *n* that is saturated.

f_{unm} = Fraction of the upstream CLN cell volume that is saturated.

f_{upn} = Wetted fraction of the upstream perimeter of the CLN – GWF interaction.

f(h) = Smooth and continuous function defining the saturated fraction of a cell.

$f(h_{ups})$ = Upstream-weighted nonlinear term of the flow equation.

g = Gravitational constant.

$\mathbf{h}$ = Vector of hydraulic heads.

h = Hydraulic head.

h_n , h_m = Hydraulic head at cell *n* and *m*.

h_p = Hydraulic head at GWF cell *p*.

$h_{\bar{n}}$ = Head at the ghost node location.

$\overline{h_n}$ = Smoothed function for expressing the downstream node head for "flow-to-dry-cell" option.

h^k = Vector of the unknown heads at current iteration level *k*.

h^{k-1} = Vector of the heads from previous iteration level *k-1*.

$HCOF_n$ = Sum of all terms that are coefficients of h_n including storage and head dependent boundary flux terms.

J_R^{k-1} = Jacobian matrix.

k = Current iteration number.

k-1 = Previous iteration number.

K = Hydraulic conductivity.

K_{nm} = Inter-cell hydraulic conductivity between cells *n* and *m*.

K_{xx} = Principal component of the hydraulic conductivity tensor in the x-direction.

K_{yy} = Principal component of the hydraulic conductivity tensor in the y-direction.

K_{zz} = Principal component of the hydraulic conductivity tensor in the z-direction.
$\hat{K}_{nm}$ = Anisotropic hydraulic conductivity value in the *n*-*m* direction for cell *n*.
$\hat{K}_{mn}$ = Anisotropic hydraulic conductivity value in the *n*-*m* direction for cell *m*.
K_{cnm} = Saturated linear conductivity of the connection between the CLN cells *n* and *m*.
K_{cn} , K_{cm} = Saturated linear conductivity of CLN cells *n* and *m*.
K_s = Hydraulic conductivity of the skin surrounding the CLN-to-GWF interface.
$K_{j\bar{n}}$ = Effective hydraulic conductivity between cell *j* and ghost location $\bar{n}$.
L_{nm} , L_{mn} = Perpendicular distances between the shared *n*-*m* interface and a point within cells *n* and *m* respectively.
L_{nj} , L_{jn} = Perpendicular distances between the shared *n*-*j* interface and a point within cells *n* and *j* respectively.
$L_{j\bar{n}}$ = Effective distance between cell *j* and ghost location $\bar{n}$.
l_{cn} , l_{cm} = Length of CLN cells *n* and *m*.
LHS_Adjust_n = Matrix correction term to adjust for ghost node contributions in column *n*.
LHS_Adjust_m = Matrix correction term to adjust for ghost node contributions in column *m*.
m = Cell index.
n = Cell index.
$\mathbf{n}$ = Outward-pointing unit normal.
$\bar{n}$ = Ghost node location.
N_m = Number of adjacent connections to GWF cell, *n*.
P = Exponent for nonlinear well loss term.
P_{nm} = Midpoint of shared face between cells *n* and *m*.
P_t = Total perimeter of the CLN cell.
P_w = Wetted perimeter of CLN cell.
P_{wu} = Wetted perimeter of CLN–GW interaction computed using the upstream head of the connection.
Q_{nm} = Volumetric flow between cells *n* and *m*.
r_n = Effective radius of CLN cell *n*.
$R^{k\text{-}1}$ = Residual vector of Newton-Raphson equation at iteration level *k*-1.
$\mathrm{R_n}$ = Residual of the balance equation for cell *n*.
r_{oh} = Effective external radius of a GWF cell for a connected horizontal CLN cell (representing the radius of influence in the Thiem equation).
r_{oz} = Effective external radius of a GWF cell for a connected vertical CLN cell (representing the radius of influence in the Thiem equation).
RHS_{GNC} = The right-hand-side term resulting from GNC expansion.
RHS_Adjust = Correction term to right-hand-side vector to adjust for ghost node contributions.
R_{cond} = Radius of the cylindrical conduit geometry for CLN domain.
RHS_n = Right-hand side value of the balance equation containing storage and boundary condition terms for cell *n*.
$\mathfrak{R}_{rh}$ = Horizontal to vertical anisotropy ratio (K_x / K_z).
$\mathfrak{R}_{rz}$ = The x:y anisotropy ratio (K_x / K_y).
RHS = Right-hand side vector.
RHS_{update} = Update to the right-hand side vector to include dry cell correction, Newton Raphson, or GNC terms.
S = the surface of a control volume.
S_f = Skin factor.
S_s , SS_n = Specific storage.
t = Time.
t-1 = Previous time step.
TOP = Top elevation of a cell.
V = Volume
V_n = Total volume of CLN cell *n*.
W = Volumetric source or sink per unit volume.

α = Angle in radians between the principal anisotropy direction and the normal to the face.

α_j = Contributing fraction of each additional contributing cell j to the head at the ghost node $\overline{n}$.

α_n = Contributing fraction of cell *n*, to the head at the ghost node $\overline{n}$.

α_{cpn} = Saturated conductance between the CLN cell *n* and GWF cell *p*.

ΔH = Horizontal cell dimension normal to the line of a horizontal CLN cell.

ΔQ_{nm}^{GNC} = Correction term to adjust the regular CVFD assembled equations to account for the head adjustment of the ghost node location.

Δt = Time step size

ΔZ = Thickness of the GWF cell normal to the line of a horizontal CLN cell.

ε = Small number (10^{-4}) over which the slope discontinuity is smoothed for the "flow to dry cell" function.

Γ_{cpn} = Volumetric flow from a connected GWF cell *p* to CLN cell *n*.

ρ = Density of water.

Ω = Small number over which the slope discontinuity is smoothed for the function defining the saturated fraction of a cell versus head.

θ = The angle that a conduit cell makes with the horizontal.

μ = Dynamic viscosity of water.

η_{nh} = The set of horizontally connected GWF cells to cell *n*.

Manuscript approved February 1, 2013

Development of MODFLOW-USG was supported in part through a research grant from AMEC.

USGS contributions to this work were funded by the USGS Groundwater Resources Program.

For more information about this publication, contact:

Office of Groundwater

U.S. Geological Survey

Mail Stop 411

12201 Sunrise Valley Drive

Reston, VA 20192

(703) 648-5001

Prepared by the Raleigh Publishing Service Center