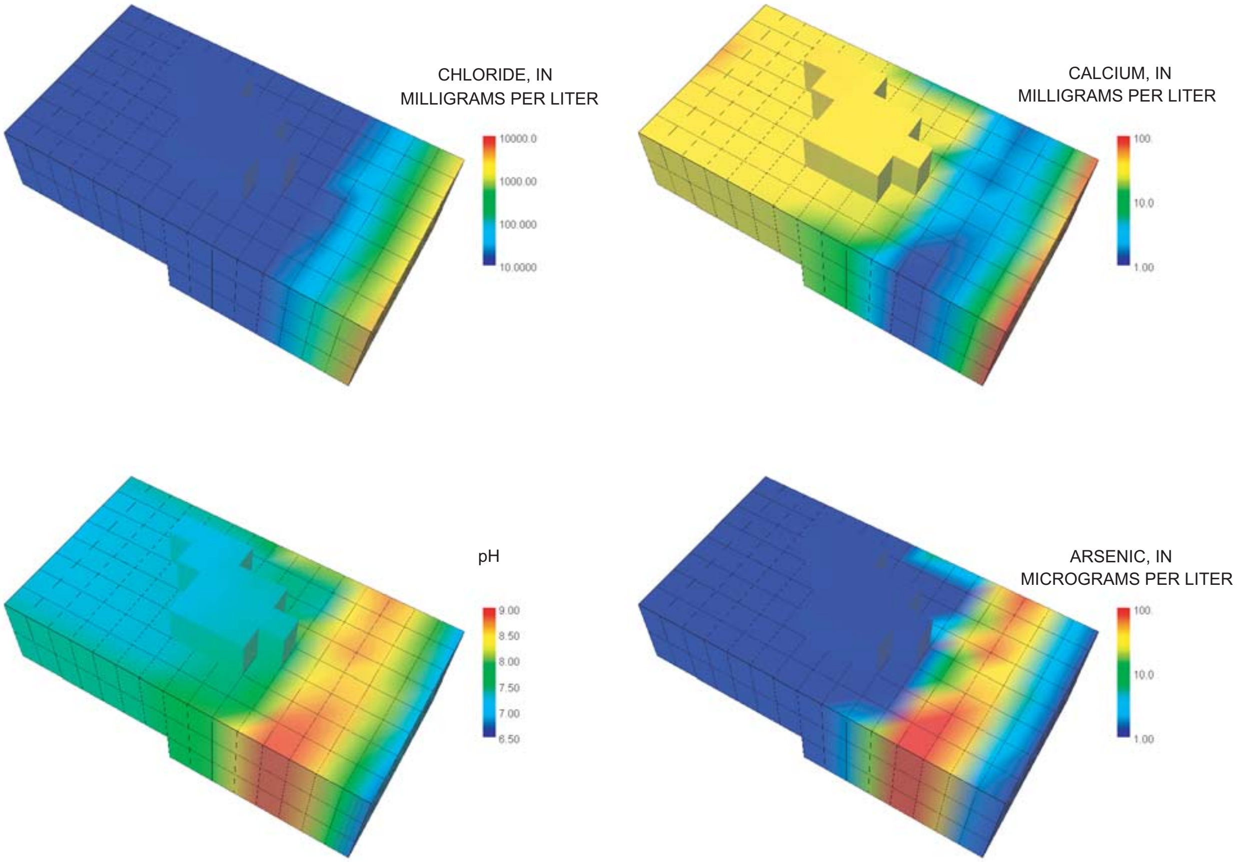
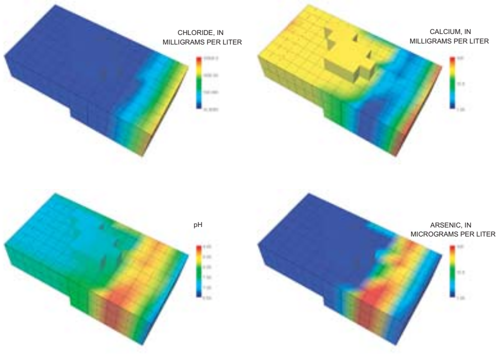
PHAST—A Program for Simulating Ground-Water Flow, Solute Transport, and Multicomponent Geochemical Reactions



Techniques and Methods 6–A8

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



COVER ILLUSTRATION: Results of PHAST simulation of the evolution of water chemistry in the Central Oklahoma aquifer (example 4 of this report). Clockwise from upper left are the distribution of chloride concentrations, calcium concentrations, arsenic concentrations and pH after 100,000 years of simulated ground-water flow and reaction. View is from the northwest looking to the southeast.

PHAST—A Program for Simulating Ground-Water Flow, Solute Transport, and Multicomponent Geochemical Reactions

By David L. Parkhurst, Kenneth L. Kipp, Peter Engesgaard, and Scott R. Charlton

Techniques and Methods 6–A8

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

U.S. Department of the Interior
Gale A. Norton, Secretary

U.S. Geological Survey
Charles G. Groat, Director

U.S. Geological Survey, Denver, Colorado: 2004

For sale by U.S. Geological Survey, Information Services
Box 25286, Denver Federal Center
Denver, CO 80225

For more information about the USGS and its products:
Telephone: 1-888-ASK-USGS
World Wide Web: http://www.usgs.gov/

Suggested citation:
Parkhurst, D.L., Kipp, K.L., Engesgaard, Peter, and Charlton, S.R., 2004, PHAST—A program for simulating ground-water flow, solute transport, and multicomponent geochemical reactions: U.S. Geological Survey Techniques and Methods 6–A8, 154 p.

Contents

Figures

Tables

Abbreviations

atmosphere	atm
cubic meter	m^3
day	d
equivalent	eq
gram	g
Joule	J
kilogram	kg
kilogram of water	kgw
liter	L
meter	m
mole	mol
milliequivalent	meq
milligram	mg
millimole	mmol
microgram	µg
micromole	µmol
parts per million	ppm
second	s
square meter	m^2
year	yr

PHAST—A Program for Simulating Ground-Water Flow, Solute Transport, and Multicomponent Geochemical Reactions

By David L. Parkhurst[1], Kenneth L. Kipp[1], Peter Engesgaard[2], and Scott R. Charlton[1]

Abstract

The computer program PHAST simulates multi-component, reactive solute transport in three-dimensional saturated ground-water flow systems. PHAST is a versatile ground-water flow and solute-transport simulator with capabilities to model a wide range of equilibrium and kinetic geochemical reactions. The flow and transport calculations are based on a modified version of HST3D that is restricted to constant fluid density and constant temperature. The geochemical reactions are simulated with the geochemical model PHREEQC, which is embedded in PHAST.

PHAST is applicable to the study of natural and contaminated ground-water systems at a variety of scales ranging from laboratory experiments to local and regional field scales. PHAST can be used in studies of migration of nutrients, inorganic and organic contaminants, and radionuclides; in projects such as aquifer storage and recovery or engineered remediation; and in investigations of the natural rock-water interactions in aquifers. PHAST is not appropriate for unsaturated-zone flow, multiphase flow, density-dependent flow, or waters with high ionic strengths.

A variety of boundary conditions are available in PHAST to simulate flow and transport, including specified-head, flux, and leaky conditions, as well as the special cases of rivers and wells. Chemical reactions in PHAST include (1) homogeneous equilibria using an ion-association thermodynamic model; (2) heterogeneous equilibria between the aqueous solution and minerals, gases, surface complexation sites, ion exchange sites, and solid solutions; and (3) kinetic reactions with rates that are a function of solution composition. The aqueous model (elements, chemical reactions, and equilibrium constants), minerals, gases, exchangers, surfaces, and rate expressions may be defined or modified by the user.

A number of options are available to save results of simulations to output files. The data may be saved in three formats: a format suitable for viewing with a text editor; a format suitable for exporting to spreadsheets and post-processing programs; or in Hierarchical Data Format (HDF), which is a compressed binary format. Data in the HDF file can be visualized on Windows computers with the program Model Viewer and extracted with the utility program PHASTHDF; both programs are distributed with PHAST.

Operator splitting of the flow, transport, and geochemical equations is used to separate the three processes into three sequential calculations. No iterations between transport and reaction calculations are implemented. A three-dimensional Cartesian coordinate system and finite-difference techniques are used for the spatial and temporal discretization of the flow and transport equations. The non-linear chemical equilibrium equations are solved by a Newton-Raphson method, and the kinetic reaction equations are solved by a Runge-Kutta or an implicit method for integrating ordinary differential equations.

The PHAST simulator may require large amounts of memory and long Central Processing Unit (CPU) times. To reduce the long CPU times, a parallel version of PHAST has been developed that runs on a multiprocessor computer or on a collection of computers that are networked. The parallel version requires Message Passing Interface, which is currently (2004) freely available. The parallel version is effective in reducing simulation times.

This report documents the use of the PHAST simulator, including running the simulator, preparing the input files, selecting the output files, and visualizing the results. It also presents four examples that verify the numerical method and demonstrate the capabilities of the simulator. PHAST requires three input files. Only the flow and transport file is described in detail in this report. The other two files, the chemistry data file and the database file, are identical to PHREEQC files and the detailed description of these files is found in the PHREEQC documentation.

---

[1] U.S. Geological Survey, Lakewood, Colorado
[2] Geological Institute, University of Copenhagen, Copenhagen, Denmark

Chapter 1. Introduction

The computer program PHAST (**PH**REEQC **A**nd H**ST**3D) simulates multi-component, reactive solute transport in three-dimensional saturated ground-water flow systems. PHAST is a versatile ground-water flow and transport simulator with capabilities to model a wide range of equilibrium and kinetic geochemical reactions. The flow and transport calculations are based on a modified version of HST3D (Kipp, 1987; 1997) that is restricted to constant fluid density and constant temperature. The geochemical reactions are simulated with the geochemical model PHREEQC (Parkhurst, 1995; Parkhurst and Appelo, 1999), which is embedded in PHAST. The combined flow, transport, and geochemical processes are simulated by three sequential calculations for each time step: first, flow velocities are calculated; then, the chemical components are transported; and finally geochemical reactions are calculated. This sequence is repeated for successive time steps until the end of the simulation.

Reactive-transport simulations with PHAST require three input files: a flow and transport data file, a chemistry data file, and a thermodynamic database file. All input data files are built with modular keyword data blocks. Each keyword data block defines a specific kind of information—for example, grid locations, porous media properties, boundary-condition information, or initial chemical composition in a zone. All spatial data are defined by zones, which are rectangular volumes. Simulation results can be saved in a variety of file formats, including Hierarchical Data Format (HDF, http://hdf.ncsa.uiuc.edu). Two utility programs are distributed with PHAST: (1) Model Viewer (Windows only), which is used to produce three-dimensional visualizations of problem definition and simulation results; and (2) PHASTHDF, which is used to extract subsets of the data stored in the HDF file.

The PHAST simulator can be run on most computer systems, including computers with Windows, Linux, and Unix operating systems. Coupled reactive-transport simulations are computer intensive, which can result in long run times on a single processor computer. A parallel (multiprocessor) version of PHAST, implemented with the Message Passing Interface (MPI), is available to be used on multiprocessor computers and on networked computer clusters (Windows, Unix, or Linux). Using the parallel version of PHAST can greatly reduce run times.

1.1. Applicability

PHAST is applicable to the study of natural and contaminated ground-water systems at a variety of scales ranging from laboratory experiments to local and regional field scales. Representative examples of PHAST applications include simulation of effluent arsenic concentrations in laboratory column experiments, migration of dissolved organic compounds from a landfill, migration of nutrients in a sewage plume in a sandy aquifer, storage of fresh water in a slightly saline aquifer, and examination of natural mineral and exchange reactions in a regional aquifer.

PHAST is not suitable for some types of reactive-transport modeling. In particular PHAST is not appropriate for unsaturated-zone flow and does not account for flow and transport of a gas phase or a non-aqueous liquid phase. PHAST is restricted to constant temperature and constant density and does not account for density-dependent flow caused by concentration gradients or temperature variations. The aqueous model for PHAST is an ion-association model that is applicable to ionic strengths to 1 molal. Thus, PHAST is not an appropriate simulator for many systems and processes, including transport of volatile organic compounds in a soil zone, hydrothermal systems, most high ionic-strength brines, or heat storage and recovery in aquifers.

1.2. Simulator Capabilities

The PHAST simulator is a general computer code with various reaction chemistry, equation-discretization, boundary-condition, source-sink, and equation-solver options. Four types of flow and reactive-transport simulations can be performed with PHAST. Listed in order from simple to complex in terms of input-data requirements and computational workload, they are steady-state simulation of ground-water flow, transient simulation of ground-water flow, steady-state simulation of flow followed by reactive transport, and transient simulation of flow with reactive transport. The reactive-transport simulation is always transient, even though it may evolve to steady-state concentration fields after sufficient simulation time. The first two types of simulations are used to model ground-water flow and to calibrate a ground-water flow model. The third type of simulation is used when the flow field can be assumed to be in steady state. The fourth type of simulation is for general reactive-transport simulations with a transient-flow field. Additional details about the four types of simulations are provided in Chapter 2 and Appendix D.

The ground-water flow and transport are defined by initial and boundary conditions for heads and concentrations. The available boundary conditions include (1) specified-head conditions where solution compositions are fixed; (2) specified-head conditions where solution compositions are associated with inflowing water; (3) flux conditions, such as precipitation recharge, with associated solution composition for inflowing water; (4) leaky conditions, such as leakage from a constant-head water table through an aquitard, with associated solution composition;(5) rivers with associated solution composition; (6) injection of specified solution compositions or pumpage by wells; and (7) free-surface boundary condition (unconfined flow). A PHAST simulation is composed of a series of simulation periods. Heads and compositions of solutions for boundary conditions are constant over each period, but time series of values may be defined for heads, solution compositions, and pumping rates to simulate transient boundary conditions.

A wide range of chemical reactions can be coupled to the transport calculations. Heterogeneous equilibrium reactions may be simulated, including equilibrium among the aqueous solution and minerals, surface complexation sites, ion exchange sites, solid solutions, and a gas phase. The most complex simulations include kinetic reactions, which require mathematical expressions that define the rates of reactions as a function of solution composition, time, or other chemical factors. In addition, geochemical simulations, which include all of the above types of reactions plus additional reaction capabilities, may be used to define chemical initial and boundary conditions for the reactive-transport simulations. Any modeling capability available in PHREEQC (including one-dimensional transport simulations) may be used to establish initial and boundary conditions for the reactive-transport simulation of PHAST. The aqueous model, equilibrium conditions, and kinetic rate expressions are defined through a data base file that can be modified by the user.

1.3. Simulator Results

PHAST produces a huge amount of information internally, including heads, velocities, component concentrations, activities and molalities of aqueous species, and saturation indices of minerals for each active node for each time step. The program offers a number of options to select the data to be saved into output files and the frequency at which they are saved. The data may be saved in tabular ASCII files suitable for viewing with a text editor, in ASCII files with tab-delimited columns of data suitable for importing into spreadsheets or plotting programs, or in a binary HDF file.

Data in the HDF file can be visualized on Windows computers with the program Model Viewer (see Appendix A). Model Viewer can be used to display boundary-condition nodes for the verification of a problem setup, to view three-dimensional fields of scalar results (head, concentration, moles of minerals, and others), and to view velocity vectors that are located at each node of the simulation region. Data can be viewed for selected time steps and data for multiple time steps can be used to produce an animation. Data can be extracted from the HDF file with a Java utility program, PHASTHDF, which runs on any computer with a Java Runtime Environment (see Appendix B).

1.4. Numerical Implementation

PHAST solves a set of partial differential equations for flow and transport and a set of nonlinear algebraic and ordinary differential equations for chemistry. The equations that are solved numerically are (1) the saturated ground-water flow equation for conservation of total fluid mass, (2) a set of solute-transport equations for conservation of mass of each solute component of a chemical-reaction system, and (3) a set of chemical-reaction equations comprising mass-balance equations, mass-action equations, and kinetic-rate equations. The ground-water flow and solute-transport equations are coupled through the dependence of advective transport on the interstitial fluid-velocity field. For constant and uniform density ground water, this coupling is one way—the fluid-velocity field affects the solute transport, but the solute transport does not affect the fluid-velocity field. The solute-transport equations and the chemical equations are coupled through the chemical concentration terms. The chemical equations are algebraic mass-balance and mass-action equations for equilibrium reactions and ordinary differential equations for kinetic reactions. The chemical equations are fully coupled through the concentration terms and must be solved simultaneously.

By using a sequential solution approach for flow, transport, and reaction calculations, numerical solutions are obtained for each of the dependent variables: potentiometric head, solute-component concentrations, species concentrations, and masses of reactants in each cell. Operator splitting is used to separate the solute-transport calculations from the chemical-reaction calculations (Yeh and Tripathi, 1989). No iterations between transport and reaction calculations within a time step are performed. Finite-difference techniques are used for the spatial and temporal discretization of the flow and transport equations. The linear finite-difference equations for flow and for transport of each solute component are solved either by a direct solver or by an iterative solver.

Operator splitting also is used to separate the chemical equilibrium reactions from the kinetic reactions. The non-linear chemical equilibrium equations are solved by a Newton-Raphson method (Parkhurst and Appelo, 1999). The kinetic-reaction equations are ordinary differential equations, which are solved by using an explicit Runge-Kutta algorithm or an implicit algorithm for stiff differential equations called CVODE (Cohen and Hindemarsh, 1996). These algorithms evaluate the reaction rates at several (possibly many) intermediate times within the overall time interval equal to the transport time step. An automatic time-step algorithm is used for controlling errors in the integration of the kinetic reaction rates and the equilibrium equations are solved at each intermediate step of the integration.

A three-dimensional Cartesian coordinate system is used to define the region for reactive-transport simulations. Although flow and transport calculations are always three dimensional, the symmetry of conceptually one- and two-dimensional regions can be used to reduce the computational workload of the geochemical calculations. For these cases, the geochemical calculations are performed on a line or plane of nodes and then copied to the corresponding symmetric parts of the simulation region.

1.5. Computer Resources

The computer code for PHAST is written in Fortran-90 and C. The code conforms to American National Standards Institute (ANSI) standards for Fortran-90 and C, with few exceptions, and has proven portable to a variety of computer operating systems, including Windows (Windows95 or later), Linux on personal computers, and Unix on Sun, Digital Equipment Corporation (DEC), Hewlett Packard (HP), and International Business Machines (IBM) computers. Mathematical expressions for kinetic reactions and some output values are calculated by using a Basic-language interpreter (with permission, David Gillespie, Synaptics, Inc., San Jose, CA, written commun., 1997).

Fully dynamic memory allocation is used by PHAST as provided by the Fortran-90 and C languages. Little effort has been expended to minimize storage requirements. Consequently, the PHAST simulator may require large amounts of memory for execution. The number of nodes defined for the simulation grid is the primary determinant of the amount of memory needed for execution. PHAST can require long Central Processing Unit (CPU) times in addition to large amounts of memory. Processing time is determined mainly by the number of nodes, the number of time steps, and the presence or absence of kinetic reactions in the simulation. For equal numbers of nodes and time steps, a problem with kinetic reactions will run more slowly than a problem without kinetics. The CPU time also increases with the number of components, aqueous species, equilibrium phases, and other reactants that are included in the problem definition.

To reduce the long CPU times of PHAST simulations, a parallel version of PHAST has been developed that runs on a multiprocessor computer or on a collection of computers that are networked. The parallel version requires libraries for MPI, a standard for message passing among processes, be installed on all computers that are used for parallel processing. Multiple processors are used only for the geochemical part of a simulation. In general, the geochemical part of the simulation requires most of the CPU time, so the parallel version is effective in reducing run times. (See Appendix C.)

1.6. Purpose and Scope

The purpose of this documentation is to provide the user with information about the capabilities and usage of the reactive-transport simulator, PHAST. Sections on running the simulator, preparing input data files, selecting output data files, and examples are provided. Sections on the output visualization tool Model Viewer, the interactive output-data extractor PHASTHDF, the parallel-computer version, and the theory and numerical implementation are provided as appendixes. The examples include a set of three verification problems for multi-component reactive transport and one demonstration example for illustration of all possible types of boundary conditions and for tutorial purposes. Although the present documentation is intended to provide a complete description of the flow and transport capabilities of PHAST, the documentation for PHREEQC (Parkhurst and Appelo, 1999) is necessary to use the chemical-reaction capabilities of PHAST. Documentation for HST3D Versions 1 and 2 (Kipp, 1987; 1997) provides additional details about the flow and transport calculations.

1.7. Acknowledgments

The contributions of Flemming Christiansen, who helped with program development and testing while at the Technical University of Denmark, and Scott McFarlane of the U.S. Geological Survey, who built and maintained the Beowulf parallel computer cluster, are gratefully acknowledged.

Chapter 2. Running the Simulator

This chapter contains information needed to run the PHAST simulator as well as some background information on program organization and flow of execution. Specific information on preparation of input data files is contained in chapters 3, 4, and the PHREEQC documentation (Parkhurst and Appelo, 1999).

2.1. Input Files

The following three data-input files are needed for simulations that include flow, transport, and reactions: (1) *prefix***.trans.dat** the flow and transport calculation data, (2) *prefix***.chem.dat** the chemical-reaction calculation data, and (3) the thermodynamic database (default name is *phast.dat*). The identification *prefix* is provided by the users and is used to identify files for a simulation. In this documentation, it is represented by "*prefix*". (Here, and throughout the documentation, bold type is used for words that must be typed as specified and italic type is used for words that must be specified by the user and for file names.) If only ground-water flow is simulated, only the *prefix***.trans.dat** file is needed. Chapter 3 and the PHREEQC documentation (Parkhurst and Appelo, 1999) contain the necessary information to create the chemistry data file and to modify the thermodynamic database file. Chapter 4 contains the necessary information to create the flow and transport data file.

2.2. Output Files

Various types of output files can be produced by running the PHAST program. Some of the output files are in a tabular format intended to be displayed or printed. Some files are intended only for post-processing by visualization or plotting programs.

Table 2.1 lists the file names in alphabetical order with a brief description of the contents of each of the output files. A more detailed description of output files is presented in Chapter 5. Except for the time unit, the units of measure of the output data are always SI (International System) metric, no matter what units were used for the input data. The time unit for output is selected by the user. Chemical concentrations are written in units of molality in all of the output files, except in the *prefix***.xyz.chem** and *prefix***.h5** files, where it is possible for the user to print concentrations in alternative units.

The files with names beginning with *prefix***.O** are in tabular format to be viewed in an editor (or to be printed) and contain values of variables at grid nodes arranged by horizontal or vertical grid slices, as chosen by the user. The files with names beginning with *prefix***.xyz** are in tab-separated columnar format, suitable for importing into spreadsheet and plotting programs. All of the output files are written in ASCII format, except for the HDF file, which is in compressed binary format. Data written to the binary HDF file can be visualized by the program Model Viewer (Appendix A) and data can be extracted to an ASCII file with the program PHASTHDF (Appendix B).

The types of transient information that may be written include the solution-method information (*prefix***.log**), the fluid and solute-dispersive conductance distributions (*prefix***.O.kd**), the potentiometric head (*prefix***.O.head**, *prefix***.xyz.head**, and *prefix***.h5**), component concentration distributions (*prefix***.O.comps**), selected chemical information (*prefix***.O.chem**, *prefix***.xyz.chem**, and *prefix***.h5**), the steady-state or final potentiometric head distribution (*prefix***.head.dat**), the velocity distribution (*prefix***.O.vel**, *prefix***.xyz.vel**, and *prefix***.h5**), the regional fluid-flow and solute-flow rates and the regional cumulative-flow results (*prefix***.O.bal**), boundary-condition flow rates (*prefix***.O.bcf**), and the well-flow data (*prefix***.O.wel**). The selection of the frequency that data are written to these files is explained in Chapter 4.

The potentiometric head field from one simulation may be used to establish the initial condition for a subsequent simulation. Potentiometric head distribution from a steady-state or transient-flow simulation can be written to a file (*prefix***.head.dat**), which can be read as the initial head conditions for subsequent runs (executions of the PHAST program). The corresponding option of using the component-concentration fields from one simulation as the initial conditions for chemistry for a subsequent simulation is not available.

Table 2.1.—List of output files that may be generated by a PHAST simulation

File name	Contents
*prefix***.h5**	Grid and boundary-condition information, potentiometric head field, velocity field, and selected chemical concentration data
*prefix***.head.dat**	Potentiometric heads at the final time step of the simulation in a form that can be read as initial head conditions in subsequent simulations
*prefix***.log**	Copies of the *prefix***.trans.dat** and *prefix***.chem.dat** input data files, solver statistics, and all error or warning messages
*prefix***.O.bal**	Fluid and chemical component regional mass balances, total balances, and balances listed by boundary-condition type
*prefix***.O.bcf**	Boundary-condition fluid and solute flow rates for cells listed by boundary-condition type
*prefix***.O.chem**	Solution concentrations, distribution of aqueous species, saturation indices, and compositions of exchangers, gas phases, kinetic reactants, phase assemblages, solid solutions, and surfaces from the internal PHREEQC calculation at the beginning of a PHAST run. Data are normally not written to this file during flow, transport, and reaction calculations.
*prefix***.O.comps**	Initial and transient total dissolved concentrations for chemical elements (components)
*prefix***.O.head**	Initial and transient potentiometric heads
*prefix***.O.kd**	Static fluid conductances and transient solute dispersive conductances at cell faces for X-, Y-, and Z-coordinate directions
*prefix***.O.probdef**	Flow and transport problem definition as specified by the input files, including array sizes, grid definition, media properties, initial conditions, static and transient boundary-condition information
*prefix***.O.vel**	Interstitial velocities for X-, Y-, and Z-coordinate directions across cell faces and interstitial velocities in the X-, Y-, and Z-coordinate directions at nodes
*prefix***.O.wel**	Static and transient well information, including well location and identification number, fluid and solute flow rates, cumulative fluid and solute flow amounts, and solute concentrations
*prefix***.xyz.chem**	Selected initial-condition and transient chemical data
*prefix***.xyz.comps**	Initial-condition and transient cell concentrations for each chemical element (component)
*prefix***.xyz.head**	Initial-condition and transient potentiometric heads
*prefix***.xyz.vel**	Steady-state or transient velocity-vector components interpolated to grid nodes
*prefix***.xyz.wel**	Transient concentration data for wells
selected_output	Rarely used. Selected chemical data from the initial call to PHREEQC to process the chemistry data file (Following the initial call to PHREEQC, selected output for initial and transient conditions are written to *prefix***.xyz.chem**.)

2.3. Program Execution

PHAST is run by using a batch file (Windows) or shell script (Unix), each referred to as a “script”. The script invokes two programs, (1) PHASTINPUT (file *phastinput* for Unix, *phastinput.exe* for Windows), which converts the flow and transport data file in keyword format into an intermediate input file, *Phast.tmp*, and (2) PHAST (file *phast-ser* for Unix, *phast-ser.exe* for Windows; “ser” for serial or single processor), which performs the flow and reactive-transport simulations. Figure 2.1 illustrates the relation among the input and output files and the execution of PHASTINPUT and PHAST within the script. The script is invoked from a command line with either one or two arguments:

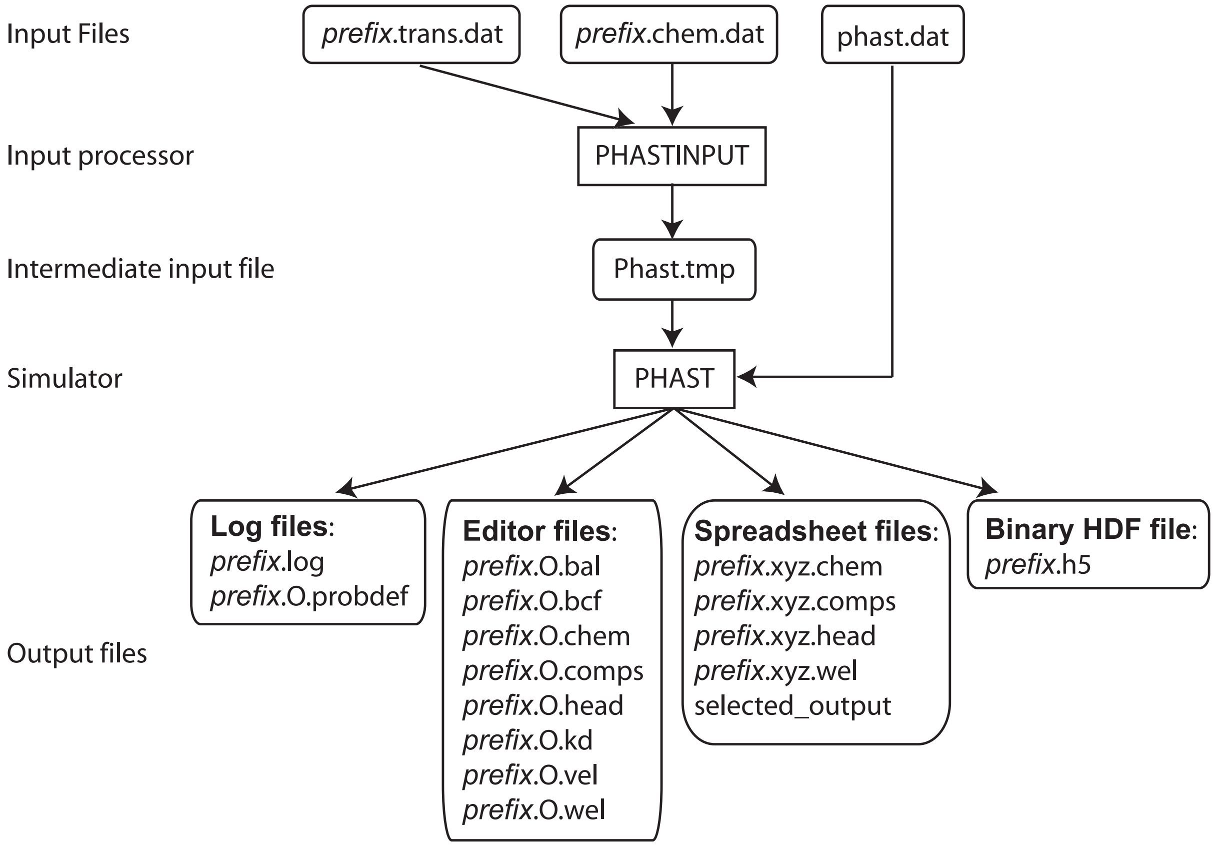


Figure 2.1.—The relation between input and output files and the execution of the two programs PHASTINPUT and PHAST.

phast *prefix* [*database_name*]

where the first argument is the prefix and the optional second argument is the thermodynamic database file name. The input files for transport and chemistry must be present in the working directory and must be named *prefix***.trans.dat** and *prefix*.**chem.dat**. If the thermodynamic database file name is not specified as the second argument, the file *phast.dat* in the database subdirectory of the installation directory for PHAST will be used. If the thermodynamic database file name is specified, the file must be present in the working directory or be specified with a complete pathname. PHASTINPUT converts time-series data read in the *prefix***.trans.dat** to a series of simulation periods over which boundary conditions and print frequencies are constant. Static data and data for each of these simulation periods are written to the file *Phast.tmp*.

When the PHAST simulator is run, the program executes the following steps: (1) The first part of the intermediate file *Phast.tmp* is read and static flow and transport data are initialized. (2) An initial call to PHREEQC is executed, which reads the thermodynamic database file, and then reads and performs the calculations specified in the chemistry data file. (3) Additional static flow and transport data are read from *Phast.tmp*, and, if requested, a steady-state flow simulation is performed. (4) Transient data for the first simulation period are read from *Phast.tmp* and the flow (if not steady state), transport, and reaction calculations are performed for the first simulation period. (5) Step 4 is repeated for each simulation period that is defined. The file *selected_output* is written (if requested) only during step 2, the initial call to PHREEQC; all other output files contain information from steps 3 and 4. The logical flow of the calculation is described further in the section D.7.1. Operator Splitting and Sequential Solution.

Chapter 3. Thermodynamic Database and Chemistry Data Files

When the program PHAST is invoked, the thermodynamic database file and the chemistry data file are used to define the chemical model, the initial and boundary-condition solutions, and the pure phases, ion exchangers, surface assemblages, and kinetic reactions that are present in each cell of the simulation region. Complete descriptions of the keyword data blocks for these files are found in the PHREEQC manual (Parkhurst and Appelo, 1999) and are not presented in this manual. Both of the chemistry data and thermodynamic database files for PHAST have identical formats to PHREEQC input and database files, and they can be generated or modified by using an interface to PHREEQC. Two interfaces are available: PHREEQCI (Charlton and Parkhurst, 2002; http://wwwbrr.cr.usgs.gov/projects/GWC_coupled/phreeqc/index.html) and PHREEQC For Windows (http://www.geo.vu.nl/users/posv/phreeqc/index.html).

The thermodynamic database file is used to specify static data for the chemical model, which includes a list of chemical elements; chemical-reaction equations that define aqueous, exchange, surface, and solid species; and the equilibrium constants for the chemical reactions. The chemistry data file is used to specify solution compositions and compositions of reactants (exchangers, minerals, surfaces, kinetic reactants, solid solutions), either directly, or indirectly through specification of geochemical reactions. Solutions, pure-phase assemblages, ion exchangers, surface assemblages, and kinetic compositions that are specified in the chemistry data file (either directly or indirectly) are used to define the initial and boundary conditions, and the sets of reactants that are present in each cell of the simulation region. Another important function of the chemistry data file is to select chemical data that are written to output files, such as pH, molalities of aqueous species, and amounts of minerals and kinetic reactants. The chemistry data file also can be used to modify the chemical model that is defined in the thermodynamic database file.

3.1. Thermodynamic Database File

The thermodynamic database file is used to specify the reactions and thermodynamic properties of the aqueous and solid chemical species that comprise the chemical model of PHAST. Although any keyword data block can be included in the thermodynamic database file, commonly, the file contains the keyword data blocks listed in table 3.1, which are used to define basic thermodynamic properties. These keyword data blocks define master species, reaction stoichiometry, and equilibrium constants for all of the aqueous-phase species, exchange species, surface species, and pure phases that can be modeled with PHAST. Four thermodynamic database files are provided with the program: the default thermodynamic database file, *phast.dat*, which is derived from PHREEQE (Parkhurst and others, 1980), and is the same as *phreeqc.dat* in the PHREEQC distribution; *wateq4f.dat*, which is derived from WATEQ4F (Ball and Nordstrom, 1991), and is consistent with *phreeqc.dat* but contains more trace elements; *minteq.dat*, which is derived from MINTEQA2 (Allison and others, 1990); and *llnl.dat*, which is the most comprehensive of the thermodynamic database files, and is derived from thermodynamic data (designation thermo.com.v8.r6.230) assembled by Lawrence Livermore National Laboratory (Wolery, 1992a, 1992b; Bethke, 1996; and Johnson and others, 1991).The elements and element valence states that are included in *phast.dat* are listed in table 3.2 along with the PHAST notation and the default formula for gram-formula weight used to convert mass to moles.

Table 3.1.—Keyword data blocks commonly used in the thermodynamic database file

Keyword data block	Function
EXCHANGE_MASTER_SPECIES	Defines names of exchange sites
EXCHANGE_SPECIES	Defines reactions for exchange sites
SOLUTION_MASTER_SPECIES	Defines names of elements
SOLUTION_SPECIES	Defines aqueous species
SURFACE_MASTER_SPECIES	Defines names of surface sites
SURFACE_SPECIES	Defines reactions for surface sites
PHASES	Defines mineral and gas dissociation reactions

Table 3.2.—Elements and element valence states included in default thermodynamic database *phast.dat*, including PHAST notation and default formula for gram formula weight

[For alkalinity, formula for gram equivalent weight is given]

Element or element valence state	PHAST notation	Formula used for default gram formula weight
Alkalinity	Alkalinity	$Ca_{0.5}(CO_3)_{0.5}$
Aluminum	Al	Al
Barium	Ba	Ba
Boron	B	B
Bromide	Br	Br
Cadmium	Cd	Cd
Calcium	Ca	Ca
Carbon	C	HCO_3
Carbon(IV)	C(4)	HCO_3
Carbon(-IV), methane	C(-4)	CH_4
Chloride	Cl	Cl
Copper	Cu	Cu
Copper(II)	Cu(2)	Cu
Copper(I)	Cu(1)	Cu
Fluoride	F	F
Hydrogen(0), dissolved hydrogen	H(0)	H
Iron	Fe	Fe
Iron(II)	Fe(2)	Fe
Iron(III)	Fe(3)	Fe
Lead	Pb	Pb
Lithium	Li	Li
Magnesium	Mg	Mg
Manganese	Mn	Mn
Manganese(II)	Mn(2)	Mn
Manganese(III)	Mn(3)	Mn
Nitrogen	N	N
Nitrogen(V), nitrate	N(5)	N
Nitrogen(III), nitrite	N(3)	N
Nitrogen(0), dissolved nitrogen	N(0)	N
Nitrogen(-III), ammonia	N(-3)	N
Oxygen(0), dissolved oxygen	O(0)	O
Phosphorous	P	P
Potassium	K	K
Silica	Si	SiO_2
Sodium	Na	Na
Strontium	Sr	Sr
Sulfur	S	SO_4
Sulfur(VI), sulfate	S(6)	SO_4
Sulfur(-II), sulfide	S(-2)	S
Zinc	Zn	Zn

3.2. Chemistry Data File

The chemistry data file defines the solutions, gas phases, and pure-phase, exchange, and surface assemblages and geochemical calculations that are used in initial and boundary conditions of the reactive-transport simulation. Also defined in the chemistry data file is the set of chemical data to be written to the *prefix***.h5** and *prefix***.xyz.chem** output data files. The keyword data blocks most commonly used in the chemistry data file to define initial and boundary conditions are listed in table 3.3. For details on the formats and options for these and all other data blocks that can be included in the chemistry data file (and thermodynamic database file), it is necessary to refer to the PHREEQC manual (Parkhurst and Appelo, 1999).

3.2.1. Chemical Initial and Boundary Conditions for Reactive Transport

Initial conditions for the reactive-transport simulation are defined with **SOLUTION**, **EXCHANGE**, **KINETICS**, **SURFACE**, **EQUILIBRIUM_PHASES**, **SOLID_SOLUTIONS**, and **GAS_PHASE** data blocks. Solution compositions and speciation calculations are defined with the **SOLUTION** data block. The composition of an exchange assemblage is defined with the **EXCHANGE** data block; sets of kinetic reactions are defined with the **KINETICS** data block, and rate expressions for the kinetic reactions are defined with the **RATES** data block; the composition of a surface assemblage is defined with the **SURFACE** data block; and the identity and amount of each phase in a pure-phase assemblage is defined with the **EQUILIBRIUM_PHASES** data block. The composition of a fixed-total-pressure or fixed-volume multicomponent gas phase is defined with the **GAS_PHASE** data block. (The gas phase feature is rarely used; gases are usually defined as fixed partial pressure phases in **EQUILIBRIUM_PHASES** data blocks.) Index numbers are defined for each solution, equilibrium-phase assemblage, exchange assemblage, gas phase, kinetic-reactant set, solid-solution assemblage, and surface assemblage. These index numbers are used in the flow and transport data file to specify initial and boundary conditions by zones. For example, solutions defined by different index numbers can be applied to a series of zones to define the spatially varying initial composition of water within an aquifer.

The units for input in the **SOLUTION** data block are concentration units, which are converted to molality for all chemical calculations. The units for input in the data blocks **EXCHANGE**, **KINETICS**, **SURFACE**, **EQUILIBRIUM_PHASES**, **SOLID_SOLUTIONS** are extensive (moles). For each cell in PHAST, the representative porous-medium volume for chemistry contains one kilogram of water, when saturated. Thus, when defining amounts of solid-phase reactants, the appropriate number of moles is numerically equal to the concentration of the reactant (moles per kilogram water, mol/kgw), assuming a saturated porous medium. Note that the concentration of the solid reactants varies spatially as porosity varies, so that if porosity is defined to vary spatially, then logically, the concentrations of solid-phase reactants also should be defined to vary spatially.

For unconfined flow (free-surface) calculations, the volume of water in a cell may vary. Thus, it is important to define the rate expressions for kinetic reactions to account for the changing volume of solution. Rate expressions for kinetic reactions introduce moles of reactants; thus, if a known rate expression has units of moles per kilogram water, it is necessary to multiply by the mass of water present [TOT("water") is the Basic-language function for mass of water] to calculate the proper number of moles of reaction to add to solution (see **KINETICS** data block in Parkhurst and Appelo, 1999).

Reaction calculations specified in the chemistry data file may be used to define initial or boundary conditions. These reaction calculations are processed during the initial call to the PHREEQC module. Reactions are defined by allowing a solution or mixture of solutions to come to equilibrium with one or more of the following reactants: an exchange assemblage, a surface assemblage, a pure-phase assemblage, a solid-solution assemblage, or a multicomponent gas phase. In addition, irreversible reactions, kinetic reactions, and reaction temperature can be specified for reaction calculations. (Although temperature does not affect flow, it does affect chemical reactions. See section D.6. Initial Conditions.) Previously defined reactants can be used in reaction calculations by **USE** data blocks. The composition of the solution, exchange assemblage, surface assemblage, pure-phase assemblage, or gas phase can be saved after a reaction calculation with the **SAVE** keyword for subsequent use in reaction calculation or for initial and boundary conditions. The **SAVE** keyword assigns an index number to the specified composition. These index numbers are used in the flow and transport file to define initial and boundary conditions for specified zones within the simulation region.

Table 3.3.—Keyword data blocks commonly used in the chemistry data file

[—, index number is not used in flow and transport data file]

Keyword data block	Function of data block in initial call to PHREEQC module	Keyword data block using index number in the flow and transport data file
END	Marks the end of definitions for initial and reaction calculations	—
EQUILIBRIUM_PHASES *index*	Defines a mineral assemblage	**CHEMISTRY_IC**
EXCHANGE *index*	Defines an assemblage of ion exchangers	**CHEMISTRY_IC**
GAS_PHASE *index*	Defines a gas phase. This option is rarely used because gas transport is not simulated. More commonly, gases are treated with fixed partial pressures in **EQUILIBRIUM_PHASES**.	**CHEMISTRY_IC**
KINETICS *index*	Defines a set of kinetic reactions	**CHEMISTRY_IC**
RATES	Defines rate expressions for kinetic reactions	—
REACTION *index*	Adds specified amounts of reactants	—
SAVE solution *index*	Saves solution composition that is the result of a reaction simulation	**CHEMISTRY_IC, FLUX_BC, LEAKY_BC, SPECIFIED_HEAD_BC, RIVER, WELL**
SAVE exchange *index* **SAVE equilibrium_phases** *index* **SAVE gas_phase** *index* **SAVE kinetics** *index* **SAVE solid_solution** *index* **SAVE surface** *index*	Saves reaction simulation results to define new compositions of ion exchangers, equilibrium phases, gas phase, kinetic reactants, solid solutions, and surfaces	**CHEMISTRY_IC**
SELECTED_OUTPUT	Defines data to be written to the *prefix*.**h5** and *prefix*.**xyz.chem** files	—
SOLID_SOLUTIONS *index*	Defines an assemblage of solid solutions	**CHEMISTRY_IC**
SOLUTION *index*	Defines a solution composition	**CHEMISTRY_IC, FLUX_BC, LEAKY_BC, SPECIFIED_HEAD_BC, RIVER, WELL**
SURFACE *index*	Defines sets of surface complexation reactions	**CHEMISTRY_IC**
TITLE	Defines a character string to annotate the calculations	—
USE *keyword index*	Selects solution, equilibrium phases, exchanger assemblage, gas phase, kinetic reactants, reaction, solid-solution assemblage, or surface assemblage to be used in a reaction calculation	—
USER_PUNCH	Defines data to be written to the *prefix*.**h5** and *prefix*.**xyz.chem** files	—

Some keyword data blocks from PHREEQC that are less commonly used have been omitted from tables 3.1 and 3.3, but all of the functionality of PHREEQC is available in the initial reaction calculations by the PHREEQC module, including all reaction and one-dimensional transport and advection capabilities. Thus, it is possible to use a reaction calculation to define a water composition that is produced by equilibrating pure water with calcite and dolomite, which can then be used as the initial water composition for an aquifer. It also is possible, for example, to simulate the percolation of a contaminant through a one-dimensional column that represents a confining layer, and save the water compositions of the column effluent to be used as time-varying input through a flux boundary condition.

3.2.2. Output of Chemical Data

PHAST simulations produce a huge amount of chemical data—including, but not limited to, the molality and activity of every aqueous species, exchange species, and surface species; the moles of every reactive mineral; the saturation index for each of a large number of minerals; and the mole transfer of every kinetic reactant—for every cell for every time step. The **SELECTED_OUTPUT** and **USER_PUNCH** data blocks are used to select a subset of the chemical results to be written to the *prefix***.h5** and *prefix***.xyz.chem** output data files. The **SELECTED_OUTPUT** data block allows selection of certain types of chemical data to be written to these two output files, including pH; pe; total molalities of specified elements; molalities and activities of specified aqueous, exchange, and surface species; saturation indices and partial pressures of specified minerals and gases; amounts and mole-transfers of equilibrium phases and kinetic reactants; and moles of gas components in the gas phase.

The **USER_PUNCH** data block provides another more versatile method to select data for the two output files. The **USER_PUNCH** data block uses an embedded Basic-language interpreter to allow calculation of chemical quantities that can then be written to the output files. The Basic-language interpreter has a full range of programming capabilities, including variables, mathematical functions, FOR loops, and subroutines, which allows calculation of almost any chemical quantity that is needed in the output files. During the reaction calculation for a cell at any given time step in the reactive-transport simulation, virtually all of the information related to the chemical calculation for the cell is available through Basic-language functions that can be used in calculations—molalities, activities, ionic strength, equilibrium constants, moles of minerals, among others. The results of these calculations can be written to the output file with a Basic-language command (PUNCH). Thus, for example, it is possible to calculate the number of grams of calcite in a cell from the number of moles of calcite, and write the mass to the two output files *(prefix***.h5** and *prefix***.xyz.chem**). Another example is to calculate the sodium sorption ratio (SAR) from the concentrations of the major cations, and write the result to the output files.

The *selected_output* file also may contain data as defined by **SELECTED_OUTPUT** and **USER_PUNCH**. However, the file only will contain results related to the first call to PHREEQC, which precedes all steady-flow, active-grid initialization, and reactive-transport calculations. Frequently, a spreadsheet file of this preliminary calculation is not useful, and the **SELECTED_OUTPUT** data block is located at the end of the *prefix***.chem.dat file**, following an **END** keyword, which results in no data being written to the *selected_output* file.

If necessary, the chemistry data file can be used to modify or augment the thermodynamic database file. If new elements, aqueous species, exchange species, surface species, or phases need to be included in addition to those defined in the thermodynamic database file, or if the stoichiometry, equilibrium constant, or activity coefficient information from the thermodynamic database file needs to be modified for a simulation, then the keyword data blocks listed in table 3.1 can be included in the chemistry data file. The data read for these data blocks in the chemistry input file will augment or supersede the data read from the thermodynamic database file. In most cases, the aqueous model and thermodynamic data for phases defined in the thermodynamic database file will not be modified, and the keywords listed in table 3.1 will not be used in the chemistry data file.

Chapter 4. Flow and Transport Data File

The flow and transport data file provides information necessary to simulate ground-water flow and solute transport. The file contains the definition of the grid, porous-media properties, initial conditions, boundary conditions, and time step. Selected properties, for example well pumping rate, boundary-condition heads, and solution compositions, may vary over the course of a simulation by definition of time series for these data.The flow and transport data file also is used to specify the units of measure used for input data, print frequencies for output files, and information related to the formulation and solution of the flow and transport finite-difference equations. All chemical compositions in the flow and transport data file, which are related to initial conditions, boundary conditions, and chemical reactions, are specified by index numbers that correspond to complete definitions in the chemistry data file (section 3.2. Chemistry Data File).

It requires only a single change in the flow and transport data file to switch from a flow, solute transport, and reaction simulation to a flow-only simulation. For flow-only simulations, the flow and transport data file is the only input file necessary for PHAST (the thermodynamic database file and chemistry data file are not used) and all data related to transport and chemistry in the flow and transport data file are ignored.

In this documentation, the term “property” refers to any numerical value used in the definition of a porous-media property, initial condition, or boundary condition. In the flow and transport file, spatially distributed properties are defined by rectangular prisms called zones. Multiple overlapping zones may be used to assemble spatially varying distributions of properties. Time-series data in boundary-condition definitions allow for changes in flow and chemical boundary conditions over the course of a simulation.

All of the data in the flow and transport data file are defined through the keyword data blocks listed in table 4.1. This chapter contains complete descriptions of data input for all of these keyword data blocks.

Table 4.1.—Keyword data blocks for the flow and transport data file

Keyword data block	Function
CHEMISTRY_IC	Defines initial conditions for chemistry, and types of reactions for the active grid region
END	Marks the end of definitions for the flow and transport data file
FLUX_BC	Defines flux boundary conditions
FREE_SURFACE_BC	Defines the simulation to have a free-surface boundary condition or not
GRID	Defines node locations
HEAD_IC	Defines initial head conditions
LEAKY_BC	Defines leaky boundary conditions
MEDIA	Defines porous media properties—porosity, hydraulic conductivity, specific storage, and dispersivity
PRINT_FREQUENCY	Defines time intervals at which data are written to the output files
PRINT_INITIAL	Writes initial conditions to the output files
PRINT_LOCATIONS	Defines zones for which data will be written to the *prefix*.**xyz.chem** and *prefix*.**O.chem** files
RIVER	Defines a river boundary condition
SOLUTE_TRANSPORT	Defines the simulation to be only a flow-only simulation or a complete reactive-transport simulation
SOLUTION_METHOD	Defines the numerical method for solving the finite-difference equations
SPECIFIED_HEAD_BC	Defines specified-head boundary conditions
STEADY_FLOW	Defines the flow field to be steady state as determined from the boundary conditions
TIME_CONTROL	Defines time step and specified times for the end of simulation periods
TITLE	Defines a character string describing the simulation
UNITS	Defines units of input data
WELL	Defines the location, open intervals, and pumping or injection rates for a well

4.1. Organization of the Flow and Transport Data File

PHAST uses keyword data blocks in the flow and transport data file to define static properties, initial conditions, and time-varying properties. Some of the data blocks define static data that apply unchanged for the entire simulation, whereas some of the data blocks are used to define both static data and transient data.

Data blocks that define the input units, grid, porous-media properties, initial conditions, numerical methods, steady-state flow, the selection of cells for writing output data, and the free-surface boundary condition apply to the entire simulation. These data blocks and the data blocks for title and printing initial conditions contain no time-varying properties. The data blocks that define only static information are **CHEMISTRY_IC**, **FREE_SURFACE_BC**, **GRID**, **HEAD_IC**, **MEDIA**, **PRINT_INITIAL**, **PRINT_LOCATIONS**, **SOLUTE_TRANSPORT**, **SOLUTION_METHOD**, **STEADY_FLOW**, **TITLE**, and **UNITS**.

Data blocks for boundary conditions include both static and transient data. The boundary-condition data blocks assign static data—for example, the type and location of all boundary conditions, the thickness of a leaky boundary, the diameter of a well, or the hydraulic conductivity of a riverbed—and transient data, which include time-series property definitions for head, solution composition, flux, and injection/pumping rate. The transient data are properties that are allowed to vary over the course of a simulation; however, the static data cannot be changed. The boundary-condition data blocks that include both static and transient data are **FLUX_BC**, **LEAKY_BC**, **RIVER**, **SPECIFIED_HEAD_BC**, and **WELL**. The **TIME_CONTROL** data block must be included in every simulation to define the time step for integration (possibly time varying), specified times to end simulation periods, and the end time for the simulation. The data block **PRINT_FREQUENCY** may include time-varying print controls.

Simulation periods are intervals of time over which all time-varying properties and print frequencies are constant. Every time any time-varying property or print frequency changes value, the old simulation period ends and a new simulation period is begun. By default, the end of a simulation period is a time when data are written to output files, but this behavior can be changed by the options in the keyword data block **PRINT_FREQUENCY.**

With one major exception, the order of keyword data blocks is arbitrary within a set of data blocks that defines a simulation period. The exception is that the order of the keyword data blocks and the order of definitions within data blocks are important for the definition of spatial data for media properties, boundary conditions, and initial conditions. For these data items, the data are assigned to cells and elements in sequential order, with subsequent definitions overwriting preceding definitions. For a given location in the grid region, the last definition in the data file that applies to that location is the one used, as described below (also see section D.5.7. Boundary-Condition Compatibility).

4.2. Spatial Data

The coordinate system for PHAST is three dimensional; X and Y represent horizontal axes and Z represents the vertical axis (allowing no tilt with respect to the gravity vector). The relation among X, Y, and Z is right-handed, in the sense that if your right thumb is pointed in the positive Z direction, then your fingers curl from the positive X axis to the positive Y axis. To establish a grid, a set of coordinate locations for nodes is defined along each axis—X, Y, and Z. The grid is the set of nodes defined by every combination from the sets of X, Y, and Z coordinate locations. Thus, nodes are distributed along a series of lines and planes parallel to the coordinate axes. The volume that encloses the grid nodes is referred to as the grid region.

The nodes define two sets of spatial volumes, called elements and cells. An element is a rectangular prism defined by exactly eight nodes that are located at the element corners (fig. 4.1). The set of elements fills the entire grid region. To conform to non-rectangular simulation regions, it is possible to eliminate parts of the grid region by designating selected elements to be inactive. The volume defined by the set of active elements is referred to as the active grid region. All nodes associated with active elements are referred to as active nodes. The volume defined by the set of inactive elements is referred to as the inactive grid region. Nodes interior to the inactive grid region are referred to as inactive nodes. Simulation results are calculated for all active nodes, including nodes internal to and on the boundary of the active grid region. Results are not calculated for any inactive nodes.

Every node in the active grid region is contained within a single cell. Cell faces either bisect the distance between adjacent nodes or lie along a plane of nodes that bounds the active grid region (fig. 4.2). The volume of a cell is composed of parts of each of the elements that meet at the node contained within the cell. Only active elements contribute volumes to a cell, so each cell contains parts of one to eight active elements. Cells are full cells (or interior cells) when the cell contains parts of eight active elements. Cells are 1/2, 1/4, or 1/8 cells if one, two, or three faces, respectively, lie along planes of nodes that bound the grid region. These and other fractional cell configurations occur at the boundaries of irregularly shaped active grid regions. Cells are undefined for nodes within the inactive grid region.

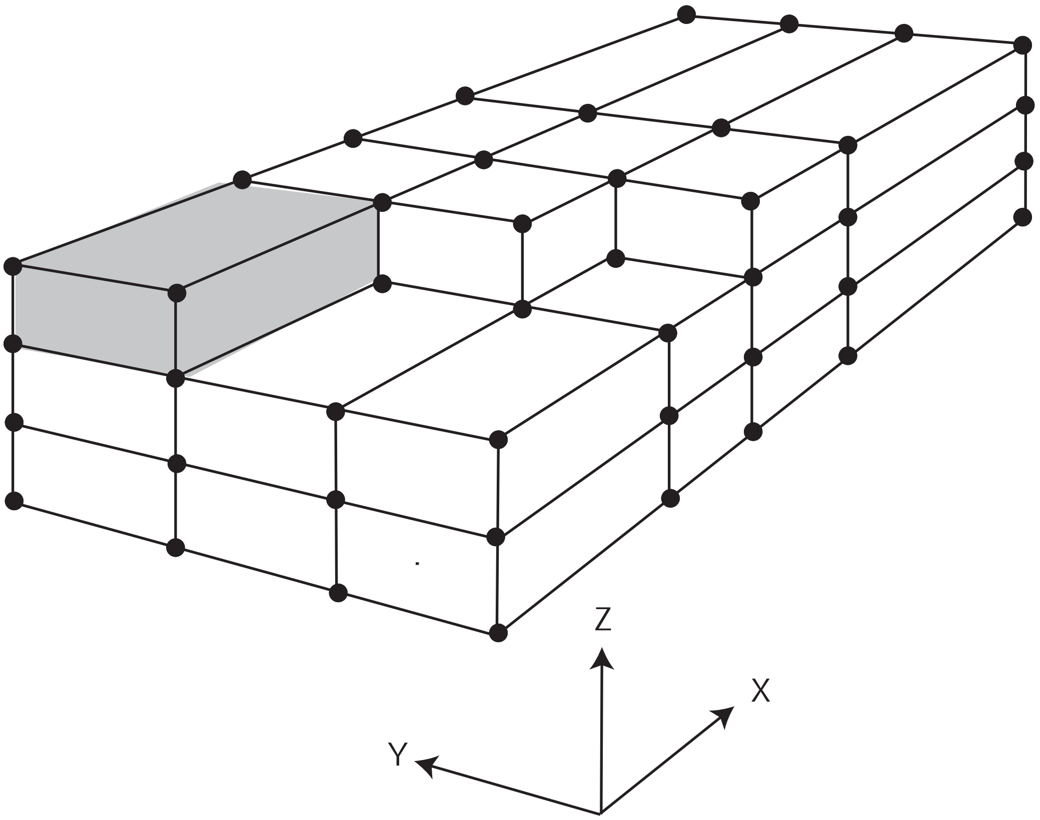


Figure 4.1.—The grid region, boundary nodes (dots), and one shaded element.

The “natural” order of elements and cells begins in the left, front, bottom corner of the grid region. Element and cell numbers increase by cycling through X, then Y, and then Z coordinates. This natural order is required whenever node-by-node input is used to define properties for a simulation.

4.2.1. Zones

Zones are rectangular prisms that are used to distribute property values within the grid region. A zone is defined by two sets of coordinates, the location of the left, front, lower corner (minimum X, minimum Y, and minimum Z values) and the location of the right, back, upper corner (maximum X, maximum Y, and maximum Z values) of a rectangular prism.

As spatial data are entered for nodes or elements, it is permissible to use overlapping zones. For example, the initial-condition head for a given node could be defined by multiple zones, possibly in multiple **HEAD_IC** data blocks. The initial head that is used for the given node in the flow simulation will be the head from the last zone in the flow and transport data file that contains that node and defines the initial head property. The same logic applies to all properties, including boundary-condition type; the last property value defined for a node will be the one used in the simulation.

It is possible to apply properties to a subset of nodes or elements within a zone by using a mask. A mask associates a 0 (or any nonpositive number) or a 1 (or any positive number) with each node in the zone. Properties are applied only to those nodes or elements with positive values of the mask. Using a mask may simplify the definition of properties that are associated with non-rectangular sets of nodes, for example a circular feature. Defining a single zone with a mask that limits the property to the nodes or elements in the circular feature may be easier than defining a large number of rectangular zones to approximate the circular feature. Masks have the disadvantage that they are grid dependent; if the grid is changed so that the number of nodes in a zone changes, then the sequence of 0s and 1s in the mask also must be changed, which is not a trivial task.

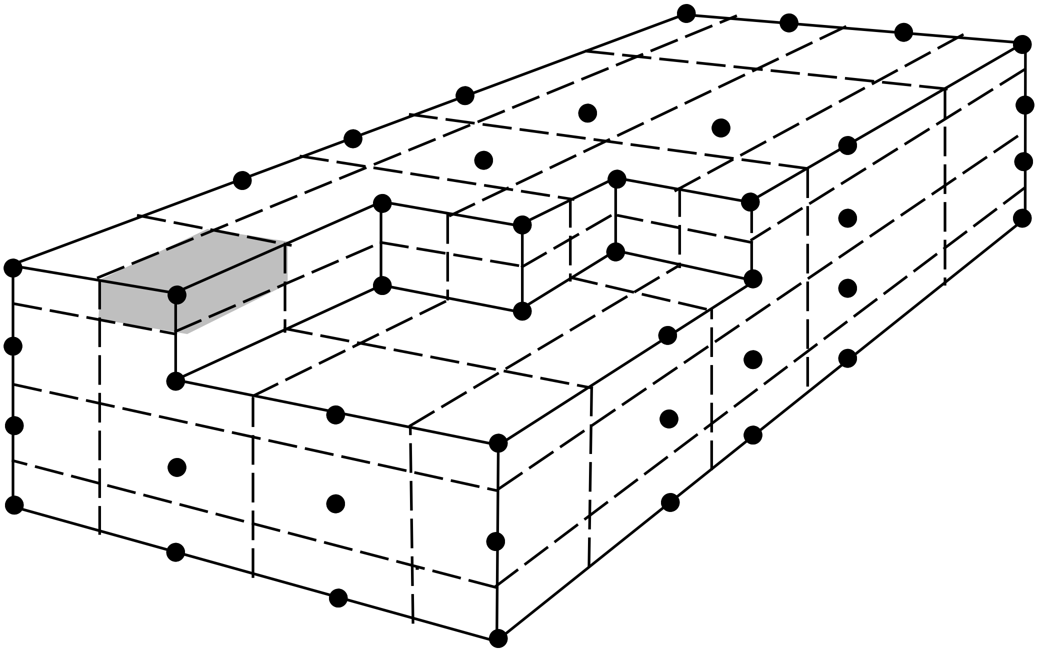


Figure 4.2.—The grid region, boundary of the active grid region (solid lines), boundary nodes (dots), cell boundaries (dashed lines), and one shaded cell.

4.2.1.1. Use of Zones for Defining Porous-Media Properties

Porous-media properties, including porosity, hydraulic conductivity, specific storage, dispersivity, and active or inactive zones, must be defined for each element. For porous-media properties, the definition of a zone selects each element whose centroid is contained in the zone. An example in two dimensions of the selection of elements by a zone is shown in figure 4.3. The properties defined for the zone will be applied to each of the elements that is selected.

Porous-media properties for a cell are defined by aggregating the properties of the active elements that are contained in the cell (fig. 4.4). A cell contains parts of one to eight active elements, thus a cell property may be derived from as many as eight different values. Porosity and specific storage are averaged in proportion to the volume of each active element that is contained in the cell to arrive at the porous-media property for the cell. Hydraulic conductivity and dispersivity are calculated for each of the six faces of a cell (for an internal cell) by averaging the properties of the active elements that contain the cell face. The averages are calculated by weighting the element properties in proportion to the area of the cell face that is within each element. A cell face is located within one to four active elements; thus cell face conductances are a weighted average derived from properties from as many as four different elements (fig. 4.4).

Frequently, it is convenient to define a uniform value for a porous-media property by using a zone that contains the entire grid region. It is permissible to use a zone that is larger than the grid region. Additional zones may then be defined that overlay different property values in different parts of the grid region to obtain a complete definition of the spatial distribution of the property. Zones may be used to define elements within the grid region to be inactive.

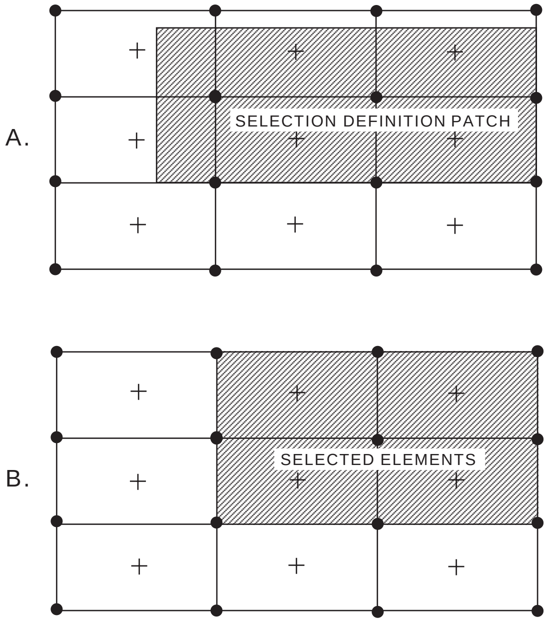


Figure 4.3.—(A) Rectangular zone used to select elements from a two-dimensional grid, and (B) the elements selected by the zone. Lines represent element boundaries.

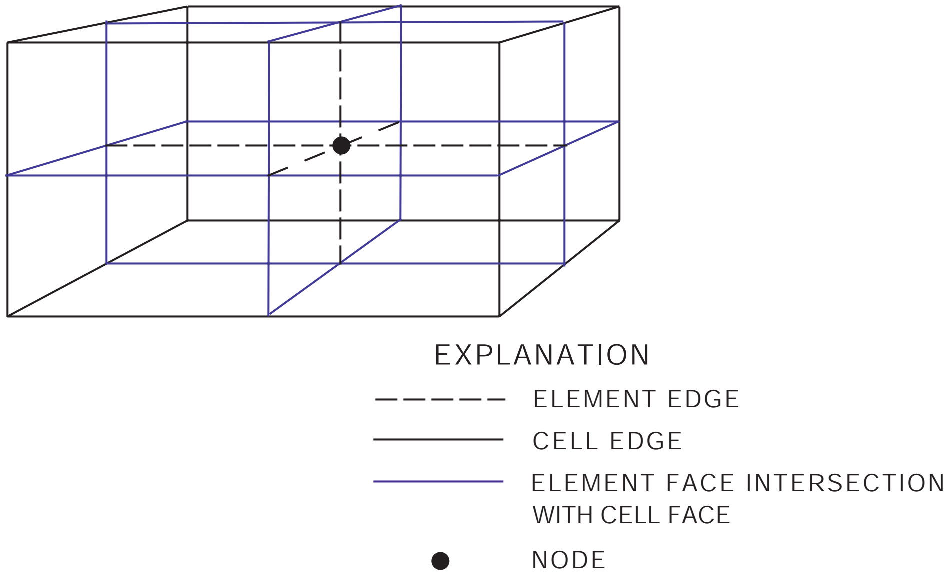


Figure 4.4.—The parts of eight elements contained within a cell and the parts of four elements located within each cell face.

4.2.1.2. Use of Zones for Defining Initial- and Boundary-Condition Properties

Initial conditions and specified-head, leaky, and flux boundary conditions are assigned by node. For these properties, the definition of a zone selects each node contained within the zone. An example in two dimensions of the selection of nodes by a zone is shown in figure 4.5. A zone also selects the set of cells that are associated with active nodes within the zone. There may be fewer cells selected than nodes selected because some of the nodes may be inactive and have no associated cell. The properties defined for the zone will be applied to each of the cells selected.

Zones must be used to define initial conditions for every node in the active grid region. Zones also are used to define boundary conditions where desired within the grid region. The definitions of well and river boundary conditions do not use zones.

For specified-flux and leaky boundary conditions, the set of nodes selected by zones must be zero, one, or two dimensional. Three-dimensional sets of nodes are not allowed for these boundary conditions because they logically apply only to cells that bound the active grid region. The sign of the flux quantity indicates whether the flux is in the positive or negative coordinate direction.

Multiple zones of each boundary-condition type may be used to specify the complete set of boundary conditions. Different boundary conditions for a single cell and its cell faces may be defined multiple times as part of different zone definitions and different keyword data blocks. The boundary condition that is used for a cell or cell face is taken from the last zone definition in the flow and transport data file that defines a boundary condition for that cell or cell face, subject to restrictions on the co-occurrence of boundary conditions (see section D.5.7. Boundary-Condition Compatibility).

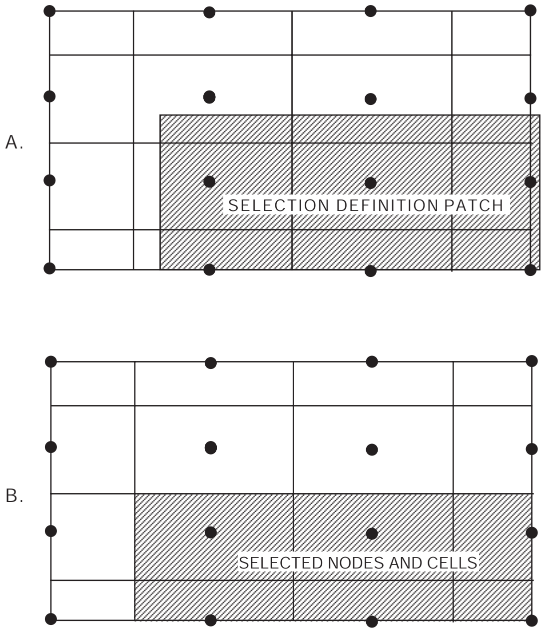


Figure 4.5.—(A) Rectangular zone used to select nodes from a two-dimensional grid, and (B) the nodes and cells selected by the zone. Lines represent cell boundaries.

4.2.1.3. Property Definitions

For each zone, spatially distributed properties can be defined by as many as five methods: (1) a single value can be applied to each node or element that is selected by the zone, (2) a list of values in natural order (section 4.2. Spatial Data) can be defined that includes exactly one value for each node or element that is selected by the zone, (3) a list of values in natural order can be read from a file that includes exactly one value for each node or element that is selected by the zone, (4) values for the property can be linearly interpolated along a specified coordinate axis by using property values at two points along the specified axis, and (5) chemical compositions can be defined as mixtures of two specified end-member compositions by defining a mixing fraction for each node that is selected by the zone. When method 4 is used to distribute properties to nodes, the node location is used for interpolation; for elements, the element centroid is used for interpolation. If the node or centroid falls outside the coordinate range defined by the two points, the value from the closest point is applied to the node or element. When method 4 is used for any chemical property, such as composition of solution, kinetic reaction, or pure-phase assemblage, the compositions specified for the two points are mixed in proportion to the distance of the node to the two points or, if the node is outside the coordinate range defined by the two points, the composition from the closest point is applied.

For methods 2, 3, and 5, a value is supplied for each node or element selected by the zone, regardless of whether the node or element is active or inactive. The property value only has effect for the active nodes or active elements; property values supplied for inactive nodes or elements are ignored.

4.2.2. Rivers and Wells

A river is defined by a series of X–Y coordinate points that locate the centerline of the river across the land surface. For each river point, the river width, river-bed-leakance parameters, and a time series of head values must be defined explicitly or implicitly by interpolation from other river points. For a vertical stack of cells, river leakage will be applied to the cell that contains the water table (free surface). Well location is defined by the X–Y coordinates of the well head. Data required for each well include depth or elevation data to define open (or screened) intervals, well diameter, numerical method for allocating flow to grid layers, and a time series of pumping or injection rate.

4.3. Transient Data

PHAST allows time series of data to be defined for certain boundary-condition parameters, time step, and print frequencies. Each time one of these values changes, a new simulation period is begun. An unlimited number of simulation periods are allowed. During a single simulation period, boundary conditions, time step, and print frequencies are constant. Properties for boundary conditions that may be transient include head, solution composition, flux, and injection or pumping rate. The following boundary-condition properties are defined by a time series of data: the head and solution composition associated with a specified-head or leaky boundary condition, the flux and solution composition associated with a flux boundary condition, the head and solution composition associated with a river point, and the pumping rate and solution composition associated with a well. Many properties of boundary conditions are static; that is, they are fixed for the entire simulation. Examples of static properties include the hydraulic conductivity in a leaky boundary condition, the width of a river associated with a river point, and the locations of the screened intervals of a well.

4.4. Description of Keyword Data Blocks

Each type of data in the flow and transport data file is defined through a keyword data block. A set of identifiers is used to organize the data that are included in the data block. In this section, each keyword data block and its identifiers are described. The initial paragraphs of this section describe conventions that are used in the flow and transport data file and in the documentation of the keyword data blocks.

The flow and transport data file is completely case insensitive. Data for all data blocks are free format; one or more spaces or tabs are used to delimit input fields. Usually the fields for an identifier must be defined on a single line, but data may be extended to multiple lines in some instances.

In this document, words (or characters) in bold must be included literally when creating input data files (although optional spellings may be permitted). Keywords are always bold uppercase (for example, **CHEMISTRY_IC**); identifiers are bold lowercase and preceded with a hyphen (for example, **-zone**). Words in *italics* are input fields that are variable and depend on user selection of appropriate values (for example, **-zone** x_1, y_1, z_1, x_1, y_2, z_2; the user must insert appropriate coordinates to define the zone). Items in brackets ([]) are optional input fields (for example, in the identifier **-u**[**niform**], everything in the brackets is optional). Mutually exclusive input fields are separated by the word "or" (for example, [*True* or *False*]). In general, the optional fields must be entered in the specified order.

For clarity, commas are used to delimit input fields in the explanations of data input; however, commas are not used in the flow and transport data file. All line numbers associated with example input for the flow and transport data files in the documentation are for identification purposes only. (Input data files for PHAST do not contain any commas or line numbering except in Basic language programs within the chemistry data file.) If multiple lines of the same type are present in an example, an additional letter is included to give the line a unique identification.

The following definitions are used in the flow and transport file:

Keywords—Input data blocks are identified with an initial keyword. This word must be spelled exactly. Several of the keywords have synonyms. For example **SPECIFIED_BC** is a synonym for **SPECIFIED_HEAD_BC**.

Identifiers—Identifiers denote the type of data within a keyword data block. Identifiers have two forms: (1) they are spelled completely and exactly or (2) they start with a hyphen followed by enough characters to define the identifier uniquely. The form with the hyphen is always acceptable. Usually, the form without the hyphen is acceptable, but in some cases the hyphen is needed to indicate the word is an identifier rather than an identical keyword. In this documentation, the hyphen is used in the flow and transport data file in all explanations and examples. Note that the hyphen in an identifier never indicates negativity.

Units—In this documentation, the units are identified by the abbreviations "L", length and "T", time. In nearly all cases, the actual units of input for the flow and transport data file are defined by specifications in the **UNITS** data block. The PHAST simulator accepts SI and most U.S. customary units for input. However, PHAST uses SI units internally and all output is in SI units, with the exception of time units. The time unit used for output-file results is selected by the user.

Comments—The number symbol (#) is used to delimit the beginning of a comment in an input file. All characters in the line that follow this character are ignored. The "#" is useful for adding comments explaining the source of various data or describing the problem set up. In addition, it is useful for temporarily removing lines from an input file. If the entire line is a comment, the line is not echoed to the log file (*prefix*.**log**). If the comment follows input data on a line, the entire line, including the comment, is echoed to the log file.

Logical line separator—A semicolon (;) is interpreted as a logical end-of-line character. This allows multiple logical lines to be entered on the same physical line. Thus, the semicolon should not be used in character fields, such as the title or other comment or description fields.

Logical line continuation—A backslash (\) followed by white space and an end-of-line [carriage return and (or) line feed] causes the input processor to ignore the end-of-line. By ignoring the end-of-line, a single logical line can be written to two physical lines. Note that only white space (tabs or spaces) can follow the backslash; if non-white space follows the backslash, the following line will not be treated as a continuation of the current line.

Repeat counter—An asterisk (*) can be used in node-by-node and element-by-element lists for property definitions to indicate repetitions of the same values. Thus, "3*0.14" is equivalent to "0.14 0.14 0.14" when a property is defined by a list of numbers.

The description of each keyword data block follows in alphabetical order. Each description includes examples of the use of the keyword and all of its identifiers, which are followed by an explanation of the data required for each identifier. Additional information is presented in notes to further explain the use of the identifiers, the interaction among identifiers, or to aid in the selection of property values. Finally, a list of the example problems that use the keyword is included at the end of the description of each keyword data block.

CHEMISTRY_IC

This keyword data block is used to define the initial conditions in the grid region, including: initial solution, and, optionally, kinetic reactions; initial phase-, exchange-, and surface-assemblage compositions; and, rarely, initial gas-phase composition (usually gases are defined as fixed partial pressures in **EQUILIBRIUM_PHASES**). The compositions are defined by index numbers that correspond to compositions defined with **SOLUTION**, **EQUILIBRIUM_PHASES**, **EXCHANGE**, **KINETICS**, **SOLID_SOLUTIONS**, **SURFACE**, and **GAS_PHASE** data blocks or through initial geochemical calculations defined in the chemistry data file that are subsequently saved with a **SAVE** data block. Chemistry initial conditions also can be defined as specified mixtures of chemical compositions on a node-by-node basis. The **CHEMISTRY_IC** data block is mandatory for all reactive-transport simulations.

Example

```
Line 0:   CHEMISTRY_IC
Line 1:         -zone       0     0     0     10    10    10
Line 2:               -solution   1
Line 3:               -equilibrium_phases     by_node
Line 4:                     < 1 2 1 3 1 1 3 4*2 >
Line 5:               -surface    file  surface.fil
Line 6:               -exchange   X     1     0.0   5     10.0
Line 7:               -gas_phase  mixture 2 4
Line 8:                     < 6*1.0 0.8 0.6 0.4 0.2 0.0 >
Line 9:               -solid_solutions        5
Line 10:              -kinetics   2
Line 11:              -mask       file        mask.file
```

Explanation

Line 0: **CHEMISTRY_IC**

CHEMISTRY_IC is the keyword for the data block; no other data are included on this line.

Line 1: **-zone** x_1, y_1, z_1, x_2, y_2, z_2

-zone—The zone selects a set of cells for which initial chemical conditions will be defined. Optionally, **zone** or **-z**[**one**].

x_1, y_1, z_1—X, Y, and Z coordinate for the left (X), front (Y), lower (Z) corner of the rectangular zone.

x_2, y_2, z_2—X, Y, and Z coordinate for the right (X), back (Y), upper (Z) corner of the rectangular zone. Units, L, are defined with the **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

Line 2: **-solution** *property*

-solution—This identifier is used to specify index numbers that define solution compositions that initially are present in the cells in the zone. For solute transport modeling, all cells in the active grid region must have a solution composition defined. Optionally, **solution** or **-s**[**olution**].

property—Index numbers are specified that correspond to solution compositions defined in the chemistry data file. Solutions or mixtures of solutions can be entered for the zone with any of five methods for defining a spatially distributed property: (1) a single index, the composition represented by the index is applied to the entire zone, (2) **by_node** followed by an index for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; an index for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), (4) **X**, **Y**, or **Z**, $index_1$, $distance_1$, $index_2$, $distance_2$, a composition is linearly interpolated from the end-member compositions defined by indices $index_1$ and $index_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the composition at the nearest point ($distance_1$ or $distance_2$) is used, or

(5) **mixture** *index*$_1$, *index*$_2$ followed by a list of mixing fractions (f_1) for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of mixing fractions must be enclosed in angle brackets "< >"; the composition associated with a node is a mixture of the two compositions—f_1 is the mixing fraction applied to composition *index*$_1$; $f_2 = 1 - f_1$ is the mixing fraction applied to composition *index*$_2$. Line 2 illustrates method 1.

Line 3: **-equilibrium_phases** *property*

-equilibrium_phases—This identifier is used to specify index numbers that define pure-phase assemblages that initially are present in the cells of the zone. A negative index number indicates that there are no equilibrium phases in the zone. Optionally, **equilibrium_phases**, **-eq**[**uilibrium_phases**], **pure_phases**, **-p**[**ure_phases**], **phases**, or **-p**[**hases**].

property—Index numbers are specified that correspond to pure-phase-assemblage compositions defined in the chemistry data file. Pure-phase assemblages or mixtures of pure-phase assemblages can be entered for the zone with any of the five methods for defining a spatially distributed property: (1) a single index, the composition represented by the index is applied to the entire zone, (2) **by_node** followed by an index for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; an index for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), (4) **X**, **Y**, or **Z**, *index*$_1$, *distance*$_1$, *index*$_2$, *distance*$_2$, a composition is linearly interpolated from the end-member compositions defined by indices *index*$_1$ and *index*$_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance*$_1$ and *distance*$_2$ along the specified coordinate direction; for locations outside the range of *distance*$_1$ to *distance*$_2$, the composition at the nearest point (*distance*$_1$ or *distance*$_2$) is used, or (5) **mixture** *index*$_1$, *index*$_2$ followed by a list of mixing fractions (f_1) for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of mixing fractions must be enclosed in angle brackets "< >"; the composition associated with a node is a mixture of the two compositions—f_1 is the mixing fraction applied to composition *index*$_1$; $f_2 = 1 - f_1$ is the mixing fraction applied to composition *index*$_2$. Lines 3 and 4 use method 2.

Line 4: *property continued*

property continued—Continuation of property definition.

Note use of a repeat factor and asterisk to enter 4 values of 2.

Line 5: **-surface** *property*

-surface—This identifier is used to specify index numbers that define surface-assemblage compositions that initially are present in the cells of the zone. A negative index number indicates that there are no surfaces in the zone. Optionally, **surface** or **-su**[**rface**].

property—Index numbers are specified that correspond to surface-assemblage compositions defined in the chemistry data file. Surface assemblages or mixtures of surface assemblages can be entered for the zone with any of the five methods for defining a spatially distributed property: (1) a single index, the composition represented by the index is applied to the entire zone, (2) **by_node** followed by an index for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; an index for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), (4) **X**, **Y**, or **Z**, *index*$_1$, *distance*$_1$, *index*$_2$, *distance*$_2$, a composition is linearly interpolated from the end-member compositions defined by indices *index*$_1$ and *index*$_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance*$_1$ and *distance*$_2$ along the specified coordinate direction; for locations outside the range of *distance*$_1$ to *distance*$_2$, the composition at the nearest point (*distance*$_1$ or *distance*$_2$) is used, or (5) **mixture** *index*$_1$, *index*$_2$ followed by a list of mixing fractions (f_1) for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of mixing fractions must be enclosed in angle brackets "< >"; the composition associated with a node is a mixture of the two compositions—f_1 is the mixing fraction applied to composition *index₁*; $f_2 = 1 - f_1$ is the mixing fraction applied to composition *index₂*. Line 5 illustrates method 3.

Line 6: **-exchange** *property*

-exchange—This identifier is used to specify index numbers that define exchange-assemblage compositions that initially are present in the cells of the zone. A negative index number indicates that there are no exchangers in the zone.Optionally, **exchange** or **-ex**[**change**].

property—Index numbers are specified that correspond to exchange-assemblage compositions defined in the chemistry data file. Exchange assemblages or mixtures of exchange assemblages can be entered for the zone with any of the five methods for defining a spatially distributed property: (1) a single index, the composition represented by the index is applied to the entire zone, (2) **by_node** followed by an index for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; an index for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), (4) **X**, **Y**, or **Z**, *index*$_1$, *distance*$_1$, *index*$_2$, *distance*$_2$, a composition is linearly interpolated from the end-member compositions defined by indices *index*$_1$ and *index*$_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance*$_1$ and *distance*$_2$ along the specified coordinate direction; for locations outside the range of *distance*$_1$ to *distance*$_2$, the composition at the nearest point (*distance*$_1$ or *distance*$_2$) is used, or (5) **mixture** *index*$_1$, *index*$_2$ followed by a list of mixing fractions (f_1) for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of mixing fractions must be enclosed in angle brackets "< >"; the composition associated with a node is a mixture of the two compositions—f_1 is the mixing fraction applied to composition *index*$_1$; $f_2 = 1 - f_1$ is the mixing fraction applied to composition *index*$_2$. Line 6 illustrates method 4.

Line 7: **-gas_phase** *property*

-gas_phase—This identifier is used to specify index numbers that define gas-phase compositions that initially are present in the cells of the zone. A negative index number indicates that there are no gas phases in the zone. Optionally, **gas_phase** or **-g**[**as_phase**].

property—Index numbers are specified that correspond to gas-phase compositions defined in the chemistry data file. Gas phases or mixtures of gas phases can be entered for the zone with any of the five methods for defining a spatially distributed property: (1) a single index, the composition represented by the index is applied to the entire zone, (2) **by_node** followed by an index for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; an index for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), (4) **X**, **Y**, or **Z**, *index*$_1$, *distance*$_1$, *index*$_2$, *distance*$_2$, a composition is linearly interpolated from the end-member compositions defined by indices *index*$_1$ and *index*$_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance*$_1$ and *distance*$_2$ along the specified coordinate direction; for locations outside the range of *distance*$_1$ to *distance*$_2$, the composition at the nearest point (*distance*$_1$ or *distance*$_2$) is used, or (5) **mixture** *index*$_1$, *index*$_2$ followed by a list of mixing fractions (f_1) for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of mixing fractions must be enclosed in angle brackets "< >"; the composition associated with a node is a mixture of the two compositions—f_1 is the mixing fraction applied to composition *index*$_1$; $f_2 = 1 - f_1$ is the mixing fraction applied to composition *index*$_2$. Lines 7 and 8 illustrate method 5.

Line 8: *property continued*

property continued—Continuation of property definition.

Note use of a repeat factor and asterisk to enter 6 values of 1.0.

Line 9: **-solid_solutions** *property*

-solid_solutions—This identifier is used to specify index numbers that define solid-solution assemblages that initially are present in the cells of the zone. A negative index number indicates that there are no solid solutions in the zone. Optionally, **solid_solution**, **solid_solutions** or **-soli**[**d_solutions**].

property—Index numbers are specified that correspond to solid-solution-assemblage compositions defined in the chemistry data file. Solid-solution assemblages or mixtures of solid-solution assemblages can be entered for the zone with any of the five methods for defining a spatially distributed property: (1) a single index, the composition represented by the index is applied to the entire zone, (2) **by_node** followed by an index for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; an index for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), (4) **X**, **Y**, or **Z**, $index_1$, $distance_1$, $index_2$, $distance_2$, a composition is linearly interpolated from the end-member compositions defined by indices $index_1$ and $index_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the composition at the nearest point ($distance_1$ or $distance_2$) is used, or (5) **mixture** $index_1$, $index_2$ followed by a list of mixing fractions (f_1) for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of mixing fractions must be enclosed in angle brackets "< >"; the composition associated with a node is a mixture of the two compositions—f_1 is the mixing fraction applied to composition $index_1$; $f_2 = 1 - f_1$ is the mixing fraction applied to composition $index_2$. Line 9 illustrates method 1.

Line 10: **-kinetics** *property*

-kinetics—This identifier is used to specify index numbers that define sets of kinetic reactants that initially are present in the cells of the zone. A negative index number indicates that there are no kinetic reactants in the zone. Optionally, **kinetics** or **-k**[**inetics**].

property—Index numbers are specified that correspond to sets of kinetic reactants defined in the chemistry data file. Kinetic-reaction sets or mixtures of kinetic-reaction sets can be entered for the zone with any of the five methods for defining a spatially distributed property: (1) a single index, the composition represented by the index is applied to the entire zone, (2) **by_node** followed by an index for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; an index for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), (4) **X**, **Y**, or **Z**, $index_1$, $distance_1$, $index_2$, $distance_2$, a composition is linearly interpolated from the end-member compositions defined by indices $index_1$ and $index_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the composition at the nearest point ($distance_1$ or $distance_2$) is used, or (5) **mixture** $index_1$, $index_2$ followed by a list of mixing fractions (f_1) for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of mixing fractions must be enclosed in angle brackets "< >"; the composition associated with a node is a mixture of the two compositions—f_1 is the mixing fraction applied to composition $index_1$; $f_2 = 1 - f_1$ is the mixing fraction applied to composition $index_2$. Line 10 illustrates method 1.

Line 11: **-mask** *property*

Use of this identifier is rare.

-mask—An array is used to include and exclude nodes from the zone definition. Optionally, **mask** or **-m**[**ask**].

property—An array of 1s (or any positive number) and 0s (or any nonpositive number) is used to include and exclude nodes from the zone definition. The array can be defined by any of four methods for defining a spatially distributed property; however, methods 1 and 4 are usually not useful: (1) a single number, 1 or 0, is used to include or exclude all of the nodes of the zone definition, (2) **by_node** followed by a 1 or 0 for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name—a 1 or 0 for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of *distance*$_1$ to *distance*$_2$, the value at the nearest point (*distance*$_1$ or *distance*$_2$) is used; if the interpolated value is less than or equal to 0 for a node, the node is excluded from the zone definition; if the interpolated value is greater than 0, the node is included in the zone definition. Line 11 illustrates method 3.

Notes

CHEMISTRY_IC defines initial conditions and must be defined for a solute-transport simulation. All non-negative index numbers must correspond to index numbers for solutions, sets of kinetic reactants, equilibrium-phase assemblages, exchange assemblages, solid-solution assemblages, surface assemblages, and gas phases that are defined in the chemistry data file. Negative index numbers indicate that the given chemical entity is not present in the zone defined. (For solute transport modeling, it is an error if a solution composition is not defined for a node.) Initial solution definitions are mandatory for all nodes in the active grid region. Sets of kinetic reactants, gas phases, equilibrium phases, and exchange, solid-solution, and surface assemblages are optional. By default, all entities (solution, set of kinetic reactants, assemblages, and gas phase) are absent from all cells. Multiple overlapping zones can be used within the **CHEMISTRY_IC** data block to define the initial conditions for the grid region. The index number for a property for a single node may be defined multiple times as part of different zone definitions. The index number used in the reactive-transport simulation for that entity for that node is the last index number defined for it in the flow and transport data file.

If property input method 4 (spatial interpolation) is used to define compositions for cells, the composition of each cell is determined by linear interpolation of the end-member compositions represented by two index numbers. The fractions applied to the end members are determined by the location of the node relative to the locations specified for the end members along the specified coordinate axis. For each type of initial condition the following are linearly interpolated: for solutions, moles of each element and temperature; for sets of kinetic reactants, the moles of each reactant; for equilibrium-phase assemblages, moles of each mineral phase; for exchange assemblages, moles of exchange sites and moles of exchanged ions; for solid-solution assemblages, moles of each component of each solid solution; for surface assemblages, moles of each type of surface site, surface area for each surface, and moles of sorbed elements; and for gas phases, moles of each gas component.

Note that the Basic-language programs defined in **RATES** data blocks in the chemistry data file and the thermodynamic data file must be written to use seconds as the unit of the time step (the TIME variable). PHAST converts time steps from the user input units for flow and transport calculations to seconds for kinetic calculations within the chemistry calculations.

Example Problems

The **CHEMISTRY_IC** data block is used in the example problems 1, 2, 3, and 4.

END

This keyword has no associated data. It is used optionally to designate the end the data input for a simulation. No input lines are processed after encountering the **END** keyword in the flow and transport data file.

Example Problems

The **END** keyword is used in the example problems 1, 2, 3, and 4.

FLUX_BC

This keyword data block is used to define the specified-flux boundary conditions. For flow-only simulations, only the fluid flux for each cell with a specified-flux boundary condition is required. For reactive-transport simulations, the index number of an associated solution also is required. The flux and index number of the associated solution may vary independently over the course of the simulation. This keyword data block is optional and only is needed if specified-flux boundaries are to be included in the simulation.

Example

```
Line 0:   FLUX_BC
Line 1:         -zone       0     10     10     10     10     10
Line 2:               -flux
Line 3:                           0      -3.5e-3
Line 3a:                          100    -2.0e-3
Line 4:               -associated_solution
Line 5                            0      X      5      0      6      10
Line 6:               -face       Z
Line 7:               -mask       file         mask.file
```

Explanation

Line 0: **FLUX_BC**

FLUX_BC is the keyword for the data block; no other data are included on this line.

Line 1: **-zone** x_1, y_1, z_1, x_2, y_2, z_2

-zone—The zone selects a set of cells for which specified-flux boundary conditions will be defined. For specified-flux boundary conditions, a zone can include a single node, a linear set of nodes, or a planar set of nodes; three-dimensional sets of nodes are not allowed. Optionally, **zone** or **-z**[**one**].

x_1, y_1, z_1—X, Y, and Z coordinate for the left (X), front (Y), lower (Z) corner of the rectangular zone.

x_2, y_2, z_2—X, Y, and Z coordinate for the right (X), back (Y), upper (Z) corner of the rectangular zone. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

Line 2: **-flux**

-flux—This identifier is used to specify a time series of volumetric fluxes for the nodes in the zone. Volumetric fluxes are applied only to cells associated with active nodes and only to the cell faces on the active grid boundary that are normal to the coordinate direction of the flux (**-face)**. The flux is defined by the property definition on line 3. A time series of flux properties may be defined by using multiple line 3s. The first *time* in the series must be zero. Optionally, **flux** or **-fl**[**ux**].

Line 3: *time* [*units*] *property*

time—Simulation time (T) at which the flux property definition (*property*) will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

property—The volumetric flux for the nodes selected by the zone can be entered with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_node** followed by a value for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets “< >”, (3) **file** followed by a file name; a value for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$; a value is linearly interpolated from $value_1$ and $value_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used. Lines 3 and 3a illustrate method 1. The flux is a signed quantity that indicates whether the flux is in the positive coordinate direction or the negative coordinate direction. Units, L/T, are defined by **-flux** identifier in the **UNITS** data block.

Line 4: **-associated_solution**

-associated_solution—This identifier is used to specify a time series of index numbers that define the solution composition for the flux of water into the active grid region through the specified-flux boundary. The composition of water flowing into the active grid is defined by the property definition of line 5. A time series of solution properties may be defined by using multiple line 5s. The first *time* in the series must be zero. Optionally, **associated_solution**, **-a**[**ssociated_solution**], **solution**, or **-s**[**olution**].

Line 5:*time* [*units*] *property*

time—Simulation time (T) at which the solution-composition property definition (*property*) will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

property—Index numbers are specified that correspond to solution compositions defined in the chemistry data file. Solutions or mixtures of solutions can be entered for the zone with any of the five methods for defining a spatially distributed property: (1) a single index, the composition represented by the index is applied to the entire zone, (2) **by_node** followed by an index for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets “< >”, (3) **file** followed by a file name; an index for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), (4) **X**, **Y**, or **Z**, $index_1$, $distance_1$, $index_2$, $distance_2$; a composition is linearly interpolated from the end-member compositions defined by indices $index_1$ and $index_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the composition at the nearest point ($distance_1$ or $distance_2$) is used, or (5) **mixture** $index_1$, $index_2$ followed by a list of mixing fractions (f_1) for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of mixing fractions must be enclosed in angle brackets “< >”; the composition associated with a node is a mixture of the two compositions—f_1 is the mixing fraction applied to composition $index_1$; $f_2 = 1 - f_1$ is the mixing fraction applied to composition $index_2$. Line 5 illustrates method 4.

Line 6: **-face** (**X**, **Y**, or **Z**)

-face—This identifier is used to specify the coordinate direction of the specified flux. If the set of nodes selected by the zone is zero or one dimensional, the coordinate direction of the flux must be specified. Optionally, **face** or **-fa**[**ce**].

X, **Y**, or **Z**—The coordinate direction of the flux.

Line 7: **-mask** *property*

Use of this identifier is rare.

-mask—An array is used to include and exclude nodes from the zone definition. Optionally, **mask** or **-m**[**ask**].

property—An array of 1s (or any positive number) and 0s (or any nonpositive number) is used to include and exclude nodes from the zone definition. The array can be defined by any of four methods for defining a spatially distributed property; however, methods 1 and 4 are usually not useful: (1) a single number, 1 or 0, is used to include or exclude all of the nodes of the zone definition, (2) **by_node** followed by a 1 or 0 for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets “< >”, (3) **file** followed by a file name—a 1 or 0 for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used; if the interpolated value is less than or equal to 0 for a node, the node is excluded from the zone definition; if the interpolated value is greater than 0, the node is included in the zone definition. Line 7 illustrates method 3.

Notes

For specified-flux boundary conditions, the set of nodes selected by zones must be zero, one, or two dimensional. Three-dimensional sets of nodes are not allowed because fluxes are normally applied only to the cells that bound the active grid region, although it is possible to apply a flux to internal cells. For two-dimensional sets of nodes, the coordinate direction of the flux can be determined from the zone definition, and the **-face** identifier is not required. For zero- and one-dimensional sets of nodes, the coordinate direction of the flux is ambiguous, and the **-face** identifier must be used to identify the coordinate direction of the flux. The sign of the flux quantity indicates whether the flux is in the positive or negative coordinate direction. Note that the hyphen in **-flux** does not indicate a negative quantity; it only indicates that the word is an identifier.

Multiple zones may be used within any **FLUX_BC** data block to define flux boundary conditions for selected cell faces within the grid region. Different boundary conditions for a single cell and its cell faces may be defined multiple times as part of different zone definitions and different keyword data blocks. The boundary condition that is used for a cell face is taken from the last zone definition in the flow and transport data file that defines a boundary condition for that cell face, subject to restrictions on the combinations of boundary conditions that are allowed. A cell can have (1) a specified-head condition only, or (2) a combination of flux, leaky, and river boundary conditions. Flux, leaky, and river boundary conditions for a cell must comply with the following restrictions: (1) a given boundary condition can be applied to only one of the cell faces, (2) rivers always apply to the positive Z cell face, (3) both leaky and river conditions cannot be applied to the positive Z cell face, and (4) both leaky and flux conditions cannot be applied to the same cell face. Both a flux and a river condition may be defined for the positive Z face of a boundary cell. See section D.5.7. Boundary-Condition Compatibility for more details.

Example Problems

The **FLUX_BC** data block is used in the example problem 4.

FREE_SURFACE_BC

This keyword data block is used to define the presence of a free surface for unconfined flow simulations. If the **FREE_SURFACE_BC** data block is not included, confined flow is simulated.

Example

```
Line 0:   FREE_SURFACE_BC          true
```

Explanation

Line 0: **FREE_SURFACE_BC** [***True*** or ***False***]

FREE_SURFACE_BC—This keyword data block is used to include or exclude a free-surface boundary condition in the simulation, which implies unconfined or confined flow, respectively. By default, confined flow is simulated. Optionally, **FREE_SURFACE**.

[***True*** or ***False***]—A value of **true** indicates that unconfined flow is simulated. A value of **false** indicates that confined flow is simulated. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Notes

This option is used to simulate confined (**FREE_SURFACE_BC false**) or unconfined (**FREE_SURFACE_BC true**) flow. If unconfined flow is simulated, the specific storage in each cell is automatically set to zero, which is equivalent to setting the fluid and matrix compressibilities to zero. A free-surface boundary condition is required for simulation of river boundary conditions. See section D.5.7. Boundary-Condition Compatibility for more details.

Example Problems

The **FREE_SURFACE_BC** data block is used in the example problems 1, 2, 3, and 4.

GRID

This keyword data block is used to define the finite-difference node locations for the simulation grid. This data block is mandatory for all simulations. The **GRID** data block contains only static data.

Example

```
Line 0:   GRID
Line 1:         -uniform    X     0.    1000. 6
Line 2:         -nonuniform       Y     0.    100.
Line 3:               400.  800.  1000.
Line 1a:        -uniform    Z     0.    10.   2
Line 4:         -overlay_uniform Z 0.0 1.0 11
Line 5:         -overlay_nonuniform Z 8.2 8.4 8.5 8.6 8.8
Line 6:         -snap Z  0.05
Line 7:         -chemistry_dimensions         XY
Line 8:         -print_orientation            XY
```

Explanation

Line 0: **GRID**

GRID is the keyword for the data block; no other data are included on this line.

Line 1: **-uniform** (**X**, **Y**, or **Z**) *minimum, maximum, number of nodes*

-uniform—This identifier is used to specify uniformly spaced nodes for the coordinate direction. **-Nonuniform** and **-uniform** are mutually exclusive for a coordinate direction. Optionally, **uniform** or **-u**[**niform**].

X, **Y**, or **Z**—Coordinate direction with uniformly spaced nodes.

minimum—Minimum node coordinate in the specified coordinate direction. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

maximum—Maximum node coordinate in the specified coordinate direction. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

number of nodes—Number of nodes in the specified coordinate direction. The number of cells in this direction is equal to the number of nodes. The number of elements in this direction is one less than the number of nodes. The minimum number of nodes is two.

Line 2: **-nonuniform** (**X**, **Y**, or **Z**) *list of node coordinates*

-nonuniform—This identifier is used to specify nonuniform node spacing for the coordinate direction. **-Nonuniform** and **-uniform** are mutually exclusive for a coordinate direction. Optionally, **nonuniform** or **-n**[**onuniform**].

X, **Y**, or **Z**—Coordinate direction with nonuniform node spacing.

list of node coordinates—List of node coordinates in the specified coordinate direction. List of node coordinates must be in ascending order. Although only one coordinate is required for **-nonuniform** input, altogether, at least two nodes must be defined for a coordinate direction by **-nonuniform**, **-overlay_uniform**, and **-overlay_nonuniform** identifiers. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block. List of node coordinates may continue on successive lines as shown in line 3.

Line 3: *list of node coordinates*

list of node coordinates—Continuation of list of node coordinates. This line must be preceded by the **-nonuniform** identifier or another continuation line for the list.

Line 4: **-overlay_uniform** (**X**, **Y**, or **Z**) *minimum*, *maximum*, *number of nodes*

-overlay_uniform—This identifier is used to specify additional uniformly spaced nodes for the specified coordinate direction. Either **-nonuniform** or **-uniform** is required for each coordinate direction; optional **-overlay_uniform** and **-overlay_nonuniform** identifiers may be defined for each coordinate direction. Multiple overlays may define the same node or nodes that are close together; nodes closer than the *snap_distance* for the coordinate (Line 6) will be merged into a single node. Optionally, **overlay_uniform** or **-o**[**verlay_uniform**].

X, **Y**, or **Z**—Coordinate direction with additional uniformly spaced nodes.

minimum—Minimum node coordinate for additional uniformly spaced nodes in the specified coordinate direction. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

maximum—Maximum node coordinate for the additional uniformly spaced nodes in the specified coordinate direction. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

number of nodes—Number of additional nodes for the specified coordinate direction.

Line 4 indicates that beginning with a node at 0.0 and ending with a node at 1.0, 11 equally spaced nodes will be added to the grid in the Z direction.

Line 5: **-overlay_nonuniform** (**X**, **Y**, or **Z**) *list of node coordinates*

-overlay_nonuniform—This identifier is used to specify additional nonuniformly spaced nodes in the specified coordinate direction. Either **-nonuniform** or **-uniform** is required for each coordinate direction; optional **-overlay_uniform** and **-overlay_nonuniform** identifiers may be defined for each coordinate direction. Multiple overlays may define the same node or nodes that are close together; nodes closer than the *snap_distance* for the coordinate (Line 6) will be merged into a single node. Optionally, **overlay_nonuniform** or **-overlay_n**[**onuniform**].

X, **Y**, or **Z**—Coordinate direction with additional nonuniformly spaced nodes.

list of node coordinates—List of one or more node coordinates in the specified coordinate direction. List of node coordinates must be in ascending order. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block. List of node coordinates may continue on successive lines.

Line 5 indicates that additional nodes are placed at 8.2, 8.4, 8.5, 8.6, and 8.8 in the Z direction.

Line 6: **-snap** (**X**, **Y**, or **Z**) *snap_distance*

-snap—This identifier is used to specify the minimum distance between nodes in the coordinate direction. Optionally, **snap** or **-s**[**nap**].

X, **Y**, or **Z**—Coordinate direction for which *snap_distance* is defined.

snap_distance—Minimum distance between nodes in the specified coordinate direction. After coordinate lists are combined and sorted for a coordinate direction, the nodes are processed in sort order. Nodes closer than *snap_distance* to a previous node will be eliminated. Default is 0.001 of specified units. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

Line 7: **-chemistry_dimensions** [**X**] [**Y**] [**Z**]

-chemistry_dimensions—This identifier is used to specify coordinate directions for which chemical calculations are performed. If, conceptually, the transport calculation is one or two dimensional, the geochemical calculations can be performed for a single line or plane of nodes and the resulting chemical compositions can be copied to the remaining symmetric lines or plane of nodes. Omitting chemical calculations in one or two coordinate directions saves substantial amounts of computation time. By default, geochemical calculations are performed on all active nodes of the three-dimensional grid region. Optionally, **chemistry_dimensions**, **transport_dimensions**, **-c**[**hemistry_dimensions**], or **-t**[**ransport_dimensions**].

[**X**] [**Y**] [**Z**]—The coordinate directions for which chemical calculations are performed. For example, "**Z**" indicates that chemical calculations are performed for one line of nodes in the Z direction, "**YZ**" indicates chemical calculation are performed on the set of active nodes in one Y–Z plane, and "**XYZ**" indicates chemical calculations are performed on all active nodes. Warning: initial and boundary conditions must be consistent with the symmetry assumed for one- or two-dimensional calculations or erroneous chemical results will be produced.

Line 8: **-print_orientation** (**XY** or **XZ**)

-print_orientation—This identifier is used to specify the orientation of the planes of data to be written to files with names that begin with *prefix*.**O**. The *prefix*.**O** files contain spatial data written as a series of planes. By default, writing is by X–Y planes. Optionally, **print_orientation** or **-p**[**rint_orientation**].

(**XY** or **XZ**)—Only two orientation options are allowed, either **XY**, indicating X–Y planes are printed or **XZ** indicating X–Z planes are printed.

Notes

Node coordinates must be defined for all three coordinate directions for all simulations. Each coordinate direction must be defined with **-uniform** or **-nonuniform**, which are mutually exclusive identifiers for each coordinate direction. The grid may be refined by adding additional nodes with the identifiers **-overlay_uniform** and **-overlay_nonuniform**. All of the nodes defined are merged into a single list and sorted; nodes within the distance defined by **-snap** of a previous node in the list are eliminated.

Flow and transport calculations always involve the entire active grid region and are thus three dimensional. The minimum number of cells (nodes) in any coordinate direction is two. The identifier **-chemistry_dimensions** is used to save computation time for one- and two-dimensional problems. If the calculation represents a one- or two-dimensional flow system and initial and boundary conditions are appropriate for one- or two-dimensional transport, the symmetry of the transport simulation can be used to reduce the number of geochemical calculations. The geochemical calculations can be performed on a single line or plane of nodes and the results copied to the other lines or plane of nodes. If the **-chemistry_dimensions** identifier is not included, geochemical calculations are performed for all active nodes. For one- and two-dimensional geochemical calculations, one or two coordinate directions are listed for the **-chemistry_dimensions** identifier; the number of nodes for the coordinate directions not listed must be exactly two.

The use of **-chemistry_dimensions** can lead to erroneous results if the initial and boundary conditions do not allow a truly one- or two-dimensional transport simulation. For example, an X–Y simulation would not have zero flow in the Z direction if a free-surface boundary condition were used, which violates the two-dimensional flow assumption. Also, one- or two-dimensional flow is not sufficient to ensure the same symmetry for transport. For example, for steady one-dimensional flow in the X direction, there are four cells in each Y–Z plane. If a contaminant is introduced into only one cell in a Y–Z plane, concentrations are not equal in each of the cells of the Y–Z plane and the transport system is not one dimensional. Thus, boundary and initial conditions must be chosen carefully to ensure one- or two-dimensional transport. The program does not check for conceptual errors in the boundary conditions. Operationally, a one- or two-dimensional simulation can be tested by removing the **-chemistry_dimensions** identifier and checking that the results are the same as the one- or two-dimensional simulation.

No set of discretization rules exists that will guarantee an accurate numerical solution with a minimum number of nodes and time steps, even for the case of uniform coefficients in the differential equations. However, the following guidelines should be considered. See Appendix D for more details.

1. If using the backward-in-space or backward-in-time differencing (**SOLUTION_METHOD** data block), verify that the grid-spacing and time-step selection do not introduce excessive numerical dispersion.
2. If using centered-in-space and centered-in-time differencing, examine the results for spatial and temporal oscillations that are caused by the time or space discretization being too coarse.
3. With reactive transport, it is important to assess the effects of spatial and temporal discretization error on the reaction chemistry that takes place by refining the spatial and temporal discretization.
4. The global-balance summary table (*prefix***.O.bal** file) may indicate that the time step is too long by exhibiting large errors for conservative constituents, particularly if the operator-splitting error is large.

Example Problems

The **GRID** data block is used in the example problems 1, 2, 3, and 4.

HEAD_IC

This keyword data block is used to define the initial head conditions in the grid region. This data block is mandatory for all simulations.

Example 1

```
Line 0:  HEAD_IC
Line 1:        -zone      0     0     0     10    10    10
Line 2:              -head      150
Line 1a:       -zone      5     5     5     10    10    10
Line 2a:             -head      200
Line 3:              -mask      file        mask.file
```

Explanation 1

Line 0: **HEAD_IC**

HEAD_IC is the keyword for the data block; no other data are included on this line.

Line 1: **-zone** $x_1, y_1, z_1, x_2, y_2, z_2$

-zone—The zone selects a set of nodes for which initial head conditions will be defined. Optionally, **zone** or **-z**[**one**].

x_1, y_1, z_1—X, Y, and Z coordinate for the left (X), front (Y), lower (Z) corner of the rectangular zone.

x_2, y_2, z_2—X, Y, and Z coordinate for the right (X), back (Y), upper (Z) corner of the rectangular zone. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

Line 2: **-head** *property*

-head—This identifier is used to specify initial heads for the zone. Optionally, **head** or **-h**[**ead**].

property—Heads for the zone can be entered with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_node** followed by a value for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name, a value for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used. Line 2 illustrates method 1. Units, L, are defined by **-head** identifier in the **UNITS** data block.

Line 3: **-mask** *property*

Use of this identifier is rare.

-mask—An array is used to include and exclude nodes from the zone definition. Optionally, **mask** or **-m**[**ask**].

property—An array of 1s (or any positive number) and 0s (or any nonpositive number) is used to include and exclude nodes from the zone definition. The array can be defined by any of four methods for defining a spatially distributed property; however, methods 1 and 4 are usually not useful: (1) a single number, 1 or 0, is used to include or exclude all of the nodes of the zone definition, (2) **by_node** followed by a 1 or 0 for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name—a 1 or 0 for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used; if the interpolated value is less than or equal to 0 for a node, the node is excluded from the zone definition; if the interpolated value is greater than 0, the node is included in the zone definition. Line 3 illustrates method 3.

Example 2

```
Line 0:   HEAD_IC
Line 1:         -water_table by_node
Line 2:               < 150.0 150.5 151.0 151.5 20*152.0 >
```

Explanation 2

Line 0: **HEAD_IC**

HEAD_IC is the keyword for the data block; no other data are included on this line.

Line 1: **-water_table** *property*

-water_table—This identifier is used to specify hydrostatic heads for the active grid region. Hydrostatic heads are specified by an array of heads, one for each node in the X–Y plane. Optionally, **water_table** or **-w**[**ater_table**].

property—Heads for the entire X–Y plane can be entered with either of two methods for defining a spatially distributed property: (1) **by_node** followed by a value for each node (active or inactive) in the X–Y plane in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", or (2) **file** followed by a file name, a value for each node (active or inactive) in the X–Y plane is read from the file in natural order (list of values may extend over multiple lines). Line 2 illustrates method 1. Units, L, are defined by **-head** identifier in the **UNITS** data block.

Notes

The **HEAD_IC** data block defines the initial head conditions. The initial head condition applies only at the beginning of the simulation, and it is not possible to redefine a head condition later in the simulation. Initial heads are mandatory for all nodes in the active grid region.

When using zones to define the initial head, multiple zones may be used within the **HEAD_IC** data block to define the initial conditions for the entire grid region. Heads for a single node may be defined multiple times as part of different zone definitions. The initial head used in the flow or reactive-transport simulation for that node is the last head defined for the node in the flow and transport data file.

When using the **-water_table** identifier, it is not permissible to define any initial head conditions with zones. The water-table heads are defined only for the uppermost X–Y plane of nodes. For each vertical stack of nodes in the grid region, the initial water-table head of the uppermost node is assigned to all nodes in the stack.

Example Problems

The **HEAD_IC** data block is used in the example problems 1, 2, 3, and 4.

LEAKY_BC

This keyword data block is used to define leaky boundary conditions. Conceptually, a leaky boundary layer is located outside of the active grid region and flow through the layer is determined by the thickness of the layer, the hydraulic conductivity of the layer, and the difference in head between the specified head at the exterior side of the layer and the head at the boundary of the active grid region. For flow-only simulations, only the parameters related to flow are required. For reactive-transport simulation, the index number of an associated solution also is required. The head at the exterior side of the layer and the index number of the associated solution may vary independently over the course of the simulation. This keyword data block is optional and is needed only if leaky boundary conditions are included in the simulation.

Example

```
Line 0:   LEAKY_BC
Line 1:         -zone       0      10     0      10     10     10
Line 2:               -head
Line 3:                            0      100
Line 4:               -associated_solution
Line 5:                            0      mixture 5 6
Line 6:                                   < 0.0 0.2 0.4 0.6 0.8 1.0 >
Line 5a:                           100    mixture 10 11
Line 6a:                                  < 1.0 0.8 0.6 0.4 0.2 0.0 >
Line 7:               -thickness          X     100   0     50    10
Line 8:               -hydraulic_conductivity       file  leaky_k.fil
Line 9:               -face               Y
Line 10:              -mask               file        mask.file
```

Explanation

Line 0: **LEAKY_BC**

LEAKY_BC is the keyword for the data block; no other data are included on this line.

Line 1: **-zone** x_1, y_1, z_1, x_2, y_2, z_2

-zone—The zone selects a set of cells for which leaky boundary conditions will be defined. For leaky boundary conditions, a zone can include a single node, a linear set of nodes, or a planar set of nodes; three-dimensional sets of nodes are not allowed. Optionally, **zone** or **-z**[**one**].

x_1, y_1, z_1—X, Y, and Z coordinate for the left (X), front (Y), lower (Z) corner of the rectangular zone.

x_2, y_2, z_2—X, Y, and Z coordinate for the right (X), back (Y), upper (Z) corner of the rectangular zone. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

Line 2: **-head**

-head—This identifier is used to specify a time series of head on the exterior side of the leaky boundary layer. The head is defined by the property definition on line 3. A time series of head properties may be defined by using multiple line 3s. The first *time* in the series must be zero. Optionally, **head** or **-he**[**ad**].

Line 3: *time* [*units*] *property*

time—Simulation time (T) at which the flux property definition (*property*) will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

property—The head on the exterior side of the leaky boundary layer can be entered for the zone with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_node** followed by a value for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets “< >”, (3) **file** followed by a file name, a value for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from *value*$_1$ and *value*$_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance*$_1$ and *distance*$_2$ along the specified coordinate direction; for locations outside the range of *distance*$_1$ to *distance*$_2$, the value at the nearest point (*distance*$_1$ or *distance*$_2$) is used. Line 3 illustrates method 1. Units, L, are defined by the **-head** identifier in the **UNITS** data block.

Line 4: **-associated_solution**

-associated_solution—This identifier is used to specify a time series of solution index numbers that associate a solution composition with any water that flows into the active grid region through the leaky boundary. The solution composition is defined by the property definition on line 5. A time series of solution properties may be defined by using multiple line 5s. The first *time* in the series must be zero. Optionally, **associated_solution**, **-a**[**ssociated_solution**], **solution**, or **-s**[**olution**].

Line 5: *time* [*units*] *property*

time—Simulation time (T) at which the flux property definition (*property*) will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

property—Index numbers are specified that correspond to solution compositions defined in the chemistry data file. Solutions or mixtures of solutions can be entered for the zone with any of the five methods for defining a spatially distributed property: (1) a single index, the composition represented by the index is applied to the entire zone, (2) **by_node** followed by an index for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets “< >”, (3) **file** followed by a file name; an index for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), (4) **X**, **Y**, or **Z**, *index*$_1$, *distance*$_1$, *index*$_2$, *distance*$_2$; a composition is linearly interpolated from the end-member compositions defined by indices *index*$_1$ and *index*$_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance*$_1$ and *distance*$_2$ along the specified coordinate direction; for locations outside the range of *distance*$_1$ to *distance*$_2$, the composition at the nearest point (*distance*$_1$ or *distance*$_2$) is used, or (5) **mixture** *index*$_1$, *index*$_2$ followed by a list of mixing fractions (f_1) for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of mixing fractions must be enclosed in angle brackets “< >”; the composition associated with a node is a mixture of the two compositions—f_1 is the mixing fraction applied to composition *index*$_1$; $f_2 = 1 - f_1$ is the mixing fraction applied to composition *index*$_2$. Lines 5, 6, 5a, and 6a illustrate method 5.

Line 6: *Property* continued.

Line 7: **-thickness** *property*

-thickness—This identifier is used to specify the thickness of the leaky boundary layer. Optionally, **thickness** or **-t**[**hickness**].

property—The thickness of the leaky boundary layer can be entered for the zone with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_node** followed by a value for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets “< >”, (3) **file** followed by a file name, a value for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, *value*$_1$, *distance*$_1$, *value*$_2$, *distance*$_2$, a value is linearly interpolated from *value*$_1$ and *value*$_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance*$_1$ and *distance*$_2$ along the specified coordinate direction; for locations outside the range of *distance*$_1$ to *distance*$_2$, the value at the nearest point (*distance*$_1$ or *distance*$_2$) is used. Line 7 illustrates method 4. Units, L, are defined by the **-leaky_thickness** identifier in the **UNITS** data block.

Line 8: **-hydraulic_conductivity** *property*

-hydraulic_conductivity—This identifier is used to specify the hydraulic conductivity of the leaky boundary layer. Optionally, **hydraulic_conductivity**, **-hy**[**draulic_conductivity**], **k**, or **-k**.

property—The hydraulic conductivity of the leaky boundary layer can be entered for the zone with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_node** followed by a value for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; a value for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, *value*$_1$, *distance*$_1$, *value*$_2$, *distance*$_2$, a value is linearly interpolated from *value*$_1$ and *value*$_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance*$_1$ and *distance*$_2$ along the specified coordinate direction; for locations outside the range of *distance*$_1$ to *distance*$_2$, the value at the nearest point (*distance*$_1$ or *distance*$_2$) is used. Line 8 illustrates method 3. Units, L/T, are defined by the **-leaky_hydraulic_conductivity** identifier in the **UNITS** data block.

Line 9: **-face** (**X**, **Y**, or **Z**)

-face—This identifier is used to specify the coordinate direction of the leakage. If the set of nodes selected by the zone is zero or one dimensional, the coordinate direction for the leakage must be specified. Optionally, **face** or **-fa**[**ce**].

X, **Y**, or **Z**—The coordinate direction of the leakage.

Line 10: **-mask** *property*

Use of this identifier is rare.

-mask—An array is used to include and exclude nodes from the zone definition. Optionally, **mask** or **-m**[**ask**].

property—An array of 1s (or any positive number) and 0s (or any nonpositive number) is used to include and exclude nodes from the zone definition. The array can be defined by any of four methods for defining a spatially distributed property; however, methods 1 and 4 are usually not useful: (1) a single number, 1 or 0, is used to include or exclude all of the nodes of the zone definition, (2) **by_node** followed by a 1 or 0 for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name—a 1 or 0 for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, *value*$_1$, *distance*$_1$, *value*$_2$, *distance*$_2$, a value is linearly interpolated from *value*$_1$ and *value*$_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance*$_1$ and *distance*$_2$ along the specified coordinate direction; for locations outside the range of *distance*$_1$ to *distance*$_2$, the value at the nearest point (*distance*$_1$ or *distance*$_2$) is used; if the interpolated value is less than or equal to 0 for a node, the node is excluded from the zone definition; if the interpolated value is greater than 0, the node is included in the zone definition. Line 10 illustrates method 3.

Notes

The zones for leaky boundary conditions must be zero, one, or two dimensional. For leaky boundary conditions with two-dimensional zones, the coordinate direction for the leakage can be determined from the zone definition and the **-face** identifier is not necessary. For zero- and one-dimensional zones the coordinate direction for the leakage is not known from the zone definition, and the **-face** identifier must be used to identify the coordinate direction of the leakage.

Multiple zones may be used within the **LEAKY_BC** data block to define leaky boundary conditions for selected cell faces within the grid region. Different boundary conditions for a single cell and its cell faces may be defined multiple times as part of different zone definitions and different keyword data blocks. The boundary condition that is used for a cell face is taken from the last zone definition in the flow and transport data file that defines a boundary condition for that cell face, subject to restrictions on the combinations of boundary conditions that are allowed. A cell can have (1) a specified-head condition only, or (2) a combination of flux, leaky, and river boundary conditions. Flux, leaky, and river boundary conditions for a cell must comply with the following restrictions: (1) a given boundary condition can be applied to only one of the cell faces, (2) rivers always apply to the positive Z cell face, (3) both leaky and river conditions cannot be applied to the positive Z cell face, and (4) both leaky and flux conditions cannot be applied to the same cell face. If a leaky and a river boundary condition are applied to the positive Z cell face, the river boundary condition is ignored. See section D.5.7. Boundary-Condition Compatibility for more details.

Example Problems

The **LEAKY_BC** data block is used in the example problem 4.

MEDIA

This keyword data block is used to define media properties, including hydraulic conductivities, porosities, specific storages, and dispersivities. These spatial properties are applied to elements, not cells (see sections 4.2. Spatial Data and D.1.2. Spatial Discretization). The **MEDIA** data block contains only static data.

Example

```
Line 0:   MEDIA
Line 1:         -zone       0     0     0     10    10    10
Line 2:               -Kx         file              kx.fil
Line 3:               -Ky         file              ky.fil
Line 1a:        -zone       0     0     0     10    10    10
Line 4:               -Kz         file              kz.fil
Line 5:               -porosity   X     0.2   0     0.3   10
Line 6:               -specific_storage             0
Line 7:               -longitudinal_dispersivity    2
Line 8:               -horizontal_dispersivity      by_element
Line 9:                     < 10*2.0 10*4.0 >
Line 10:              -vertical_dispersivity        0.2
Line 1b:        -zone       4     0     0     5     0     0
Line 11:              -active     0
Line 12:              -mask       file              mask.file
```

Explanation

Line 0: **MEDIA**

MEDIA is the keyword for the data block; no other data are included on this line.

Line 1: **-zone** x_1, y_1, z_1, x_2, y_2, z_2

-zone—The zone selects a set of elements for which media properties will be defined. Optionally, **zone** or **-z**[**one**].

x_1, y_1, z_1—X, Y, and Z coordinate for the left (X), front (Y), lower (Z) corner of the rectangular zone.

x_2, y_2, z_2—X, Y, and Z coordinate for the right (X), back (Y), upper (Z) corner of the rectangular zone. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

Line 2: **-Kx** *property*

-Kx—This identifier is used to specify the hydraulic conductivity in the X direction for the elements in the zone. Optionally, **Kx**, **Kxx**, or **-Kx**[**x**].

property—The hydraulic conductivity in the X direction can be entered for the zone with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_element** followed by a value for each element in the zone (active or inactive) in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; a value for each element in the zone (active or inactive) is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each element centroid in the zone; interpolation is proportional to the fractional distance of the element centroid between $distance_1$ and $distance_2$ along the specified coordinate direction; for centroids outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used. Line 2 illustrates method 3. Units, L/T, are defined by the **-hydraulic_conductivity** identifier in the **UNITS** data block.

Line 3: **-Ky** *property*

-Ky—This identifier is used to specify the hydraulic conductivity in the Y direction for the elements in the zone. Optionally, **Ky**, **Kyy**, or **-Ky**[**y**].

property—The hydraulic conductivity in the Y direction can be entered for the zone with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_element** followed by a value for each element in the zone (active or inactive) in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; a value for each element in the zone (active or inactive) is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each element centroid in the zone; interpolation is proportional to the fractional distance of the element centroid between $distance_1$ and $distance_2$ along the specified coordinate direction; for centroids outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used. Line 3 illustrates method 3. Units, L/T, are defined by the **-hydraulic_conductivity** identifier in the **UNITS** data block.

Line 4: **-Kz** *property*

-Kz—This identifier is used to specify the hydraulic conductivity in the Z direction for the elements in the zone. Optionally, **Kz**, **Kzz**, or **-Kz**[**z**].

property—The hydraulic conductivity in the Z direction can be entered for the zone with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_element** followed by a value for each element in the zone (active or inactive) in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; a value for each element in the zone (active or inactive) is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each element centroid in the zone; interpolation is proportional to the fractional distance of the element centroid between $distance_1$ and $distance_2$ along the specified coordinate direction; for centroids outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used. Line 4 illustrates method 3. Units, L/T, are defined by the **-hydraulic_conductivity** identifier in the **UNITS** data block.

Line 5: **-porosity** *property*

-porosity—This identifier is used to specify the porosity for the elements in the zone. Optionally, **porosity** or **-p**[**orosity**].

property—The porosity can be entered for the zone with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_element** followed by a value for each element in the zone (active or inactive) in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; a value for each element in the zone (active or inactive) is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each element centroid in the zone; interpolation is proportional to the fractional distance of the element centroid between $distance_1$ and $distance_2$ along the specified coordinate direction; for centroids outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used. Line 5 illustrates method 4. Porosity is dimensionless.

Line 6: **-specific_storage** *property*

-specific_storage—This identifier is used to specify the specific storage for the elements in the zone. Optionally, **specific_storage**, **storage**, **-s**[**pecific_storage**], or **-s**[**torage**].

property—The specific storage can be entered for the zone with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_element** followed by a value for each element in the zone (active or inactive) in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; a value for each element in the zone (active or inactive) is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each element centroid in the zone; interpolation is proportional to the fractional distance of the element centroid between $distance_1$ and $distance_2$ along the specified coordinate direction; for centroids outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used. Line 6 illustrates method 1. Units, 1/L, are defined by the **-specific_storage** identifier in the **UNITS** data block.

Line 7: **-longitudinal_dispersivity** *property*

-longitudinal_dispersivity—This identifier is used to specify the dispersivity in the direction of the flow-velocity vector for the elements in the zone. Optionally, **longitudinal_dispersivity**, **dispersivity_longitudinal**, **long_dispersivity**, or **-l**[**ongitudinal_dispersivity**], **-d**[**ispersivity_longitudinal**], **-l**[**ong_dispersivity**].

property—Dispersivity in the direction of the flow-velocity vector can be entered for the zone with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_element** followed by a value for each element in the zone (active or inactive) in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; a value for each element in the zone (active or inactive) is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each element centroid in the zone; interpolation is proportional to the fractional distance of the element centroid between $distance_1$ and $distance_2$ along the specified coordinate direction; for centroids outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used. Line 7 illustrates method 1. Units, L, are defined by the **-dispersivity** identifier in the **UNITS** data block.

Line 8: **-horizontal_dispersivity** *property*

-horizontal_dispersivity—This identifier is used to specify the horizontal transverse dispersivity for the elements in the zone. Optionally, **horizontal_dispersivity**, **dispersivity_horizontal**, **-h**[**orizontal_dispersivity**], or **-dispersivity_h**[**orizontal**].

property—The horizontal transverse dispersivity can be entered for the zone with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_element** followed by a value for each element in the zone (active or inactive) in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; a value for each element in the zone (active or inactive) is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each element centroid in the zone; interpolation is proportional to the fractional distance of the element centroid between $distance_1$ and $distance_2$ along the specified coordinate direction; for centroids outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used. Line 8 illustrates method 2. Units, L, are defined by the **-dispersivity** identifier in the **UNITS** data block.

Line 9: *property continued*

property continued—Continuation of property definition.

Note use of repeat factors and asterisks to enter 10 values of 2.0 and 10 values of 4.0.

Line 10: **-vertical_dispersivity** *property*

-vertical_dispersivity—This identifier is used to specify the vertical transverse dispersivity for the elements in the zone. Optionally, **vertical_dispersivity**, **dispersivity_vertical**, **-v**[**ertical_dispersivity**], or **-dispersivity_v**[**ertical**].

property—The vertical transverse dispersivity can be entered for the zone with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_element** followed by a value for each element in the zone (active or inactive) in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; a value for each element in the zone (active or inactive) is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each element centroid in the zone; interpolation is proportional to the fractional distance of the element centroid between $distance_1$ and $distance_2$ along the specified coordinate direction; for centroids outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used. Line 10 illustrates method 1. Units, L, are defined by the **-dispersivity** identifier in the **UNITS** data block.

Line 11: **-active** *property*

-active—This identifier is used to specify elements within the zone to be active or inactive. Optionally, **active** or **-a**[**ctive**].

property—Active and inactive elements are defined with values of 1 or positive numbers (active) and 0 or nonpositive numbers (inactive). The active and inactive elements can be defined for the zone by any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_element** followed by a value for each element in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; a value for each element in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each element centroid in the zone; interpolation is proportional to the fractional distance of the centroid location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used; if the interpolated value is less than or equal to 0 for a centroid, the element is inactive; if the interpolated value is greater than 0 for a centroid, the element is active. Line 11 illustrates method 1.

Line 12: **-mask** *property*

Use of this identifier is rare.

-mask—An array is used to include and exclude elements from the zone definition. Optionally, **mask** or **-m**[**ask**].

property—An array of 1s (or any positive number) and 0s (or any nonpositive number) is used to include and exclude elements from the zone definition. The array can be defined for the zone by any of four methods for defining a spatially distributed property; however, methods 1 and 4 are usually not useful: (1) a single number, 1 or 0, is used to include or exclude all of the elements of the zone definition, (2) **by_element** followed by a 1 or 0 for each element (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name—a 1 or 0 for each element (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each element centroid in the zone; interpolation is proportional to the fractional distance of the centroid location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used; if the interpolated value is less than or equal to 0 for an element, the element is excluded from the zone definition; if the interpolated value is greater than 0, the element is included in the zone definition. Line 12 illustrates method 3.

Notes

Media properties are defined by element, not by cell. The number of elements is one fewer in each coordinate direction than the number of nodes. The number of elements listed using property input methods 2 and 3 must match exactly the number of elements with centroids that are within the zone definition. When using method 3 or 4 to define a porous media property, a value must be supplied for each element in the zone, even if an element has been specified to be inactive. However, porous media properties assigned to inactive elements will have no effect on the simulation.

Multiple zones may be used within the **MEDIA** data block to media properties within the grid region. Different media properties for a single element may be defined multiple times as part of different, overlapping zone definitions. An individual media property that is used for an element is the last zone definition that defines that media property for that element.

By definition of active and inactive zones, it is possible to model a configuration other than the entire grid region defined by the **GRID** data block. Zones specified to be inactive are logically removed from the grid region. The degree to which the active grid region conforms to a desired three-dimensional shape is limited by the spatial discretization of the grid and the effort necessary to define a sufficient number of zones to describe the shape accurately. Elements are specified to be inactive by the use of the **-active** identifier in the **MEDIA** data block. The parts of cells that are contained in inactive elements are not included in the active grid region. If all elements that join at a node are inactive, then that node is inactive and is removed from the simulation.

To allow for the definition of an active grid region with a regional topography of land surface, inactive elements are allowed above active elements. That is, the potentially saturated region of an unconfined flow simulation does not have to possess a flat upper surface. Inactive elements may not be defined for one-dimensional simulations (**-chemistry_dimensions** in **GRID** data block). If unconfined flow is simulated (**FREE_SURFACE_BC** data block), the specific storage is reset to zero regardless of the value entered in the **MEDIA** data block.

Dispersivity is a mixing parameter with a magnitude that is dependent on the scale of observation distance over which solutes are transported in the flow system. A review of field data by Gelhar and others (1992) shows that the longitudinal dispersivity (**-longitidinal_dispersivity**) is on the order of 10 percent of the longitudinal transport distance, whereas horizontal transverse dispersivity (**-horizontal_dispersivity)** is on the order of 1 percent of the longitudinal transport distance, and vertical transverse dispersivity (**-vertical_dispersivity**) is on the order of 0.001 to 0.1 percent of the longitudinal transport distance.

Example Problems

The **MEDIA** data block is used in the example problems 1, 2, 3, and 4.

PRINT_FREQUENCY

This keyword data block is used to select the results to be written to output files and the frequencies at which the results are written. This data block and all identifiers are optional. The data for time “0 days” in the example below indicates the default settings. The effect of the default settings is to write data at the end of each simulation period for all identifiers with “end” or nonzero values, and to not write data for identifiers with values of zero or “false”. Explicit definition of an identifier in this data block will override the default setting. The times specified in this data block indicate times during the simulation when print frequencies will be changed. Only print frequencies explicitly specified following a time specification are changed. A print frequency, default or explicitly specified, remains in effect during a simulation until a specified time when the frequency for the identifier is explicitly changed.

Example

```
Line 0:   PRINT_FREQUENCY
Line 1:   0 days
Line 2:         -bc_flow_rates               0     days
Line 3:         -boundary_conditions         False
Line 4:         -components                  0     days
Line 5:         -conductances                0     yr
Line 6:         -flow_balance                1     step
Line 7:         -force_chemistry_print       0     step
Line 8:         -HDF_chemistry               1     yr
Line 9:         -HDF_heads                   1     yr
Line 10:        -HDF_velocities              1     yr
Line 11:        -heads                       end
Line 12:        -progress_statistics         1     step
Line 13:        -save_final_heads            False
Line 14:        -velocities                  0     yr
Line 15:        -wells                       end
Line 16:        -xyz_chemistry               0     days
Line 17:        -xyz_components              0     yr
Line 18:        -xyz_heads                   0     yr
Line 19:        -xyz_velocities              0     yr
Line 20:        -xyz_wells                   0     step
Line 1a:  1 yr
Line 16a:       -xyz_chemistry               0.1   yr
```

Explanation

Line 0: **PRINT_FREQUENCY**

PRINT_FREQUENCY is the keyword for the data block; no other data are included on this line.

Line 1: *time*, [*units*]

time—Simulation time at which to change print frequencies. Any print frequencies defined in lines following this time specification, and preceding the next time specification, or the end of the data block, will become effective at this time during the simulation. If any identifiers are included, this time will mark the beginning of a simulation period.

units—*Units* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Default units are defined by **-time** identifier in **UNITS** data block.

Line 2: **-bc_flow_rates** *frequency*, [*time unit* or **step**]

-bc_flow_rates—This identifier is used to control writing flow rates for all boundary-condition cells to the file *prefix***.O.bcf**. By the default setting, no flow rates are written to the file *prefix***.O.bcf**. Optionally, **bc_flow_rates** or **-bc_**[**flow_rates**].

frequency—Frequency at which flow rates are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient data will be written to the file *prefix*.**O**.**bcf**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 3: **-boundary_conditions** [***True*** or ***False***]

-boundary_conditions—This identifier is used to control writing heads, component concentrations, fluxes, and other boundary-condition information to the file *prefix***.O.probdef**. These data are written at most once per simulation period. By the default setting, no boundary-condition data are written to the file *prefix***.O.probdef**. Optionally, **boundary_conditions**, **boundary**, **bc**, **-b**[**c**], or **-b**[**oundary_conditions**].

True or ***False***—**True** causes writing of boundary-condition information to the file *prefix***.O.probdef**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 4: **-components** *frequency*, [*time unit* or **step**]

-components—This identifier is used to control writing total chemical element (component) data for each cell to the file *prefix*.**O**.**comps**. By the default setting, no component data are written to the file *prefix*.**O**.**comps**. Optionally, **component**, **components**, or **-com**[**ponents**].

frequency—Frequency at which component concentration data are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient data will be written to the file *prefix*.**O**.**comps**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 5: **-conductances** *frequency*, [*time unit* or **step**]

-conductances—This identifier is used to specify the frequency for writing static fluid and transient solute dispersive conductances for each cell face to the file *prefix***.O.kd**. By the default setting, no static or transient conductances are written to the file *prefix***.O.kd**. Optionally, **conductances, conductance,** or **-cond**[**uctances**].

frequency—Frequency at which conductances are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient data will be written to the file *prefix*.**O**.**kd**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 6: **-flow_balance** *frequency*, [*time unit* or **step**]

-flow_balance—This identifier is used to specify the frequency for writing flow-balance information to the file *prefix***.O.bal**. By the default setting, flow-balance information is written to the file *prefix***.O.bal** at the end of each simulation period. Optionally, **flow_balance** or **-f**[**low_balance**].

frequency—Frequency at which flow-balance data are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient data will be written to the file *prefix*.**O**.**bal**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 7: **-force_chemistry_print** *frequency*, [*time unit* or **step**]

-force_chemistry_print—This identifier is used to specify the frequency for writing detailed chemical descriptions of the composition of the solution and all reactants for each cell to the file *prefix***.O.chem**. *Warning*: this file could exceed file-size limits of the operating system because a long description of the chemistry in each cell is written for each selected time step. Writing this information may be useful for debugging, for small problems, or if the **PRINT_LOCATIONS** data block is used to limit the set of cells for which data are written. Data written to the file *prefix***.O.chem** also can be limited by the options of the **PRINT** data block of the chemistry data file. By the default setting, no detailed chemical descriptions are written to the file *prefix***.O.chem**. Optionally, **force_chemistry**, **force_chemistry_print**, or **-fo**[**rce_chemistry_print**].

frequency—Frequency at which chemistry data are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient data will be written to the file *prefix*.**O.chem**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 8: **-HDF_chemistry** *frequency*, [*time unit* or **step**]

-HDF_chemistry—This identifier is used to specify the frequency for writing chemistry data to the file *prefix*.**h5**. Chemistry data to be written to the file *prefix*.**h5** are defined in the **SELECTED_OUTPUT** and **USER_PUNCH** data blocks of the chemistry data file. By the default setting, chemistry data are written to the file *prefix*.**h5** at the end of each simulation period. Optionally, **hdf_chemistry**, **hdf_concentration**, **hdf_concentrations**, **-hdf_c**[**oncentrations**], or **-hdf_c**[**hemistry**].

frequency—Frequency at which chemistry data are written to the HDF file. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no chemistry data will be written to the file *prefix*.**h5**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 9: **-HDF_heads** *frequency*, [*time unit* or **step**]

-HDF_heads—This identifier is used to specify the frequency for writing heads to the file *prefix*.**h5**. By the default setting, heads are written to the file *prefix*.**h5** at the end of each simulation period. Optionally, **hdf_head**, **hdf_heads**, or **-hdf_h**[**eads**].

frequency—Frequency at which head data are written to the HDF file. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no head data will be written to the file *prefix*.**h5**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 10: **-HDF_velocities** *frequency*, [*time unit* or **step**]

-HDF_velocities—This identifier is used to specify the frequency for writing X, Y, and Z velocities to the file *prefix*.**h5**. By the default setting, velocities are written to the file *prefix*.**h5** at the end of each simulation period. Optionally, **hdf_velocity**, **hdf_velocities**, **-hdf_v**[**elocities**], or **-hdf_v**[**elocity**].

frequency—Frequency at which velocities at nodes are written to the HDF file. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no velocity data will be written to the file *prefix*.**h5**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 11: **-heads** *frequency*, [*time unit* or **step**]

-heads—This identifier is used to specify the frequency for writing potentiometric heads to the file *prefix*.**O.head**. By the default setting, heads are written to the file *prefix*.**O.head** at the end of each simulation period.Optionally, **head**, **heads**, or **-h**[**eads**].

frequency—Frequency at which head data are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient head data will be written to the file *prefix*.**O.head**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 12: **-progress_statistics** *frequency*, [*time unit* or **step**]

-progress_statistics—This identifier is used to specify the frequency for writing solver statistics, including solution-method information, number of iterations, and maximum changes in head and component concentrations (due to transport), to the file *prefix*.**log** and to the screen. By the default setting, solver statistics are written to the file *prefix*.**log** and screen at the end of each simulation period. Optionally, **progress_statistics**, **-pr**[**ogress_statistics**], **solver_statistics** or **-solv**[**er_statistics**].

frequency—Frequency at which solver statistics are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no data on solver iterations will be written to the file *prefix*.**log** or to the screen.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 13: **-save_final_heads** [***True*** or ***False***]

-save_final_heads—This identifier is used to control writing heads to the file *prefix***.head.dat** at the end of the simulation. The file *prefix***.head.dat** is an ASCII file than can be used for initial head conditions in subsequent simulations. Initial heads can be read from the file by using a zone that includes the entire grid region and “**-head file** *prefix***.head.dat**” in **HEAD_IC** data block. By the default setting, no head data are written to the file *prefix***.head.dat**. Optionally, **save_head**, **save_heads**, **save_final_heads**, **-sa**[**ve_heads**], or **-sa**[**ve_final_heads**].

True or ***False***—**True** writes heads at the end of the simulation to the file *prefix***.head.dat**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 14: **-velocities** *frequency*, [*time unit* or **step**]

-velocities—This identifier is used to specify the frequency for writing interstitial velocities at cell boundaries and interpolated velocities at nodes to the file *prefix***.O.vel**. By the default setting, no velocities are written to the file *prefix***.O.vel**. Optionally, **velocities**, **velocity** **-v**[**elocities**], or **-v**[**elocity**].

frequency—Frequency at which velocities are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient data will be written to the file *prefix*.**O.vel**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 15: **-wells** *frequency*, [*time unit* or **step**]

-wells—This identifier is used to specify the frequency for writing transient well information, including fluid and solute flow rates, cumulative fluid and solute flow amounts, and solute concentrations, to the file *prefix***.O.wel**. Data are written in the order of the well sequence numbers. By the default setting, well information is written to the file *prefix***.O.wel** at the end of each simulation period. Optionally, **wells** or **-w**[**ells**].

frequency—Frequency at which transient well information is written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient data will be written to the file *prefix*.**O.wel**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 16: **-xyz_chemistry** *frequency*, [*time unit* or **step**]

-xyz_chemistry—This identifier is used to specify the frequency for writing selected chemical data to the file *prefix***.xyz.chem**. The **SELECTED_OUTPUT** and **USER_PUNCH** data blocks of the chemistry data file are used to select data that are written to the file *prefix***.xyz.chem**. Cells for which results are to be written can be restricted with the **PRINT_LOCATIONS** data block. By the default setting, no data are written to the file *prefix***.xyz.chem**. Optionally, **concentrations**, **selected_output**, **selected_outputs**, **xyz_chemistry**, **-c**[**oncentrations**], **-se**[**lected_outputs**], or **-xyz_ch**[**emistry**].

frequency—Frequency at which transient concentration data are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient data will be written to the file *prefix***.xyz.chem**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 17: **-xyz_components** *frequency*, [*time unit* or **step**]

-xyz_components—This identifier is used to specify the frequency for writing component (chemical element) concentrations to the file *prefix***.xyz.comps**. By the default setting, no component concentrations are written to the file *prefix***.xyz.comps**. Optionally, **xyz_component**, **xyz_components**, or **-xyz_c**[**omponents**].

frequency—Frequency at which component data are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient data will be written to the file *prefix*.xyz**.comps**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 18: **-xyz_heads** *frequency*, [*time unit* or **step**]

-xyz_heads—This identifier is used to specify the frequency for writing heads to the file *prefix***.xyz.head**. By the default setting, no heads are written to the file *prefix***.xyz.head**. Optionally, **xyz_head**, **map_head**, **-xyz_h**[**ead**], or **-map_h**[**ead**].

frequency—Frequency at which head data are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient head data will be written to the file *prefix***.xyz.head**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 19: **-xyz_velocities** *frequency*, [*time unit* or **step**]

-xyz_velocities—This identifier is used to specify the frequency for writing interpolated velocities at cell nodes to the file *prefix***.xyz.vel**. By the default setting, no velocities are written to the file *prefix***.xyz.vel**. Optionally, **xyz_velocity**, **map_velocity**, **-xyz_v**[**elocity**], or **-map_v**[**elocity**].

frequency—Frequency at which velocities at cell nodes are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient data will be written to the file *prefix***.xyz.vel**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Line 20: **-xyz_wells** *frequency*, [*time unit* or **step**]

-xyz_wells—This identifier is used to specify the frequency for writing a time-series of concentrations for each well to the file *prefix***.xyz.wel**. By the default setting, no concentrations are written to the file *prefix***.xyz.wel**. Optionally, **xyz_well**, **xyz_wells**, **well_time_series**, **-xyz_w**[**ells**], **-well_**[**time_series**], **wells_time_series**, or **-wells_**[**time_series**].

frequency—Frequency at which concentrations are written. Frequency can be an interval of time or a number of time steps between writing results to the file, or the string “end”, which will cause writing at the end of the simulation period. If *frequency* is zero, no transient data will be written to the file *prefix***.xyz.wel**.

time unit or **step**—*Time unit* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Alternatively, if **step** is entered, results are written every *frequency* time steps in the simulation. Default units are defined by **-time** identifier in **UNITS** data block.

Notes

The **PRINT_FREQUENCY** data block controls writing of data during the simulation (times greater than zero), but it does not control printing data at the beginning of the simulation (time zero). The **PRINT_INITIAL** data block is used to control writing of time-zero information to output files, including initial conditions and media properties. Print frequencies may be changed at any time in the simulation by specifying the time at which the frequencies are to change (*time*), followed by the identifiers and frequencies that are to change. A new simulation period is begun for every time (*time*) that is specified.

User-specified units for the various print frequencies need not be the same. One **PRINT_FREQUENCY** data block may contain multiple time units and (or) the **step** unit. If time units are used for a print frequency, and the simulation time does not fall on an even multiple of the print frequency, then when the simulation time is within one time step of an even multiple of the print frequency, the time step will be decreased to reach the target simulation time at which printing is to occur. The time step will revert to the original time step until a smaller time step is needed to reach another target time for printing or the end of a simulation period.

A value of 0 (zero) for *frequency* suppresses writing of the specified data. By the default settings, print frequencies are set to “end” so that writing will occur at the end of each simulation period, except for frequencies defined by **-components**, **-conductances**, **-force_chemistry_print**, **-xyz_components**, **-xyz_heads**, **-xyz_velocities**, **-wells**, and **-xyz_wells**, which are set to 0 to suppress writing. Once set by default, or explicitly following a specified *time*, a print frequency remains in effect until the next *time* for which the print frequency is explicitly defined.

When using **steps** for a print frequency, writing to the files is determined by the number of time steps since the beginning of the simulation, not from the beginning of the current simulation period. For example, if a print frequency is specified to be every 2 steps beginning at time 0, every 3 steps beginning at time 5, and time step (**TIME_CONTROL**) is 1, the printing will occur at time steps 2 and 4 (evenly divisible by 2), 5 (last time step of the first simulation period), 6 and 9 (evenly divisible by 3), and 10 (last time step of the second simulation period). Similarly, when a frequency is specified in time units, printing will occur when the total time from the beginning of the simulation is evenly divisible by the specified frequency and at the end of each simulation period.

When steady-state flow is simulated (**STEADY_FLOW true**), nonzero print frequencies for the identifiers **-head** (*prefix***.O.head**), **-flow_balance** (*prefix***.O.bal**), **-velocity** (*prefix***.O.vel)**, **-hdf_heads** (*prefix***.h5**), **-hdf_velocities** (*prefix***.hdf.vel**), **-xyz_heads** (*prefix***.xyz.head**), and **-xyz_velocities** (*prefix***.xyz.vel**) are treated in a special way. During the iterations to achieve steady-state flow, heads are written to the *prefix***.O.head** file for every steady-state iteration; no data are written to this file during the transient part of the simulation. During the iterations to achieve steady-state flow, balances are written to the *prefix***.O.bal** file for every steady-state iteration; during the transient part of the simulation, the print frequency defined by **-flow_balance** is used to determine when data are written to the *prefix***.O.bal** file. For steady-flow simulations, velocities are written to the *prefix***.O.vel**, *prefix***.h5**, and *prefix***.xyz.vel** files only once if the *frequency* for the identifier corresponding to the file is nonzero for any simulation period within the flow and transport data file. Similarly, heads are written to their respective files only once. Printing of steady-flow heads and velocities also can be requested in the **PRINT_INITIAL** data block.

The files with names beginning with *prefix***.O** are data formatted to be printed or viewed on a screen. The orientation of the printout of the spatially distributed properties in the *prefix***.O** files, either X–Y or X–Z planes, is controlled by the **-print_orientation** identifier in the **GRID** data block. The files *prefix***.xyz.chem**, *prefix***.xyz.comps**, *prefix***.xyz.head**, *prefix***.xyz.vel**, and *prefix***.xyz.wel** are written in tab-separated columns to facilitate importing into spreadsheets and post-processing programs for graphical display.

Example Problems

The **PRINT_FREQUENCY** data block is used in the example problems 1, 2, 3, and 4.

PRINT_INITIAL

This keyword data block is used to print the initial and static flow and transport data to various output files. The output controlled by this data block is useful for verifying that media properties, and initial and boundary conditions have been defined correctly. The example below indicates the default settings for each of the identifiers. The effect of the default setting is to not write data for identifiers with values of “false” and to write data for identifiers with values of “true” in the example. Options in this data block apply only to initial conditions and have no effect on printing data during a simulation (simulation time greater than zero).

Example

```
Line 0:   PRINT_INITIAL
Line 1:         -boundary_conditions          false
Line 2:         -components                   false
Line 3:         -conductances                 false
Line 4:         -echo_input                   true
Line 5:         -fluid_properties             true
Line 6:         -force_chemistry_print        false
Line 7:         -HDF_chemistry                true
Line 8:         -HDF_heads                    true
Line 9:         -HDF_steady_flow_velocities   true
Line 10:        -heads                        true
Line 11:        -media_properties             false
Line 12:        -solution_method              true
Line 13:        -steady_flow_velocities       false
Line 14:        -wells                        true
Line 15:        -xyz_chemistry                false
Line 16:        -xyz_components               false
Line 17:        -xyz_heads                    false
Line 18:        -xyz_steady_flow_velocities   false
Line 19:        -xyz_wells                    false
```

Explanation

Line 0: **PRINT_INITIAL**

PRINT_INITIAL is the keyword for the data block; no other data are included on this line.

Line 1: **-boundary_conditions** [***True*** or ***False***]

-boundary_conditions—This identifier is used to control writing initial boundary-condition information to the file *prefix***.O.probdef**, including data for specified-head, specified-flux, and leaky boundary conditions and the concentrations of all components in solutions related to boundary conditions. By the default setting, no initial boundary-condition information is written to the file *prefix***.O.probdef**. Optionally, **boundary_conditions**, **boundary**, **bc**, **-b**[**oundary_conditions**], or **-b**[**c**].

True or ***False***—**True** writes static boundary-condition data to the file *prefix***.O.probdef**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 2: **-components** [***True*** or ***False***]

-components—This identifier is used to control writing initial indices and mixing fractions for solutions, equilibrium phases, exchangers, surfaces, gas phases, solid solutions, and kinetic reactions that define initial conditions for the simulation and initial component concentrations to the file *prefix***.O.comps**. By the default setting, no initial indices, mixing fractions, and component concentrations are written to the file *prefix***.O.comps**. Optionally, **component**, **components**, or **-c**[**omponents**].

True or ***False***—**True** writes initial component concentrations and indices and mixing fractions to the file *prefix***.O.comps**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 3: **-conductances** [***True*** or ***False***]

-conductances—This identifier is used to control writing static fluid-conductance factors and initial-condition conductances to the file *prefix***.O.kd**. By the default setting, no conductance factors or initial-condition conductances are written to the file *prefix***.O.kd**. Optionally, **conductance**, **conductances**, or **-con**[**ductances**].

True or ***False***—**True** writes conductance data to the file *prefix***.O.kd**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 4: **-echo_input** [***True*** or ***False***]

-echo_input—This identifier is used to control writing of lines from the flow and transport data file to the file *prefix***.log** as they are processed. The option takes effect as soon as it is encountered in the flow and transport data file. (Writing lines from the chemistry data file to the file *prefix***.log** is controlled by **-echo_input** in the **PRINT** data block of the chemistry data file.) By the default setting, lines from the flow and transport data file are written to the file *prefix***.log**. Optionally, **echo_input** or **-e**[**cho_input**].

True or ***False***—**True** writes lines from the flow and transport data file to the file *prefix***.log** as they are processed; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 5: **-fluid_properties** [***True*** or ***False***]

-fluid_properties—This identifier is used to control writing fluid property data to the file *prefix***.O.probdef**, including compressibility, molecular diffusivity, viscosity, and density. By the default setting, fluid properties are written to the file *prefix***.O.probdef**. Optionally, **fluid_properties**, **fluid**, or **-f**[**luid_properties**].

True or ***False***—**True** writes input fluid properties to the file *prefix***.O.probdef**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 6: **-force_chemistry_print** *frequency*, (*time unit* or **step**)

-force_chemistry_print—This identifier is used to control writing detailed chemical descriptions of the composition of the solution and all reactants for each cell to the file *prefix***.O.chem**. *Warning*: this file could be huge because it will produce a long description of the chemistry for each cell. Writing this information may be useful for debugging, for small problems, or if the cells for which writing results are restricted by cell selections made in the **PRINT_LOCATIONS** data block. Data written to the file can be restricted by options within the **PRINT** data block of the chemistry data file. By the default setting, no chemical descriptions of solution and reactant compositions are written to the file *prefix***.O.chem**. Optionally, **force_chemistry**, **force_chemistry_print**, or **-fo**[**rce_chemistry_print**].

True or ***False***—**True** writes detailed chemical description of each cell to the file *prefix***.O.chem**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 7: **-HDF_chemistry** [***True*** or ***False***]

-HDF_chemistry—This identifier is used to control writing chemistry data at time zero to the file *prefix***.h5**. Data to be written are defined in **SELECTED_OUTPUT** and **USER_PUNCH** data blocks of the chemistry data file. By the default setting, chemistry data at time zero are written to the file *prefix***.h5**. Optionally, **hdf_chemistry**, **hdf_concentration**, **hdf_concentrations**, **-hdf_c**[**hemistry**], or **-hdf_c**[**oncentrations**].

True or ***False***—**True** writes initial chemistry data to the file *prefix***.h5**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 8: **-HDF_heads** [***True*** or ***False***]

-HDF_heads—This identifier is used to control writing initial heads to the file *prefix***.h5**. By the default setting, initial heads are written to the file *prefix***.h5**. Optionally, **hdf_head**, **hdf_heads**, or **-hdf_h**[**eads**].

True or ***False***—**True** writes initial heads to the file *prefix***.h5**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 9: **-HDF_steady_flow_velocities** [***True*** or ***False***]

-HDF_steady_flow_velocities—This identifier is used to control writing steady-flow velocities to the file *prefix***.h5**. This option has meaning only if steady-state flow is specified in the **STEADY_FLOW** data block. By the default setting, steady-flow velocities are written to the file *prefix***.h5**. Optionally, **hdf_steady_flow_velocity**, **hdf_steady_flow_velocities**, **-hdf_s**[**teady_flow_velocity**], **-hdf_s**[**teady_flow_velocities**], **hdf_ss_velocity**, **hdf_ss_velocities**, **-hdf_s**[**s_velocity**], or **-hdf_s**[**s_velocities**].

True or ***False***—**True** writes steady-flow velocities to the file *prefix***.h5**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 10: **-heads** [***True*** or ***False***]

-heads—This identifier is used to control writing initial heads to the file *prefix***.O.head**. By the default setting, initial heads are written to the file *prefix***.O.head**. Optionally, **head**, **heads**, or **-h**[**eads**].

True or ***False***—**True** writes initial heads to the file *prefix***.O.head**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 11: **-media_properties** [***True*** or ***False***]

-media_properties—This identifier is used to control writing problem definition data for all the media properties for porous media zones, including element zone definitions, porosities, hydraulic conductivities, specific storages, and dispersivities to the file *prefix***.O.probdef**. By the default setting, no problem definition data are written to the file *prefix***.O.probdef**. Optionally, **media_properties**, **media**, **medium**, **-m**[**edia_properties**], or **-m**[**edium**].

True or ***False***—**True** writes input media properties to the file *prefix***.O.probdef**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 12: **-solution_method** [***True*** or ***False***]

-solution_method—This identifier is used to control writing input data related to the solution method for flow and transport equations to the file *prefix***.O.probdef**, including type of solver used and relevant solver parameters. By the default setting solution-method data are written to the file *prefix***.O.probdef**. Optionally, **solution_method**, **method**, **-s**[**olution_method**], or **-met**[**hod**].

True or ***False***—**True** writes data for the solution method to the file *prefix***.O.probdef**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 13: **-steady_flow_velocities** [***True*** or ***False***]

-steady_flow_velocities—This identifier is used to control writing steady-flow velocities to the file *prefix***.O.vel**. This option has meaning only if steady-state flow is specified in the **STEADY_FLOW** data block. By the default setting, no steady-flow velocities are written to the file *prefix***.O.vel**. Optionally, **steady_flow_velocity**, **steady_flow_velocities**, **-st**[**eady_flow_velocity**], **-st**[**eady_flow_velocities**], **ss_velocity**, **ss_velocities**, **-ss**[**_velocity**], or **-ss**[**_velocities**].

True or ***False***—**True** writes steady-flow velocities to the file *prefix***.O.vel**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 14: **-wells** [***True*** or ***False***]

-wells—This identifier is used to control writing static well information, including location, diameter, screened intervals, and well indices to the file *prefix***.O.wel**. By the default setting, static well information is written to the file *prefix***.O.wel**. Optionally, **wells**, or **-w**[**ells**].

True or ***False***—**True** writes static well information to the file *prefix***.O.wel**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 15: **-xyz_chemistry** [***True*** or ***False***]

-xyz_chemistry—This identifier is used to control writing initial chemistry data to the file *prefix***.xyz.chem**. The **SELECTED_OUTPUT** and **USER_PUNCH** data blocks of the chemistry data file are used to select data that are written to the file *prefix***.xyz.chem**. Cells for which results are to be written can be restricted with the **PRINT_LOCATIONS** data block. By the default setting, no initial chemistry data are written to the file *prefix***.xyz.chem**. Optionally, **xyz_chemistry** or **-xyz_ch**[**emistry**].

True or ***False***—**True** writes initial chemistry data to the file *prefix***.xyz.chem**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 16: **-xyz_components** [***True*** or ***False***]

-xyz_components—This identifier is used to control writing initial component concentrations to the file *prefix***.xyz.comps**. By the default setting, no initial component concentrations are written to the file *prefix***.xyz.comps**. Optionally, **xyz_component**, **xyz_components**, or **-xyz_c**[**omponents**].

True or ***False***—**True** writes initial component concentrations to the file *prefix***.xyz.comps**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 17: **-xyz_heads** [***True*** or ***False***]

-xyz_heads—This identifier is used to control writing initial heads to the file *prefix***.xyz.head**. By the default setting, no initial heads are written to the file *prefix***.xyz.head**. Optionally, **xyz_head**, **xyz_heads**, or **-xyz_h**[**eads**].

True or ***False***—**True** writes initial heads to the file *prefix***.xyz.head**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 18: **-xyz_steady_flow_velocities** [***True*** or ***False***]

-xyz_steady_flow_velocities—This identifier is used to control writing velocities from the steady-flow calculation to the file *prefix*.**xyz.vel**. This option has meaning only if steady-state flow is specified in the **STEADY_FLOW** data block. By the default setting, no steady-flow velocities are written to the file *prefix*.**xyz.vel**. Optionally, **xyz_steady_flow_velocity**, **xyz_steady_flow_velocities**, **-xyz_s**[**teady_flow_velocity**], **-xyz_s**[**teady_flow_velocities**], **xyz_ss_velocity**, **xyz_ss_velocities**, **-xyz_s**[**s_velocity**], or **-xyz_s**[**s_velocities**].

True or ***False***—**True** writes steady-flow velocities to the file *prefix*.**xyz.vel**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 19: **-xyz_wells** [***True*** or ***False***]

-xyz_wells—This identifier is used to control writing initial concentrations at wells to the file *prefix*.**xyz.wel**. By the default setting, no initial concentrations at wells are written to the file *prefix*.**xyz.wel**. Optionally, **xyz_well**, **xyz_wells**, or **-xyz_w**[**ells**].

True or ***False***—**True** writes initial concentrations to the file *prefix*.**xyz.wel**; **false** suppresses writing. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Notes

Default settings at the beginning of a simulation are **true** for **-echo_input**, **-fluid_properties**, **-HDF_chemistry**, **-HDF_heads**, **-HDF_steady_flow_velocities**, **-heads**, **-solution_method**, and **-wells**; default settings at the beginning of a simulation are false for all other identifiers. The **PRINT_INITIAL** options are used only once, prior to any transient calculations. The files with names beginning with *prefix*.**O** are data formatted to be printed or viewed on a screen. The orientation of the printout of the spatially distributed properties in the *prefix*.**O** files, either X–Y or X–Z planes, is controlled by the **-print_orientation** identifier in the **GRID** data block. The files *prefix*.**xyz.chem**, *prefix*.**xyz.comps**, *prefix*.**xyz.head**, *prefix*.**xyz.vel**, and *prefix*.**xyz.wel** are written in tab-separated columns to facilitate importing into spreadsheets and post-processing programs for graphical display.

Example Problems

The **PRINT_INITIAL** data block is used in the example problems 3 and 4.

PRINT_LOCATIONS

This keyword data block is used to limit printing of results to the chemistry output file (*prefix*.**O.chem**) and (or) the spreadsheet file *prefix*.**xyz.chem** to a subset of cells within the grid region. In the absence of this data block, results are printed for all active nodes, if printing is enabled. The **PRINT** data block of the chemistry data file specifies the data to be written to the file *prefix*.**O.chem**. The **SELECTED_OUTPUT** and **USER_PUNCH** data blocks in the chemistry data file are used to specify the data to be included in the file *prefix*.**xyz.chem**. The **PRINT_LOCATIONS** data block contains only static data.

Example

```
Line 0:   PRINT_LOCATIONS
Line 1:         -chemistry
Line 2:               -zone       0     0     0     10    10    10
Line 3:                     -print      0
Line 2a:              -zone       10    0     10    10    10    10
Line 3a:                    -print      1
Line 4:         -xyz_chemistry
Line 5:               -sample X 2
Line 5a:              -sample Y 2
Line 2b:              -zone       0     0     0     10    10    10
Line 3b:                    -print      1
Line 6:                     -mask       file        mask.file
```

Explanation

Line 0: **PRINT_LOCATIONS**

PRINT_LOCATIONS is the keyword for the data block; no other data are included on this line. Optionally, **PRINT_LOCATION**.

Line 1: **-chemistry**

-chemistry—Zone definitions following this identifier are used to select nodes for which results are written to the file *prefix*.**O.chem**. Optionally, **chemistry** or **-c**[**hemistry**].

Line 2: **-zone** $x_1, y_1, z_1, x_2, y_2, z_2$

-zone—The zone selects a set of nodes for which printing will be turned on or off. Optionally, **zone** or **-z**[**one**].

x_1, y_1, z_1—X, Y, and Z coordinate for the left (X), front (Y), lower (Z) corner of the rectangular zone.

x_2, y_2, z_2—X, Y, and Z coordinate for the right (X), back (Y), upper (Z) corner of the rectangular zone. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

Line 3: **-print** *property*

-print—This identifier is used to control writing results for cells selected by the zone to the file *prefix*.**O.chem** (when **-print** follows **-chemistry**) or the file *prefix*.**xyz.chem** (when **-print** follows **-xyz_chemistry**). Optionally, **print** or **-p**[**rint**].

property—A value of 1 or positive number indicates results will be written to the specified file for nodes selected by the zone; a value of 0 or nonpositive number indicates results will not be written. Values for **-print** can be entered with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_node** followed by a value for each node within the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; a value for each node in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used; if the interpolated value is less than or equal to 0 for a node, results will not be printed for the node; if the interpolated value is greater than 0, results will be printed for the node. Lines 3, 3a, and 3b illustrate method 1.

Line 4: **-xyz_chemistry**

-xyz_chemistry—Zone definitions following this identifier are used to select nodes for which results are written to the file *prefix*.**xyz.chem**. Optionally, **xyz_chemistry** or **-x**[**yz_hemistry**].

Line 5: **-sample** (**X**, **Y**, or **Z**) *sample frequency*

-sample—This identifier is used to specify a subgrid for writing results to the file *prefix*.**O.chem** (when **-sample** follows **-chemistry**) or the file *prefix*.**xyz.chem** (when **-sample** follows **-xyz_chemistry**). Optionally, **sample**, **sample_grid**, **thin**, **thin_grid**, **-s**[**ample_grid**] or **-t**[**hin_grid**].

X, **Y**, or **Z**—The coordinate direction for which a subset of grid nodes will be selected.

sample frequency—Writing to the specified file will occur for the first and last nodes in the coordinate direction (provided these nodes are active); between the first and last nodes, results will be printed at node intervals of *sample frequency*. If *sample frequency* is 2, then data will be written for every other node. To preserve the edges of the grid region, the nodes are selected by working to the interior of the grid region from each end of the grid in the specified coordinate direction.

Line 6: **-mask** *property*

-mask—An array is used to include and exclude nodes from the zone definition. Optionally, **mask** or **-m**[**ask**].

property—An array of 1s (or any positive number) and 0s (or any nonpositive number) is used to include and exclude nodes from the zone definition. The array can be defined by any of four methods for defining a spatially distributed property; however, methods 1 and 4 are usually not useful: (1) a single number, 1 or 0, is used to include or exclude all of the nodes of the zone definition, (2) **by_node** followed by a 1 or 0 for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets “< >”, (3) **file** followed by a file name—a 1 or 0 for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used; if the interpolated value is less than or equal to 0 for a node, the node is excluded from the zone definition; if the interpolated value is greater than 0, the node is included in the zone definition. Line 6 illustrates method 3.

Notes

Frequency of printing to the files *prefix*.**O.chem** and *prefix*.**xyz.chem** is controlled by the respective identifiers **-chemistry** and **-xyz_chemistry** in the **PRINT_FREQUENCY** data block. By default, when printing to one of these files is enabled, printing will occur for all active nodes (**MEDIA** data block, identifier **-active**) for the dimensions for which chemistry is calculated (**GRID** data block, identifier **-chemistry_dimensions**). The **PRINT_LOCATIONS** data block may be used to limit the nodes selected for printing by specifying **-print 0** for some nodes. It may be convenient to define **-print 0** for all nodes in the grid region, and then define **-print 1** for selected nodes. The number of nodes selected for writing chemical data to the file *prefix*.**O.chem** should be kept to a minimum to avoid generating an output that could exceed file-size limits of the computer operating system.

If both **-sample_grid** and **-print** identifiers are defined, the **-print** definitions supersede the **-sample_grid** definitions. If a node is included in more than one zone, the value specified for **-print** in the last definition will apply.

Example Problems

The **PRINT_LOCATIONS** data block is used in the example problem 4.

RIVER

This keyword data block is used to define boundary conditions that describe a single river. This data block is optional and only is needed if river boundary conditions are included in the simulation. A river is defined by a series of X–Y coordinate points, which define line segments that locate the centerline of the river across the land surface. For each river point, the river width, head, and river-bed-leakance parameters must be defined, either by explicit definitions or by interpolation from other river points. Leakage into and out of the aquifer through the riverbed is calculated from these parameters. Head and solution composition may vary over the course of the simulation. For a vertical stack of cells, any river leakage enters or leaves the cell that contains the water table. Multiple **RIVER** data blocks are used to define multiple rivers in the grid region.

Example

```
Line 0:   RIVER 1 Rubicon River
Line 1:         -point  155.0     3633. # X, Y location of 1st river point
Line 2:               -head
Line 3:                           0     day   275.
Line 3a:                          1.5   day   276.
Line 4:               -solution
Line 5:                           0     4
Line 5a:                          1.8   5
Line 6:               -width                  125.
Line 7:               -bed_hydraulic_conductivity         1.5e-2
Line 8:               -bed_thickness          1.6
Line 9:               -depth                  3.5
Line 1a:        -point  165.0     3663.  # X, Y location of 2nd river point
Line 1b:        -point  175.0     3603.  # X, Y location of 3rd river point
Line 2a:              -head
Line 3b:                          0     274.
Line 3c:                          2.1   274.5.
Line 4a:              -solution
Line 5b:                          0     4
Line 6a:              -width                  150.
Line 7a:              -bed_hydraulic_conductivity         1.5e-2
Line 8a:              -bed_thickness          2.
Line 10:              -bottom                 271.5
```

Explanation

Line 0: **RIVER** *number,* [*description*]

RIVER is the keyword for the data block.

number—positive number to designate this river. Default is 1.

description—optional character field that identifies the river.

Line 1: **-point** *X*, *Y*

-point—The X–Y coordinate location of a river point is defined. Line 1 may be repeated as many times as needed to define the entire length of the river. At least two points must be defined. The series of points in sequence define the river by a series of line segments. Optionally, **point**, **-p[oint]**, **node**, or **-n**[**ode**].

X—X coordinate location of a river point. Units, L, are defined by the **-horizontal_grid** identifier in the **UNITS** data block.

Y—Y coordinate location of a river point. Units, L, are defined by the **-horizontal_grid** identifier in the **UNITS** data block.

Line 2: **-head**

-head—This identifier is used to specify the head at the X–Y river-point location. Units, L, are defined by the **-head** identifier in the **UNITS** data block. The head is defined on line 3. A time series of heads may be defined by using multiple line 3s. The first *time* in the series must be zero. Optionally, **head**, or **-he**[**ad**].

Line 3: *time* [*units*] *head*

time—Simulation time (T) at which the head definition (*head*) will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

head—Head at the river point starting at *time*.

Line 4: **-solution**

-solution—This identifier is used to specify the solution index number for the X–Y river-point location. Solution index numbers correspond to solution compositions defined in the chemistry data file. The composition of water at the river point is defined in line 5. A time series of solution compositions may be defined by using multiple line 5s. The first *time* in the series must be zero. Optionally, **solution**, **associated_solution**, **-s**[**olution**], or **-a**[**ssociated_solution**].

Line 5: *time* [*units*] *solution*

time—Simulation time (T) at which the solution composition will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

solution—Solution index number for the river point starting at *time*.

Line 6: **-width** *width*

-width—This identifier is used to specify the width of the river at the X–Y river-point location. Units, L, are defined by the **-horizontal_grid** identifier in the **UNITS** data block. Optionally, **width**, or **-w**[**idth**].

width—Width of the river at the river point.

Line 7: **-bed_hydraulic_conductivity** *bed_hydraulic_conductivity*

-bed_hydraulic_conductivity—This identifier is used to specify the hydraulic conductivity for the riverbed at the X–Y river-point location. Units, L/T, are defined by the **-river_bed_hydraulic_conductivity** identifier in the **UNITS** data block. Optionally, **bed_hydraulic_conductivity**, **bed_k**, **hydraulic_conductivity**, **k**, **-bed_h**[**ydraulic_conductivity**], **-bed_k**, **-hy**[**draulic_conductivity**], or **-k**.

bed_hydraulic_conductivity—Hydraulic conductivity of the riverbed at the river point.

Line 8: **-bed_thickness** *bed_thickness*

-bed_thickness—This identifier is used to specify the thickness of the riverbed at the X–Y river-point location. Units, L, are defined by the **-river_bed_thickness** identifier in the **UNITS** data block. Optionally, **bed_thickness**, **thickness**, **-bed_t**[**hickness**], or **-t**[**hickness**].

bed_thickness—Thickness of the riverbed at the river point.

Line 9: **-depth** *depth*

-depth—This identifier is used to specify the depth of the river at the X–Y river-point location and is alternative to the **-bottom** identifier. The depth of the river is subtracted from the initial head defined for the river point to define the elevation of the bottom of the river at that point. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block. Optionally, **depth**, or **-dep**[**th**].

depth—Depth of the river at the river point.

Line 10: **-bottom** *bottom*

-bottom—This identifier is used to specify the elevation of the top of the riverbed at the X–Y river-point location and is alternative to the **-depth** identifier. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block. Optionally, **bottom**, **z**, **river_bottom**, **-bo**[**ttom**], **-z**, or **-r**[**iver_bottom**].

bottom—Elevation of the top of the riverbed at the river point.

Notes

River points can be defined in upstream or downstream order. Multiple rivers are defined by using multiple **RIVER** data blocks, where each river is uniquely identified by the integer following the **RIVER** keyword. A sequential set of two points within a **RIVER** data block define a line segment. The head, width, and river-bed elevation, hydraulic conductivity, and thickness at each point are used to quantify river leakage to or from the aquifer. The elevation of the river bottom can be defined in two ways: it can be defined explicitly with **-river_bottom** identifier, or it can be defined by a river depth (**-depth**) that will be subtracted from the head (**-head**) that applies at time zero. For reactive-transport simulation, a solution composition is associated with each point that defines the composition of water leaking from the river into the aquifer.

Parameters may be defined explicitly for a river point, or they may be interpolated from other river points. The head, width, and river-bed elevation or depth, hydraulic conductivity, thickness, and solution (for reactive transport simulations) must be defined for the first and last points of the river. For a point between the first and last points, a parameter can be defined explicitly with its identifier (**-head**, for example), or it can be interpolated from the nearest river points upstream and downstream where the parameter has been explicitly defined. Linear interpolation is performed by the ratio of the distance of the interpolation point from one of the explicitly defined points and the total distance along the river line segments that lie between the two explicitly defined points. Solution compositions at river points are determined by explicit definition (**-solution** identifier) or by mixing explicitly defined solutions at other river points by mixing fractions determined by linear interpolation.

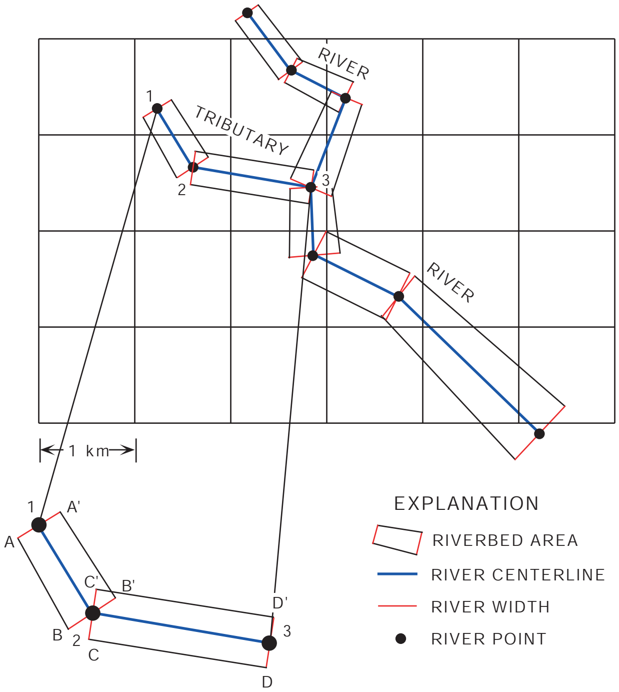


Figure 4.6.—Discretization of river segments for a river and its tributary. Initial river-bed areas are shown before fill-in of gaps and removal of overlapping areas.

Figure 4.6 shows an example of points that define a river and its tributary and the initial set of polygons that are processed to define the river in each cell that the polygons intersect (see section D.5.5. River-Leakage Boundary for details). The line segment (fig. 4.6, line segment 1–2) joining two sequential river points forms the centerline of a trapezoid. At each of the two river points, the width of the river is defined by a line segment perpendicular to the centerline (fig. 4.6, lines A-A’ and B-B’). A trapezoid is formed by connecting the endpoints of the two river-width line segments (fig. 4.6, lines A-B and A’-B’). Several steps are involved in processing the data at a series of river points to arrive at the parameter values that describe the river in each grid cell (see section D.5.5. River-Leakage Boundary). The trapezoids are modified to add areas between adjacent trapezoids and to remove duplicated areas. These modified polygons are intersected with the grid cells, and properties of the river areas in each grid cell are interpolated from the properties at the river points.

Complete explicit or implicit definition of all properties for all river points is required for time zero. Implicit definition is by interpolation. A time-sequence of values can be defined for the head (**-head**) and the solution composition (**-solution**) for a river point. All interpolation is repeated at each time that head or solution change at any river point. At the beginning of each simulation period, terms that incorporate both the transient head and solution composition and the static area and river-bed definition for each cell are incorporated into the flow and transport equations.

A river boundary condition always applies to the positive Z face of a cell. Both a river and a flux boundary condition may be defined for the positive Z face of a cell; however, a river and a leaky boundary condition cannot both be defined for the positive Z face of a cell. If both a river and a specified-head boundary condition are defined for a cell, the cell is a specified-head boundary cell and the river boundary condition has no effect. Rivers cannot be used in confined flow simulations. See section D.5.7. Boundary-Condition Compatibility for more details.

Example Problems

The **RIVER** data block is used in the example problem 4.

SOLUTE_TRANSPORT

This keyword data block is a switch used to toggle transport and reaction calculations on or off. If the **SOLUTE_TRANSPORT** data block is not included, the simulation is assumed to be a reactive-transport simulation. If **SOLUTE_TRANSPORT** is set to false, any keyword data blocks or identifiers related to transport or chemistry in the flow and transport data file are ignored. Molecular diffusivity is the only parameter that can be defined in this data block and the diffusivity only is pertinent to transport calculations. The diffusivity is a scalar value that is uniform over the entire grid region; it is not spatially variable. The **SOLUTE_TRANSPORT** data block contains only static data.

Example

```
Line 0:  SOLUTE_TRANSPORT               true
Line 1:        -diffusivity             1e-9        # m^2/s
```

Explanation

Line 0: **SOLUTE_TRANSPORT** [***True*** or ***False***]

SOLUTE_TRANSPORT—This keyword is used to switch the program from a reactive-transport simulation to a flow-only simulation. By default, a simulation includes flow, transport, and reaction calculations.

[***True*** or ***False***]—A value of **true** indicates that transport and reaction calculations are made. A value of **false** indicates that only flow is simulated and no transport or reaction calculations are to be done. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 1: **-diffusivity** *diffusivity*

-diffusivity—This identifier is used to specify molecular diffusivity of all chemical solutes. The diffusivity only applies when **SOLUTE_TRANSPORT** is defined to be **true**. Default is 10^{-9} square meter per second (m^2/s). Optionally, **diffusivity** or **-d**[**iffusivity**].

diffusivity—Molecular diffusivity, m^2/s.

Notes

The diffusivity is used in simulations only if the transport equations are solved, that is, when **SOLUTE_TRANSPORT** is true. The units for diffusivity are square meters per second (m^2/s); no option for alternative units is available.

Example Problems

The **SOLUTE_TRANSPORT** data block is used in the example problems 1, 2, 3, and 4.

SOLUTION_METHOD

This keyword data block is used to select the linear equation solver that is used to solve the finite-difference flow and transport equations and to set parameters related to the solver and the formulation of the finite-difference equations. The two solvers available are a D4 Gaussian elimination direct solver and a restarted generalized conjugate-gradient (ORTHOMIN) iterative solver (section D.7.2. Linear-Equation Solvers for Flow and Transport Finite-Difference Equations). If this data block is not included, the iterative solver with default parameter values is used. The **SOLUTION_METHOD** data block contains only static data.

The choice of equation solver depends on consideration of the memory storage requirement and computational workload. Both factors are related to the size of the problem being simulated and to the rate of convergence of the iterative method. The importance of each factor depends on the computer system used. Thus, it is difficult to give a definitive formula for the selection of the solver. Small problems of a few hundred nodes are most efficiently solved with the direct solver. The iterative solver usually is more efficient for simulations with several thousand nodes or more. The PHAST simulator writes out the storage requirements for the solver selected into the *prefix*.**log** file.

Example 1

```
Line 0:   SOLUTION_METHOD
Line 1:         -iterative_solver       true
Line 2:         -tolerance              1e-10
Line 3:         -save_directions        10
Line 4:         -maximum_iterations     500
Line 5:         -space_differencing     0.0
Line 6:         -time_differencing      1.0
Line 7:         -cross_dispersion      false
```

Explanation 1

Line 0: **SOLUTION_METHOD**

SOLUTION_METHOD is the keyword for the data block; no other data are included on this line.

Line 1: **-iterative_solver** [***True*** or ***False***]

-iterative_solver—This identifier is used to specify whether the iterative solver will be used for the flow and transport equations. By default, the program will use the iterative solver. Optionally, **iterative_solver**, **iterative**, or **-iterativ**[**e_solver**].

True or ***False***—**True**, use the iterative solver; **false**, use the direct solver. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 2: **-tolerance** *tolerance*

-tolerance—This identifier is used to specify a tolerance for the iterative solver. The iterative solver has converged to a numerical solution when the scaled residual is smaller than the tolerance. Default is $1x10^{-10}$ (-). Optionally, **tolerance**, or **-to**[**lerance**].

tolerance—Convergence criterion for the iterative solver.

Line 3: **-save_directions** *number*

-save_directions—This identifier is used to specify the number of search directions that are saved between restarts of the iterative solver. Default is 10. Optionally, **save_directions**, **save**, or **-sa**[**ve_directions**].

number—Number of saved search directions.

Line 4: **-maximum_iterations** *number*

-maximum_iterations—This identifier is used to specify the maximum number of iterations that are allowed for the iterative solver. Default is 500. Optionally, **maximum_iterations**, **maximum**, **iterations**, **-m**[**aximum_iterations**], or **-iterati**[**ons**].

number—Maximum number of iterations for the iterative solver.

Line 5: **-space_differencing** *weight*

-space_differencing—This identifier is used to specify the weighting used in spatial differencing for the advective term in the transport equation. Default is upstream weighting. Optionally, **space_differencing**, **space**, or **-sp**[**ace_differencing**].

weight—The weighting can range from 0.0 (upstream-in-space differencing) to 0.5 (centered-in-space differencing).

Line 6: **-time_differencing** *weight*

-time_differencing—This identifier is used to specify the time differencing for the flow and transport equations during transient calculations. [This weighting factor does not apply to steady-state flow equations where fully implicit weighting (*weight* = 1.0) is used.] Default is fully implicit (1.0). Optionally, **time_differencing**, **time**, or **-ti**[**me_differencing**].

weight—Value can range from 0.5 (centered-in-time differencing) to 1.0 (fully implicit or backwards-in-time differencing).

Line 7: **-cross_dispersion** [***True*** or ***False***]

-cross_dispersion—This identifier is used to include cross-dispersion terms in the transport equations. By default, cross-dispersion terms are not included. Optionally, **cross_dispersion**, or **-c**[**ross_dispersion**].

True or ***False***—**True**, include cross-dispersion terms in transport equations; **false**, do not include cross-dispersion terms in transport equations. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Notes 1

By default, the iterative solver is used, which is a restarted ORTHOMIN method that solves the Schur complement, reduced matrix, which is preconditioned by triangular factorization. See section D.7.2. Linear-Equation Solvers for Flow and Transport Finite-Difference Equations for details on the implementation of the iterative solver. The identifiers, **-tolerance**, **-save_directions**, and **-maximum_iterations** apply only to the iterative solver. *Tolerance* is the convergence criterion for the iterative solver; if the Euclidean norm of the relative residual vector is less than *tolerance*, the iterative solution to the linear equations has converged. It is not feasible to determine this tolerance parameter on the basis of the problem specifications and the desired accuracy of the solution. A value for *tolerance* can be determined empirically by examining the magnitude of the changes in heads and concentrations with successively smaller values of *tolerance*. Experience with test problems has shown that a tolerance of 10^{-6} to 10^{-12} is necessary to obtain 3 or 4 digit agreement with the direct solver.

The solver does not seem to be very sensitive to the number of search directions (**-save_directions**) that are retained between restarts of the ORTHOMIN solver. The default value of 10 seems to work satisfactorily for most problems. If solver convergence requires more than about 100 iterations, the number of retained search directions should be increased to 15 or greater. However, the number of search directions should not be set to a very large number because, as the number of search directions is increased, the amount of memory needed to save the vectors of the search directions increases.

The limit on the number of iterations (**-maximum_iterations**) prevents runaway conditions when the convergence rate becomes very slow. Iteration counts are written to the *prefix*.**log** file as controlled by **-progress_statistics** of the **PRINT_FREQUENCY** data block. For some simulations, the user may need to double or triple the default maximum of 500 iterations. However, more than a few hundred iterations indicates that adjustments probably need to be made in the spatial or temporal discretization or that there is an incompatibility in the problem formulation.

Default weighting is upstream-in-space (**-space_differencing** *0.0*) and backward-in-time (fully implicit, **-time_differencing** *1.0*). For steady-flow simulations, the time differencing scheme for the flow equation is always fully implicit. For transient-flow simulations with reactive transport, the same time weighting (**-time_differencing**) is used for both the flow and the transport equations. The flow equation does not have an advective term and thus does not use a spatial differencing factor. For additional information on the effects of weighting in time and space on oscillations and numerical dispersion, see sections D.1.2. Spatial Discretization, D.1.3. Temporal Discretization, and D.8. Accuracy from Spatial and Temporal Discretization.

If cross-dispersion terms (see section D.3. Property Functions and Transport Coefficients) are used (**-cross_dispersion** true), the solution to the transport equations may result in negative concentrations, which will be set to zero in the chemical calculations. Setting negative concentrations to zero will increase the mass of a solute in the system and increase global mass-balance errors.

Example 2

```
Line 0:  SOLUTION_METHOD
Line 1:        -direct_solver          true
Line 2:        -space_differencing     0.5
Line 3:        -time_differencing      0.5
Line 4:        -cross_dispersion     False
```

Explanation 2

Line 0: **SOLUTION_METHOD**

SOLUTION_METHOD is the keyword for the data block; no other data are included on this line.

Line 1: **-direct_solver** [***True*** or ***False***]

-direct_solver—This identifier is used to specify whether the direct solver will be used to solve the flow and transport equations. By default, the program will use the iterative solver. Optionally, **direct_solver**, **direct**, or **-d**[**irect_solver**].

True or ***False*****—True**, use the direct solver; **false**, use the iterative solver. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 2: **-space_differencing** *weight*

Same as example 1, line 5.

Line 3: **-time_differencing** *weight*

Same as example 1, line 6.

Line 4: **-cross_dispersion** [***True*** or ***False***]

Same as example 1, line 7.

Notes 2

By default, the iterative solver is used, but the direct solver is faster for small problems (as many as 1000 nodes). The direct solver does not have the iteration-termination (convergence-tolerance) errors of the iterative solver and, unlike the iterative solver, the direct solver has no additional parameters that need to be defined.

Example Problems

The **SOLUTION_METHOD** data block is used in the example problems 1, 2, 3, and 4.

SPECIFIED_HEAD_BC

This keyword data block is used to define specified-head boundary conditions. For flow-only simulations, only the heads are required for each specified-head boundary-condition node. For reactive-transport simulations, the index number of a fixed solution composition or an associated solution composition also is required for each specified-head boundary-condition node. The head and solution index number of the associated solution may vary independently over the course of the simulation. This keyword data block is optional and only is needed if specified-head boundary conditions are included in the simulation.

Example

```
Line 0:  SPECIFIED_HEAD_BC
Line 1:        -zone      0     10    10    10    10    10
Line 2:              -head
Line 3:                   0     100
Line 3a:                  10    101.5
Line 4:              -associated_solution
Line 5:                   0     6
Line 5a:                  5     7
Line 1a:       -zone      0     0     10    10    0     10
Line 2a:             -head
Line 3b:                  0     102
Line 6:              -fixed_solution
Line 7:                   0     6
Line 8:              -mask                  file        mask.file
```

Explanation

Line 0: **SPECIFIED_HEAD_BC**

SPECIFIED_HEAD_BC is the keyword for the data block; no other data are included on this line.

Line 1: **-zone** x_1, y_1, z_1, x_2, y_2, z_2

-zone—The zone selects a set of cells for which specified-head boundary conditions will be defined. Optionally, **zone** or **-z**[**one**].

x_1, y_1, z_1—X, Y, and Z coordinate for the left (X), front (Y), lower (Z) corner of the rectangular zone.

x_2, y_2, z_2—X, Y, and Z coordinate for the right (X), back (Y), upper (Z) corner of the rectangular zone. Units, L, are defined by **-horizontal_grid** (X and Y coordinates) and **-vertical_grid** (Z coordinate) identifiers in the **UNITS** data block.

Line 2: **-head**

-head—This identifier is used to specify the head at the specified-head boundary. The head is defined by the property definition on line 3. A time series of head properties may be defined by using multiple line 3s. The first *time* in the series must be zero. Optionally, **head** or **-he**[**ad**].

Line 3: *time* [*units*] *property*

time—Simulation time (T) at which the head property definition (*property*) will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

property—The head at the boundary can be entered with any of four methods for defining a spatially distributed property: (1) a single value for the zone, (2) **by_node** followed by a value for each node in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets “< >”, (3) **file** followed by a file name, a value for each node in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or *distance*$_2$) is used. Units, L, are defined by the **-head** identifier in the **UNITS** data block. Lines 3, 3a, and 3b illustrate method 1.

Line 4: **-associated_solution**

-associated_solution—This identifier is used to specify a time series of index numbers that define the solution composition for the flux of water into the active grid region through the specified-head boundary. The composition of water flowing into the active grid is defined by the property definition of line 5. A time series of solution properties may be defined by using multiple line 5s. The first *time* in the series must be zero. Optionally, **associated_solution**, **associated**, or **-a**[**ssociated_solution**].

Line 5:*time* [*units*] *property*

time—Simulation time (T) at which the solution-composition property definition (*property*) will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be "seconds", "minutes", "hours", "days", or "years" or an abbreviation of one of these units.

property—Index numbers are specified that correspond to solution compositions defined in the chemistry data file. Solutions or mixtures of solutions can be entered for the zone with any of the five methods for defining a spatially distributed property: (1) a single index, the composition represented by the index is applied to the entire zone, (2) **by_node** followed by an index for each node in the zone in natural order; an index is explicitly defined for each node in the zone (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; an index for each node in the zone is read from the file in natural order (list of values may extend over multiple lines), (4) **X**, **Y**, or **Z**, *index*$_1$, *distance*$_1$, *index*$_2$, *distance*$_2$, a composition is linearly interpolated from the end-member compositions defined by indices *index*$_1$ and *index*$_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance*$_1$ and *distance*$_2$ along the specified coordinate direction; for locations outside the range of *distance*$_1$ to *distance*$_2$, the composition at the nearest point (*distance*$_1$ or *distance*$_2$) is used, or (5) **mixture** *index*$_1$, *index*$_2$ followed by a list of mixing fractions (f_1) for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of mixing fractions must be enclosed in angle brackets "< >"; the composition associated with a node is a mixture of the two compositions—f_1 is the mixing fraction applied to composition *index*$_1$; $f_2 = 1 - f_1$ is the mixing fraction applied to composition *index*$_2$. Lines 5 and 5a illustrate method 1.

Line 6: **-fixed_solution**

-fixed_solution—This identifier is used to specify a time series of index numbers that define fixed solution compositions for the specified-head boundary. When this identifier is used, the concentrations at the specified-head boundary cells do not vary with the flux of water in or out of the boundary cells. The composition of water at the specified-head boundary cells is defined by the property definition of line 7. A time series of solution properties may be defined by using multiple line 7s. The first *time* in the series must be zero. Optionally, **fixed_solution**, **fixed**, or **-f**[**ixed_solution**].

Line 7: *time* [*units*] *property*

time—Simulation time (T) at which the solution-composition property definition (*property*) will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be "seconds", "minutes", "hours", "days", or "years" or an abbreviation of one of these units.

property—Index numbers are specified that correspond to solution compositions defined in the chemistry data file. Solutions or mixtures of solutions can be entered for the zone with any of the five methods for defining a spatially distributed property: (1) a single index, the composition represented by the index is applied to the entire zone, (2) **by_node** followed by an index for each node in the zone in natural order; an index is explicitly defined for each node in the zone (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name; an index for each node in the zone is read from the file in natural order (list of values may extend over multiple lines), (4) **X**, **Y**, or **Z**, *index*$_1$, *distance*$_1$, *index*$_2$, *distance*$_2$, a composition is linearly interpolated from the end-member compositions defined by indices *index*$_1$ and *index*$_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance*$_1$ and *distance*$_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the composition at the nearest point ($distance_1$ or $distance_2$) is used, or (5) **mixture** $index_1$, $index_2$ followed by a list of mixing fractions (f_1) for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of mixing fractions must be enclosed in angle brackets "< >"; the composition associated with a node is a mixture of the two compositions—f_1 is the mixing fraction applied to composition $index_1$; $f_2 = 1 - f_1$ is the mixing fraction applied to composition $index_2$. Line 7 illustrates method 1.

Line 8: **-mask** *property*

Use of this identifier is rare.

-mask—An array is used to include and exclude nodes from the zone definition. Optionally, **mask** or **-m**[**ask**].

property—An array of 1s (or any positive number) and 0s (or any nonpositive number) is used to include and exclude nodes from the zone definition. The array can be defined by any of four methods for defining a spatially distributed property; however, methods 1 and 4 are usually not useful: (1) a single number, 1 or 0, is used to include or exclude all of the nodes of the zone definition, (2) **by_node** followed by a 1 or 0 for each node (active or inactive) in the zone in natural order (list of values may extend over multiple lines); the list of values must be enclosed in angle brackets "< >", (3) **file** followed by a file name—a 1 or 0 for each node (active or inactive) in the zone is read from the file in natural order (list of values may extend over multiple lines), or (4) **X**, **Y**, or **Z**, $value_1$, $distance_1$, $value_2$, $distance_2$, a value is linearly interpolated from $value_1$ and $value_2$ for each node in the zone; interpolation is proportional to the fractional distance of the node location between $distance_1$ and $distance_2$ along the specified coordinate direction; for locations outside the range of $distance_1$ to $distance_2$, the value at the nearest point ($distance_1$ or $distance_2$) is used; if the interpolated value is less than or equal to 0 for a node, the node is excluded from the zone definition; if the interpolated value is greater than 0, the node is included in the zone definition. Line 8 illustrates method 3.

Notes

A choice between two identifiers, **-fixed_solution** and **-associated_solution**, is used to define boundary conditions for the concentrations of chemical components at specified-head boundaries. With the **-fixed_solution** identifier, the concentrations of components in the boundary cell are equal to the concentrations in the solution with the specified index number or the mixture of solutions as defined with method 4 of the property definition, regardless of reactions or flow into or out of the cell. With the **-associated_solution** identifier, the concentrations of the components in the boundary cell may vary as determined by reaction and transport. If flow is out of the active grid region through the specified-head boundary cell during a time step, the concentrations in the water that exits the active grid region are equal to the concentrations in the boundary cell. If flow enters the active grid region through the boundary of the active grid during a time step, the concentrations in the water that enters the cell are equal to the concentrations in the associated solution with the specified index number or the mixture of solutions as defined with method 4 of the property definition.

Multiple zones may be used within **SPECIFIED_HEAD_BC** data blocks to define boundary conditions within the grid region. Different boundary conditions for a single cell (node) may be defined multiple times as part of different zone definitions in different keyword data blocks. The boundary condition that is used for a cell is the last zone definition that defines a boundary condition for that cell. No other boundary conditions can be applied at a specified-head cell, including leaky, flux, and river boundary conditions. See section D.5.7. Boundary-Condition Compatibility for more details.

Example Problems

The **SPECIFIED_HEAD_BC** data block is used in the example problems 1, 2, 3, and 4.

STEADY_FLOW

This keyword data block is used to specify whether flow conditions are steady state and to define criteria for the determination that a steady-state flow condition is attained. If the simulation is defined to have steady-state flow, then an initial flow-only calculation is performed to determine the steady-state velocities, and these velocities are used in the transport equations throughout the simulation. Steady-flow is calculated by time stepping with the finite-difference flow equations until (1) heads are constant and (2) flow-in equals flow-out within specified tolerances. The data in this data block apply only to the initial calculation to obtain steady-state flow.

Example

```
Line 0:   STEADY_FLOW true
Line 1:         -head_tolerance              1e-5
Line 2:         -flow_balance_tolerance      0.001
Line 3:         -minimum_time_step           1     s
Line 4:         -maximum_time_step           1000  day
Line 5:         -head_change_target          100.
Line 6:         -iterations                  500
```

Explanation

Line 0: **STEADY_FLOW [*True*** or ***False***]

STEADY_FLOW—This keyword is used to specify whether flow conditions for the simulation are steady state. If the simulation is a reactive-transport simulation (**SOLUTE_TRANSPORT true**), then the steady-state velocities will be used for all transport calculations. By default, the simulation is assumed to be transient flow.

True or ***False*—True**, an initial calculation will be performed to calculate steady-state velocities; **false**, transient velocities will be calculated at each time step of the simulation. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Line 1: **-head_tolerance** *tolerance*

-head_tolerance—This identifier is used to specify a head tolerance for determining when flow is at steady state. Head changes for a time step must be less than this tolerance for the flow system to be at steady state. Default is 10^{-5} (units are specified with **-head** identifier in the **UNITS** data block). Optionally, **head_tolerance**, **head_tol**, or **-h**[**ead_tolerance**].

tolerance—Tolerance for head changes. Units, L, are defined by **-head** identifier in the **UNITS** data block.

Line 2: **-flow_balance_tolerance** *tolerance*

-flow_balance_tolerance—This identifier is used to specify a relative flow-balance tolerance for determining when flow is at steady state. The fractional flow balance must be less than this tolerance for the flow system to be at steady state. The fractional flow balance is the difference between the inflow and outflow rates divided by the average of the inflow and outflow rates for the current time step. Default, 0.001 (-). Optionally, **flow_tol**, **flow_tolerance**, **flow_balance_tol**, **flow_balance_tolerance**, **f**[**low_tolerance**], or **-f**[**low_balance_tolerance**].

tolerance—Fractional flow-balance tolerance. Units are dimensionless.

Line 3: **-minimum_time_step** *time_step*, [*units*]

-minimum_time_step—This identifier is used to specify the initial time step used in calculating steady-state flow and also is the minimum time step allowed for automatic time stepping in the steady-state flow calculation. Default is the value for **-time_step** defined in **TIME_CONTROL** divided by 1,000. Optionally, **minimum**, **minimum_time**, **minimum_time_step**, or **mi**[**nimum_time_step**].

time_step—Initial and minimum time step for steady-flow calculation. Units, T, are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for time step can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

Line 4: **-maximum_time_step** *time_step*, [*units*]

-maximum_time_step—This identifier is used to specify the maximum time step used in the steady-state flow calculation. Default is the value for **-time_step** defined in **TIME_CONTROL** times 1,000. Optionally, **maximum**, **maximum_time**, **maximum_time_step**, or **ma**[**ximum_time_step**].

time_step—Maximum time step for steady-state flow calculation. Units, T, are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for time step can be "seconds", "minutes", "hours", "days", or "years" or an abbreviation of one of these units.

Line 5: **-head_change_target** *target*

-head_change_target—This identifier is used to specify the target head change for a single time step in the steady-flow calculation. The time step will be decreased or increased during the steady-flow calculation to try to achieve head changes of *target* over each time step. Default is 0.3 times the thickness of the grid region. Optionally, **head_change**, **head_change_target**, **head_target**, **-head_c**[**hange_target**], or **-head_ta**[**rget**].

target—Target head change. Units, L, are defined by **-head** identifier in the **UNITS** data block.

Line 6: **-iterations** *maximum_iterations*

-iterations—This identifier is used to specify the maximum number of iterations (time steps) that will be used in attempting to attain a steady-flow conditions. Default is 100 iterations. Optionally, **iterations** or **-i**[**terations**].

maximum_iterations—Maximum number of iterations (time steps) used in attempting to attain a steady-flow condition.

Notes

STEADY_FLOW can be specified with or without transport calculations (**SOLUTE_TRANSPORT)**. If **STEADY_FLOW** is true, an additional flow-only calculation is performed at the beginning of the PHAST simulation to determine the steady-state head field and associated steady-state velocity field. The calculation ensures that heads are constant within the tolerance specified by **-head_tolerance** and global flow balance is satisfied within a relative tolerance specified by **-flow_balance_tolerance**. See section D.1.4. Automatic Time-Step Algorithm for Steady-State Flow Simulation for details. The steady-state velocity field is used in all transport calculations for the simulation. The **STEADY_FLOW** true option saves CPU time by eliminating the solution of the ground-water flow equation and recalculation of the same velocity field at each time step.

The difference equations for unconfined flow (**FREE_SURFACE_BC true**) contain explicit terms, which are calculated on the basis of conditions from the previous time step: for example, fraction of cell saturated, fraction of well screen receiving water, and the vertical location of the cell receiving river flux. These explicit terms may cause instability (for example, oscillations) in the numerical method, such that the method does not converge to a steady-state flow condition. The only remedy available is to use smaller time steps so that the explicit terms change slowly. To ensure a sufficiently small time step, it may be necessary to start with a very small time step (**-minimum_time_step**) and limit the maximum time step (**-maximum_time_step**). If time steps are small, it also may be necessary to increase the number of iterations (**-iterations**) to allow the method to slowly approach the steady-state solution.

The automatic time-step algorithm will fail if the maximum change in head after the second steady-state time step exceeds the target change in head (**-head_change_target** *target*). For the first two steady-state time steps the minimum time step (**-minimum_time_step** *time_step*) is used. Therefore, the simulator fails because it cannot reduce the time step to less than the minimum, but the head changes are still too large. The remedy to this problem is to increase the maximum acceptable change in head (**-head_change_target** *target*), reduce the minimum allowable time step (**-minimum_time_step** *time_step*), or adjust the initial head conditions to be closer to the head boundary conditions.

Example Problems

The **STEADY_FLOW** data block is used in the example problems 1 and 4.

TIME_CONTROL

This keyword data block is used to define the time steps for the simulation and times at which simulation periods end. The time step that applies at time zero must be defined, but a new time step may be defined to apply beginning at any time in the simulation. Times at which to end simulation periods may be explicitly defined in this data block; however, the number of simulation periods also depends on the time series definitions in **FLUX_BC**, **LEAKY_BC**, **PRINT_FREQUENCY**, **RIVER**, **SPECIFIED_HEAD_BC**, and **WELL** data blocks. The last time defines the end of the simulation. The time specified for the end of each simulation period is measured from the beginning of the simulation. This keyword data block is mandatory for all simulations.

Example

```
Line 0:   TIME_CONTROL
Line 1:         -time_step
Line 2:               0           1     day
Line 2a:              10    day   5     day
Line 3:         -time_end
Line 4:               1     yr
Line 4a:              2     yr
```

Explanation

Line 0: **TIME_CONTROL**

TIME_CONTROL is the keyword for the data block; no other data are included on this line.

Line 1: **-time_step**

-time_step—This identifier is used to specify that time-step data will be entered. Optionally, **step**, **time_step**, **delta**, **delta_time**, **-s**[**tep**], **-time_s**[**tep**], or **-d**[**elta_time**].

Line 2: *time*, [*units*], *time_step*, [*units*]

time—Simulation time (T) at which the new time step will be used. The first *time* in a series of line 2s must be zero. Line 2 may be appended to line 1 or other line 2s. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

time_step—Time-step length (T), units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time_step* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

Line 3: **-time_end**

-time_end—This identifier is used to specify a series of times at which simulation periods end. The last time in the series marks the end of the simulation. All times are elapsed time from the beginning of the simulation. The number of simulation periods also depends on the time series definitions in **FLUX_BC**, **LEAKY_BC**, **PRINT_FREQUENCY**, **RIVER**, **SPECIFIED_HEAD_BC**, and **WELL** data blocks. Optionally, **time_change**, **change_time**, **end**, **end_time**, **time_end**, **-time_c**[**hange**], **-c**[**hange_time**],**-e**[**nd_time**], or **-time_e**[**nd**]**.**

Line 4: *time_end*, [*units*]

time_end—Time at which a simulation period ends. The last line 4 indicates the time of the end of the simulation. *Time* (T) units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

Notes

A **TIME_CONTROL** data block, including both identifiers, is mandatory. If a time unit is specified for a time variable (*time*, *time_step*, or *time_end*), then the specified unit overrides the default unit (**-time**, **UNITS** keyword data block). The units specified for each variable need not be the same. The first *time* in the **-time_step** series must be zero. Every *time_end* marks the end of a simulation period; the final *time_end* marks the end of the simulation. Every time less than the final *time_end* that is included in a time series for *time_step* or in any time series in **FLUX_BC**, **LEAKY_BC**, **PRINT_FREQUENCY**, **RIVER**, **SPECIFIED_HEAD_BC**, and **WELL** data blocks will mark the end of one simulation period and the beginning of another.

The time step, in addition to grid discretization, is critical for obtaining a sufficiently accurate numerical solution to the flow and transport equations. Criteria for estimating appropriate time steps for different finite-difference weightings are discussed in section D.1.3. Temporal Discretization.

The difference equations for unconfined flow (**FREE_SURFACE_BC true**) contain explicit terms, which are calculated on the basis of conditions from the previous time step: for example, fraction of cell saturated, fraction of well screen receiving water, and the vertical location of the cell receiving river flux. These explicit terms may cause instability (for example, oscillations) in the numerical method. The only remedy available is to use smaller time steps (*time_step*) so that the explicit terms change slowly.

Example Problems

The **TIME_CONTROL** data block is used in the example problems 1, 2, 3, and 4.

TITLE

This keyword data block is used to define two title lines that are written to each of the output files, except the *prefix*.**xyz** and *prefix*.**h5** files.

Example

```
Line 0:   TITLE
Line 1:   line 1
Line 1a:  line 2
```

Explanation

Line 0: **TITLE**

TITLE is the keyword for the data block.

Line 1: *title*

title—The title may continue on as many lines as desired; however, only the first two lines are written to output files. Lines are read and saved as part of the title until a keyword begins a line or until the end of the file.

Notes

The **TITLE** data block is used to identify the simulation in the output files, except the *prefix*.**xyz** and *prefix*.**h5** files. Be careful not to begin a line of the title with a keyword because that marks the end of the **TITLE** data block. If more than one title keyword data block is included, only the lines from the last **TITLE** data block will appear in the output files.

Example Problems

The **TITLE** data block is used in the example problems 1, 2, 3, and 4.

UNITS

This keyword data block is used to specify the units of measure for the input data. Internal to PHAST, all flow and transport calculations use SI units. The **UNITS** data block provides the information necessary to convert the input data to SI units. Output units are always SI except for time, which is output in the unit defined by the **-time** identifier of this keyword data block. The **UNITS** data block contains only static data.

Example

```
Line 0:   UNITS
Line 1:         -time                          years
Line 2:         -horizontal_grid               km
Line 3:         -vertical_grid                 ft
Line 4:         -head                          ft
Line 5:         -hydraulic_conductivity        m/d
Line 6:         -specific_storage              1/ft
Line 7:         -dispersivity                  m
Line 8:         -flux                          m/s
Line 9:         -leaky_hydraulic_conductivity        m/s
Line 10:        -leaky_thickness               km
Line 11:        -well_diameter                 in
Line 12:        -well_flow_rate                gpm
Line 13:        -river_bed_hydraulic_conductivity          m/s
Line 14:        -river_bed_thickness           m
```

Explanation

Line 0: **UNITS**

UNITS is the keyword for the data block; no other data are included on this line.

Line 1: **-time** *time_units*

-time—This identifier is used to specify the default unit of time for input and output data.

time_units—Units for time (T). Time units can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Optionally, **time** or **-t**[**ime**].

Line 2: **-horizontal_grid** *units*

-horizontal_grid—This identifier is used to specify the units of distance for input data related to the X and Y coordinate directions, which include distances in the **GRID** data block and all **-zone** definitions. Optionally, **horizontal_grid**, **horizontal**, or **-ho**[**rizontal_grid**].

units—Input units of measure for horizontal distance in the X and Y coordinate directions (L). Distance units can be U.S. customary (“inches”, “feet”, or “miles”), metric (“millimeters”, “centimeters”, “meters”, or “kilometers”), or an abbreviation of one of these units.

Line 3: **-vertical_grid** *units*

-vertical_grid—This identifier is used to specify the units of distance for input data related to the Z coordinate direction, which include vertical distances in the **GRID** data block and all **-zone** definitions. Optionally, **vertical_grid**, **vertical**, or **-v**[**ertical_grid**].

units—Input units for vertical distance (L). Units of distance can be U.S. customary (“inches”, “feet”, or “miles”), metric (“millimeters”, “centimeters”, “meters”, or “kilometers”), or an abbreviation of one of these units.

Line 4: **-head** *units*

-head—This identifier is used to specify the units of head for input data in all keyword data blocks. Optionally, **head** or **-he**[**ad**].

units—Input units for head (L). Head units can be U.S. customary (“inches”, “feet”, or “miles”), metric (“millimeters”, “centimeters”, “meters”, or “kilometers”), or an abbreviation of one of these units.

Line 5: **-hydraulic_conductivity** *units*

-hydraulic_conductivity—This identifier is used to specify the units of hydraulic conductivity for input data in the **MEDIA** data block. Optionally, **hydraulic_conductivity**, **K**, **-hy**[**draulic_conductivity**], or **-K**.

units—Input units for hydraulic conductivity (L/T). Length (L) can be U.S. customary (“inches”, “feet”, or “miles”), metric (“millimeters”, “centimeters”, “meters”, or “kilometers”), or an abbreviation of one of these units. Time (T) can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Units must include the divide symbol, “/”, and must contain no spaces.

Line 6: **-specific_storage** *units*

-specific_storage—This identifier is used to specify the units of specific storage for input data in the **MEDIA** data block. Optionally, **specific_storage**, **storage**, **-s**[**pecific_storage**], or **-s**[**torage**].

units—Input units for specific storage (1/L). Length (L) can be U.S. customary (“inches”, “feet”, or “miles”), metric (“millimeters”, “centimeters”, “meters”, or “kilometers”), or an abbreviation of one of these units. Units must include the “1/”, and must contain no spaces.

Line 7: **-dispersivity** *units*

-dispersivity—This identifier is used to specify the units of dispersivity for input data in the **MEDIA** data block. Optionally, **dispersivity**, **alpha**, **-d**[**ispersivity**], or **-a**[**lpha**].

units—Input units for dispersivity (L). Dispersivity units can be U.S. customary (“inches”, “feet”, or “miles”), metric (“millimeters”, “centimeters”, “meters”, or “kilometers”), or an abbreviation of one of these units.

Line 8: **-flux** *units*

-flux—This identifier is used to specify the units of fluid flux for input data in **FLUX_BC** data blocks. Optionally, **flux** or **-f**[**lux**].

units—Input units for fluid flux (L/T). Length (L) can be U.S. customary (“inches”, “feet”, or “miles”), metric (“millimeters”, “centimeters”, “meters”, or “kilometers”), or an abbreviation of one of these units. Time (T) can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Units must include the divide symbol, “/”, and must contain no spaces.

Line 9: **-leaky_hydraulic_conductivity** *units*

-leaky_hydraulic_conductivity—This identifier is used to specify the units of hydraulic conductivity for input data in **LEAKY_BC** data blocks. Optionally, **leaky_hydraulic_conductivity**, **leaky_K**, **-leaky_h**[**ydraulic_conductivity**], or **-leaky_K**.

units—Input units for hydraulic conductivity (L/T) for **LEAKY_BC** data blocks. Length (L) can be U.S. customary (“inches”, “feet”, or “miles”), metric (“millimeters”, “centimeters”, “meters”, or “kilometers”), or an abbreviation of one of these units. Time (T) can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Units must include the divide symbol, “/”, and must contain no spaces.

Line 10: **-leaky_thickness** *units*

-leaky_thickness—This identifier is used to specify the units of thickness for the leaky boundary for input data in **LEAKY_BC** data blocks. Optionally, **leaky_thickness**, **thickness**, **-leaky_t**[**hickness**], or **-t**[**hickness**].

units—Input units for thickness of the leaky boundary (L). Thickness units can be U.S. customary (“inches”, “feet”, or “miles”), metric (“millimeters”, “centimeters”, “meters”, or “kilometers”), or an abbreviation of one of these units.

Line 11: **-well_diameter** *units*

-well_diameter—This identifier is used to specify the units of the diameter or radius of wells for input data in **WELL** data blocks. Optionally, **well_diameter** or **-well_d**[**iameter**].

units—Input units for well diameter or radius, L. The diameter or radius units can be U.S. customary (for example “inches” or “feet”), metric (for example “millimeters”, “centimeters”, or “meters”), or an abbreviation of one of these units.

Line 12: **-well_flow_rate** *units*

-well_flow_rate—This identifier is used to specify the units of flow rates for input data in **WELL** data blocks. Optionally, **well_flow_rate**, **well_pumpage**, **-well_f**[**low_rate**], or **-well_p**[**umpage**].

units—Input units for flow rate (L^3/T). Flow-rate units can be U.S. customary (for example, “gal/min”, “gpm”, “ft^3/sec”, or “ft3/sec”), metric (for example, “Liter/minute”, “meters^3/day”, or “meters3/day”), or an abbreviation of one of these units.

Line 13: **-river_bed_hydraulic_conductivity** *units*

-river_bed_hydraulic_conductivity—This identifier is used to specify the units of hydraulic conductivity for input data in **RIVER** data blocks. Optionally, **river_bed_hydraulic_conductivity**, **river_bed_k**, **river_k**, **-river_bed_h**[**ydraulic_conductivity**], **-river_bed_k**, or **-river_k**.

units—Input units for hydraulic conductivity (L/T) of riverbeds. Length (L) can be U.S. customary (“inches”, “feet”, or “miles”), metric (“millimeters”, “centimeters”, “meters”, or “kilometers”), or an abbreviation of one of these units. Time (T) can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units. Units must include the divide symbol, “/”, and must contain no spaces.

Line 14: **-river_bed_thickness** *units*

-river_bed_thickness—This identifier is used to specify the units of thickness of the riverbed for input data in **RIVER** data blocks. Optionally, **river_bed_thickness**, **river_thickness**, **-river_bed_t**[**hickness**], or **-river_t**[**hickness**].

units—Input units for thickness of the riverbed (L). Units of thickness for the riverbed can be U.S. customary (for example, “inches”, “feet”, or “miles”), metric (for example, “millimeters”, “centimeters”, “meters”, or “kilometers”), or an abbreviation of one of these units.

Notes

The **UNITS** data block must be defined for all flow and transport simulations. The definitions in this data block specify the units for input data for all keyword data blocks in the flow and transport data file. Note that all definitions apply to the input data. All output data are in SI units, except time. The unit of time defined with the **-time** identifier is used for both input and output data.

Example Problems

The **UNITS** data block is used in the example problems 1, 2, 3, and 4.

WELL

This keyword data block is used to define an injection or pumping well. The location, diameter, open intervals, pumping or injection rate, and solution composition must be defined for the well. The pumping or injection rate and the chemical composition of injected water may vary over the course of a simulation. This data block is optional and only is needed if a well boundary condition is included in the simulation. Multiple **WELL** data blocks are used to define all of the wells in the grid region.

Example

```
Line 0:   WELL 122 Metropolis Injection Well 122
Line 1:         1766.       2356.
Line 2:         -injection_rate
Line 3:               0           0.0
Line 3a:              1     yr    4.5
Line 4:         -solution
Line 5:               0           16
Line 5a:              0.5   yr    17
Line 6:         -diameter               12
Line 7:         -elevation        101.  107.
Line 7a:        -elevation        143.  153.
Line 7b:        -elevation        175.  183.
Line 0a:  WELL 165 Metropolis Supply Well 165
Line 1a:        1833.       2320.
Line 8:         -pumping_rate
Line 9:               0           4.5
Line 9a:              2           3.5
Line 10:        -land_surface_datum     292.
Line 11:        -radius                 12
Line 12:        -depth            42.   47.
Line 12a:       -depth            99.   103.
Line 13:        -allocate_by_head_and_mobility            true
```

Explanation

Line 0: **WELL** *number*, [*description*]

WELL is the keyword for the data block.

number—positive number that identifies this well. Default is 1.

description—optional character field that identifies the well.

Line 1: *X*, *Y*

X—X location of the well. Units, L, are defined by the **-horizontal_grid** identifier in the **UNITS** data block.

Y—Y location of the well. Units, L, are defined by the **-horizontal_grid** identifier in the **UNITS** data block.

Line 2: **-injection_rate**

-injection_rate—This identifier is used to specify the rate of fluid injection into the well. The injection rate is defined on line 3. A time series of injection rates may be defined by using multiple line 3s. The first *time* in the series must be zero. Optionally, **injection_rate**, **injection**, or **-i**[**njection_rate**].

Line 3: *time* [*units*] *injection_rate*

time—Simulation time (T) at which the injection rate will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

injection_rate—Rate of fluid injection into the well starting at *time*. Units, L^3/T, are defined by the **-well_flow_rate** identifier in the **UNITS** data block.

Line 4: **-solution**

-solution—This identifier is used to specify a solution index number that defines the composition of water injected into the well. The identifier is required for each injection well. Solution index numbers correspond to solution compositions defined in the chemistry data file. The composition of injected water is defined on line 5. A time series of solution indices may be defined by using multiple line 5s. The first *time* in the series must be zero. Optionally, **solution**, **associated_solution**, **-s**[**olution**], or **-a**[**ssociated_solution**].

Line 5: *time* [*units*] *solution*

time—Simulation time (T) at which the solution composition will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

solution—Solution index number for injected water composition starting at *time*.

Line 6: **-diameter** *diameter*

-diameter—This identifier is used to specify the diameter of the well. Optionally, **diameter**, or **-di**[**ameter**].

diameter—Diameter of the well. Units, L, are defined by the **-well_diameter** identifier in the **UNITS** data block.

Line 7: **-elevation** $elevation_1$, $elevation_2$

-elevation—This identifier is used to specify the top and bottom elevations of an open interval in the well. The top and bottom elevations may be entered in either order. Multiple **-elevation** identifiers are used to define all open intervals for the well. Optionally, **elevation**, **elevations**, or **-e**[**levations**].

$elevation_1$—Elevation of start of open interval in the well. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block.

$elevation_2$—Elevation of end of open interval in the well. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block.

Line 8: **-pumping_rate**

-pumping_rate—This identifier is used to specify the pumping rate for the well. The pumping rate is defined on line 9. A time series of pumping rates may be defined by using multiple line 9s. Optionally, **pumping_rate**, **pumping**, **pumpage**, **-pu**[**mping_rate**], or **-pu**[**mpage**].

Line 9: *time* [*units*] *pumping_rate*

time—Simulation time (T) at which the pumping rate will take effect. Units are defined explicitly with *units*; default units are defined by **-time** identifier in **UNITS** data block.

units—Units for *time* can be “seconds”, “minutes”, “hours”, “days”, or “years” or an abbreviation of one of these units.

pumping_rate—Pumping rate (L^3/T) starting at *time*. Units are defined by the **-well_flow_rate** identifier in the **UNITS** data block.

Line 10: **-land_surface_datum** *land_surface_datum*

-land_surface_datum—This identifier is used to specify the elevation of the well datum. This data item is required only if open intervals are defined with the **-depth** identifier. Optionally, **land_surface_datum**, **lsd**, **-l**[**and_surface_datum**], or **-l**[**sd**].

land_surface_datum—Elevation of land surface at the well. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block.

Line 11: **-radius** *radius*

-radius—This identifier is used to specify the radius of the well. Optionally, **radius**, or **-r**[**adius**].

radius—Radius of the well. Units, L, are defined by the **-well_diameter** identifier in the **UNITS** data block.

Line 12: **-depth** $depth_1$, $depth_2$

-depth—This identifier is used to specify the top and bottom of an open interval as depths below the well datum. The two depths may be entered in either order. The **-land_surface_datum** identifier must be used to specify the elevation of the well datum. Multiple **-depth** identifiers are used to define all open intervals for the well. Optionally, **depth**, **depths**, or **-de**[**pths**].

$depth_1$—Depth of top of open interval in the well. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block.

$depth_2$—Depth of bottom of open interval in the well. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block.

Line 13: **-allocate_by_head_and_mobility** [***True*** or ***False***]

-allocate_by_head_and_mobility—This identifier is used to specify the method by which water is distributed to the aquifer from the well or received from the aquifer to the well. The well-bore flow-rate can be allocated by the product of mobility and head difference or by mobility only (Kipp, 1987, p. 34 and p. 122). By default, allocation is by mobility only. Optionally, **allocation_by_head_and_mobility**, **allocate_by_head_and_mobility**, **head_and_mobility**, **-al**[**location_by_head_and_mobility**], **-al**[**location_by_head_and_mobility**], or **-h**[**ead_and_mobility**].

[***True*** or ***False***]—**True** allocates flow by mobility and head difference, **false** allocates flow by mobility only. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t**[**rue**] or **f**[**alse**].

Notes

Multiple wells are defined by using multiple **WELL** data blocks, where each well is uniquely identified by the integer following the **WELL** keyword. The well location, geometry, and flow rate must be defined for each well, including X–Y coordinate location, diameter or radius, injection or pumping rate, open intervals and solution composition. Only the flow rate (**-injection_rate** or **-pumping_rate**) and the solution composition (**-solution**) for a well may vary during a simulation. Open intervals for a well could include screened intervals, perforated intervals, and open-hole intervals. The elevation of open intervals in a well can be defined in two ways: (1) the elevations of the top and bottom of an interval can be defined explicitly with the **-elevation** identifier, or (2) the well datum (elevation of a measuring point) can be defined with the **-land_surface_datum** identifier and the depths to the top and bottom of each open interval can be defined with the **-depth** identifier. If the well is an injection well, the flow rate is defined with the **-injection_rate** identifier as a positive number and an associated solution composition (**-solution** identifier) must be specified for reactive-transport simulations. If the well is a pumping well, the flow rate is defined with the **-pumping_rate** identifier as a positive number. A well can be changed from an injection well to a pumping well (or the reverse) in the time series definition for injection or pumping rate by changing the sign of the rate.

Wells in corner cells or side cells are assumed to be located exactly at the corner or at the mid-side of the cell, respectively. The well flow rate into or out of the active grid region is specified by the user. The program does not adjust the well flow rate to account for symmetry at boundaries. Thus, if the simulation region is one-quarter or one-half of a symmetric region (see Example 2 for an example of the use of symmetry) then the user must reduce the well flow rate by a factor of one-quarter or one-half, respectively, to account for the actual flow into the active grid region.

By default, allocation of well-bore flow to the open intervals is by a formula that depends only on mobility. Alternatively, the **-allocate_by_head_and_mobility** identifier can be used to allocate well-bore flow by a formula that uses the product of mobility and head difference between the well and the aquifer (Kipp, 1987, p. 34 and p. 122). For numerical stability, wells should have at least a small flow rate.

Example Problems

The **WELL** data block is used in the example problem 4.

Chapter 5. Output Files

PHAST has many options to control writing data to output files. Many of the output files are in paragraph or tabular form to be displayed or printed. Some files are columnar data or binary, and are intended for post-processing by plotting, statistical, and visualization programs. All of the output files are written in ASCII text format, except for the HDF file, which is in compressed binary format. Programs are included in the PHAST distribution package to visualize and extract data from the HDF file.

5.1. Content of Output Files

Each output file is listed in this section with a description of the types of data that are written to that file. In addition, the keyword data blocks and identifiers that control the content and print frequency are listed. Files are listed alphabetically by suffix.

*prefix***.h5**—Grid and boundary-condition information, heads, nodal velocities, and selected chemical concentration data are written to this HDF file. Grid node locations and the identity of nodes with each boundary-condition type also are written to this file. Boundary-condition types include specified head, flux, leaky, river, and well. In addition, heads, nodal velocities, and chemical data can be written to this file at varying frequencies. Basic-language programs can be used to calculate other chemical quantities that are written to this file. This binary file is intended to be used with the data visualization program, Model Viewer (Appendix A) and the data extraction program PHASTHDF (Appendix B).

Writing of initial-condition heads and chemical data is controlled by the identifiers **-HDF_heads** and **-HDF_chemistry** in the **PRINT_INITIAL** data block. By default, initial-condition heads and chemical data are written to the file. If steady-state flow is simulated, the steady-state velocities at nodes (see section D.10. Nodal Velocity Calculation) can be written to the HDF file by using the **-HDF_steady_flow_velocities** identifier of the **PRINT_INITIAL** data block. By default, steady-state flow velocities are not written to the file. During the simulation, heads, nodal velocities, and chemical data are written at selected frequencies as defined by the **-HDF_heads**, **-HDF_velocities**, and **-HDF_chemistry** identifiers in the **PRINT_FREQUENCY** data block. By default, the heads, velocities, and chemistry data are written to the HDF file at the end of each simulation period. If steady-state flow is simulated, the heads and velocities will be written only once.

The chemical data to be written to the HDF file are defined in the **SELECTED_OUTPUT** data block of the chemistry data file and by Basic-language programs in the **USER_PUNCH** data block of the chemistry data file.

*prefix***.head.dat**—This ASCII file is intended to be used for initial head conditions in subsequent simulations. At the end of the simulation, heads for each node in the grid region are written to this file. These heads can be read for a subsequent simulation by using the **file** option to define values for the **-head** identifier in the **HEAD_IC** data block.

Writing to the file is controlled by the **-save_final_heads** identifiers of the **PRINT_FREQUENCY** data block. By default, final heads are not written to the file.

*prefix***.log**—This file is used to document the progress of the simulation. The *prefix***.trans.dat** and *prefix***.chem.dat** input data files are echoed to this file along with any error and warning messages resulting from processing these input files. Where possible, error messages immediately follow the echoed line from the input file that contains the error. Additional log messages indicate the progress of the simulation through simulation periods and time steps. Error and warning messages generated during the simulation calculations are written to this file. This ASCII file is intended to be viewed with a text editor.

The identifiers **-echo_input** in **PRINT_INITIAL** data block and in the **PRINT** data block of the chemistry data file can be used to disable echoing input data to the log file; by default, echoing is enabled. The **-progress_statistics** identifier in **PRINT_FREQUENCY** data block can be used to control the time-step frequency for detailed printing of progress statistics to the log file. By default, progress statistics are written at the end of each simulation period.

*prefix***.O.bal**—Fluid and chemical-component global mass balances for the time step, cumulative balances, and balances listed by boundary-condition type are written to this file. This ASCII file is intended to be viewed with a text editor.

The frequency of writing data to the file is specified with the **-flow_balance** identifier in the **PRINT_FREQUENCY** data block. By default, the flow-balance information is written at the end of each simulation period.

*prefix***.O.bcf**—Fluid and solute flow rates through each boundary-condition cell are listed in this file by boundary-condition type. Data in the file are formatted as planes of nodes, either X–Y or X–Z, as defined by **-print_orientation** in the **GRID** data block. This ASCII file is intended to be viewed with a text editor.

The frequency of writing data to the file is specified with the **-bc_flow_rates** identifier in the **PRINT_FREQUENCY** data block. By default, boundary-condition flow rates are not written to the file.

*prefix***.O.chem**—Solution concentrations, distribution of aqueous species, saturation indices, and compositions of exchangers, gas phases, kinetic reactants, phase assemblages, solid solutions, and surfaces are written to this file while processing the chemistry data file at the beginning of a PHAST simulation. Data also may be printed for cells during the subsequent reactive-transport part of the simulation. By default, these data are not written during the reactive-transport simulation because this file could exceed file-size limits for the computer operating system. However, for diagnostic purposes, for small problems, or for selected locations, it is possible to enable printing during the reactive-transport simulation. This ASCII file is intended to be viewed with a text editor.

Writing to the file can be enabled and the frequency of writing specified by using the **-force_chemistry_print** identifier in the **PRINT_FREQUENCY** data block. By default, data for all cells will be written to this file, once printing is enabled with the **-force_chemistry_print** identifier. The **PRINT_LOCATIONS** data block can be used to restrict the set of cells for which data will be written. In addition, the **PRINT** and **USER_PRINT** data blocks in the chemistry input file can be used to choose the types of data that are written to the file.

*prefix***.O.comps**—Initial-condition chemical information and transient total dissolved component concentrations are written to this file. Initial-condition information includes the indices and mixing fractions for solutions, equilibrium phases, exchangers, surfaces, gas phases, solid solutions, and kinetic reactions that define chemical conditions at the beginning of simulation and the initial component concentrations. Component concentrations may be written to the file during the reactive-transport simulation. Data in the file are formatted as planes of nodes, either X–Y or X–Z, as defined by **-print_orientation** in the **GRID** data block. This ASCII file is intended to be viewed with a text editor.

Initial-condition chemical information is written at the beginning of the simulation as specified by the **-components** identifier in the **PRINT_INITIAL** data block. By default, initial-condition information is not written to the file. The frequency of writing transient data during the reactive-transport simulation is specified with the **-components** identifier in the **PRINT_FREQUENCY** data block. By default, transient component concentrations are not written to the file.

*prefix***.O.head**—Initial and transient heads for each node are written to this file. If steady-state flow is simulated, heads are written only once during the simulation. Data in the file are formatted as planes of nodes, either X–Y or X–Z, as defined by **-print_orientation** in the **GRID** data block. This ASCII file is intended to be viewed with a text editor.

Initial heads are written at the beginning of the simulation as defined by the **-heads** identifier in **PRINT_INITIAL** data block. By default, initial heads are written to this file. The frequency of writing transient heads is specified by the **-heads** identifier in the **PRINT_FREQUENCY** data block. By default, heads are written to the file at the end of each simulation period.

*prefix***.O.kd**—Static fluid-conductance factors, fluid conductances (Kipp, 1987, p. 104), and transient solute dispersive conductances (Kipp, 1987, p. 110) are written to this file for positive X, Y, and Z cell faces for each cell. Data in the file are formatted as planes of nodes, either X–Y or X–Z, as defined by **-print_orientation** in the **GRID** data block. This ASCII file is intended to be viewed with a text editor.

Static fluid-conductance factors are written at the beginning of the simulation as specified by the **-conductances** identifier in **PRINT_INITIAL** data block. By default, conductance factors are not written to the file. The frequency of writing transient conductances is specified by **-conductances** identifier in the **PRINT_FREQUENCY** data block. By default, transient conductances are not written to the file.

*prefix***.O.probdef**—Static and initial-condition flow and transport information that relates to the problem definition for the simulation is written to this file. The *prefix***.O.probdef** file is used to verify that the input data properly define the simulation. Information written to the file includes array sizes, grid definition, permeability distribution, porosity distribution, dispersivity distribution, fluid properties, initial head distribution, indices for initial conditions in each cell (solutions, pure phases, exchangers, kinetic reactions, solid solutions, surfaces, and gas phases), initial component concentrations in each cell, specific storage distribution, static and transient boundary-condition information, cell volumes, and component concentrations for fixed and associated solutions for each boundary-condition cell. Transient boundary-condition information may be written to the file during the simulation. Data in the file that are spatially distributed are formatted as planes of nodes, either X–Y or X–Z, as defined by **-print_orientation** in the **GRID** data block. This ASCII file is intended to be viewed with a text editor.

The identifiers **-boundary_conditions**, **-fluid_properties**, and **-media_properties** in the **PRINT_INITIAL** data block can be used to select problem definition data to be printed in this file. By default, only the fluid properties are written to the file. Writing of transient boundary-condition information is specified by using the **-boundary_conditions** identifier in the **PRINT_FREQUENCY** data block. By default, transient boundary-condition information is not written to the file.

*prefix***.O.vel**—Interstitial velocity vector components across cell faces and velocity vector components at grid nodes (see section D.10. Nodal Velocity Calculation) are written to this file. If steady-state flow is simulated, velocity vector components are written only once during the simulation. Data in the file are formatted as planes of nodes, either X–Y or X–Z, as defined by **-print_orientation** in the **GRID** data block. This ASCII file is intended to be viewed with a text editor.

If steady-state flow is simulated (**STEADY_FLOW** data block), the **-steady_flow_velocities** identifier in **PRINT_INITIAL** data block can be used to control printing of velocity vector components to the file. By default, steady-state velocities are not written to the file. For transient-flow simulations, the frequency for writing transient velocity data is specified by the **-velocities** identifier in the **PRINT_FREQUENCY** data block. By default, transient velocities are not written to the file.

*prefix***.O.wel**—Static and transient well information is written to this file, including well location, well identification number, fluid and solute flow rates, cumulative fluid and solute flow amounts, solute concentrations, and injection and production rates per node for each well. This ASCII file is intended to be viewed with a text editor.

Static data are written at the beginning of the simulation as defined by the **-wells** identifier in **PRINT_INITIAL** data block. By default, static well data are written to the file. The frequency for writing transient well data to the file is specified by the **-wells** identifier in the **PRINT_FREQUENCY** data block. By default, transient well data are written to the file at the end of each simulation period.

*prefix***.xyz.chem**—Selected initial-condition and transient chemical data for solutions, pure-phase assemblages, surface assemblages, exchange assemblages, kinetic reactions, solid solutions, and gas phases are written to this file for all active nodes in natural order (see section 4.2. Spatial Data). Basic-language programs can be used to calculate other chemical quantities that are written to this file. Data in the file are formatted in tab-delimited columns with one row per node; columns of X, Y, and Z coordinates and time begin each row. This ASCII file is intended to be used for post-processing simulation results by user-supplied spreadsheet, statistical, plotting, and visualization programs.

Initial-condition data are written to the file at the beginning of the simulation as defined by the **-xyz_chemistry** identifier in **PRINT_INITIAL** data block. By default, no initial-condition chemical data are written to the file. The frequency for writing transient chemical data to the file is specified with the **-xyz_chemistry** identifier in the **PRINT_FREQUENCY** data block. By default, no transient chemical data are written to the file. The **PRINT_LOCATIONS** data block in the flow and transport data file can be used to specify the cells for which data will be printed. By default, chemical data will be printed for all cells in the active grid region.

The chemical data to be written to the *prefix***.xyz.chem** file are defined in the **SELECTED_OUTPUT** data block of the chemistry data file and by Basic-language programs in the **USER_PUNCH** data block of the chemistry data file.

*prefix***.xyz.comps**—Initial-condition and transient concentrations for each component (chemical element) are written to this file for all nodes in natural order (see section 4.2. Spatial Data). Data in the file are formatted in tab-delimited columns with one row per node; columns with X, Y, and Z coordinates and time begin each row. This ASCII file is intended to be used for post-processing simulation results by user-supplied spreadsheet, statistical, plotting, and visualization programs.

Initial-condition component data are written at the beginning of the simulation as defined by the **-xyz_components** identifier in **PRINT_INITIAL** data block. By default, initial component data are not written to the file. The frequency for writing transient chemical component data to the file is specified with the **-xyz_components** identifier in the **PRINT_FREQUENCY** data block. By default, transient chemical data are not written to the file.

*prefix***.xyz.head**—Initial-condition and transient potentiometric heads for all nodes are written to this file in natural order (see section 4.2. Spatial Data). If steady-state flow is simulated, heads are written at most once during the simulation. Data in the file are formatted in tab-delimited columns with one row per node; columns with X, Y, and Z coordinates and time begin each row. This ASCII file is intended to be used for post-processing simulation results by user-supplied spreadsheet, statistical, plotting, and visualization programs.

Initial-condition heads are written at the beginning of the simulation as defined by the **-xyz_heads** identifier in **PRINT_INITIAL** data block. By default, initial heads are not written to the file. The frequency for writing transient heads to the file is specified with the **-xyz_heads** identifier in the **PRINT_FREQUENCY** data block. By default, transient heads are not written to the file.

*prefix***.xyz.vel**—Steady-state or transient velocity-vector components interpolated to grid nodes (see section D.10. Nodal Velocity Calculation) are written to this file for all nodes in natural order (see section 4.2. Spatial Data). If steady-state flow is simulated, velocity vector components are written at most once during the simulation. Data in the file are formatted in tab-delimited columns with one row per node; columns with X, Y, and Z coordinates and time begin each row. This ASCII file is intended to be used for post-processing simulation results by user-supplied spreadsheet, statistical, plotting, and visualization programs.

If steady-state flow is simulated (**STEADY_FLOW** data block), the **-xyz_steady_flow_velocities** identifier in **PRINT_INITIAL** data block can be used to control printing of velocity vector components to the file. By default, steady-state velocities are not written to the file. For transient-flow simulations, the frequency for writing transient velocity data is specified by the **-xyz_velocities** identifier in the **PRINT_FREQUENCY** data block. By default, transient velocities are not written to the file.

*prefix***.xyz.wel**—Initial and transient concentration data for wells are written to this file, including component concentrations, pH, and alkalinity. Data in the file are formatted in tab-delimited columns with one row per well; columns with X and Y well coordinates, vertical datum, time, and well number begin each row. This ASCII file is intended to be used for post-processing simulation results by user-supplied spreadsheet, statistical, plotting, and visualization programs.

The **-xyz_wells** identifier in **PRINT_INITIAL** data block can be used to control printing of initial component concentrations for wells to the file. By default, initial component concentrations are not written to the file. The frequency for writing transient component concentrations to the file is specified with the **-xyz_wells** identifier in the **PRINT_FREQUENCY** data block. By default, transient component concentrations are not written to the file.

selected_output—Selected chemical data for solutions, pure-phase assemblages, surface assemblages, exchange assemblages, kinetic reactions, solid solutions, and gas phases are written to this file, but only during the processing of the chemistry data file from the initial call to PHREEQC at the beginning of the simulation. This ASCII file is formatted in tab-delimited columns of data. The file is rarely used because it does not contain any results from the reactive-transport part of the simulation.

The file name for this file is defined with the **-file** identifier in the **SELECTED_OUTPUT** data block of the chemistry data file. Data items to be written are selected in the **SELECTED_OUTPUT** data block of the chemistry data file and by Basic-language statements defined in the **USER_PUNCH** data block of the chemistry data file.

5.2. Selection of Data for Chemical Output Files

Because of the many chemical components and species in reactive-transport modeling, it is unwieldy to write all of the chemical data to the *prefix*.**xyz.chem** and *prefix*.**h5** files. It is necessary to select the data that are written to these files with the **SELECTED_OUTPUT** and **USER_PUNCH** data blocks in the chemistry data file (*prefix*.**chem.dat**). If more than one **SELECTED_OUTPUT** or **USER_PUNCH** data blocks are included in the chemistry data file, the last data block of each type determines the chemical data to be written during the reactive-transport simulation to both the *prefix*.**xyz.chem** and *prefix*.**h5** files.

The **SELECTED_OUTPUT** data block has identifiers that select printing of specific quantities, such as the total moles of an element in solution, the activity of an aqueous species, the saturation index of a mineral, and several other quantities. The quantities are selected by lists that are specified in the data block. The **USER_PUNCH** data block allows mathematical manipulation of chemical quantities and printing of data items through Basic-language statements. Through the use of these two data blocks, the huge amount of output data produced by a simulation can be limited to the most important values, and files can be limited to manageable sizes.

5.3. Output Files for Post-Processing

The file *prefix*.**h5** contains potentiometric head, velocity components, and concentrations of chemical species for the active grid region at user-specified time levels. These data can be extracted to produce files from which statistics, contour maps, iso-surface plots, or vector maps of velocity fields can be generated with user-supplied software. The interactive utility program, PHASTHDF, is used to extract subsets of data from the HDF file. It is possible to extract data for specified subregions of the active grid region and for specified simulation times. PHASTHDF generates ASCII files with the same format as *prefix*.**xyz** files. The capabilities and use of PHASTHDF are described in Appendix B.

A modified version of the Model Viewer visualization software (Hsieh and Winston, 2002) is included in the distribution of PHAST. The software is used to visualize three-dimensional concentration, head, and velocity fields from PHAST simulations that are stored in the *prefix*.**h5** file. The capabilities and use of Model Viewer are described in Appendix A.

An alternative way to obtain data in a form suitable for post-processing is to write data to files *prefix*.**xyz.head**, *prefix*.**xyz.chem**, *prefix*.**xyz.comps**, and *prefix*.**xyz.vel**, which have columns of data with one row for each active node. The columns of these files are tab delimited and the first columns are X, Y, and Z coordinate locations for the node, the time, and a 1 or 0 to indicate whether the cell for the node is wet or dry. This format for the data allows easy importation into spreadsheet, statistical, plotting, or visualization programs. To limit the file size of the *prefix*.**xyz.chem** file, a subset of nodes can be specified with the **PRINT_LOCATIONS** data block and only data from nodes in the subset will be written. No similar facility exists for the other *prefix*.**xyz** files.

Concentration data for chemical components, pH, and alkalinity at wells can be written at selected times in columnar format to the file *prefix*.**xyz.wel**. The first columns of the file are the X, Y coordinate location of the well, the well datum, the time, and the well number. The data that are output apply to the well datum, which is the elevation of the top of the uppermost open interval in the well (Kipp, 1987). For a pumping well, the water at the well datum is a mixture of water from all of the production zones of the well, weighted by the flow rates for the zones.

5.4. Diagnostic Output Files

Diagnostic messages, initial conditions, and problem definition information are written to files to be able to verify problem definition and to debug simulation failures. To identify what is happening when things go wrong, the PHAST simulator writes diagnostic output to the file *prefix*.**log**. All warnings and errors are written to this file, including those detected while processing the input files and those that occur during the flow, transport, and reaction calculations. The *prefix*.**log** file also contains simulation progress and information on the solution of the finite-difference equations for each time step. The equation-solution information includes iteration counts for the solver (if applicable) and maximum changes in the dependent variables written out at an interval requested by the user (**PRINT_FREQUENCY** data block, **-progress_statistics**).

The first set of data in the *prefix*.**O.head** and *prefix*.**O.comps** files contains the initial-condition distributions of the variables for potentiometric head and component concentrations, respectively, if requested by the user (**PRINT_INITIAL** data block, **-heads** and **-components**). These files can be used to verify that the initial conditions have been defined correctly. The *prefix*.**O.probdef** file contains information related to the problem definition, which is useful for verifying that the problem formulation is correct. This information includes locations of nodes, parameters for the fluid and porous medium, initial conditions, and boundary conditions.

The boundary-condition index array contains indices of boundary-condition types and excluded cells for the simulation grid. This array is written to the *prefix*.**O.probdef** file if requested (**PRINT_INITIAL** data block, **-boundary_conditions**). The array can help the user locate errors in specification of boundary-condition type and spatial distribution. The 9-digit index numbers in table 5.1 apply to individual boundary types, with one exception: the combination of a flux and river leakage on a Z-face. A given index number may have a combination of the values in the table for flux, leakage, and river leakage conditions. For example flux on the X-face and leakage on the Y-face would be indicated by an index number of 230000000. A cell face can have at most one boundary type associated with it (except that flux and river leakage can coexist on a Z-face). Flux and leakage boundaries can be applied to at most one face. River leakage always applies to the Z-face. An index number of -1 indicates that a cell is outside the active grid region.

Table 5.1.—Format of the boundary-condition index number for each type of boundary condition

[Specified flux, leakage, and river boundaries can be used in a single cell; the index number for the cell can be obtained by adding the appropriate indices from the table. A cell face can have at most one boundary type associated with it. Flux and leakage boundaries can be applied to at most one face. River leakage always applies to the Z-face]

Boundary condition index number	Meaning
100000100	Specified head, specified solution composition
100000000	Specified head, associated solution composition
200000000	Specified flux through the X-face
020000000	Specified flux through the Y-face
002000000	Specified flux through the Z-face
300000000	Leakage through the X-face
030000000	Leakage through the Y-face
003000000	Leakage through the Z-face
006000000	River leakage (Z-face)
008000000	Flux and river leakage (Z-face)
-1	Inactive cell

The *prefix*.**O.kd** file contains conductance factors for flow, flow conductances, and solute-dispersive conductances for transport, if requested (**PRINT_INITIAL** data block, **-conductances**; **PRINT_FREQUENCY** data block, **-conductances**). The solute-dispersive conductances are functions of the velocity field and, thus, may be transient. This file is used primarily for diagnostic purposes in the later stages of tracking down a problem.

Chapter 6. Examples

A large number of example problems have been simulated to (1) confirm the numerical accuracy of the PHAST program for simulating coupled flow, transport, and chemical reactions by comparing results with analytical solutions, hand calculations, or published numerical results and (2) test and demonstrate most of the capabilities of the program. Seven of these verification examples are listed with no further documentation; PHAST simulations successfully reproduced the results of these relatively simple problems. However, four examples are presented in detail in the following sections for verification and tutorial purposes. The first two examples are one- and three-dimensional simulations of reactive solutes for which analytical solutions exist. The third example is a problem from the literature that includes kinetic biological processes. The final example is a simplified field-scale problem that uses all of the available boundary conditions and all of the commonly used chemical processes (except kinetics)—ion exchange, surface complexation, and mineral equilibria. The flow and transport data files and the chemistry data files for all four examples are included in tables (and in the *examples* directory in the distribution of the program) and selected results are shown. These example files can be used as templates for other reactive-transport simulations.

6.1. Verification Examples

The following examples were simulated with PHAST to verify that the program performs calculations correctly. Results from PHAST were compared with either published tables of numerical results, numerical output from computer evaluations of published analytical solutions, or published graphical results. All the verification examples were simulated by PHAST with satisfactory agreement with the published results; no further discussion of these examples is included.

Ground-water flow examples:

- Unconfined flow in one dimension with precipitation recharge. Analytical solution is presented in Bear (1972, p. 366).
- Single-well injection into two-dimensional rectangular region, confined flow. Finite-difference numerical solution is presented in Wang and Anderson (1982, p. 79).
- Well pumping from one layer of a two-layer leaky aquifer. Analytical solution is presented in Cheng and Morohunfola (1993, p. 791).

Solute-transport examples:

- Solute transport in one-dimensional column with advection and dispersion. Analytical solution is presented in Lapidus and Amundson (1952, p. 984).
- Single-well injection of tracer into two-dimensional rectangular region. Method-of-characteristics numerical solution is presented in Konikow and others (1996, p. 49).

Reactive-transport examples:

- Solute transport with decay in one-dimensional column. Method-of-characteristics numerical solution is presented in Konikow and others (1996, p. 45).
- Aerobic biodegradation combined with nitrate-reducing metabolism in a one-dimensional column. Numerical solution is presented in Kindred and Celia (1989, section 4.4, p. 1155). This problem is included in the set of example data sets distributed with PHAST. This example differs from the example in section 6.4. Example 3: Aerobic Consumption of a Substrate with Biomass Growth because the growth of biomass is not considered.

6.2. Example 1: Pulse Source of Chemical Constituent that Undergoes Sorption and Decay

An analytical solution for the concentration of a constituent as a function of time and distance following a pulse injection into a one-dimensional column is given by de Marsily (1986, p. 268). The constituent simultaneously undergoes linear decay and linear sorption. This example compares PHAST simulation results with the analytical solution for an injection with the column characteristics and the chemical parameters given in table 6.1. A solution containing the constituent is injected into the column for 60 s (seconds); a solution without the constituent is then injected for an additional 60 s.

In this example, the prefix “ex1” is used for the input file names. The chemistry for the simulation is defined in the *ex1***.chem.dat** file and the flow and transport characteristics of the simulation are defined in the *ex1***.trans.dat** file.The third input file that is needed for a reactive-transport simulation is the thermodynamic database file, which is named *phast.dat* by default and is the same as a PHREEQC thermodynamic database file.

Table 6.1.—Chemical and physical parameters for example 1

[cm, centimeter; s, second; m, meter; mol, mole; kgw, kilogram water]

Parameter	Value
Column length, cm	12.0
Interstitial velocity, cm/s	.1
Porosity, unitless	.1
Longitudinal dispersivity, cm	.1
Sorption constant, K_d, (mol/kgw)/(mol/kgw)	1.0
Decay rate constant, 1/s	.01

The chemistry data file for this example is given in table 6.2. Typically, the data base file *phast.dat* is used for the definition of most reactions and thermodynamic data. However, because of the idealized nature of this example problem, it is convenient to define all of the chemical characteristics of the chemical system within the chemistry data file (except for water and its associated species, which are found in the thermodynamic database file). The first data block, **SOLUTION_MASTER_SPECIES**, defines a new "element", A, which will be used as the reactive constituent. The **SOLUTION_SPECIES** data block defines a single uncharged[1] aqueous species for the element.

Table 6.2.—Chemistry data file for example 1

```
TITLE
.       Example 1.--Pulse of solute undergoing sorption and decay
.       Analytical solution from Quantitative Hydrogeology, de Marsily
#       Kd = 1 mol sorbed/mol dissolved
#       K  = 0.01 1/s   Decay constant
SOLUTION_MASTER_SPECIES
        A       A       0       A       1
SOLUTION_SPECIES
        A = A
        log_k   0
SURFACE_MASTER_SPECIES
        Surf    Surf
SURFACE_SPECIES
        Surf = Surf
                log_k 0
        Surf + A = SurfA
                log_k   -10
SOLUTION 1
END
SOLUTION 2
        A        1.0 mmol/kgw
END
SURFACE 1
        -no_edl
```

[1] If the constituent were an ion, additional issues would be raised. **KINETICS** removes the element and water is split to form dissolved O_2 or H_2 and hydronium or hydroxide ions to maintain charge balance. Thus, unwanted pH and redox effects will be obtained if a major cation undergoes significant decay. It is advisable to remove an oppositely charged element or replace the ionic element with another like ionic element, so that charge balance is maintained.

```
        Surf    1e10
END
KINETICS 1
        A_decay
        -formula A 1.0
        -m 0
RATES
        A_decay
        -start
10 rate = -(TOT("A") + MOL("SurfA")) * .01
20 moles = rate * TIME
30 SAVE moles
        -end
END
SELECTED_OUTPUT
        -file ex1.dummy.sel
        -reset false
USER_PUNCH
        -headings A SurfA SurfA/A
10 PUNCH TOT("A")*1000
20 PUNCH MOL("SurfA")*1000
30 IF TOT("A") > 0 THEN PUNCH MOL("SurfA")/TOT("A") ELSE PUNCH -1
END
```

The **SURFACE_MASTER_SPECIES** data block defines a surface that will sorb constituent A. The **SURFACE_SPECIES** data block defines two surface complexes; the sites of the surface will be either an uncomplexed site, denoted by "Surf", or a site complexed with constituent A, denoted "SurfA". The empty **SOLUTION** *1* data block defines pure water by default. This pure water solution will be used to fill the column initially and also to flush the column for the last 60 s of the simulation. **SOLUTION** *2* defines a solution containing 1 mmol/kgw (millimole per kilogram water) of constituent A. This solution will be injected into the column for the first 60 s of the simulation. **SURFACE** *1* defines the name of a surface sorption site and the number of sorption sites available to react.

PHREEQC does not have an explicit mechanism for simulating linear sorption. However, it is possible to define the number of surface sites and equilibrium constants for surface species in such a way that linear sorption is simulated accurately. Consider the reaction and log form of the mass-action equation for the sorption of A on surface Surf:

$$Surf + A = SurfA \text{ with } Log(K) = Log(SurfA) - Log(A) - Log(Surf)\,. \tag{6.1}$$

The mass-action equation holds strictly for activities. (Note that for charged surface species, by default, an electrostatic term is included in the mass-action equation; **-no_edl** in the **SURFACE** *1* data block eliminates that term and also precludes the need to define the surface area parameters for the surface.) For an uncharged aqueous species (A), activity in dilute solutions is nearly equal to molality. For surface species, activity is equal to mole fraction. If the number of sites is large relative to the amount of A that sorbs, 10^{10} moles for example, the mole fraction of $Surf$ is very nearly 1 and the mole fraction of $SurfA$ is the number of moles of $SurfA$ divided by 10^{10}. For this example, K_d is 1, $\frac{\frac{moles(SurfA)}{kgwater}}{(A)} = 1$, where (A) is molality. Substituting and rewriting equation 6.1 gives

$$Log(K) = Log\left(\frac{moles(SurfA)}{kgwater}\right) + Log(kgwater) - Log(10^{10}) - Log(A) - 0\text{, and} \tag{6.2}$$

$$Log(K) + 10 - Log(kgwater) = Log\left(\frac{\frac{moles(SurfA)}{kgwater}}{(A)}\right) = Log(K_d) = 0\,. \tag{6.3}$$

The mass of water in each cell is fixed at 1 kg (kilogram), so the term $Log(kgwater)$ is zero. Therefore, $Log(K)$ must be -10 to obtain the required K_d. These definitions are implemented in **SURFACE_SPECIES** and **SURFACE** *1* data blocks.

The **KINETICS** *1* data block defines the name of a rate expression, “A_decay”, that removes constituent A from solution. The **RATES** data block defines the rate of the kinetic reaction “A_decay”. The rate expression defines a rate that is proportional to the total number of moles of A in the system (dissolved plus sorbed). Because the chemical calculation is performed for one kilogram of water, the rate has the same proportionality to concentration.

The **SELECTED_OUTPUT** and **USER_PUNCH** data blocks define selected data to be written to output files. The **SELECTED_OUTPUT** data block is used to control writing selected-output data to as many as three files: *ex1.dummy.sel* (as defined with **-file** in the **SELECTED_OUTPUT** data block in this example), *ex1***.xyz.chem**, and *ex1***.h5**. In the **SELECTED_OUTPUT** data block, **-reset** *false* eliminates all default writing to the selected-output files. In this example, all data to be written to selected-output files are defined in the **USER_PUNCH** data block. To understand which data go to each of the three selected output files, it is necessary to know the sequence of calculations in PHAST. PHAST runs the chemistry data file as a PHREEQC simulation before it begins to simulate flow and transport. During this initial PHREEQC simulation, any results specified by **SELECTED_OUTPUT** and **USER_PUNCH** data blocks will be written to the selected-output file as defined by the **-file** identifier (*ex1.dummy.sel*). However, in this example, these data blocks are located at the end of the file after an **END** keyword, and no chemical simulations are defined after this point. Therefore, the file *ex1.dummy.sel* will be empty. After the PHREEQC simulation, PHAST begins the flow and transport simulation. During the flow and transport simulation, PHAST writes all selected output to the file *ex1***.xyz.chem** and to the file *ex1***.h5**. The identifiers **-xyz_chemistry** and **-hdf_chemistry** in the **PRINT_FREQUENCY** data block of the flow and transport data file are used to specify the frequencies at which data are written to these two files.

The flow and transport data file (table 6.3) defines all system geometry, spatial discretization, characteristics of the porous media, boundary conditions, initial conditions, time steps, and duration of the simulation. The flow and transport data file is linked to the chemistry data file by the index numbers of solutions, surface, and kinetics that are used as initial conditions (**CHEMISTRY_IC** in the flow and transport data file) and the index numbers of solutions used as boundary conditions (**SPECIFIED_HEAD_BC** data block in the flow and transport data file).

Table 6.3.—Flow and transport data file for example 1

```
TITLE
.       Example 1.--Pulse of solute undergoing sorption and decay
.       Analytical solution from Quantitative Hydrogeology, de Marsily
#       1D region       12 cm
#       Velocity       0.1 cm/s
#       Porosity       0.1
#       Dispersivity     1 cm    (isotropic)
#       60 s pulse, 60 s chaser
SOLUTE_TRANSPORT                           true
STEADY_FLOW                                true
UNITS
        -time                               sec
        -horizontal                          cm
        -vertical                            cm
        -head                                cm
        -hydraulic_cond                    cm/s
        -specific_storage                  1/cm
        -dispersivity                        cm
GRID
        -uniform X  0 12                     61
        -uniform Y  0  1                      2
        -uniform Z  0  1                      2
        -chemistry_dimensions                 X
        -print_orientation                   XY
MEDIA
        -zone   0 0 0 12 1 1
                -Kx                        0.12
```

```
                -Ky                         0.12
                -Kz                         0.12
                -porosity                    0.1
                -storage                       0
                -longitudinal_dispersivity  0.1
                -horizontal_dispersivity    0.1
                -vertical_dispersivity      0.1
FREE_SURFACE_BC false
SPECIFIED_HEAD_BC
        -zone   0 0 0 0 1 1
                -head                          0         1
                -associated_solution           0         2
                                              60         1
        -zone   12 0 0 12 1 1
                -head                          0       0.0
                -associated                    0         1
HEAD_IC
        -zone   0 0 0 12 1 1
                -head   X 1.0 0.0 0.0 12.0
CHEMISTRY_IC
        -zone   0 0 0 12 1 1
                -solution                      1
                -surface                       1
                -kinetics                      1
SOLUTION_METHOD
        -direct
        -space                               0.5
        -time                                0.5
TIME_CONTROL
        -time_step                             0       0.4 sec
        -time_end                            120 sec
PRINT_FREQUENCY
        0
        -hdf_chemistry                        10 sec
        -xyz_chemistry                         0 sec
        60
        -velocity                             60 sec
        -xyz_chemistry                        60 sec
        -force_chemistry_print                60 sec
END
```

The first data block in the flow and transport file is **TITLE**, which simply defines two lines to be written to the output files to identify the simulation. The **SOLUTE_TRANSPORT** data block is used to specify that this is not only a flow simulation, but transport and reactions are to be simulated as well. The flow field is defined to be steady state for the duration of the simulation with the **STEADY_FLOW** data block. A steady-state velocity field will be calculated once from the boundary conditions. The velocity field will then be constant for all time steps of the transport calculation. The **UNITS** data block is required to set the units of measure for all of the input quantities. On output, all units are in SI, including seconds for time, as specified with the **-time** identifier.

The **GRID** data block defines the locations of nodes that discretize the simulation region. The grid is 12 cm (centimeter) (X) by 1 cm (Y), by 1 cm (Z), with two nodes in the Y and Z directions and 61 nodes equally spaced in the X direction. The grid region begins at the first node (at X=0 cm) and ends at the last node (at X=12 cm). The flow and transport calculations are always performed on a three-dimensional grid. However, for symmetric systems (both flow and chemistry must be symmetric), the chemistry can be run on a subset of nodes, either a line (one dimensional) or plane (two dimensional), and the results copied to the remaining symmetric parts of the grid region. The identifier **-chemistry_dimensions X** indicates that chemical reactions will be calculated for one line of nodes in the X direction, and then copied to the other three lines of nodes in the X direction. The **-print_orientation** identifier specifies that X–Y planes of data will be written to the files with names beginning with *ex1***.O**.

The **MEDIA** data block specifies the properties of the porous medium. In this example, a single zone, inclusive of the entire grid region, is used to define uniform porous-medium properties throughout the grid region. Multiple zones could be used to represent heterogeneity of porous media properties within the grid region. An interstitial velocity of 0.1 cm/s (centimeter per second) is desired for this example simulation. From Darcy's Law (eq. D.3), if the head gradient is defined to be 1/12 by the boundary conditions (**SPECIFIED_HEAD_BC** data block) and the porosity is 0.1, then a hydraulic conductivity of 0.12 cm/s will produce the desired interstitial velocity of 0.1 cm/s.

The example simulates flow in a confined aquifer (column), which is specified by inactivating the free-surface boundary condition (**FREE_SURFACE_BC** *false*). The flow is prescribed by specified heads at both ends of the column to achieve the desired head gradient. The **SPECIFIED_HEAD_BC** data block defines two zones. At the plane of nodes at X=0 cm, the head is specified to be 1 cm and beginning at time 0 s, any inflowing solution will have the composition defined by solution 2. At 60 s, the composition of water entering at X=0 cm changes to solution 1. At the plane of nodes at X=12 cm, the head is specified to be 0 cm and any inflowing solution will have the composition defined by solution 1 for the entire simulation. Recall that solution 2 is defined in the chemistry data file to contain constituent A, whereas solution 1 is defined to be pure water. For confined flow, specified-head values may be relative to any elevation datum, without regard to the Z datum of the coordinate system. (For unconfined flow, heads must be defined relative to the Z datum of the coordinate system.)

The initial head is defined to be linear in the X direction, ranging from 1 cm at X=0 cm to 0 cm at X=12 cm by the **HEAD_IC** data block. This initial head distribution is exactly the steady-state head distribution for this system, but any initial head distribution should result in the same steady-state head distribution. The **CHEMISTRY_IC** data block is used to define the initial composition of water throughout the grid region and the types of chemistry that are active in each cell. In this example, the grid region initially contains pure water (solution 1), and every cell has surface and kinetic chemical reactions. The index numbers for the identifiers **-solution**, **-surface**, and **-kinetics** refer to the index numbers of solutions, surfaces, and kinetic reactants defined in the chemistry data file.

The **SOLUTION_METHOD** data block specifies that the flow and transport equations are solved by the direct method, which is appropriate for the small number of nodes in this problem. The finite-difference approximations to the flow and transport differential equations use centered-in-time and centered-in-space weighting (**-time** and **-space** in **SOLUTION_METHOD** data block). This weighting results in second-order truncation errors, which contain no numerical dispersion, but may introduce oscillations in the numerical solutions for head and concentration. In this example, the Peclet number is 2 (eq. D.5), which indicates that there will be no oscillations due to the spatial discretization (see section D.1.2. Spatial Discretization). The **TIME_CONTROL** data block specifies the time step (**-time_step**) of 0.4 s will be used from 0 s until the end of the simulation. The time at the end of the simulation (**-time_end**) is specified to be 120 s. For this example, the oscillation criterion (eq. D.8) resulting from temporal discretization is $\Delta t = 0.4 < \frac{\Delta x^2}{v\alpha} = \frac{0.2^2}{0.1(0.1)} = 4$, which indicates that there should be no oscillations (see section D.1.3. Temporal Discretization).

The **PRINT_FREQUENCY** data block specifies the frequency of writing data to the output files. Starting at 0 s, the default print frequencies will be used for all data, except that chemistry data will be written to the HDF file after every 10 s of simulation (**-HDF_chemistry** *10 sec*), and no data will be written to the *ex1***.xyz.chem** file (**-xyz_chemistry** *0 sec*). At 60 s, print frequencies are specified to be 60 s for the *ex1***.O.vel** (**-velocity**), *ex1***.xyz.chem** (**-xyz_chemistry**), and *ex1***.O.chem** (**-force_chemistry_print**) data files (**PRINT_FREQUENCY** data block). The print frequency for the HDF file is not redefined at 60 s, so the print frequency from 0 s (10 s print frequency) remains in effect until the end of the simulation.The **END** statement marks the end of the definitions for the flow and transport data file.

The time-series definitions for the **SPECIFIED_HEAD_BC** and **PRINT_FREQUENCY** data blocks specify a change at time 60 s. Thus, the total number of simulation periods is two, from 0 s to 60 s, and from 60 s to the end of the simulation at 120 s (**-time_end** in the **TIME_CONTROL** data block).

The results of the simulation are written to several output files. The velocity field, which is written to the *ex1***.O.vel** file and also can be viewed in Model Viewer from data written in the *ex1***.h5** file, is uniform at 0.001 m/s (meters per second) or 0.1 cm/s (centimeters per second). The head field is listed in *ex1***.O.head** and also stored in the *ex1***.h5** file. Descriptions of the chemistry in each cell are written to *ex1***.O.chem**. Selected output (as defined in the chemistry data file), including the concentration of A and SurfA, is written to *ex1***.xyz.chem** and also to the *ex1***.h5** file. The concentration of A from the PHAST simulation as a function of distance at 120 s is plotted in figure 6.1 along with the analytical solution. The simulation used centered-in-time and centered-in-space weighting for the difference equations. Results of a simulation using upstream-in-space (**-space_differencing** 0) and backward-in-time (**-time_differencing** 1) also are plotted for comparison. The PHAST results for centered-in-time and centered-in-space weighting are similar to the analytical solution. The discrepancy with the analytical solution is caused predominantly by the operator-splitting error because the spatial and temporal discretization errors should be minimized by using

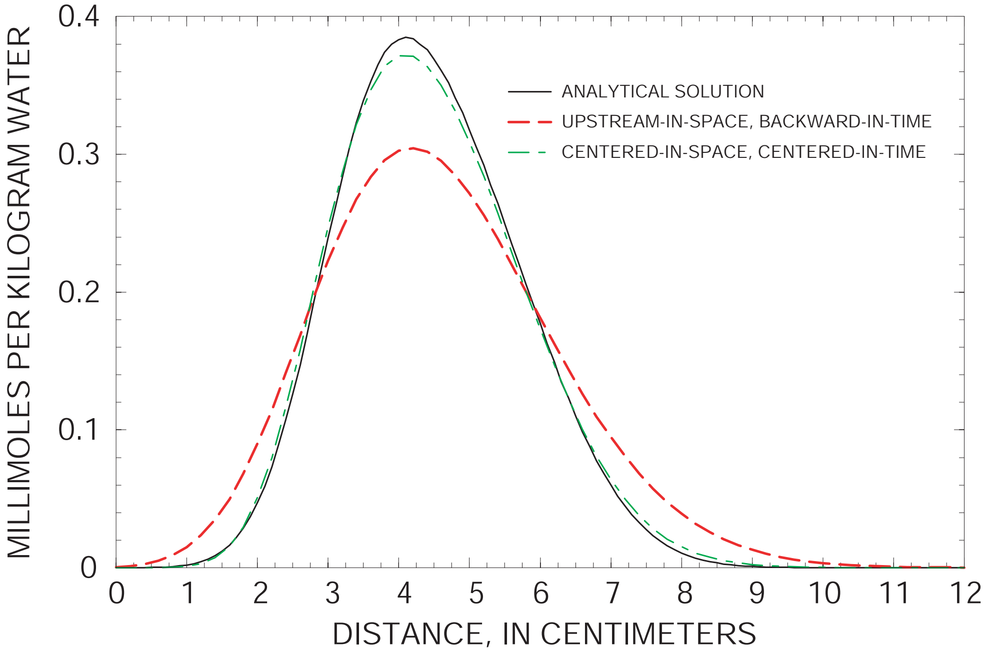


Figure 6.1.—Example 1: Analytical solution for concentration at 120 seconds of a pulse of solute that undergoes linear sorption and linear decay compared to numerical solutions using (1) centered-in-space and centered-in-time weighting and (2) upstream-in-space and backward-in-time weighting.

centered-in-time and centered-in-space weighting. The operator-splitting error for linear sorption and centered-in-time weighting can be estimated by equation D.31, $\alpha_{nos} = \frac{D_{nos}}{v} = \frac{(0.1)(0.4)}{(2)(2)}((2)(0.5)(2)-1) = 0.01$, which indicates that the numerical dispersivity caused by operator splitting is one tenth of the specified dispersivity [0.1 m (meters)]. For this weighting, the numerical dispersivity caused by operator splitting (and time discretization) can be reduced most effectively by decreasing the time step.

The PHAST results for backward-in-time and upstream-in-space weighting show additional numerical dispersion. Here, numerical dispersion is caused by errors due to spatial and temporal discretization in addition to the operator-splitting error. The total numerical dispersion is estimated from equations D.7, D.10, and D.31,

$\alpha_{ns} + \alpha_{nt} + \alpha_{nos} = \frac{0.2}{2} + \frac{(0.1)(0.4)}{2} + \frac{(0.1)(0.4)}{(2)(2)}((2)(1)(2)-1) = 0.1 + 0.02 + 0.03 = 0.15$. Thus, for backward-in-time and upstream-in-space weighing, the numerical dispersivity is larger than specified dispersivity (0.1 m), which accounts for the large deviation from the analytical solution. For this weighting, the numerical solution can be improved both by refining the spatial discretization and by decreasing the time step.

6.3. Example 2: Chain of Four Kinetically Decaying Reactants

Example 2 simulates a four-species decay chain as a function of time and space, with the first species in the decay series introduced over a rectangular patch at the upstream end of a flow system; the flow is one dimensional and uniform in the X direction, but the transport simulation is three dimensional because the chemical species disperse in all directions. A method for deriving an analytical solution for any number of species in a chain of first-order decay reactions is presented by Sun and others (1999). They present an analytical solution for this example (Sun and others, 1999, example 6.3, p. 436) that is based on their method for chain decay and the approximate analytical solution for a single species given by Domenico (1987). We derived an alternative analytical solution for comparison with PHAST simulation results that uses the method of Sun and others (1999), but implements the analytical solution for a single species given by Wexler (1992). The Wexler (1992) analytical series solution is

Table 6.4.—Chemical and physical parameters for example 2

[m, meter; d, day]

Parameter	Value
Simulation region length, m	100.0
Simulation region width, m	41
Simulation region height, m	25
Interstitial velocity, m/d	.2
Porosity, unitless	.1
Longitudinal dispersivity, m	1.5
Horizontal transverse dispersivity, m	.3
Vertical transverse dispersivity, m	.1
Decay rate constant A to B, 1/d	.05
Decay rate constant B to C, 1/d	.02
Decay rate constant C to D, 1/d	.01
Decay rate constant D, 1/d	.005
Source-patch location (Y direction) m	15 to 26
Source-patch location (Z direction), m	10 to 15
Duration of simulation, d	400

regarded as more accurate than the single term approximation of Domenico (1987). The parameters for this example are given in table 6.4. Simulation and analytical results are presented for a 400-day simulation.

The chemistry data file is presented in table 6.5. All of the chemical information for the four species of the chain decay series are defined in the chemistry data file. **SOLUTION_MASTER_SPECIES** and **SOLUTION_SPECIES** data blocks define the name and a single aqueous species for each of four elements: [A], [B], [C], and [D]. (Using brackets in the names avoids conflicts with elements carbon (C) and boron (B), which are defined in the thermodynamic database file.) **SOLUTION** *1* is defined as pure water and **SOLUTION** *2* contains 1 mmol/kgw of [A], which will be applied over the rectangular patch at the upstream end of the grid region.

Table 6.5.—Chemistry data file for example 2

```
SOLUTION_MASTER_SPECIES
        [A]    [A]    1       1       1
        [B]    [B]    1       1       1
        [C]    [C]    1       1       1
        [D]    [D]    1       1       1
SOLUTION_SPECIES
        [A] = [A]
                log_k   0
        [B] = [B]
                log_k   0
        [C] = [C]
                log_k   0
        [D] = [D]
                log_k   0
```

```
SOLUTION 1
END
SOLUTION 2
        units   mmol/kgw
        pe      12.0    O2(g)   -0.67
        [A]     1.0
END
RATES
        [A]_decay
        -start
10 rate = -TOT("[A]") * .05/(3600*24)
20 moles = rate * TIME
30 SAVE moles
        -end
        [B]_decay
        -start
10 rate = -TOT("[B]") * .02/(3600*24)
20 moles = rate * TIME
30 SAVE moles
        -end
        [C]_decay
        -start
10 rate = -TOT("[C]") * .01/(3600*24)
20 moles = rate * TIME
30 SAVE moles
        -end
        [D]_decay
        -start
10 rate = -TOT("[D]") * .005/(3600*24)
20 moles = rate * TIME
30 SAVE moles
        -end
END
KINETICS 1
        [A]_decay
                -formula [A] 1 [B] -1.0
                -m 0
        [B]_decay
                -formula [B] 1 [C] -1.0
                -m 0
        [C]_decay
                -formula [C] 1 [D] -1.0
                -m 0
        [D]_decay
                -formula [D] 1
                -m 0
END
SELECTED_OUTPUT
        -file ex2.dummy.sel
        -reset false
USER_PUNCH
        -heading A B C D
        -start
10 PUNCH TOT("[A]")*1000
20 PUNCH TOT("[B]")*1000
30 PUNCH TOT("[C]")*1000
40 PUNCH TOT("[D]")*1000
        -end
END
```

The **RATES** data block defines rate expressions for the decay of each species in the decay series. Each rate expression converts decay constants in units of per day to units of per second and multiplies by the respective species concentration to obtain the rate of decay. In the **KINETICS** *1* data block, the stoichiometry of each decay reaction is defined. The sign of the stoichiometric coefficient in **-formula** in **KINETICS** combined with the sign of the mole transfer calculated in the **RATES** expression (*30* **SAVE** *moles*) gives the direction of the transfer relative to the solution. In this example, *moles* as calculated in **RATES** is negative. Therefore, constituents that have a positive coefficient for **-formula** identifier in the **KINETICS** data block will be removed from solution (*moles* times coefficient is negative); constituents that have a negative coefficient in **-formula** will be added to the solution (*moles* times coefficient is positive). Thus, the rate expression "[A]_decay" converts [A] to [B]; "[B]_decay" converts [B] to [C]; "[C]_decay" converts [C] to [D]; and "[D]_decay" removes [D]. The **SELECTED_OUTPUT** and **USER_PUNCH** data blocks define data to be written to output files as described in example 1.

For the definitions of the flow and transport data file (table 6.6), it is useful to take advantage of the symmetry of the problem definition. A horizontal plane of symmetry is located at Z=12.5 m, and a vertical plane of symmetry is located at Y=20.5 m. All concentration, head, and velocity fields on either side of each of these two planes are mirror images. Thus, it is possible to run a simulation on a grid region that is one quarter of the original problem (Y=0 to 20.5 m, Z=0 to 12.5 m) and reflect the results to the other three quadrants. In this example, all properties, boundary conditions, and initial conditions are defined for the original problem (Y=0 to 41 m, Z = 0 to 25 m). Only the grid, which defines the actual simulation region, is restricted to the front lower quadrant (Y=0 to 20.5 m, Z=0 to 12.5 m).

Table 6.6.—Flow and transport data file for example 2

```
TITLE
.       3D analytic, continuous injection, 4 species chain decay
.       Sun, Peterson, Clement (1999) J. Cont. Hyd., Example 6.3
.       Documentation compares PHAST with chain decay from
.       Sun and others combined with Wexler (1992) solution
.       to patch source
SOLUTE_TRANSPORT                            true
UNITS
        -time                               day
        -horizontal                         m
        -vertical                           m
        -head                               m
        -hydraulic_conductivity             m/d
        -specific_storage                   1/m
        -dispersivity                       m
GRID
# Symmetry is used, so that grid is only defined on 1/4 of YZ face
        -uniform X  0.0    100    26        # 4.0  m spacing
        -uniform Y  0.0    20.5   2
        -overlay_uniform Y
# nodes are set so that cell face is at 15 m to allow source patch definition
                    0.5    19.5   20        # 1.0  m spacing
        -uniform Z  0.0    12.5   2
        -overlay_uniform Z
# nodes are set so that cell face is at 10 m to allow source patch definition
                    0.5    11.5   12        # 1.0  m spacing
        -print_orientation XY
MEDIA
        -zone   0 0 0 100 41 25
# V = (1/e)*K*dh/dx, .2 = 1/.1*K*1/100, K = 2
                -Kx                         2
                -Ky                         2
                -Kz                         2
                -porosity                   0.1
                -storage                    0
                -longitudinal_dispersivity  1.5
                -horizontal_dispersivity    0.45
```

```
                -vertical_dispersivity      0.15
FREE_SURFACE_BC false
SPECIFIED_HEAD_BC
# first define x=0 head, inflow to be pure water
        -zone   0 0 0 0 41 25
                -head                       0   1
                -fixed_solution             0   1
# now define patch source of contaminant A
        -zone   0 15 10 0 26 15
                -fixed_solution             0   2
# define x=100 head
        -zone   100 0 0 100 41 25
                -head                       0   0
                -associated_solution        0   1
HEAD_IC
        -zone   0 0 0 100 41 25
                -head   X 1.0 0.0 0.0 100.0
CHEMISTRY_IC
        -zone   0 0 0 100 41 25
                -solution                   1
                -kinetics                   1
SOLUTION_METHOD
        -iterative                          true
        -tolerance                          1e-14
        -space                              0.5
        -time                               0.5
        -cross_dispersion                   false
PRINT_FREQUENCY
        0
        -hdf_chemistry                      400 day
        -vel                                400 day
        -xyz_chemistry                      400 day
        -xyz_head                           400 day
        -xyz_velocity                       400 day
TIME_CONTROL
        -time_step                          0   10  day
        -time_end                           400 day
END
```

The uses of the **TITLE**, **SOLUTE_TRANSPORT**, **UNITS**, **FREE_SURFACE_BC**, and **PRINT_FREQUENCY** data blocks are the same as described in described in example 1. For demonstration purposes, steady-state flow (**STEADY_FLOW** data block) is not specified, so head and velocity are calculated at each time step of the simulation, even though the flow field is specified by the initial and boundary conditions to be at steady state.

To ensure that the rectangular patch over which species [A] is introduced into the region corresponds exactly the edges of cell faces, it is necessary to define cell boundaries that coincide with Y=15 m and Z=10 m. Cell boundaries occur halfway between nodes, so for equally spaced nodes 1 m apart, Y nodes need to be placed at Y=14.5 m and Y=15.5 m. Similarly, Z nodes need to be placed at 9.5 m and 10.5 m. The **GRID** data block defines the locations of these nodes and the extent of the grid region by using a combination of uniform grid spacings. For the Y dimension, nodes are placed at Y=0 m and Y=20.5 m with the **-uniform** *Y* identifier. Nodes are placed at 0.5, 1.5, 2.5, …19.5 m, which satisfy the required cell boundary criterion, by overlaying additional equally spaced nodes with the **-overlay_uniform** *Y* identifier. Similarly, **-uniform** *Z* and **-overlay_uniform** *Z* identifiers place nodes in the required locations in the vertical. Nodes in the X direction have four-meter spacing from 0 to 100 m.

An interstitial velocity of 0.2 m/d (meter per day) is used in the analytical solution for this example simulation. From Darcy's Law (eq. D.3), if the head gradient is defined to be 0.01 (1 m per 100 m) by the boundary conditions and the porosity is 0.1, then a hydraulic conductivity of 2 m/d will produce the desired interstitial velocity of 0.2 m/d. The **MEDIA** data block defines the hydraulic conductivity, porosity, storage, and the three dispersivities (table 6.4) from Sun and others (1999). The **SPECIFIED_HEAD_BC** data block defines the head to be 1 m at the upstream end of the grid region and 0 m at the downstream end of the grid region to produce a head gradient of 0.01 for time 0 s onward. Also in the **SPECIFIED_HEAD_BC** data block, the water composition at the boundary (**-fixed_solution**) is defined to be pure water (**SOLUTION** *1* in chemistry data file) from time 0 s onward, except over a rectangular patch at X=0. For the patch from Y=15 to 26 m and Z=10 15 m, the water composition is specified to be solution 2 (water containing [A], **SOLUTION** *2* in chemistry data file) from time 0 s onward.

The initial head condition is defined to be linear from 1 m at X=0 m to 0 m at X=100 (**HEAD_IC** data block), which is consistent with the desired steady-state flow system. The initial water composition within the grid region is specified by the **CHEMISTRY_IC** data block to be pure water (**-solution** *1*) and four kinetic decay reactions are defined by the identifier **-kinetics** *1*, where *1* corresponds to the definition of **KINETICS** *1* in the chemistry data file.

The iterative linear-equation solver is specified to be used for the finite-difference equations with the **SOLUTION_METHOD** data block and the identifier **-tolerance** is used to define the relative residual tolerance for convergence of the iterative solver. Centered-in-space (**-space**) and centered-in-time (**-time**) weightings are used to form the finite-difference equations. The **TIME_CONTROL** data block specifies the time step to be 10 d (days), starting at time 0 s. The time at the end of the simulation is 400 d (**-time_end**). The time step is less than the estimate of the time step needed to avoid oscillations due to temporal discretization (see section D.1.3. Temporal Discretization), $\Delta t = 10 < \frac{4^2}{1.5(0.2)} = 53$ d. The cell Peclet number (eq. D.5) in the X direction is $\frac{4}{1.5} = 2.7$ (see section D.1.2. Spatial Discretization), which is greater than 2, the criterion to avoid oscillations (eq. D.8) due to spatial discretization; however, in this example no oscillations appear in the results. Cross-dispersive terms are not included in the difference equations (**-cross_dispersion** *false*). The cross-dispersion identifier was not necessary because the default is not to include the cross-dispersive terms. Cross-dispersive terms were not used in this example because the analytical solution from Wexler (1992) does not include cross dispersion.

Results of the simulation at 400 d for the plane Z=12.5 are shown in figure 6.2 along with the analytical solution. The similarity of results indicates that PHAST is performing the calculations correctly. The minor discrepancies between the two sets of curves are ascribed to truncation errors and operator-splitting errors in PHAST, which could be decreased by decreasing spatial and temporal discretization.

6.4. Example 3: Aerobic Consumption of a Substrate with Biomass Growth

Example 3 simulates flow in a one-dimensional column that contains a bacterial community that consumes oxygen and degrades a substrate. PHAST is used to model the kinetics of aerobic bacterial growth in a flowing system, while considering cell death and growth limitation by oxygen and substrate availability. The example is taken from Kindred and Celia (1989, section 4.1, p. 1154).

The rate equations for oxygen consumption are given by

$$R_{oxygen,\,O} = V_{max}(O)\frac{X}{I_b}\left(\frac{O}{K_h(O)+O}\right), \tag{6.4}$$

$$R_{oxygen,\,S} = \frac{Y_S}{Y_O}V_{max}(S)\frac{X}{I_b}\left(\frac{S}{K_h(S)+S}\right), \text{ and} \tag{6.5}$$

$$R_{oxygen} = Minimum(R_{oxygen,\,O}, R_{oxygen,\,S}), \tag{6.6}$$

where $R_{oxygen,\,O}$ is the limiting rate of oxygen consumption due to oxygen availability; $R_{oxygen,\,S}$ is the limiting rate of oxygen consumption due to substrate availability; R_{oxygen} is the limiting rate of oxygen consumption, which is the minimum of the two rates, $R_{oxygen,\,O}$ and $R_{oxygen,\,S}$; O, S, and X are the concentrations of oxygen, substrate, and aerobic biomass; $V_{max}(O)$ and $K_h(O)$ are the kinetic parameters for oxygen; $V_{max}(S)$ and $K_h(S)$ are the kinetic parameters for the substrate; Y_O and

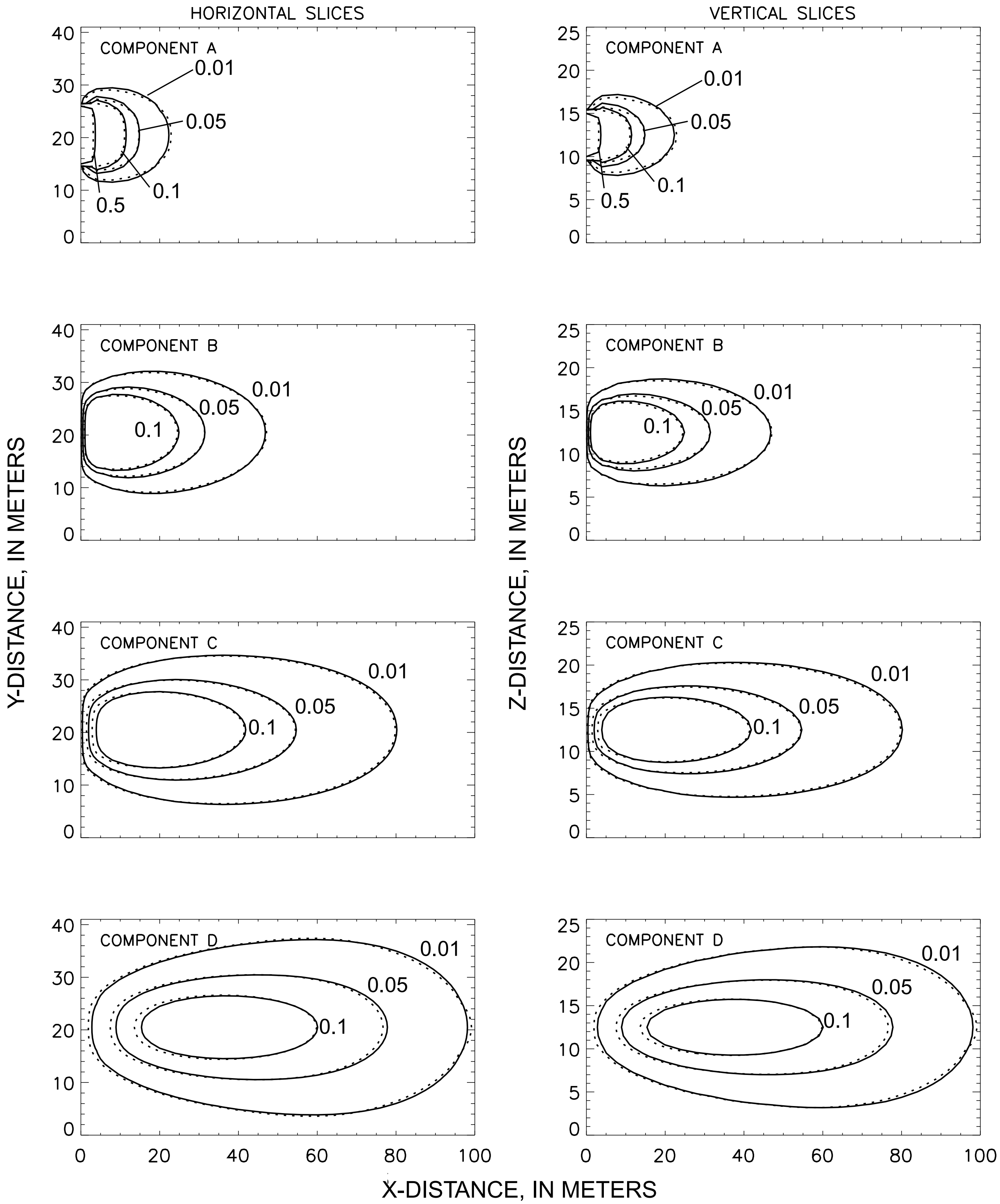


Figure 6.2.—Example 2: Analytical solution (solid lines) compared to simulation results (dashed lines) for concentrations at 400 days and Z=12.5 meters of a chain of decay products. Component A decays sequentially to components B, C, and D.

Y_S are yield coefficients for oxygen and organic substrate, and I_b is the biomass inhibition factor. $I_b = 1 + \frac{X}{k_{biomass}}$, where $k_{biomass}$ is the aerobic biomass inhibition constant. The rate of biomass production is

$$R_{biomass} = Y_O R_{oxygen} - k_m X, \tag{6.7}$$

where k_m is the specific death rate or maintenance coefficient for aerobic biomass. The parameters for equations 6.4-6.7 and the flow and transport parameters for the simulation are given in table 6.7.

Table 6.7.—Chemical and physical parameters for example 3

[mg, milligrams; L, liter; m, meter; s, second; d, day]

Parameter	Value
$V_{max}(S)$, d^{-1}	1
$K_h(S)$, mg/L	.1
Y_S, L/mg	.25
$V_{max}(O)$, d^{-1}	1
$K_h(O)$, mg/L	.1
Y_O, L/mg	.125
k_m, d^{-1}	.01
$k_{biomass}$, mg/L	1
X_0, initial biomass, mg/L	.2
O_0, initial dissolved oxygen, mg/L	3.0
Porosity, unitless	.38
Dispersivity, m	.2
Hydraulic conductivity, m/s	1.e-4
Column length, m	50
Time simulated, d	37
Interstitial velocity, m/d	1.0

The chemistry data file for this example is given in table 6.8. The **TITLE**, **SELECTED_OUTPUT**, and **USER_PUNCH** data blocks are used in the same ways as the previous two examples. The **SOLUTION_MASTER_SPECIES** and **SOLUTION_SPECIES** data blocks define a new chemical component to represent the substrate that is used by the bacteria. **SOLUTION** *1* defines the composition of the water initially present in the column, which contains no substrate; **SOLUTION** *2* defines the water that is introduced into the column, which contains 10 mg/L (milligram per liter) of the substrate. All concentrations are entered as mg/L.

Table 6.8.—Chemistry data file for example 3

```
TITLE
        Kindred and Celia, WRR, 1989, v. 25, p. 1154
        Problem 4.1. Aerobic Biodegradation
SOLUTION_MASTER_SPECIES
        Substrate       Substrate       0       1.0     1.0
SOLUTION_SPECIES
        Substrate = Substrate
        log_k   0.0
SOLUTION 1  Initial condition
        units   mg/L
        pH      7.0
        pe      8
        Substrate       0.0
        O(0)            3
SOLUTION 2  Infilling
        units   mg/L
        pH      7.0
        pe      8
        Substrate       10.
        O(0)            3
END
RATES
Aerobic
        -start
200 vmaxO = 1/(24*3600)
210 vmaxS1 = 1/(24*3600)
240 Kh1O = .1
250 Kh1S1 = .1
290 kbio = 1
320 km = .01/(24*3600)

410 O = TOT("O(0)")*16*1000
430 S1 = TOT("Substrate")*1000
450 x1 = KIN("aerobic_biomass")
470 Ib = 1 + x1 / kbio

500    REM  **  Oxygen rate ***
510 rateO = (vmaxO * x1 / Ib) * (O / ( kh1O + O))
520    REM  **  Substrate rate **
530 rateS1 = 2*(vmaxS1 * x1 / Ib) * (S1 / ( kh1S1 + S1))
560 rate = rateO
570 if rateS1 < rate then rate = rateS1
590 mg_O = -TIME*rate
600 SAVE mg_O
        -end

Aerobic_biomass
        -start
200 vmaxO = 1/(24*3600)
210 vmaxS1 = 1/(24*3600)
240 Kh1O = .1
250 Kh1S1 = .1
290 kbio = 1
320 km = .01/(24*3600)

410 O = TOT("O(0)")*16*1000
430 S1 = TOT("Substrate")*1000
```

```
450 x1 = KIN(“aerobic_biomass”)
470 Ib = 1 + x1 / kbio

520    REM  **  Substrate rate ***
530 rateS1 = .25*(vmaxS1 * x1 / Ib) * (S1 / ( kh1S1 + S1))
540    REM  **  Oxygen rate ***
550 rateO = 0.125*(vmaxO * x1 / Ib) * (O / ( kh1O + O))
560 rate = RateO
570 if rateS1 < rate then rate = rateS1
590 mg_O_biomass = -TIME*(rate - km * x1)
600 SAVE mg_O_biomass
        -end

KINETICS 1
Aerobic
#     mg_O * 1/gfw * 1/1000 = mg_O * 1/(16*1000) = mg_O * 6.25e-5
#     mg_O * (mg_S/mg_O)* 1 / gfw * 1/1000 = mg_O * 0.5 * 1/(1*1000) = mg_O * 5e-4
        -formula  O 6.25e-5 Substrate 5e-4
        -m 0
Aerobic_biomass
        -formula  Substrate  0
        -m 0.2
KINETICS 2
Aerobic_biomass
        -formula  Substrate  0
        -m 0.2
SELECTED_OUTPUT
        -file ex3.dummy.sel
        -reset false
USER_PUNCH
-headings Substrate O(0) O_biomass
10 PUNCH TOT(“Substrate”)*1000,  TOT(“O(0)”)*16*1000
20 PUNCH KIN(“Aerobic_biomass”)
END
```

Two rate expressions are defined in the **RATES** data block, which are named “Aerobic” and “Aerobic_biomass”. The Basic-language statements for the “Aerobic” rate expression calculate the rate of oxygen consumption as defined by equations 6.4-6.6. The Basic-language statements for the “Aerobic_biomass” rate expression calculate the rate of biomass production (or decay) as defined by equation 6.7. The **KINETICS** *1* data block defines the stoichiometry of the reactions and the amount of reactants initially present in the rest of the column ($X > 0$), which includes biomass growth (“Aerobic_biomass”) and oxygen and substrate consumption (“Aerobic”). The **KINETICS** *2* data block defines the stoichiometry of the reactions and the amount of reactants that will be present in the cell at $X = 0$; biomass growth (“Aerobic_biomass”) is included [Kindred and Celia (1989, show aerobic biomass at $X = 0$], but oxygen and substrate consumption (“Aerobic”) are not included to maintain the specified-head boundary condition in this cell for dissolved oxygen and substrate concentrations. Considering the definitions in **KINETICS** and **RATES** for “Aerobic”, the sign and magnitude of the transfers need explanation. First, the sign of the mass transfer calculated in the “Aerobic” rate expression (the quantity *mg_O* in the **RATES** data block) is negative and the coefficients (**-formula**) in the **KINETICS** data blocks are positive. Therefore, the product of mass transfer (*mg_O* from **RATES**) and coefficient (from **KINETICS**, **-formula**) is negative and the “Aerobic” kinetic reaction removes oxygen and substrate from solution. The rate expressions from Kindred and Celia (1989) calculate mole transfer in terms of mg, which is assumed to be equal to mg/kgw; however, for PHAST, mass-transfers must have units of moles. The coefficients in the identifier -**formula** have been adjusted to provide molar mass-transfer quantities. If mg_O is the mass-transfer calculated by the **RATES** expression, the mole transfer is $mg\_\mathrm{O}\frac{1}{f_O 1000} = mg\_\mathrm{O}\frac{1}{(16)1000} = 6.25x10^{-5} mg\_\mathrm{O}$, where f_O is the gram formula weight of elemental oxygen g/mol (grams per mole). The number of milligrams of substrate removed is $\frac{Y_O}{Y_S} mg\_O$, and the number of moles of substrate removed is $\frac{Y_O}{Y_S} mg\_O \frac{1}{f_S 1000} = \frac{0.5}{(1)1000} mg\_O = 5x10^{-4} mg\_O$, where f_S is the gram formula weight of substrate (g/mol). Thus, the coefficients in -formula for "Aerobic" in the **KINETICS** data blocks are 6.25x10^{-5} for oxygen and 5x10^{-4} for substrate.

The coefficient for Substrate in **-formula** for "Aerobic_biomass" in the **KINETICS** data blocks is zero. Thus, the "Aerobic_biomass" reaction does not affect the solution composition in any way, but the mass transfer (*mg_O_biomass*) does affect the calculation through the cumulative amount of a reactant that is present. Initially, the amount of biomass is set to 0.2 mg (**-m** identifier) per kilogram of water. As biomass increases or decreases according to the *mg_O_biomass* calculated in the rate expression, the variables KIN("Aerobic_biomass") and "M" (only available within the Basic-language program for "Aerobic_biomass") are updated. Note the sign convention for calculating the cumulative moles of reactant: positive *mg_O_biomass* (from **RATES** expressions) decreases the amount of reactant (biomass) and negative *mg_O_biomass* increases the amount of reactant (biomass). (For dissolved Substrate, the gram formula weight was defined to be 1.0, which means that moles equals grams.) At any time in the simulation, the cumulative amount of biomass is known through either of the two variables KIN("Aerobic_biomass") or "M".

The flow and transport file corresponding to the problem description is given in table 6.9. The **TITLE**, **UNITS**, **SOLUTE_TRANSPORT**, **MEDIA**, and **FREE_SURFACE_BC** data blocks have the same functions as previously described in the first two examples. A 50-m column is defined with 101 nodes at a uniform spacing of 0.5 m (**GRID** data block). Because the example is one-dimensional for flow and transport and symmetric in the Y and Z directions, chemistry needs to be calculated for only one line of nodes in the X direction (**-chemistry_dimensions** *X*). The head gradient is calculated such that the interstitial velocity is 1 m/d. By assuming a hydraulic conductivity of 1x10^{-4} m/s and a porosity of 0.38, Darcy's law (eq. D.3) is used to determine that a decrease in head of 2.199074 m over the 50-m grid region will result in an interstitial velocity of 1 m/d. Specified heads are defined at each end of the column (**SPECIFIED_HEAD_BC** data block) to maintain the head gradient and any water entering the column at the upstream end will have the composition of solution 2 (water containing substrate) as defined in the chemistry data file. Both the heads and solution compositions for the specified-head boundary conditions apply from time 0 d to the end of the simulation. The initial head condition is defined to be consistent with the boundary conditions and the steady-state flow at the specified velocity; the initial head is 2.199074 m at X=0, decreasing linearly to 0 m at X=50 m (**HEAD_IC** data block). The **CHEMISTRY_IC** data block defines the water composition and reactants that initially are present in the column. Solution 1 fills the column and a set of kinetic reactions (**KINETICS** *1*, in the chemistry data file) are present throughout the column, except at the inlet (X=0). At the inlet, biomass is allowed to accumulate, but the "Aerobic" reaction, which consumes oxygen, is not included because the concentration of oxygen is specified by the problem definition to be fixed at 3.0 mg/L at the boundary.

Table 6.9.—Flow and transport data file for example 3

```
TITLE
        Kindred and Celia, WRR, 1989, v. 25, p. 1154
        Problem 4.1. Aerobic Biodegradation
UNITS
        -time                                   days
        -horizontal_grid                        meters
        -vertical_grid                          meters
        -head                                   meters
        -hydraulic_conductivity                 m/s
        -specific_storage                       1/m
        -dispersivity                           m
SOLUTE_TRANSPORT                                true
GRID
        -uniform x     0.0      50.     101
        -uniform y     0.0      1.0     2
        -uniform z     0.0      1.0     2
        -chemistry_dimensions                   X
        -print_orientation                      XZ
MEDIA
```

```
        -zone   0. 0. 0. 50.  1. 1.
                -porosity                      0.38
                -long_dispersivity             0.2
                -horizontal_dispersivity       0.2
                -vertical_dispersivity         0.2
                -Kx                            1e-4
                -Ky                            1e-4
                -Kz                            1e-4
                -specific_storage              0
HEAD_IC
#velocity 1 m/day    v*por/K*L = delta H
# 1/(24*3600)*.38*50./1e-4 = 2.199074
        -zone   0. 0. 0. 100. 1. 1.
                -head   X   2.199074 0.  0. 50.
SPECIFIED_HEAD_BC
        -zone   0. 0. 0. 0.  1.  1.
                -head                          0       2.199074
                -associated_solution           0       2
        -zone   50. 0. 0. 50.  1.  1.
                -head                          0       0.
                -associated_solution           0       1
FREE_SURFACE_BC                                false
CHEMISTRY_IC
        -zone   0. 0. 0. 50. 1. 1.
                -solution                      1
                -kinetics                      1
        -zone   0. 0. 0. 0. 1. 1.
                -kinetics                      2
SOLUTION_METHOD
        -direct_solver                         true
        -space_differencing                    0.5
        -time_differencing                     0.5
PRINT_INITIAL
        -components                            true
TIME_CONTROL
        -delta_time                            0       0.2 day
        -end_time                              37  day
PRINT_FREQUENCY
        0
        -xyz_chemistry                         10  day
        -hdf_chemistry                         1   day
        -hdf_velocity                          100 day
END
```

The **SOLUTION_METHOD** data block specifies use of the direct linear-equation solver and centered-in-space, centered-in-time weighting for the finite-difference equations. Oscillations are possible with this weighting because the cell Peclet number (2.5, eq. D.5) is greater than 2.0 (see section D.1.2. Spatial Discretization) for this node spacing (0.5 m) and dispersivity (0.2). The time step (0.2 d) exceeds the time criterion for possible oscillations (0.125 d, eq. D.8) (see section D.1.3. Temporal Discretization). The **PRINT_INITIAL** data block specifies that initial concentrations for each component will be written to the *ex3***.O.comps** file. The print frequencies for the *ex3***.xyz.chem** and *ex3***.h5** files are specified in the **PRINT_FREQUENCY** data block. The duration of the simulation and the size of the time step are defined in the **TIME_CONTROL** data block.

Results of the simulation (fig. 6.3) appear to be identical to results presented in figure 1 of Kindred and Celia (1989) (not reproduced here). No oscillations are observed in the results, which justifies the use of the centered-in-time, centered-in-space weighting for the difference equations with the spatial and temporal discretization chosen.

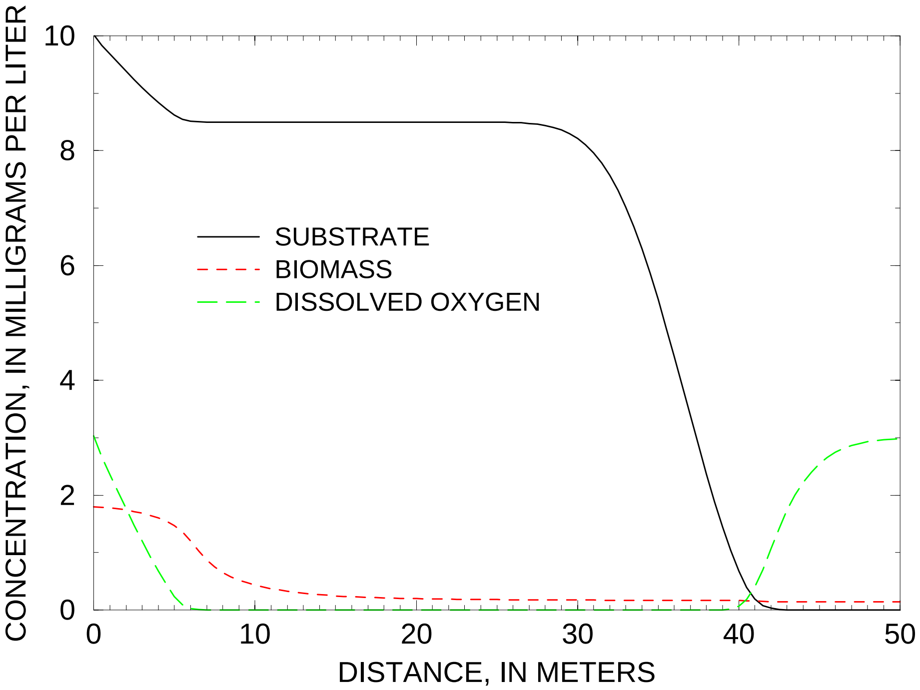


Figure 6.3.—Example 3: Concentrations of substrate, biomass, and dissolved oxygen at 37 days.

6.5. Example 4: Simulation of Regional-Scale Transport and Reactions in the Central Oklahoma Aquifer

Example 4 presents a reactive-transport model that simulates the evolution over geologic time of water compositions that are found in the Central Oklahoma aquifer. The model simulates flow, transport, and reactions at a regional scale [90 km (kilometers) by 48 km] for an aquifer with both confined and unconfined regions and a complex three-dimensional flow pattern. Assuming the aquifer initially contained brines similar to those found at depth in the area, the chemical evolution of the aquifer over geologic time is hypothesized to result from a constant influx of fresh water from precipitation and chemical reactions with the aquifer material. The simulated chemical evolution provides an explanation for differing water compositions between confined and unconfined parts of the aquifer and the large naturally occurring arsenic concentrations that are found in the confined part of the aquifer. Most of the flow and transport capabilities and the phase-equilibrium, cation-exchange, and surface-complexation reaction capabilities of PHAST are used in this example. Because of the variety of boundary conditions and chemical capabilities used, the input files are useful templates for setting up other field-scale reactive-transport simulations.

The geochemistry of the aquifer has been described in Parkhurst and others (1996). Two predominant water types occur in the aquifer, a calcium magnesium bicarbonate water with pH in the range of 7.0 to 7.5 in the unconfined part of the aquifer and a sodium bicarbonate water with pH in the range of 8.5 to 9.2 in the confined part of the aquifer. In addition, marine-derived sodium chloride brines exist below the aquifer and presumably in fluid inclusions and dead-end pore spaces within the aquifer. Large concentrations of arsenic, selenium, chromium, and uranium occur naturally within the aquifer, but only arsenic is considered in this simulation. Figure 6.4 shows that locations of large arsenic concentrations are almost exclusively within the confined part of the aquifer, which also is the location of high-pH, sodium bicarbonate water.

The conceptual model for the calculation assumes that brines initially filled the aquifer. The aquifer contains calcite, dolomite, clays with cation exchange capacity, and hydrous ferric oxide surfaces. The initial compositions of the cation exchanger and surfaces are in equilibrium with the brine, which contains arsenic. Arsenic is initially sorbed on the hydrous ferric oxide surfaces. The aquifer is assumed to be recharged with rainwater that is concentrated by evaporation and equilibrated with calcite and dolomite in the vadose zone. This water then enters the saturated zone and reacts with calcite and dolomite in the presence of the cation exchanger and hydrous ferric oxide surfaces. A period of 100,000 years of flushing the brine-filled aquifer with fresh water is simulated.

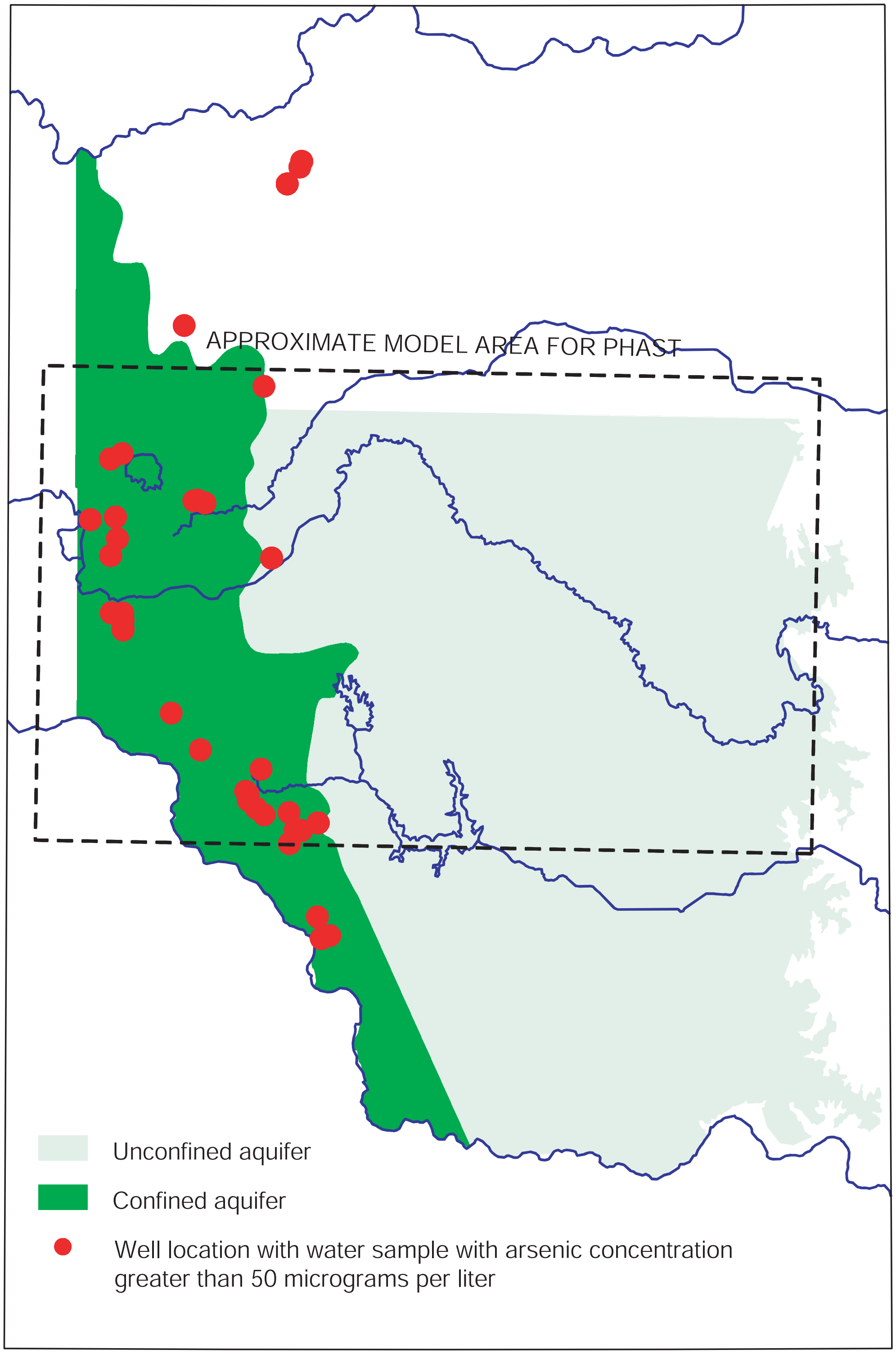


Figure 6.4.—Locations in the Central Oklahoma aquifer with arsenic concentrations greater than 50 micrograms per liter and PHAST model area.

In general, the ion-association model of PHAST is not appropriate for brine calculations. However, a brine is a logical initial condition for the chemical composition of the aquifer water in this example. The inclusion of a brine initial condition is justified on the basis that concentrations quickly decrease to levels appropriate for the ion association model and that the brine has a sodium chloride composition. Chloride salts were used to fit the activity coefficient parameters of the major cations in the aqueous model of PHAST and so thermodynamic calculations should be appropriate to higher ionic strengths in chloride dominated systems than in systems dominated by other anions.

6.5.1. Initial Conditions

Parkhurst and others (1996) provide data from which it is possible to estimate the number of moles of calcite, dolomite, and cation exchange sites in the aquifer per liter of water. Point counts from thin sections from aquifer core material indicate the weight percent for calcite ranges from 0 to 2 percent and dolomite from 0 to 7 percent, with dolomite more abundant in all samples. Again from point counts of thin sections, porosity is estimated to be 0.22. Measurements of cation exchange capacity for the clay ranged from 10 to 50 meq/100 g (milliequivalents per 100 grams), with average clay content of 30 percent. For the simulation in this example, calcite was assumed to be present at 0.1 weight percent and dolomite at 3 weight percent, which, assuming a rock density of 2.7 kg/L (kilogram per liter), corresponds to 0.1 mol/L (mole per liter) for calcite and 1.6 mol/L for dolomite. The number of cation exchange sites was estimated to be 0.5 eq/L (equivalents per liter).

The amount of arsenic on the surface of hydrous ferric oxides was estimated from sequential extraction data on core samples (Mosier and others, 1991). Arsenic concentrations in the solid phases generally ranged from 10 to 20 ppm, which correspond to 1.3 to 2.6 mmol/L (millimole per liter) arsenic. The number of surface sites was estimated from the amount of extractable iron in sediments, which ranged from 1.6 to 4.4 percent (Mosier and others, 1991). A content of 2 percent iron for the sediments corresponds to 3.4 mol/L of iron. However, most of the iron is contained in goethite and hematite, which have fewer surface sites than hydrous ferric oxides. The fraction of iron in hydrous ferric oxides, as opposed to other ferric oxides, was arbitrarily assumed to be 0.2. Thus, a total of 0.68 mol of iron was assumed to be in hydrous ferric oxides, and using a value of 0.2 for the number of sites per mole of iron, a total of 0.14 mol of sites per liter was used in the calculations. A gram formula weight of 89 was used to estimate that the mass of hydrous ferric oxides was 30 g/L (gram per liter). The specific surface area for hydrous ferric oxides was assumed to be 600 m^2/g (square meter per gram).

6.5.2. Chemistry Data File

The chemistry data file for this example simulation is presented in table 6.10. The default thermodynamic database file, *phast.dat* (which is equivalent to *phreeqc.dat* in the PHREEQC distribution), was augmented with additional thermodynamic data for arsenic and surface complexation in the chemistry data file. The **SURFACE_MASTER_SPECIES** data block defines a new surface named “Surf”, which is not present in the thermodynamic data base. The **SURFACE_SPECIES** data block defines all of the surface-complexation reactions for the surface *Surf*. **SOLUTION_MASTER_SPECIES**, and **SOLUTION_SPECIES** data blocks explicitly define the aqueous reactions of arsenic; these species are not present in *phast.dat*, but the definitions would supersede any definitions of these species had they been present in the thermodynamic database file. The reactions and equilibrium constants for surface species and arsenic aqueous species are taken from Dzombak and Morel (1990).

The water entering the saturated zone of the aquifer was assumed to be the composition of rain water concentrated twenty fold (**SOLUTION** *1*) to account for evapotranspiration and then equilibrated with calcite and dolomite at a vadose-zone partial pressure of carbon dioxide (P_{CO_2}) of $10^{-1.5}$ atm (atmosphere). A batch-reaction calculation (**USE** and **EQUILIBRIUM_PHASES** *1* data blocks) is defined and the resulting solution is stored as solution 1 (**SAVE solution** *1*).

The composition of the brine that represents the water that initially filled the aquifer was selected from Parkhurst and others (1996) and is defined in the **SOLUTION** *2* data block. A pure-phase assemblage containing calcite and dolomite is defined with the **EQUILIBRIUM_PHASES** *2* data block. The brine and pure-phase assemblage are allowed to react to equilibrium and the result of the batch-reaction equilibration is stored as solution 2 (**SAVE solution** *2*).

The number of cation exchange sites is defined with the **EXCHANGE** *2* data block and the number of surface sites, specific surface area, and mass of surface material are defined with the **SURFACE** *2* data block. Both the initial exchange and the initial surface compositions are specified to be in equilibrium with the brine (**-equil solution** *2*). The concentration of arsenic in the brine is set to 0.01 µmol/kgw (micromole per kilogram water), which results in a total of approximately 8 mmol arsenic on the surface (from the *ex4***.O.chem** file). This concentration on the surface is greater than the upper end of the range of the sequential extraction data.

The **SELECTED_OUTPUT** and **USER_PUNCH** data blocks are used to specify the data that will be written to the *ex4***.h5** and *ex4***.xyz.chem** files. Writing of data selected by default is suppressed (**-reset false**) and only pH is selected for writing (**-pH**) in the **SELECTED_OUTPUT** data block. Concentrations of calcium, magnesium, sodium, chloride, carbon, and sulfur are specified to be printed in units of mg/L and arsenic is specified to be printed in units of µg/L (microgram per liter) in the Basic-language statements of the **USER_PUNCH** data block.

Table 6.10.—Chemistry data file for example 4

```
SURFACE_MASTER_SPECIES
        Surf    SurfOH
SURFACE_SPECIES
        SurfOH = SurfOH
                log_k   0.0
        SurfOH  + H+ = SurfOH2+
                log_k   7.29
        SurfOH = SurfO- + H+
                log_k   -8.93
        SurfOH + AsO4-3 + 3H+ = SurfH2AsO4 + H2O
                log_k   29.31
        SurfOH + AsO4-3 + 2H+ = SurfHAsO4- + H2O
                log_k   23.51
        SurfOH + AsO4-3 = SurfOHAsO4-3
                log_k   10.58
SOLUTION_MASTER_SPECIES
        As       H3AsO4         -1.0     74.9216          74.9216
SOLUTION_SPECIES
#H3AsO4 primary master species
        H3AsO4 = H3AsO4
        log_k           0.0
#H2AsO4-             482
        H3AsO4 = AsO4-3 + 3H+
        log_k   -20.7
#HAsO4-2                483
        H+ + AsO4-3 = HAsO4-2
        log_k   11.50
#AsO4-3              484
        2H+ + AsO4-3 = H2AsO4-
        log_k           18.46
SOLUTION 1 20 x precipitation
        pH      4.6
        pe      4.0     O2(g)   -0.7
        temp    25.
        units   mmol/kgw
        Ca      .191625
        Mg      .035797
        Na      .122668
        Cl      .133704
        C       .01096
        S       .235153         charge
END
USE solution 1
EQUILIBRIUM_PHASES 1
        Dolomite        0.0     1.6
        Calcite         0.0     0.1
        CO2(g)          -1.5    100.
SAVE solution 1
END
SOLUTION 2 Brine
```

```
        pH      5.713
        pe      4.0     O2(g)   -0.7
        temp    25.
        units   mol/kgw
        Ca      .4655
        Mg      .1609
        Na      5.402
        Cl      6.642           charge
        C       .00396
        S       .004725
        As      .01 umol/kgw
END
USE solution 2
EQUILIBRIUM_PHASES 2
        Dolomite        0.0     1.6
        Calcite         0.0     0.1
SAVE solution 2
END
EXCHANGE 2
        -equil with solution 2
        X       0.5
SURFACE 2
        -equil solution 2
        SurfOH          0.14     600.    30.
END
SELECTED_OUTPUT
        -file ex4.dummy.sel
        -reset false
        -pH
USER_PUNCH
# Prints concentrations in mg/kgw to ex4.xyz.chem
-heading        Ca      Mg      Na      Cl      C(4)    SO4     As
10 PUNCH TOT(“Ca”)*1e3*40.08
20 PUNCH TOT(“Mg”)*1e3*24.312
30 PUNCH TOT(“Na”)*1e3*23.
40 PUNCH TOT(“Cl”)*1e3*35.45
50 PUNCH TOT(“C(4)”)*1e3*61.   # as HCO3-
60 PUNCH TOT(“S(6)”)*1e3*96.   # as SO4
70 PUNCH TOT(“As”)*1e6*74.296  # ug/L
END
```

6.5.3. Flow and Transport Data File

The flow and transport file (table 6.11) specifies the input units, grid, the porous media properties, boundary conditions, initial conditions, print control, time step, and duration of the simulation through a series of keyword data blocks. A two-line title is defined with the **TITLE** data block. Units for all of the *input* data are defined in the **UNITS** data block; all *output* data are in SI units, with the exception of time, which will be years as specified by the **-time** identifier. A grid (**GRID** data block) containing 16 nodes in the X direction (6,000-m node spacing), 9 nodes in the Y direction (6,000-m node spacing), and 5 nodes in the Z direction (100-m node spacing) is used to represent the part of the Central Oklahoma aquifer selected for modeling. The northern and southern boundaries of the model are near rivers that provide satisfactory boundary conditions. The eastern boundary of the model coincides with the eastern extent of the geologic units of the aquifer. The extent of fresh water in the aquifer is used to set the western boundary of the model.

Table 6.11.—Flow and transport data file for example 4

```
TITLE
 Central Oklahoma aquifer,
 demonstration of PHAST
UNITS
        -time                                years
        -horizontal_grid                     meters
        -vertical_grid                       meters
        -head                                meters
        -hydraulic_conductivity              m/s
        -specific_storage                    1/m
        -dispersivity                        m
        -leaky_k                             m/sec
        -leaky_thickness                     m
        -flux                                m/yr
        -river_bed_thickness                 m
        -river_bed_hydraulic_conductivity    m/s
        -well_diameter                       in
        -well_flow_rate                      L/day
GRID
        -uniform x    0    90000    16
        -uniform y    0    48000    9
        -uniform z    0    400      5
        -print_orientation                   XY
SOLUTE_TRANSPORT true
STEADY_FLOW true
        -head_tolerance                      1e-6
        -flow_balance_tolerance              1e-3
MEDIA
        -zone   0. 0. 0. 90000.  48000. 400.
                -porosity                    0.22
                -long_dispersivity           2000.
                -horizontal_dispersivity     50.
                -vertical_dispersivity       50.
                -Kx                          1.5e-5
                -Ky                          1.5e-5
                -Kz                          1.5e-7
                -storage                     0
        # Make bottom, east zone inactive
        -zone   48000. 0. 0. 90000. 48000. 100.
                -active                      0

RIVER 1 Little River
        -point  44000. 15000.
                -width                       200.
                -depth                       1.
                -bed_thickness               1.
                -bed_hydraulic_conductivity  1.
                -head                        0         335.
                -solution                    0         1
        -point  44000. 0.
        -point  90000. 0.
                -width                       200.
                -depth                       1.
                -bed_thickness               1.
                -bed_hydraulic_conductivity  1.
                -head                        0         275.
                -solution                    0         1
```

```
RIVER 2 North Fork River
        -point  30000. 36000.
                -width                        200.
                -depth                        1.
                -bed_thickness                1.
                -bed_hydraulic_conductivity   1
                -head                         0         335.
                -solution                     0         1
        -point  30000. 48000.
        -point  90000. 48000.
                -width                        200.
                -depth                        1.
                -bed_thickness                1.
                -bed_hydraulic_conductivity   1
                -head                         0         280.
                -solution                     0         1
RIVER 3 North Canadian River
        -point  60000. 30000.
                -width                        200.
                -depth                        1.
                -bed_thickness                1.
                -bed_hydraulic_conductivity   1
                -head                         0         350.
                -solution                     0         1
        -point  90000. 20000.
                -width                        200.
                -depth                        1.
                -bed_thickness                1.
                -bed_hydraulic_conductivity   1
                -head                         0         305.
                -solution                     0         1
FLUX_BC
        -zone   30000. 3000. 400.  90000. 45000. 400.
                -flux                         0         -0.055
                -associated_solution          0         1
SPECIFIED_HEAD_BC
        # Lake Stanley Draper
        -zone   30000. 14000 300. 32000.  20000.  400.
                -head                         0         348.
                -associated_solution          0         1
LEAKY_BC
        -zone   0. 48000. 0. 29000. 48000. 400.
                -hydraulic                    1.5e-5
                -thickness                   30000.
                -head                         0         305.0
                -associated                   0         1
       -zone 0 0 0 39000 0 400
                -hydraulic_conductivity       1.5E-5
                -thickness                    20000
                -head                         0         320
                -associated_solution          0         2
FREE_SURFACE_BC true
WELL 1 Observation well 1 in arsenic zone
        12000 36000
                -diameter       2                               # inches
                -pumping                      0         1       # L/day
                -elevation      90 110                          # m
HEAD_IC
        -zone   0. 0. 0. 90000. 48000. 400.
```

```
                -head   380.
CHEMISTRY_IC
        -zone   0. 0. 0. 90000. 48000. 400.
                -solution               2
                -equilibrium_phases     2
                -exchange               2
                -surface                2
SOLUTION_METHOD
        -iterative_solver       true
        -tolerance              1e-10
        -space_differencing     0.
        -time_differencing      1.
TIME_CONTROL
        -delta_time     0               2000    yr
        -end_time                       100000  yr
PRINT_INITIAL
        -velocities             true
        -xyz_head               true
        -xyz_ss_velocities      true
PRINT_FREQUENCY
        0
        -xyz_chemistry          50000   yr
        -hdf_chemistry          2000    yr
        -xyz_well               2000    yr
        -save_final_heads       true
PRINT_LOCATIONS
# Don't print bottom layer to ex4.xyz.chem file
        -xyz_chemistry
               -zone   0 0 0  90000 48000 0
               -print 0
END
```

Both flow and reactive-transport are simulated, which is specified with **SOLUTE_TRANSPORT** *true*. The **STEADY_FLOW** data block specifies that the program perform an initial flow calculation to obtain the steady-state head and velocity fields, which will be used for all transport calculations during the simulation. Criteria for attainment of steady flow are defined by the identifiers in the **STEADY_FLOW** data block. The properties of the porous media are defined in the **MEDIA** data block. The longitudinal dispersivity (2,000 m) and horizontal and vertical transverse dispersivities (50 m) are set arbitrarily to be less than the node spacing. The dispersivity specifies the amount of physical dispersion for the simulation; however, numerical dispersion also is present in the numerical solution as discussed below. The **MEDIA** data block excludes nodes in the eastern part of the lowest layer from the active grid region (**-active** *0*), so that the flow system thins toward the eastern edge.

Boundary conditions are defined with the **RIVER**, **FLUX_BC**, **SPECIFIED_HEAD_BC**, **LEAKY_BC**, and **WELL** data blocks. This example demonstrates the use of all of these data blocks; the nodes affected by these boundary conditions are shown in figure 6.5. Three rivers are defined in the **RIVER** data blocks. The location of each river is defined with X, Y locations. Head, width, depth, river-bed thickness, river-bed hydraulic conductivity, and associated solution are defined for the beginning and ending points for each river; in this example values for intermediate points are not specified, but are calculated by interpolation along the line segments that connect the river points. All rivers are expected to be gaining streams for this simulation, but if any water recharges from the river to the aquifer, its composition will be the composition of solution 1 (**-solution** *1* is used for all river end points). The hydraulic conductivity of the riverbed is large (**-bed_hydraulic_conductivity**), which will cause the head in the aquifer to be nearly equal to the head in the river for each cell that contains a river. All heads and solution compositions for the rivers are defined to apply from time 0 yr (year) through the end of the simulation.

A flux boundary condition (**FLUX_BC** data block) represents precipitation of 0.055 m/yr (meter per year) recharge over the eastern two-thirds of the aquifer with water composition defined by solution 1. Note the sign of the flux is negative, which indicates that the flux is in the negative Z direction. The **SPECIFIED_HEAD_BC** data block is used in this simulation to represent a lake. The western part of the northern boundary face is defined to have a leaky boundary condition (**LEAKY_BC** data block) on the north and south boundaries of the simulation region. These leaky boundaries allow flow out of the grid region in response to the difference between the head at the boundary and a specified head at a given distance from the boundary. If flow into the active grid

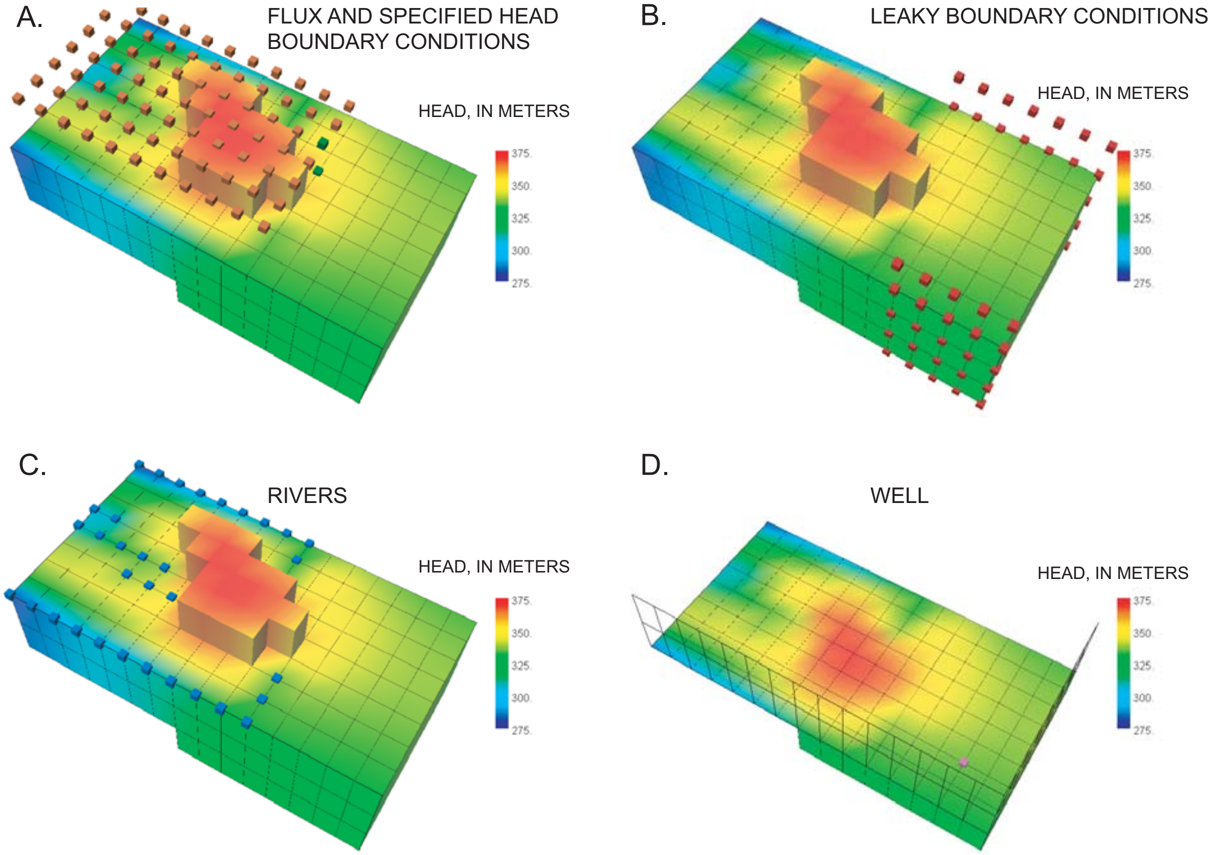


Figure 6.5.— Example 4, locations of boundary conditions: (A) flux boundary-condition nodes are shown in orange and specified-head boundary-condition nodes are shown in green; (B) leaky boundary-condition nodes; (C) river boundary-condition nodes; and (D) node of cell containing the well screen (head field is cropped to show the location of the node). All panels show color rendition of head distribution in the simulation region. View is from the northwest looking to the southeast.

region should occur through these boundaries, the composition of the water will be given by solution 1. All heads, solution compositions, and fluxes are defined to apply from time 0 yr through the end of the simulation.

Unconfined flow with a water-table boundary condition is simulated (**FREE_SURFACE_BC** *true*). One well is included (**WELL** *1* data block) in a zone that has large arsenic concentration; the well is pumped at a small rate, which is constant from 0 yr through the end of the simulation. The well is included to demonstrate how a well could be used to monitor concentrations at a given point in the aquifer. Component concentrations for wells are written to the file *ex4***.xyz.wel**. More detailed information for a node can be extracted from the *ex4***.h5** and *ex4***.xyz.chem** files, but the *ex4***.xyz.wel** file contains fewer data items and thus can be written at a higher frequency without exceeding file-size limitations for the computer operating system.

Initial conditions for head are defined in the **HEAD_IC** data block. Although this initial-condition head distribution is uniform (380 m), the initial steady-state flow calculation will calculate the head distribution to achieve steady-state flow for the specified boundary conditions. Initial conditions for chemistry (**CHEMISTRY_IC** data block) include not only the composition of the water in the active grid region, but also the initial amounts of reactive minerals, the amount and composition of the cation exchanger, and the number of sites and composition of a reactive surface in each cell. These compositions are not defined directly in the flow and transport file, but by index numbers that refer to definitions in the chemistry data file. In this simulation, initial chemical conditions are uniform throughout the active grid region, but heterogeneous distributions may be defined by using multiple zone definitions in the **CHEMISTRY_IC** data block.

The **SOLUTION_METHOD** data block is used to specify that the iterative linear-equation solver is used in this example. The difference equations are formulated with upstream-in-space and backward-in-time weighting (as opposed to centered-in-time and centered-in-space weighting for examples 1–3), which avoids the possibility of oscillations due to spatial and temporal discretization. However, this differencing scheme introduces numerical dispersion into the solution to the flow and transport equations. For this example, the maximum calculated velocity is about 15 m/yr, but typically, velocities are about 1 m/yr or less; the time step is 2,000 yr; and the horizontal discretization is 6,000 m. The numerical dispersivity (eq. D.7 and D.10) is on the order of $\frac{6000}{2} + \frac{1(2000)}{2} = 4,000$ m (see sections D.1.2. Spatial Discretization and D.1.3. Temporal Discretization), which is larger than the physical dispersivity specified in the **MEDIA** data block. Thus, the simulations have an effective dispersivity that is approximately three times the specified dispersivity. To reduce the numerical dispersion, the grid and time step would need to be refined.

The length of simulation (100,000 yr) and uniform time step (2,000 yr) for the transient transport and reaction simulation are defined with **TIME_CONTROL** data block. The length of time simulated was determined by the time needed for the simulated water compositions to resemble the observed distribution of water compositions in the aquifer. Finally, **PRINT_INITIAL** specifies the data for the active grid region that are to be printed after initialization, and **PRINT_FREQUENCY** specifies the transient data to be printed at specified time intervals. The **PRINT_LOCATIONS** data block allows printing to the *prefix***.xyz.chem** and *prefix***.O.chem** files to be included or excluded for specified zones. In this example, the bottom layer of the model (Z=0) is excluded from printing to the *ex4***.xyz.chem** file.

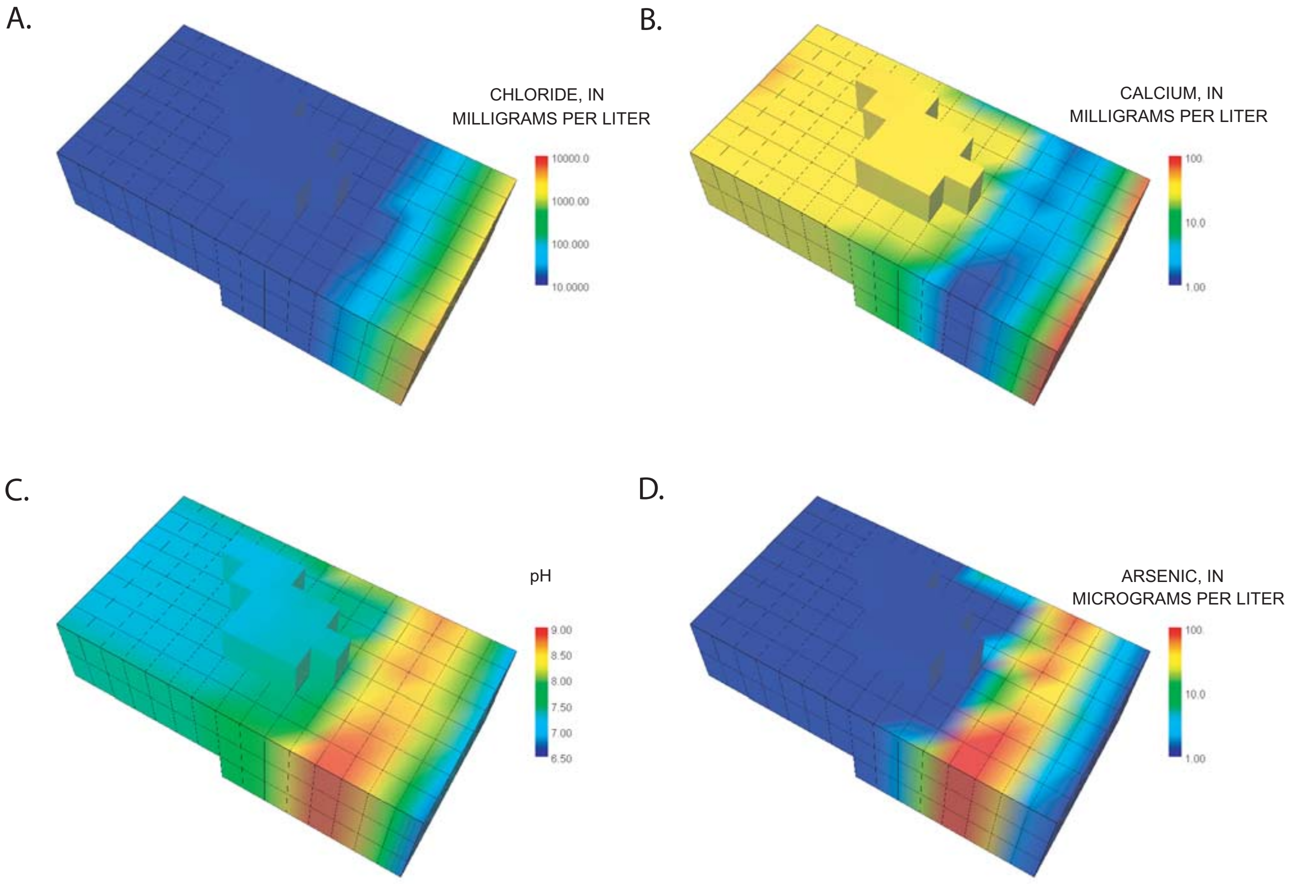


Figure 6.6.—Example 4, distribution of: (A) chloride concentrations, (B) calcium concentrations, (C) pH, and (D) arsenic concentrations after 100,000-year simulation of the evolution of water in the Central Oklahoma aquifer. View is from the northwest looking to the southeast.

6.5.4. Results

Selected results at the end of the 100,000-year simulation are plotted in figure 6.6. The three-dimensional view of the active grid region is from the northwest, which shows a part of the grid region is missing—the inactive grid region—at the eastern end of the bottom layer. Also, much of the top layer of cells are not shown because the cells are dry. The block of nodes that is shown for the top layer is a coarse representation of the water-table mound in the central part of the aquifer.

Figure 6.6A shows the concentration of chloride in the active grid region. After 100,000 years, the large-concentration brine has been flushed from the eastern two-thirds of the active grid region, but concentrations of chloride exceeding1,000 mg/L remain at the western edge of the active grid region.

Initially, the exchange sites in each active cell were in equilibrium with the brine, which caused sodium to be the dominant cation on the exchanger. The yellow areas in the eastern two-thirds of the active grid region in figure 6.6B represent areas where sodium has been removed from the exchanger, and calcium and magnesium are the dominant cations in solution and on the exchanger. In the western one-third of the active grid region, brines have been mostly removed or greatly diluted, but sodium persists as the dominant cation on the exchanger. The blue areas of figure 6.6B correspond to high pH zones in figure 6.6C. The pH is high in these zones because of dissolution of calcite and dolomite, which is enhanced because of the exchange of calcium and magnesium for sodium on the exchanger.

The zones of high pH correspond to large arsenic concentrations in figure 6.6D. The carbonate and exchange reactions drive the pH to higher than 8.5 and arsenic concentrations increase to greater than 100 µg/L in the red zones. In the blue zones of figure 6.6D, the concentrations of arsenic are small, either because the exchanger has been flushed of sodium and the pH has stabilized at about 7.0 to 7.5, or because low pH values associated with the brines still persist in the western part of the active grid region.

The transport calculations show three types of water in the aquifer, the remnants of the initial brine, sodium bicarbonate water, and calcium and magnesium bicarbonate water, all of which are similar to the observed water types in the aquifer. Arsenic concentrations are consistent with values observed in the aquifer. The time of evolution to calcium-magnesium dominated exchange sites is dependent on the number of exchange sites. A larger number of exchange sites will produce lower maximum pH values and lower maximum arsenic concentrations at the end of a 100,000-year simulation. The calculation is sensitive to the stability constant for the surface-complexation reactions for arsenic (**SURFACE_SPECIES** data block in the chemistry data file), the number of surface complexation sites (**SURFACE** data block in the chemistry data file), and the concentration of arsenic in the brine; a decrease in the log K for the predominant arsenic complexation reaction, a decrease in the number of sites, or a decrease in the arsenic concentration in the brine tends to decrease the maximum arsenic concentrations. The large arsenic concentrations are dependent of the flow field; changes in the flow model cause the position of the peak arsenic concentrations to occur at different times and in different locations. Finally, the peaks in arsenic concentrations are affected by the effective dispersivity (specified plus numerical) in the calculation; smaller effective dispersivity tends to increase the maximum arsenic concentrations.

The model results, which are based largely on measured values and literature thermodynamic data, provide a satisfactory explanation of the variation in major ion chemistry, pH, and arsenic concentrations within the Central Oklahoma aquifer. However, the time scale for the simulation (100,000 yr) is much shorter than the length of time over which the aquifer probably evolved (millions of years) and the initial amount of sorbed arsenic in the model is greater than the selective extractions from modern sediments. It is the nature of modeling that even satisfactory results give rise to more detailed questions and suggest new hypotheses about the ground-water system. One of the keys to successful modeling is to know when to stop.

Chapter 7. Notation

This section presents the definitions of the notation used in this report. Units used internally by PHAST and for the output files are given where applicable.

7.1. Roman Characters

$\boldsymbol{A}$ matrix of coefficients of the difference equations (appropriate units)

$\boldsymbol{A}^{-1}$ inverse of the matrix of coefficients of the difference equations (appropriate units)

a activity

$\boldsymbol{b}$ right-hand-side vector of the difference equations (appropriate units)

b_L thickness of the semi-confining layer (m)

b_{Lm} thickness of the semi-confining layer for cell *m* (m)

C_f flow storage factor (m^2)

Cr Courant number (-)

C_s component storage factor (m^3)

c_j total aqueous concentration of component *j* (mol/kgw)

$\hat{c}_j$ total aqueous concentration of component *j* in the source fluid (mol/kgw)

$c_j^{(*)}$ intermediate total aqueous concentration of component *j* (mol/kgw)

$\bar{c}_e$ concentration of solid reactant e (mol/kgw)

$\boldsymbol{D}$ dispersion coefficient tensor (m^2/s)

D_M effective molecular diffusion coefficient (m^2/s)

D_{nos} numerical dispersion coefficient from operator splitting (m^2/s)

D_{ns} numerical dispersion coefficient from spatial discretization (m^2/s)

D_{nt} numerical dispersion coefficient from temporal discretization (m^2/s)

D_{ij} dispersion coefficient tensor element (m^2/s)

$\boldsymbol{e}^{(\nu)}$ error in the solution vector at iteration ν (appropriate units)

f_j gram formula weight of component *j* (g/mol)

g gravitational acceleration (m/s^2)

gfw gram formula weight (g/mol)

h potentiometric head (m)

h_b potentiometric head at the boundary of the simulation region (m)

h_e potentiometric head at the external boundary of the semi-confining layer (m)

h_{em} potentiometric head at the external boundary of the semi-confining layer for cell *m* (m)

h_m potentiometric head at the boundary of the simulation region for cell *m* (m)

I_b biomass inhibition factor (-)

$\boldsymbol{K}$ hydraulic conductivity tensor (m/s)

K equilibrium constant (-)

K_d linear sorption distribution constant [(mol/kgw)/(mol/kgw)];

K_e linear equilibrium sorption coefficient (m^3/kgw)

$K_h(O)$ half-saturation constant for oxygen (mg/L)

$K_h(S)$ half-saturation constant for substrate (mg/L)

K_L hydraulic conductivity of the semi-confining layer (m/s)

K_{Lm} hydraulic conductivity of the semi-confining layer for cell *m* (m/s)

$k_{biomass}$ aerobic biomass inhibition constant (mg/L)

kgw, $kgwater$ kilogram water (kg)

k_m specific death rate or maintenance coefficient for aerobic biomass (d^{-1})

M_j total mass of component *j* in the aqueous phase (kg)

m molality (mol/kgw)

N total number of cells

N_c number of chemical components

N_E number of heterogeneous equilibrium reactions

N_K number of kinetic reactions

N_s number of chemical species

(*n*) index of the discrete time value (superscript)

n_j total number of moles of component *j* in the aqueous phase (mol)

n_p number of cells assigned to processor p

O concentration of dissolved oxygen (mg/L)

P number of processors

P_{CO_2} partial pressure of carbon dioxide (atm)

Pe Peclet number (-)

p pressure (Pa)

Q_{Lm} volumetric flow rate at a leakage boundary for cell *m* (m^3/s)

q source flow rate intensity (m^{3}s^{-1}m^{-3})

q_L volumetric flux at a leakage boundary (m^3 m^{-2} s^{-1})

R retardation factor for linear sorption (-)

$R_{biomass}$ rate of biomass production (mg L^{-1} d^{-1})

R_i rate of formation of component *i* by all kinetic reactions (mol/s)

R_k rate of kinetic reaction k (mol kgw^{-1} s^{-1})

R_{oxygen} limiting rate of oxygen consumption (mg L^{-1} d^{-1})

$R_{oxygen,\,O}$ limiting rate of oxygen consumption due to oxygen availability (mg L^{-1} d^{-1})

$R_{oxygen, S}$ limiting rate of oxygen consumption due to substrate availability (mg L^{-1} d^{-1})

$\boldsymbol{r}^{(\nu)}$ residual vector at iteration ν (appropriate units)

S concentration of substrate (mg/L)

S_{BLm} area of the leakage boundary face of cell *m* (m^2)

S_s storage coefficient (m^{-1})

$Surf$ surface complexation site

$SurfA$ element A sorbed on a surface complexation site

T time for all processors to complete the reaction task (s)

T_i total component concentration in aqueous and solid phases (mol/L) or mass fraction (-)

$\hat{T}_p$ time per cell for the reaction task for processor p (s/cell)

t time (s)

$V_{max}(O)$ maximum rate parameter for oxygen consumption (d^{-1})

$V_{max}(S)$ maximum rate parameter for substrate consumption (d^{-1})

$\boldsymbol{v}$ interstitial velocity vector (m/s)

v_i interstitial velocity component in the *i* coordinate direction (m/s)

w_j mass fraction of component *j* (-)

X concentration of aerobic biomass (mg/L)

$\boldsymbol{x}^{(\nu)}$ solution vector at iteration ν (appropriate units)

Y_O yield coefficient oxygen (L/mg)

Y_S yield coefficient substrate (L/mg)

z elevation coordinate (m)

7.2. Greek Characters

α dispersivity (m)

α_L longitudinal dispersivity (m)

α_{ns} numerical dispersivity from spatial discretization (m)

α_{nt} numerical dispersivity from temporal discretization (m)

α_T transverse dispersivity (m)

α_{T_H} horizontal transverse dispersivity (m)

α_{T_V} vertical transverse dispersivity (m)

γ activity coefficient (kgw/mol)

δ_{ij} Kronecker delta function (-)

δh_m change in potentiometric head at the boundary of the simulation region over the time step for cell *m* (m)

ε saturated porosity (-)

θ weighting factor for time differencing (-)

(ν) iteration counter (superscript)

ρ water density (kg/m^3)

ρ_b dry bulk density of porous medium (kg/m^3)

τ convergence tolerance (-)

υ_p stoichiometric coefficient of a product in a chemical reaction (-)

υ_r stoichiometric coefficient of a reactant in a chemical reaction (-)

$\upsilon_{j,e}^{E}$ stoichiometric coefficient of component j in heterogeneous equilibrium reaction e (-)

$\upsilon_{j,k}^{K}$ stoichiometric coefficient of component j in kinetic reaction k (-)

7.3. Mathematical Operators and Special Functions

Δx cell size (m)

Δt time step (s)

L_f spatial discretization of $\nabla \bullet \boldsymbol{K}\nabla$ (m^{-1} s^{-1})

L_s spatial discretization of $\rho\nabla \bullet \boldsymbol{D}\nabla - \rho\nabla\bullet\boldsymbol{v}$ (kg m^{-3} s^{-1})

∇ gradient operator (m^{-1}) scalars and divergence operator for vectors

$|\ |$ magnitude of a vector

$\|\ \|$ Euclidean norm for a vector or Frobenius norm for a matrix

References Cited

Allison, J.D., Brown, D.S., and Novo-Gradac, K.J., 1990, MINTEQA2/PRODEFA2—A geochemical assessment model for environmental systems—version 3.0 user's manual: Athens, Georgia, Environmental Research Laboratory, Office of Research and Development, U.S. Environmental Protection Agency, 106 p.

Aziz, Khalid and Settari, Antonin, 1979, Petroleum reservoir simulation: Barking, Essex, England, Applied Science, 476 p.

Ball, J.W., and Nordstrom, D.K., 1991, WATEQ4F—User's manual with revised thermodynamic data base and test cases for calculating speciation of major, trace and redox elements in natural waters: U.S. Geological Survey Open-File Report 90-129, 185 p.

Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and van der Vorst, H., 1994, Templates for the solution of linear systems—Building blocks for iterative methods: Philadelphia, PA, Society for Industrial and Applied Mathematics, 112 p.

Bear, Jacob, 1972, Dynamics of fluids in porous media: New York, American Elsevier, 764 p.

Behie, G.A. and Forsyth, P.A., 1984, Incomplete factorization methods for fully implicit simulation of enhanced oil recovery: SIAM Journal of Scientific and Statistical Computation, v. 5, pp. 543-560.

Bethke, C.M., 1996, Geochemical reaction modeling, concepts and applications: New York, Oxford University Press, 397 p.

Burnett, R.D. and Frind, E.O., 1987, Simulation of contaminant transport in three dimensions, 2. Dimensionality effects: Water Resources Research, v. 23, no. 4, pp. 695-705.

Burns, Greg, Daoud, Raja, and Vaigl, James, 1994, LAM—An open cluster environment for MPI, *in* J.W. Ross, ed., Proceedings of Supercomputing Symposium '94: University of Toronto, pp. 379-386.

Charlton, S.R., and Parkhurst, D.L., 2002, PHREEQCI—A graphical user interface to the geochemical model PHREEQC: U.S. Geological Survey Fact Sheet FS-031-02, 2 p.

Cheng, A.H.-D., and Morohunfola, O.K., l993, Multilayered leaky aquifer systems, 1, Pumping well solutions: Water Resources Research, v. 29, no. 8, pp. 2787-2800.

Cohen, S.D., and Hindemarsh, A.C., 1996, CVODE, A stiff/nonstiff ODE solver in C: Computers in Physics, vol. 10, no. 2, p. 138.

Domenico, P.A., 1987, An analytical model for multidimensional transport of a decaying contaminant species: Journal of Hydrology, v. 91, pp. 49-58.

Dongarra, J.J., Duff, I.S., Sorensen, D.C., and van der Vorst, H.A., 1991, Solving linear systems on vector and shared memory computers: Philadelphia, Society for Industrial and Applied Mathematics, 256 p.

Dzombak, D.A., and Morel, F.M.M., 1990, Surface complexation modeling—Hydrous ferric oxide: New York, John Wiley, 393 p.

Eagleson, P.S., 1970, Dynamic hydrology: New York, McGraw Hill, 462 p.

Elman, H.C., l982, Iterative methods for large sparse nonsymmetric systems of linear equations: Yale University, New Haven, Conn., Ph.D., 190 p.

Fletcher, C.A.J., 1991, Computational Techniques for Fluid Dynamics, vol.1, 2nd ed., New York, Springer-Verlag, 401 p.

Foster, I.T., 1995, Designing and building parallel programs: Reading, Mass., Addison-Wesley, 381 p.

Garabedian, S.P., LeBlanc, D.R., Gelhar, L.W., and Celia, M.A., 1991, Large-scale natural gradient tracer test in sand and gravel, Cape Cod, Massachusetts, 2. Analysis of spatial moments for a nonreactive tracer: Water Resources Research, v. 27, no. 5, pp. 911-924.

Gelhar, L.W., Welty, C., and Rehfeldt, K.R., 1992, A critical review of data on field-scale dispersion in aquifers: Water Resources Research, v. 28, no. 7, pp. 1955-1974.

Gropp, W.D., and Lusk, Ewing, 1996, User's guide for MPICH, a portable implementation of MPI: Argonne, Ill.,Mathematics and Computer Science Division, Argonne National Laboratory, ANL-96/6, 70 p.

Gropp, W., Lusk, E., and Skjellum, A., 1995, Using MPI—Portable parallel programming with the message passing interface: Cambridge, Mass., MIT Press, 307 p.

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

Johnson, J.W., Oelkers, E.H., and Helgeson, H.C., 1991, SUPCRT92—A software package for calculating the standard molal thermodynamic properties of minerals, gases, aqueous species, and reactions from 1 to 5000 bars and 0^o to 1000^o C: Livermore, Calif., Earth Sciences Department, Lawrence Livermore Laboratory, 168 p.

Kindred, J.S., and Celia, M.A., 1989, Contaminant transport and biodegradation 2. Conceptual model and test simulations: Water Resources Research, v. 25, no. 6, pp. 1149-1159.

Kipp, K.L., 1987, HST3D—A computer code for simulation of heat and solute transport in three-dimensional ground-water flow systems: U.S. Geological Survey Water-Resources Investigations Report 86-4095, 517 p.

_____, 1997, Guide to the revised heat and solute transport simulator HST3D—Version 2: U.S. Geological Survey Water-Resources Investigations Report 97-4157, 149 p.

Kipp, K.L., Russell, T.F., and Otto, J.S., 1992, D4Z—A new renumbering for iterative solution of ground-water flow and solute-transport equations, *in* Russell, T.F., Ewing, R.E., Brebbia, C.A., Gray, W.G., and Pinder, G.F., eds., Computational methods in water resources IX, v. 1, Numerical methods in water resources: Boston, Computational Mechanics Publications and London, Elsevier Applied Science, pp. 495-502.

Kipp, K.L., Russell, T.F., and Otto, J.S., 1994, Three-dimensional D4Z renumbering for iterative solution of ground-water flow and transport equations, *in* Peters, A., Wittum, G., Herrling, B., Meissner, U., Brebbia, C.A., Gray, W.G., and Pinder, G.F., eds., Computational methods in water resources X, v. 2: Dordrecht, The Netherlands, Kluwer Academic Publishers, pp. 1417-1424.

Kirkner, D.J. and Reeves, H., 1988, Multicomponent mass transport with homogeneous and heterogeneous chemical reactions: Effect of the chemistry on the choice of numerical algorithm, 1, Theory: Water Resources Research, v. 24, no. 10, pp. 1719-1729.

Konikow, L.F., Goode, D.J., and Hornberger, G.Z., 1996, A three-dimensional method-of-characteristics solute-transport model (MOC3D): U.S. Geological Survey Water-Resources Investigations Report 96-4267, 87 p.

Lapidus, L. and Amundson, N.R., l952, Mathematics of adsorption in beds, 6., The effect of longitudinal diffusion and ion exchange in chromatographic columns: Journal of Physical Chemistry, v. 56, pp. 984-988.

Marsily, Ghislain de, 1986, Quantitative hydrogeology—Groundwater hydrology for engineers: New York, Academic Press, 440 p.

Meijerink, J.A. and van der Vorst, H.A., 1977, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix: Mathematics of Computation, v. 31, no. 137, pp. 148-162.

Mosier, E.L., Papp, C.S.E., Motooka, J.M., Kennedy, K.R., and Riddle, G.O., 1991, Sequential extraction analyses of drill core samples, Central Oklahoma aquifer: U.S. Geological Survey Open-File Report 91-347, 42 p.

Parkhurst, D.L., 1995, User's guide to PHREEQC—A computer program for speciation, reaction-path, advective-transport, and inverse geochemical calculations: U.S. Geological Survey Water-Resources Investigations Report 95-4227, 143 p.

Parkhurst, D.L., and Appelo, C.A.J., 1999, User's guide to PHREEQC (Version 2)—A computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations: U.S. Geological Survey Water-Resources Investigations Report 99-4259, 312 p.

Parkhurst, D.L., Christenson, Scott, and Breit, G.N., 1996, Ground-water-quality assessment of the Central Oklahoma aquifer—Geochemical and geohydrologic investigations: U. S. Geological Survey Water-Supply Paper 2357-C, 101 p.

Parkhurst, D.L., and Kipp, K.L., 2002, Parallel processing for PHAST—A three-dimensional reactive-transport simulator, *in* Hassanizadeh, S.M., Schlotting, R.J., Gray, W.H., and Pinder, G.F., eds., Computational methods in water resources—Developments in water science: Amsterdam, Elsevier, no. 47, v. 1, pp. 711-718.

Parkhurst, D.L., Thorstenson, D.C., and Plummer, L.N., 1980, PHREEQE—A computer program for geochemical calculations: U.S. Geological Survey Water-Resources Investigations Report 80-96, 195 p. (Revised and reprinted August, 1990.)

Patankar, S.V., 1980, Numerical heat transfer and fluid flow: New York, Hemisphere Publishing Corporation, 197 p.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., 1989, Numerical recipes: The art of scientific computing: Cambridge, U.K., Cambridge University Press, 702 p.

Price, H.S. and Coats, K.H., 1974, Direct methods in reservoir simulation: Transactions of the Society of Petroleum Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, v. 257, pp. 295-308.

Reed, M.H., 1982, Calculation of multicomponent chemical equilibria and reaction processes in systems involving minerals, gases, and an aqueous phase: Geochimica et Cosmochimica Acta, v. 46, pp. 513-528.

Robson, S.G., 1974, Feasibility of digital water-quality modeling illustrated by application at Barstow, California: U.S. Geological Survey Water-Resources Investigations Report 73-46, 66 p.

_____, 1978, Application of digital profile modeling techniques to ground-water solute-transport at Barstow, California: U.S. Geological Survey Water-Supply Paper 2050, 28 p.

Scheidegger, A.E., 1961, General theory of dispersion in porous media: Journal of Geophysical Research, v. 66, no. 10, pp. 3273-3278.

Steefel, C.I. and MacQuarrie, K.T.B., 1996, Approaches to modeling of reactive transport in porous media, *in* Lichtner, P.C., Steefel, C.I., and Oelkers, E.H., eds., Reactive transport in porous media, Reviews in mineralogy v. 34: Washington, D.C., Mineralogical Society of America, pp. 83-130.

Stoer, J., and Bulirsch, R., 1993, Introduction to numerical analysis, 2nd edition: New York, Springer-Verlag, 660 p.

Sun, Y., Petersen, J.N., and Clement, T.P., 1999, Analytical solutions for multiple species reactive transport in multiple dimensions: Journal of Contaminant Hydrology, v. 35, no. 4, pp. 429-440.

Wang, H.F. and Anderson, M.P., 1982, Introduction to groundwater modeling—Finite difference and finite element methods: New York, W.H. Freeman and Co., 237 p.

Wexler, E.J., 1992, Analytical solutions for one-, two-, and three-dimensional solute transport in ground-water systems with uniform flow: Techniques of Water-Resources Investigations of the U.S. Geological Survey, Book 3, Chapter B7, 190 p.

Wolery, T.J., 1992a, EQ3/EQ6, a software package for geochemical modeling of aqueous systems, package overview and installation guide (version 7.0): Lawrence Livermore National Laboratory Report UCRL-MA-110662, part 1, 66 p.

Wolery, T.J., 1992b, EQ3NR, a computer program for geochemical aqueous speciation-solubility calculations—Theoretical manual, user's guide, and related documentation (version 7.0): Lawrence Livermore National Laboratory Report UCRL-MA-110662, part 3, 246 p.

Yeh, G.T., and Tripathi, V.S., 1989, A critical evaluation of recent developments in hydrogeochemical transport models of reactive multichemical components: Water Resources Research, v. 25, no. 1, pp. 93-108.

Zhu, Chen, and Anderson, Greg, 2002, Environmental applications of geochemical modeling: Cambridge, UK, Cambridge University Press, 284 p.

Appendix A. Three-Dimensional Visualization of PHAST Simulation Results

Results of PHAST simulations may be visualized by using a modified version of the Model Viewer software (Hsieh and Winston, 2002), which runs only under Windows operating systems. Model Viewer has capabilities to rotate, translate, and slice a three-dimensional visualization of PHAST simulation results. The modified Model Viewer software and documentation (PDF file) of the original Model Viewer (without description of any features related to PHAST) are included in the distribution package for PHAST.

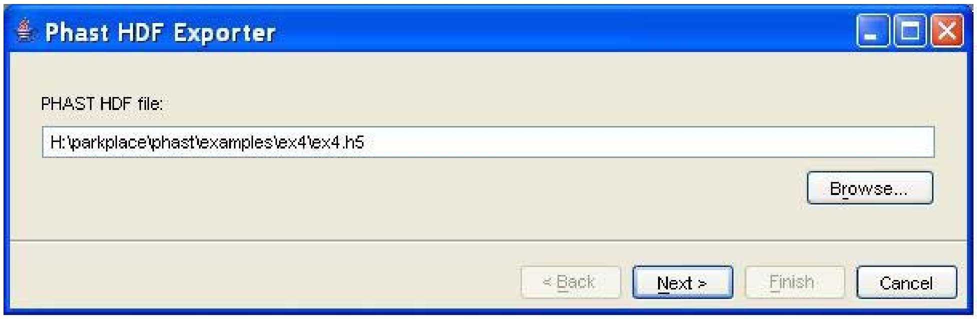
The modified version of Model Viewer displays data that is read from the PHAST HDF output file (*prefix***.h5**). The data written to the HDF output file may include heads, velocities, and chemical data as well as boundary-condition features. The scalar quantities for chemistry and head can be visualized as solid representations (see fig. 6.6 for examples) or isosurfaces (Hsieh and Winston, 2002). Interstitial ground-water velocities can be visualized as vectors at each model node. In addition, model features, including specified-head, specified-flux, leaky, river, and well boundary nodes can be as different colored nodes visualized with Model Viewer.

The active grid region can be dissected parallel to coordinate axes to visualize the interior structure of scalar fields. The color range (color bar) corresponding to the numerical range of scalar values can be adjusted. Options exist to display grids, axes, and boundaries of the grid region, and to change many other aspects of the visualizations. For any variable that can be visualized, an animation may be displayed that shows a scalar (or vector) field for each time step that has been saved to the HDF output file. Visualizations can be exported to graphics files to include in other documents, and animations can be exported to make movies.

Model Viewer can be run from the “Start” menu of Windows, by a shortcut written to the desktop, or by double clicking on the executable file in a file folder. By default in the PHAST distribution, the Model Viewer executable is located in the directory C:\Program Files\Usgs\phast-x.x\Model Viewer\bin\modview.exe, where “x.x” represents the version number of PHAST.

Appendix B. Using PHASTHDF to Extract Data from the HDF Output File

The HDF output file (*prefix***.h5**) is in compressed, binary format (http://hdf.ncsa.uiuc.edu), which cannot be read by text editors. The contents of the HDF output file can be extracted with the HDF utility *h5dump* and written to an ASCII text file, but the file format produced by the utility is not convenient for use with other programs. PHASTHDF is a graphical, interactive utility program included with the PHAST distribution that is used to extract results from the HDF output file. PHASTHDF is written in the Java language, which can be run on any type of computer that has a Java Runtime Environment. PHASTHDF generates an ASCII text file, referred to as the export file, which is in the same format as *prefix***.xyz** files. The format of an export file is suitable for importing into spreadsheets, plotting programs, statistics programs, text editors, or for manipulation by scripting languages such as PERL or AWK. Export files generated by PHASTHDF contain a set of lines—one line for each active node—for each time step that is selected; time steps for inclusion in the export file are selected from interactive screens of the PHASTHDF program. The lines of the export file include columns for X, Y, and Z coordinates of a node, the time, a cell-saturation identifier that indicates whether the cell for the node is dry (0) or not (1), followed by columns for the scalar and velocity data that are selected from the appropriate screens of the PHASTHDF program.



PHAST HDF file:

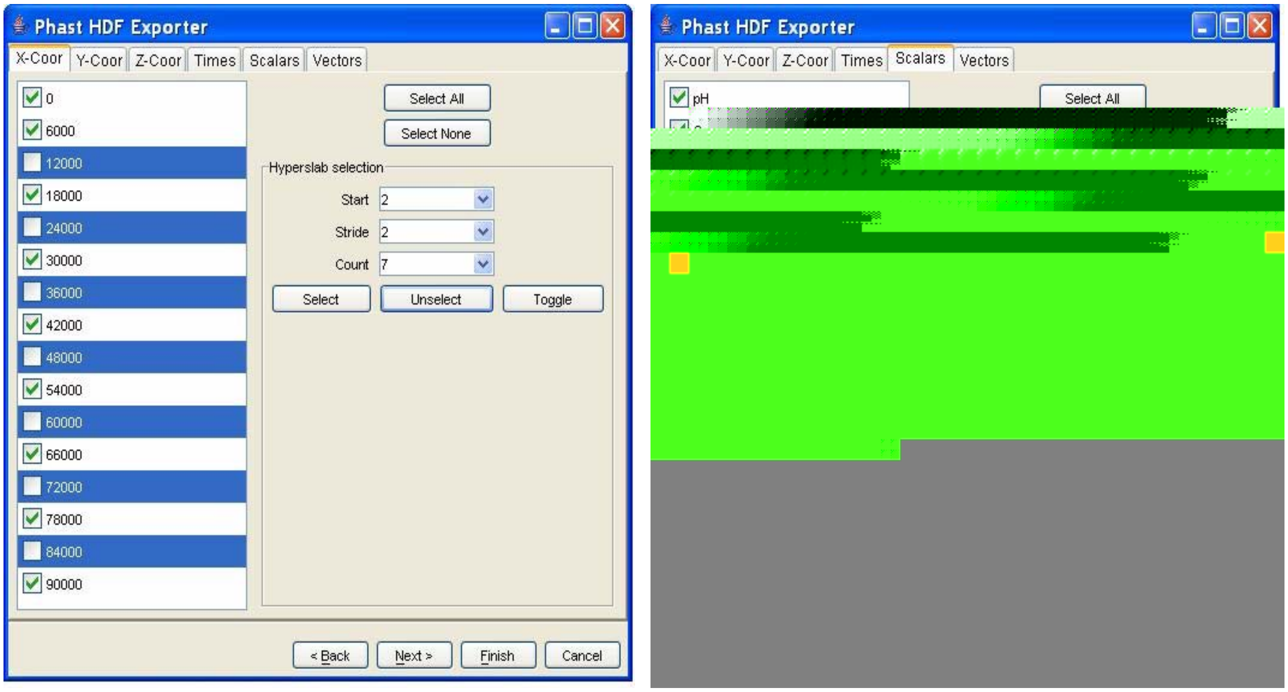
H:\parkplace\phast\examples\ex4\ex4.h5

Figure B.1.—PHASTHDF input screen for selection of HDF output file from which data are to be extracted.

Running PHASTHDF consists of three steps: (1) selecting the HDF output file from which to extract data (fig. B.1), (2) selecting the X, Y, and Z coordinates of node locations, the time steps, the scalars, and vectors to be extracted (fig. B.2), and (3) specifying the name of the export file (fig. B.3).

Figure B.2 shows the second screen for PHASTHDF, which allows selection of the nodes, time steps, scalars, and vectors that will be extracted from the *prefix***.h5** file. This screen has five tabs that change the content of the screen. By default, all data contained in the HDF file—all scalars and vectors, all nodes, and all time steps—are selected for extraction. Extracting the complete set of data may generate a large text file, which could exceed the file-size limitation of the computer operating system. Normally, the screens corresponding to the five tabs are used to specify a subset of data that can be exported to a text file of manageable size.

A hyperslab is a set of nodes that is selected by specifying individual X, Y, and Z node locations. The screen on the left of figure B.2 corresponds to the “X-Coor” tab. In the figure, a subset of X coordinates has been selected by using the “Hyperslab selection” check boxes. Beginning with the third coordinate (Start=2 indicates the offset from the first coordinate), every other node (Stride=2) for a total of seven nodes (Count=7) has been unselected. Nodes with these X coordinates will not appear in the export file. Selecting the Y-Coor or Z-Coor tab produces a screen for the respective coordinate that is identical in function to the screen for the X-Coor tab. The HDF output file contains a series of time planes; the “Times” tab allows selection of a subset of time planes to be extracted and written to the export file.



Phast HDF Exporter

X-Coor | Y-Coor | Z-Coor | Times | Scalars | Vectors

☑ 0
☑ 6000
☐ 12000
☑ 18000
☐ 24000
☑ 30000
☐ 36000
☑ 42000
☐ 48000
☑ 54000
☐ 60000
☑ 66000
☐ 72000
☑ 78000
☐ 84000
☑ 90000

Hyperslab selection

Start 2
Stride 2
Count 7

Phast HDF Exporter

X-Coor | Y-Coor | Z-Coor | Times | Scalars | Vectors

☑ pH

Figure B.2.—PHASTHDF input screens associated with the first and fourth of five tabs that allow selection of model nodes and chemical data to be extracted from the HDF output file.

The screen on the right of figure B.2 corresponds to the “Scalars” tab. Selecting this tab produces a list of all of the scalar data that have been stored in the HDF file. It is possible to include or exclude each item of the list individually. For every item that is checked, a column of data will be written to the export file. In this example, all of the data except fluid head will be written to the export file. Similarly, the “Vector” tab allows selection of any combination of the X, Y, and Z components of the interstitial velocities at nodes to be written to the export file.

The final screen of PHASTHDF is used to specify the name for the export file (fig. B.3). When the “Finish” button is clicked, the selected data will be written to the export file.

Figure B.3.—PHASTHDF input screen used to specify the file name for the export file.

Appendix C. Parallel-Processing Version of PHAST

The numerical simulation of ground-water flow and solute transport with geochemical reactions is computationally intensive and often requires the use of large vector or massively parallel computers. Three-dimensional simulation of reactive transport with dozens of components and tens of thousands of grid nodes can take days of computing time on desktop workstations or personal computers (PCs). Speeding up multi-component reactive-transport simulation in a workstation or PC environment makes reactive-transport simulations accessible to a much wider group of ground-water modelers.

One class of parallel computer that is potentially available to ground-water modelers is a cluster of workstations or PCs on a private, high-speed, local area network (LAN). If the Linux operating system is used on the cluster, the cluster is known as a Beowulf cluster. Clusters of computers running Unix, Macintosh, and Windows operating systems also are possible. Another class of parallel computer that is readily available is a single workstation, server, or PC containing multiple (2–8) processors. A parallel version of PHAST has been written that can be run on all of these classes of parallel systems.

C.1. Parallelization of PHAST

Typically, the CPU time spent on the chemistry calculation is at least twice the time spent in the flow and transport calculation, and usually the chemistry calculation requires 90 to 99 percent of the CPU time. A PHAST demonstration problem run on a single processor required 98 percent of the simulation time for the reaction-chemistry calculations; only 2 percent of the time was used for flow and transport calculations (Parkhurst and Kipp, 2002). Thus, parallelization has been applied to the reaction part of the simulator because it could lead to the greatest reduction in CPU time.

The strategy for parallelization of PHAST is based on the parallel machine model of Foster (1995), which is a number of von Neumann computers (nodes) coupled by an interconnection network. Each computer executes its own program (process) on its own local data and communicates with other computers by sending messages over the network or within the parallel workstation. The memory is distributed among the processors. This configuration forms a distributed-memory multiple-instruction, multiple-data (MIMD) computer.

The parallel programming model (Foster, 1995) consists of tasks connected by message queues called channels. One or more tasks can be mapped to a processor node. The Message Passing Interface (MPI) (Gropp and others, 1995) variant of the task/channel model, as implemented by the Local Area Multicomputer (LAM) (Burns and others, 1994) project and MPICH (Gropp and Lusk, 1996), were used for communication. Message passing programs create a number of processes at program startup. Each process executes the same program on its own data following a single program multiple data (SPMD) model. A set of tasks is distributed among the available processes.

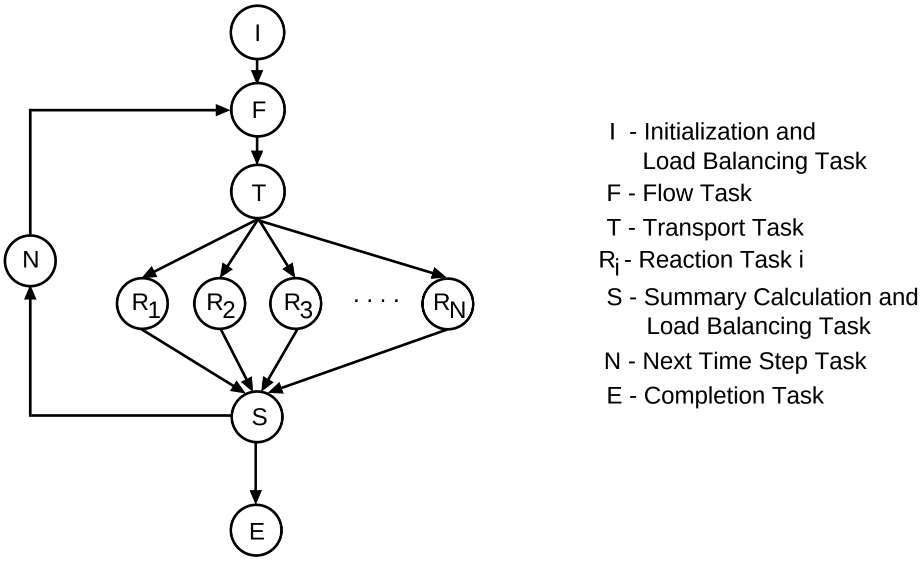


Figure C.1.—Flow-control diagram for parallel-processing version of PHAST.

For the parallel version of PHAST, a fixed number of processes are started at initialization of a simulation to execute the necessary tasks. The tasks for a PHAST simulation include initialization, flow, transport, cell-by-cell reaction, summary calculation and load balancing, and completion. The flow and reactive-transport tasks are logically separate and sequential in constant-density flow. The operator-splitting algorithm in PHAST further separates functions into transport tasks, one transport calculation for each component, and chemical-reaction tasks, one for each cell in the active grid region. The data requirements for the reaction tasks are independent for each cell.

Figure C.1 shows the flow-control graph (Dongarra and others, 1991) for the partitioning of tasks for a single time step. The nodes of the graph represent tasks and the arrows represent execution dependencies. Data dependencies do not exist for a task with no incoming arrows. The parallel structure of the reaction tasks is apparent. Additional time steps are indicated by the feedback loop. Data communication is required between the flow, transport, reaction, and summary calculation and load-balancing tasks. The parallel version has a communication pattern that is

- Local with each reaction task communicating only with the transport task;
- Structured with a partitioning that has dynamic mapping;
- Static with reaction and transport tasks as fixed communication partners; and
- Synchronous with communication after the sequential tasks of transport, reaction, and summarization.

Parallel communication includes scattering from the transport task to the reaction tasks and gathering from the reaction tasks to the summary calculation task. Additional communication is needed for the load-balancing task, which is included in the summary calculation task. The flow and transport tasks are combined and are run on a single, root processor. Cell-reaction tasks are combined into sets of reaction tasks, which are mapped by the root process—one set to each process (including the root process). Flexibility is maintained by allowing the size of each set of reaction tasks to vary for each time step.

The root process determines the sets of reaction tasks and the mapping of the sets to the processes by a load-balancing algorithm that has a static and a dynamic component. The static component of the algorithm consists of randomly ordering the cells during the initialization task. Much of the computational load may be in a reaction front located in a set of contiguous cells. The randomization is intended to distribute the cells from computationally intensive areas uniformly among all of the processes. To adjust for the heterogeneity in processor capability, a small calculation is run initially on each processor to measure its relative speed. The results are used to establish the initial number of cells in each set of reaction tasks.

The dynamic component of the algorithm consists of rebalancing the computational workload at the end of each time step. The sets of reaction tasks are revised by adjusting the number of cells assigned to each reaction-task set. The revision of the number of cells in each set causes cell calculations to be moved from one process to another to equalize the reaction-task computation time for each process. When a cell is moved from one process to another, all of the chemical information for a cell is sent from the old process to the new process by data messages. The number of cells to assign to each process is estimated by finding a set of n_p integers such that all processes require the same amount of time, T, to complete their set of reaction tasks:

$$n_p \hat{T}_p = T; \text{ for } p = 1, ..., P; \text{ subject to the constraint that } \sum_{p=1}^{P} n_p = N, \tag{C.1}$$

where n_p is the number of cells for process p; $\hat{T}_p$ is the average time per cell for a reaction task for process p (the average time to complete one cell calculation during the last chemical calculation); P is the number of processes; and N is the total number of cells. Recognizing that

$$n_p = n_1 \frac{\hat{T}_1}{\hat{T}_p}, \text{ for all } p = 1, ..., P \ , \tag{C.2}$$

and substituting into $\sum_{p=1}^{P} n_p = N$, the following relation is obtained:

$$n_1 = \frac{N}{\sum_{p=1}^{P} \frac{\hat{T}_p}{\hat{T}_1}}, \qquad \text{(C.3)}$$

which determines the number of cells to assign to process 1. The number of cells to assign to all other processes can be determined by equation C.2, with some accommodation to account for the fact that each n_p must be an integer. An optimal load balance will satisfy equations C.1, and reduce the idle time for each of the processors to zero during the reaction-task calculation. A damping factor is used to eliminate flutter in rebalancing; only half of the number of cells to be remapped are actually transferred to a different process at any rebalancing step. The time to rebalance the load using this algorithm, including communication time to move cell information, is minimal, typically consuming much less than 1 percent of the total computation time.

The parallel algorithm for PHAST is a coarse-grain parallelization in that the ratio of computation time to communication time is large. Overall efficiency, which includes the communication time of distributing data to and gathering data from each process, frequently is in the range of 80 to 95 percent. This efficiency implies that a network of 10 identical single-processor computers can run the same simulation about 8 to 9 times faster than a single computer with one processor.

The results of a parallel simulation should be nearly identical to the results of the same simulation run on a single processor. However, slight differences in the results between the parallel and single-processor versions are caused by rounding and truncation of numbers in the message passing between the parallel processes and by differences in implementation of floating-point arithmetic among the processors of a cluster. The chemistry data, flow and transport, and thermodynamic database files used to run the single-processor version are the same as those used to run the parallel version, and the same output files are produced. The only differences in the content of the output files are some additional data related to the efficiency of the parallel processing that are written to the screen and to the log file.

C.2. Running the Parallel Version

To be able to run the parallel version of PHAST, it is necessary to install a version of MPI on a network of computers (or a single multiprocessor computer). LAM and MPICH are public domain versions of MPI and other proprietary versions of MPI are available (Burns and others, 1994, http://www.lam-mpi.org; http://www-unix.mcs.anl.gov/mpi/mpich). A compiled parallel version of PHAST for LAM on Linux computers and MPICH (Gropp and Lusk, 1996) for Windows NT, Windows 2000, and Windows XP computers (other Windows operating systems cannot run MPICH) is provided in the distribution packages. It also is possible to run MPI on a heterogeneous cluster of Unix computers (SUN, IBM, Linux, and others). This requires building PHAST (compiling and linking to libraries) for each of the operating systems on the network that are used for PHAST simulation.

Installing and running MPI can be difficult. Properly setting up the communication that is needed among computers, including the sets of network files that each computer can access, is not a simple task. It is beyond the scope of this manual to describe all of the ways that MPI can be installed and configured. However, two examples are provided to demonstrate how the parallel version of PHAST can be run under some simplifying assumptions on a Beowulf cluster and a cluster of PCs running Windows.

For the first example, consider a Beowulf cluster of Linux operating system computers on which LAM has been installed and an MPI test problem runs correctly. It is assumed that all of the computers are able to read from and write to a directory named */z/wulf1/home/geek/phast/examples/ex4*. It is further assumed that the executables for PHASTINPUT and the LAM parallel version of PHAST (found in */bin* of the installation directory) are copied into this directory with the names *phastinput* and *phast-lam*. A file named *hosts*, created by the user, is in this directory, which lists the names of a number of computers on the network that will be used for the simulation. Finally, the input files *ex4.chem.dat*, *ex4.trans.dat*, and *phast.dat* also are in the directory. The parallel version of PHAST for LAM can be run with the following sequence of commands:

```
cd /z/wulf1/home/geek/phast/examples/ex4
phastinput ex4
lamboot hosts
mpirun N -wd /z/wulf1/home/geek/phast/examples/ex4 phast-lam
wipe hosts
```

The *phastinput* command runs PHASTINPUT, which reads the data input files and generates an intermediate file named *Phast.tmp*, the same as for single-processor execution. The *lamboot* command initiates the LAM daemon on each computer listed in the *hosts* file. The *mpirun* command initiates the program *phast-lam* on each computer for which a daemon was initiated (*N* option). Each process (program) reads from and writes to the working directory (*-wd* option). Note that the program *phast-lam*, which is run on each computer, also is found in the working directory. The final command (*wipe*) removes the LAM daemon from each computer. Many temporary files are written by each process into the shared working directory, but will be deleted if the job completes successfully. The procedure for running PHAST with LAM can be simplified by the use of shell scripts and LAM schema files, but use of these files is not discussed in this documentation.

For the second example, consider a network of computers running the Windows NT, Windows 2000, or Windows XP operating system. MPICH has been installed on the computers and an MPI test problem runs correctly. In the process of running a test problem, the computers available to MPICH will be defined by the user when running the MPICH Configuration Tool. It is assumed that all of the computers are able to read from and write to a shared directory named *\\hostname\mpich_share*, which points to the local directory *c:\phast\examples\ex4* residing on the computer named *hostname*. It is further assumed that the executables for PHASTINPUT (named *phastinput.exe*), the MPICH parallel version of PHAST (named *phast-mpich.exe*), *zlib.dll*, and *hdf5dll.dll* (all found in */bin* of the installation directory) are copied into this shared directory. Finally, the input files *ex4.chem.dat*, *ex4.trans.dat,* and *phast.dat* also are in the directory. The parallel version of PHAST for MPICH can be run on *hostname* with the following sequence of commands entered in a DOS command-line window:

```
cd c:\phast\examples\ex4
phastinput ex4
mpirun -np 10 -dir \\hostname\mpich_share \\hostname\mpich_share\phast-mpich
```

The *phastinput* command runs PHASTINPUT, which reads the data input files and generates an intermediate file named *Phast.tmp*, the same as for single-processor execution. The *mpirun* command initiates ten instances of the program *phast-mpich* distributed among the computers for which MPICH has been configured (*-np* 10). Each process (program) reads from and writes to the shared directory. Many temporary files are written into the shared directory for each process, but will be deleted if the job completes successfully. The process of running PHAST with MPICH may be simplified by writing batch scripts, adding directories to the path environmental variable, and by defining additional environmental variables, but use of these features is not discussed in this documentation.

Appendix D. Theory and Numerical Implementation

PHAST is a simulator of multi-component, reactive transport in three dimensions derived from the flow and solute-transport model HST3D and the geochemical-reaction model PHREEQC. Multi-component solute transport and chemical reactions are simultaneous processes. However, in PHAST, the solute-transport and chemical-reaction calculations are separated by operator splitting and performed by a sequential solution approach without iteration.

The governing equations for a reactive solute-transport simulator are a set of partial differential equations describing ground-water flow and solute transport for each aqueous component, a set of non-linear algebraic equations describing equilibrium chemical reactions, and a set of ordinary differential equations describing rates of kinetic chemical reactions. Multiple aqueous chemical constituents are present in a natural or contaminated ground-water flow system. A subset of the aqueous chemical constituents is chosen to be components. The number of components required is the minimum number of species from the total set necessary to describe the chemical composition of the system (Reed, 1982). Components as defined for this simulator are not the same as thermodynamic components used in connection with phase rule calculations (see Zhu and Anderson, 2002, pp. 51-55). A set of aqueous species is chosen as master species (one for each component), such that all other species (aqueous and solid) can be formed by chemical reactions from these master species. The total aqueous concentrations of the chemical components are the dependent variables of the transport equations, and the activities of the master species and the number of moles of solid reactants are the dependent variables for the chemical equations.

This appendix covers the essential equations, features, and methods used in the PHAST simulator including: the assumptions underlying the flow and transport equations and their discretization; the equilibrium and kinetic reaction equations; the property functions and transport coefficients; the well-source conditions; the boundary and initial conditions; the operator-splitting approach and sequential solution of flow, transport, and chemical equations; accuracy of solutions to the coupled equations; the global balance calculation; and the nodal velocity calculation. The ground-water flow and component-transport differential equations and boundary conditions are summarized, and those that are implemented differently than in HST3D are described in detail. No additions or modifications to the numerical methods of PHREEQC were necessary for the chemical-reaction calculations of PHAST; minor modifications were made to the output functions of PHREEQC. The PHREEQC manual (Parkhurst and Appelo, 1999) covers the numerical implementation details for the chemical-reaction calculations, and only a brief summary will be included here.

D.1. Flow and Transport Equations

The general saturated ground-water flow and component solute-transport equations solved by the PHAST simulator are based on those of the HST3D simulator described in Kipp (1987, 1997). In HST3D, the equations were written for variable fluid density and viscosity and for a single component solute-transport equation including linear sorption and decay. For PHAST, the simulator has been restricted to constant density and viscosity and the linear sorption and decay terms are eliminated. The following assumptions determine the constitutive equation set for PHAST:

- Ground water fully saturates the porous medium within the region of ground-water flow, referred to as the simulation region.
- Ground-water flow is described by Darcy's law.
- Ground water and the porous medium are compressible under confined flow conditions.
- Ground water and the porous medium are incompressible under unconfined flow conditions.
- Ground water has constant, uniform density and viscosity.
- Isothermal conditions exist in the simulation region with respect to flow and transport.
- Porosity and permeability are functions of space.
- The coordinate system is orthogonal and aligned with the principal directions of the permeability tensor so that this tensor is diagonal for anisotropic media.
- The coordinate system is right-handed with the Z-axis pointing vertically upward.
- Dispersive-mass fluxes of the bulk fluid from spatial-velocity fluctuations are neglected.
- Contributions to the total fluid mass balance from pure-solute mass sources within the region are neglected.
- Multiple solute components are present.

- Chemistry is defined by a set of chemical reactions, both equilibrium and kinetic.
- Changes in the ratio of solid reactants to mass of water caused by porosity changes in confined systems are neglected.
- Chemical reactions do not induce porosity or permeability changes.
- Dispersivity values are the same for each chemical component.

These assumptions simplify the flow and transport equations relative to those in HST3D, but expand the chemical equation set to include multiple components and chemical reactions.

The ground-water flow equation used in PHAST is given in Kipp (1987). The dependent variable internal to the simulator is pressure; however, given the assumptions of PHAST, equivalent equations can be written by using potentiometric head as the dependent variable. The term “pressure” will be used in connection with some of the discussion of the flow equation and boundary conditions, although potentiometric head is used for data input and output. The ground-water flow equation using potentiometric head (Bear, 1972) is as follows:

$$S_s\frac{\partial h}{\partial t} = \nabla \bullet \boldsymbol{K} \nabla h + q, \qquad \text{(D.1)}$$

with $$h = \frac{p}{\rho g} + z, \qquad \text{(D.2)}$$

where S_s is the storage coefficient (per meter, m^{-1}); h is the potentiometric head (m); t is time (s); ∇ is the gradient operator (m^{-1}) for scalars and the divergence operator for vectors; $\boldsymbol{K}$ is the hydraulic conductivity tensor (m/s); q is the source flow rate intensity ($m^3 s^{-1} m^{-3}$); p is the pressure (Pa); ρ is the water density (kg/m^3); g is the gravitational acceleration (m/s^2); and z is the elevation coordinate (m).

Darcy’s law relates the specific discharge field to the gradient of the potentiometric head. Thus, the interstitial velocity is

$$\boldsymbol{v} = -\frac{\boldsymbol{K}}{\varepsilon}\nabla h, \qquad \text{(D.3)}$$

where $\boldsymbol{v}$ is the interstitial velocity (m/s) and ε is the saturated porosity (-).

The equation for conservation of a single solute aqueous component is based on the following assumptions:

- Pressure diffusion is neglected.
- Solute transport by local, interstitial, velocity-field fluctuations and mixing at pore junctions is described by a hydrodynamic dispersion-coefficient tensor.
- Forced diffusion by gravitational, electrical, and other fields is neglected.
- No pure diffusive solute sources occur in the fluid or porous matrix phases.

The general form of the reactive solute-transport equation for component *j* in the aqueous phase with both equilibrium and kinetic reactions (modified from Bear, 1972; Kirkner and Reeves, 1988) is as follows:

$$\frac{\partial}{\partial t}(\varepsilon\rho c_j) = \nabla \bullet \varepsilon \boldsymbol{D} \nabla \rho c_j - \nabla \bullet \varepsilon \boldsymbol{v} \rho c_j - \sum_{e=1}^{N_E} \upsilon^E_{j,e} \frac{\partial}{\partial t}(\varepsilon\rho\bar{c}_e) + \sum_{k=1}^{N_K} \upsilon^K_{j,k} \varepsilon\rho R_k + q\varepsilon\rho\hat{c}_j \,; \quad j = 1, \ldots, N_c, \qquad \text{(D.4)}$$

where c_j is the total aqueous concentration of component *j* (mol/kgw); $\boldsymbol{D}$ is the dispersion-coefficient tensor (m^2/s); N_E is the number of heterogeneous equilibrium reactions; $\upsilon^E_{j,e}$ is the stoichiometric coefficient of component *j* in heterogeneous equilibrium reaction *e* (-); $\bar{c}_e$ is the concentration of solid reactant e (mol/kgw); N_K is the number of kinetic reactions; $\upsilon^K_{j,k}$ is the stoichiometric coefficient of component *j* in kinetic reaction *k* (-); R_k is the rate of kinetic reaction k [mol kgw^{-1} s^{-1} (moles per kilogram water per second)]; $\hat{c}_j$ is the total aqueous concentration of component *j* in the source water (mol/kgw); and N_c is the number of chemical components in the system. Equation D.4 is written for each aqueous component. Note that the reaction rate, R_k, may be a function of aqueous species of components other than j and the R_k and $\bar{c}_e$ may appear in multiple transport equations. Thus, the reaction rate and the rate of solid species formation provide terms that couple the set of reactive-transport differential equations D.4.

The classical method of finite differences (Patankar, 1980) is used to discretize the partial-differential equations and boundary conditions in space and time. Several options are available for the differencing, which are discussed in the following sections. The selection of components for PHAST also is discussed.

D.1.1. Components

For PHAST, the term component is used to refer to an entity that has an aqueous mole-balance equation in a PHREEQC calculation. PHREEQC has an aqueous mole-balance equation for each element that is present in the system, including hydrogen and oxygen, and a mole-balance equation that conserves charge. Therefore hydrogen, oxygen, and charge imbalance are transported, plus any additional elements. It is possible to extend the definition of an "element" to include an organic compound, for example atrazine, or a single redox state of an element, for example ferrous iron, by appropriate definitions in **SOLUTION_MASTER_SPECIES** and **SOLUTION_SPECIES** data blocks. It also is possible to formulate mass balance on individual isotopes through the use of the isotope capabilities of PHREEQC, for example carbon-13. If these definitions are made, then atrazine, ferrous iron, or carbon-13 could be components in a PHAST simulation.

A component may be present in the form of many different aqueous species; however, the transport equation is applied to the total aqueous concentration (sum of moles in all aqueous species) for each component. As in HST3D (Kipp, 1987; 1997) concentration units for the transport equation in PHAST are mass fraction. Mass fraction is calculated for each component as $w_j = \frac{M_j}{1 + M_j} = \frac{0.001 n_j f_j}{1 + 0.001 n_j f_j}$, where w_j is mass fraction of component *j* (-), M_j is the total mass of component *j* in the aqueous phase (kg), n_j is the total number of moles of component *j* in the aqueous phase (mol), and f_j is the gram formula weight of component *j* (g/mol). Note that the masses of other components do not affect the mass fraction of a given component and that the mass of water is assumed to be exactly 1 kg. This formulation is compatible with the single-component transport formulation inherited from the program HST3D (Kipp, 1987; 1997) and is necessary to conserve mass between the transport calculations and the reaction calculations.

The components hydrogen and oxygen are handled in a special way. The two quantities that are transported are the "excess" hydrogen and oxygen relative to 1 kg of water. In other words, the mass of excess hydrogen is calculated by taking the total mass of hydrogen in solution and subtracting the mass of hydrogen in 1 kg of water, similarly for oxygen. PHREEQC allows the mass of water to vary, so the total mass of water in a solution may not be identically 1.0 kg, but variable water mass is not consistent with the constant density assumption for the flow and transport equations. To avoid mass-balance errors, the transport calculations use the mass of each component in solution relative to an assumed 1 kg of water. However, the chemical calculations use the total mass of water, including water that is produced or consumed by reactions. The chemical calculations reconstitute the total hydrogen and oxygen, and implicitly the total mass of water, by adding the excess hydrogen and oxygen from transport to 1.0 kg of water.

D.1.2. Spatial Discretization

Although the spatial discretization is described by Kipp (1987, 1997), a brief summary of grid construction and terminology is given here for reference. The simulation region is a bounded volume of three-dimensional space, often irregularly shaped, over which the flow and transport equations are solved. A right-handed, three-dimensional Cartesian coordinate system is applied to the simulation region with the Z-axis pointing in the opposite direction of the gravitational force. A cylindrical-coordinate system is not available in PHAST. Spatial discretization is done by constructing a grid of node points that covers the simulation region (fig. D.1). This grid is formed by specifying the distribution of node locations in each of the three coordinate directions. Every combination of X, Y, and Z nodal coordinate values gives the location of all the node points in the three-dimensional grid. The grid occupies a rectangular prism of space, referred to as the grid region. The grid region completely contains the simulation region. A set of nodes in the grid is selected to approximate the shape of the simulation region and nodes in the set are referred to as boundary nodes. Boundary nodes and nodes internal to the simulation region form the set of active nodes. The remaining nodes of the grid are inactive nodes. Boundaries of the active grid region are planes that are normal to one of the coordinate directions and connect boundary nodes. Simulation results are computed for all active nodes.

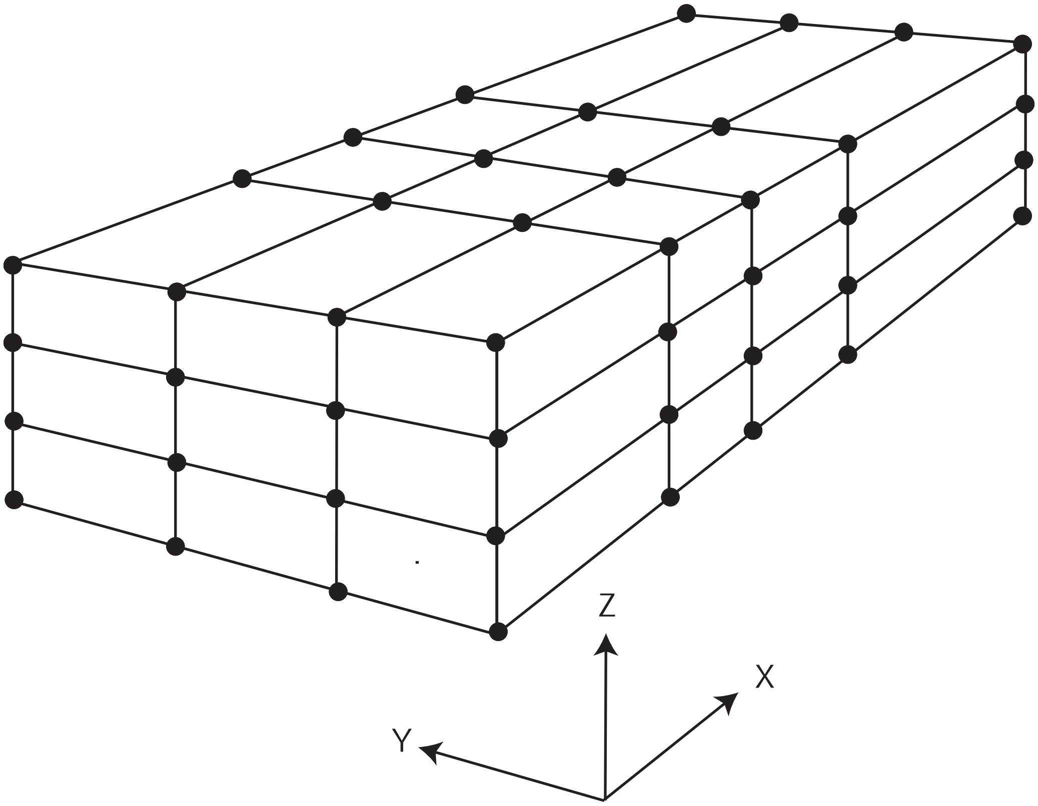


Figure D.1.—Grid of node points that covers the simulation region.

The grid region can be partitioned into elements. An element is the volume of a rectangular prism defined by eight nodes that are adjacent in the grid, with one node at each corner of the element (fig. 4.1). The set of elements fills the entire grid region. An element is referred to as active if all of its nodes are in the active grid region, either internal or on the boundary. An element is referred to as inactive if any of its nodes are in the inactive grid region.

The active grid region also can be partitioned into cells (fig. 4.2). A cell is the volume associated with an active node. For interior nodes, a cell is a rectangular prism with cell boundaries that are planes bisecting the distance between adjacent node points. Interior cells are referred to as full cells and contain portions of eight elements that meet at the cell node. Cells associated with boundary nodes are referred to as fractional cells. From one to seven active elements may meet at a boundary node; thus, a fractional boundary cell may consist of various fractions of the full cell (1/8 to 7/8). Parts of one to three boundary faces of a fractional cell do not lie between adjacent node points, but, rather, are boundary planes of the active grid region that intersect the cell node (fig. 4.2). Each of the grid region boundary faces of a given fractional cell is perpendicular to a different coordinate axis. Cells are not defined for inactive nodes.

The type of grid that is described here and is used by PHAST is called a point-distributed grid. A common alternative method for constructing the grid is to specify the locations of the planes that form the cell boundaries. Their intersections form the cells, and the node points are located in the center of each cell. This type of grid is called a cell-centered or block-centered grid.

An element has uniform porous-media properties: porosity, specific storage coefficient, hydraulic conductivity, and dispersivity. However, a cell contains portions of one to eight active elements (fig. 4.4); thus it can contain as many as eight different values for a porous-media property. The finite-difference equations are formed by integrating the partial differential equations over each cell volume, as explained in Kipp (1987). In this integration, the variations in porous-media properties within a cell are arithmetically averaged in proportion to their active element volume in the cell (fig. 4.4). Fluid and solute-dispersive conductances are calculated from porous-media properties by averaging the properties over a given cell face in proportion to the area of the cell face that is within each active element. A cell face is within parts of one to four active elements (fig. 4.4); thus a conductance is calculated from property values of as many as four different elements.

Variable weighting for spatial discretization of the advective term in the transport equations is available, ranging from centered-in-space differencing to upstream-in-space differencing (**SOLUTION_METHOD** data block, **-space_differencing**). Centered-in-space differencing has the potential for causing oscillations in the solutions. This can be particularly problematic for the chemical-reaction calculations if negative concentrations are computed. In PHAST, negative concentrations are set to zero and ignored in the chemical-reaction calculation, but this leads to mass-balance errors. Analysis of one-dimensional flow and transport with constant coefficients (Fletcher, 1991, pp. 286-293) shows that numerical oscillation does not occur if

$$\frac{\Delta x}{\alpha} \le 2 \text{ or } Pe \le 2, \qquad \text{(D.5)}$$

where Δx is the cell size (m); α is the dispersivity (m); Pe is the cell Peclet number ($Pe = v_x \frac{\Delta x}{D_{xx}} = \frac{\Delta x}{\alpha}$)(-); v_x is the component of velocity (m/s) in the X direction; and D_{xx} is the dispersion-coefficient tensor element (m^2/s) appropriate for one-dimensional flow and transport in the X direction.

Upstream-in-space differencing introduces artificial numerical dispersion into the transport solutions. Again, analysis of one-dimensional flow and transport with constant coefficients (Fletcher, 1991, pp. 302-303) yields a numerical dispersion coefficient and a numerical dispersivity given by

$$D_{ns} = \frac{v_x \Delta x}{2} \text{ and} \qquad \text{(D.6)}$$

$$\alpha_{ns} = \frac{\Delta x}{2}, \qquad \text{(D.7)}$$

where D_{ns} is the numerical dispersion coefficient from spatial discretization (m^2/s) and α_{ns} is the numerical dispersivity from spatial discretization (m). These terms result from the truncation error of the finite-difference approximation.

D.1.3. Temporal Discretization

Variable weighting for temporal discretization is available (**SOLUTION_METHOD** data block, **-time_differencing**). Well-known specific cases include fully implicit (backward-in-time) differencing and Crank-Nicholson (centered-in-time) differencing. Centered-in-time differencing also has the potential for causing oscillations in the solutions. Again, this can be particularly problematic for the chemical-reaction calculations if negative concentrations are computed.

Fully implicit temporal discretization is always used for computing a steady-state solution to the flow equation. This removes the possibility of oscillations in the solution and allows for large time steps to be taken as steady state is approached. For transient-flow simulation, the specified weighting for temporal discretization is used for both the flow equation and the transport equations.

Analysis of one-dimensional flow and transport with constant coefficients shows that numerical oscillation does not occur for centered-in-time differencing (with centered-in-space differencing) if

$$\frac{D_{xx}\Delta t}{\Delta x^2} = \frac{\alpha v_x \Delta t}{\Delta x^2} \le 1 \text{ or } Cr \le Pe, \qquad \text{(D.8)}$$

where Δt is the time step (s) and Cr is the cell Courant number ($Cr = \frac{v_x \Delta t}{\Delta x}$) (-). This criterion is not as stringent as that for spatial oscillation, and in some cases the method is stable even if the criterion is exceeded.

Backward-in-time differencing also introduces artificial numerical dispersion into the transport solutions. Again, analysis of one-dimensional flow and transport with constant coefficients yields the numerical dispersion coefficient and numerical dispersivity from temporal discretization given by

$$D_{nt} = \frac{v_x^2 \Delta t}{2} \text{ and} \tag{D.9}$$

$$\alpha_{nt} = \frac{v_x \Delta t}{2}, \tag{D.10}$$

where D_{nt} is the numerical dispersion coefficient from temporal discretization (m^2/s) and α_{nt} is the numerical dispersivity from temporal discretization (m). These terms result from the truncation error of the finite-difference approximation.

The total numerical dispersivity is the sum of the spatial (eqn. D.7) and temporal (eqn. D.10) numerical dispersivities. For numerical accuracy in transport simulations, the total numerical dispersivity should be much less than the physical dispersivity. Thus, for upstream-in-space and backward-in-time differencing,

$$\frac{\Delta x}{2} + \frac{v_x \Delta t}{2} \ll \alpha \text{ or in dimensionless form:} \tag{D.11}$$

$$Pe(1 + Cr) \ll 2\,. \tag{D.12}$$

If upstream-in-space and centered-in-time differencing are used, then the temporal dispersivity is zero and accuracy depends on the relation:

$$\frac{\Delta x}{2} \ll \alpha\,. \tag{D.13}$$

If centered-in-space and backward-in-time differencing are used, then the spatial dispersivity is zero and accuracy depends on the relation:

$$\frac{v_x \Delta t}{2} \ll \alpha\,. \tag{D.14}$$

Although these equations are from a simplified, restricted one-dimensional analysis, they provide guidance for discretization of more complex three-dimensional simulations of transport.

D.1.4. Automatic Time-Step Algorithm for Steady-State Flow Simulation

A simulation option is available that calculates the steady-state flow field by simulating transient flow until steady state is reached.The time steps are automatically adjusted during the steady-flow simulation, which significantly reduces computation time compared to the use of fixed time steps. Therefore, the ground-water flow simulator uses an empirical algorithm for automatic time stepping based on Aziz and Settari (1979, p. 403) as adapted by Kipp (1997, p. 13).

For the automatic time-step option, the user may specify the target change in potentiometric head over a time step that is considered acceptable (**STEADY_FLOW** data block, **-head_change_target**) as well as the maximum (**STEADY_FLOW** data block, **-maximum_time_step**) and minimum (**STEADY_FLOW** data block, **-minimum_time_step**) time steps allowed. At the beginning of each time step, adjustment is made to the size of the time step to try to attain the target change in head. If the maximum change in head for the last time step exceeds the target change in head or is less than 90 percent of the target change in head, the time step is adjusted by the ratio of the target head change to the change in head for the last time step. However, the new time step is constrained by the specified minimum and maximum time-step limits, and further, it may not grow by more than a factor of two at each adjustment. Because the actual changes in the potentiometric head during a simulation are not directly proportional to the changes in time-step length, the specified target head change can be considerably exceeded when this linear predictor is used for the adjustment of the time step. Therefore, an additional test is made at the end of each time step. If the maximum change in head is greater than 1.5 times the target head change, the time step is repeated with a shorter time-step length, which is calculated as explained above.

The minimum allowed time step, optionally set by the user, is maintained for the first two steps of the simulation. If the maximum change in the head during the second time step exceeds the target head change, the simulator stops because it cannot adjust the time step below the minimum allowed value.

D.2. Chemical-Reaction Equations

Both equilibrium and kinetic reactions are simulated in PHAST. Homogeneous equilibrium in the aqueous phase is described with an ion-association aqueous model that includes mass-action equations for ion pairs and complexes, and activity coefficients based on Debye-Hückel theory. Heterogeneous equilibrium reactions are based on equality of activities among phases. For kinetic chemical reactions, the rates of reactions are user-defined functions of solution composition. The equations describing equilibrium reactions and the formulation of kinetic chemical reactions are given in detail in Parkhurst and Appelo (1999).

The chemical equations form a set of non-linear algebraic equations for equilibrium reactions and ordinary differential equations in time for kinetic reactions. The differential and algebraic equations are coupled through the total aqueous concentration of components. However, the chemical reactions in each cell of the simulation region are solved independently. For a given cell, no information about chemical compositions in other cells is used in the simulation of equilibrium and kinetic reactions.

D.2.1. Equilibrium Reactions

The set of equilibrium reactions includes aqueous-complexation, mineral-phase, ion-exchange, surface-complexation, solid-solution, and gas-phase (rarely useful in PHAST) equilibria. All equilibrium reactions are based on mass-action equations that relate activities of species to an equilibrium constant:

$$K = \frac{\prod_{products} a_p^{\upsilon_p}}{\prod_{reactants} a_r^{\upsilon_r}}, \tag{D.15}$$

where K is the equilibrium constant, a is activity, and υ_r and υ_p are stoichiometric coefficients of the reactants and products in the chemical equation. Equilibrium constants are functions of temperature. Activities of aqueous species are related to molalities through activity coefficients:

$$a = \gamma m, \tag{D.16}$$

where γ is the activity coefficient and m represents molality. The activity coefficients are functions of species charge and ionic strength, and are derived from extensions of the Debye-Hückel activity-coefficient theory. Activities of pure minerals are assumed to be fixed at 1.0. Activities of gas components are equal to their partial pressures. Activities of solid-solution components are equal to their mole fraction or optionally a function of mole fraction. Activities of ion-exchange species are equal to their equivalent fractions, possibly times an activity coefficient that is a function of ionic strength. Activities of surface complexes are equal to mole fraction of the site type (Parkhurst and Appelo, 1999).

D.2.2. Component Mole Balance

For any time step in the flow and transport simulation, the total number of moles of each component in each cell is defined by the sum of moles of component in solution and the moles of component in the solids. At the beginning of the chemistry calculation for a cell, the total aqueous concentration of a component in solution is defined by the concentration resulting from the transport calculation. The moles of component in the solids includes the sum of moles of the component in mineral and gas phases, exchangers, surfaces, and solid solutions in the saturated part of the cell. During the chemistry calculation, the moles of component in solution are augmented by kinetic reactions, after which, equilibrium is recalculated among solution and solids to produce a new total aqueous concentration. The total aqueous concentrations of components at the end of the chemistry calculation are then used in the next transport step after adjustment for any change in saturation in the cell (see section D.5.6. Free-Surface Boundary).

D.2.3. Kinetic Reactions

Kinetic reactions (**KINETICS** data block in the chemistry data file) are defined with rate equations that are a function of solution composition. The user defines the rate equations by using Basic-language statements (**RATES** data block in the chemistry data file), which allow general definitions of rate expressions without recompilation of the computer code (Parkhurst and Appelo, 1999).

D.3. Property Functions and Transport Coefficients

Before the flow and component transport equations can be solved, information about the fluid properties, porous-media properties, and transport coefficients needs to be provided. As mentioned previously, fluid density and viscosity are constant and uniform in the PHAST simulator. Porosity is a function of the fluid potentiometric head in the case of confined flow (Kipp, 1987, p. 24) and is assumed to be constant in the case of unconfined flow. For the capacitance or storage term in the confined flow equation, the compressibility of the water and the porous medium is accounted for by the specific storage (Eagleson, 1970, p. 270; Kipp, 1987, p. 25). The specific storage includes both fluid and porous matrix compressibility, and thus is a hybrid property of both fluid and porous matrix. The fluid and porous matrix are assumed incompressible for the case of unconfined flow.

The variation in the ratio of water to solid that is caused by changes in porosity under confined flow conditions is relatively small and is neglected in the chemical calculations. Thus, for confined systems, the concentration of a solid (mol/kgw) does not vary because of changes in storage. Concentrations of solids and sorbed components only change by heterogeneous chemical reaction. (For unconfined systems, concentrations of solids also will vary as the water table rises and falls within a cell, see section D.5.6. Free-Surface Boundary.)

Hydrodynamic dispersion is characterized by the hydrodynamic dispersion-coefficient tensor (Bear, 1972, Scheidegger, 1961). For an isotropic porous medium, two parameters describe the dispersion-coefficient tensor, the longitudinal dispersivity and the transverse dispersivity. The dispersion-coefficient tensor is a function of space, through the dispersivity coefficients, and is a function of space and time through the interstitial velocity. Mathematically,

$$D_{ij} = (\alpha_L - \alpha_T)\frac{v_i v_j}{|\boldsymbol{v}|} + \alpha_T |\boldsymbol{v}| \delta_{ij} + D_M \delta_{ij}, \tag{D.17}$$

where D_{ij} is the dispersion-coefficient tensor element (m^2/s); v_i is the interstitial velocity in the *i* coordinate direction (m/s); $|\boldsymbol{v}|$ is the magnitude of the interstitial velocity vector (m/s); α_L is the longitudinal dispersivity (m); α_T is the transverse dispersivity (m); D_M is the effective molecular diffusion coefficient (m^2/s); and δ_{ij} is the Kronecker delta function, $\delta_{ij} = 1$, for $i = j$ and $\delta_{ij} = 0$, for $i \neq j$. The dispersion-coefficient tensor is a nine-element tensor that is symmetric. This tensor links the solute dispersive fluxes to the concentration gradients. The off-diagonal elements give rise to cross-dispersive fluxes, which are dispersive fluxes in one coordinate direction driven by concentration gradients in another coordinate direction.

Several field studies have indicated that transverse dispersion in the vertical direction is much smaller than transverse dispersion in the horizontal direction (Robson, 1974, 1978; Garabedian and others, 1991; and Gelhar and others, 1992) for the case of a stratified porous medium with flow parallel to the stratification. Burnett and Frind (1987) formulated an *ad hoc* anisotropic dispersion-coefficient tensor for this case involving horizontal and vertical transverse dispersivities. The modified dispersion-coefficient tensor is

$$D_{xx} = \alpha_L \frac{v_x^2}{|\boldsymbol{v}|} + \alpha_{T_H} \frac{v_y^2}{|\boldsymbol{v}|} + \alpha_{T_V} \frac{v_z^2}{|\boldsymbol{v}|} + D_M, \tag{D.18}$$

$$D_{yy} = \alpha_L \frac{v_y^2}{|\boldsymbol{v}|} + \alpha_{T_H} \frac{v_x^2}{|\boldsymbol{v}|} + \alpha_{T_V} \frac{v_z^2}{|\boldsymbol{v}|} + D_M, \tag{D.19}$$

$$D_{zz} = \alpha_L \frac{v_z^2}{|\boldsymbol{v}|} + \alpha_{T_V} \frac{v_x^2}{|\boldsymbol{v}|} + \alpha_{T_V} \frac{v_y^2}{|\boldsymbol{v}|} + D_M, \tag{D.20}$$

$$D_{xy} = D_{yx} = (\alpha_L - \alpha_{T_H})\frac{v_x v_y}{|\boldsymbol{v}|}, \tag{D.21}$$

$$D_{xz} = D_{zx} = (\alpha_L - \alpha_{T_V})\frac{v_x v_z}{|\boldsymbol{v}|}, \text{ and} \tag{D.22}$$

$$D_{yz} = D_{zy} = (\alpha_L - \alpha_{T_V})\frac{v_y v_z}{|\boldsymbol{v}|}, \tag{D.23}$$

where α_{T_H} is the horizontal transverse dispersivity (m) and α_{T_V} is the vertical transverse dispersivity (m). Here the terms horizontal and vertical refer to the axes directions of the Cartesian coordinate system. This formulation is used in the PHAST simulator and it reduces to the isotropic form when $\alpha_{T_H} = \alpha_{T_V}$.

Cross-derivative dispersive-flux terms may be included or excluded in a PHAST simulation. The more rigorous treatment of including the cross-derivative terms involves an explicit calculation (**SOLUTION_METHOD** data block, **-cross_dispersion true**). Explicit calculation of the cross-dispersive flux terms is lagged one iteration in the solution cycle of the solute-transport equations; this lagging preserves the seven-point stencil in the finite-difference equations, which enables the use of the same linear equation solvers for the transport equations as are used for the flow equation. When cross-derivative dispersive flux terms are included in the solution method, two iterations of transport equation solution are done for each time step to compensate for the lagged terms. The cross-derivative dispersive flux terms are recalculated for each iteration on the basis of the concentration conditions existing at the end of the previous iteration. These lagged terms are incorporated into the right-hand-side of the finite-difference transport equations. This treatment requires storage of the nine finite-difference dispersion-coefficient terms for each cell.

In the solution of the transport equations, strong diagonal flows through a cell can cause the cross-dispersive solute fluxes to produce negative concentrations. Therefore, an alternative treatment of the cross-derivative dispersion terms consists of neglecting the cross-derivative dispersion fluxes entirely. No iterations of the component transport solution are required. For robustness of the PHAST simulator, the default option is to neglect the cross-derivative dispersive flux terms.

For PHAST, it is assumed that dispersion of each aqueous component can be described by the same set of dispersivity values. This yields the same set of dispersion coefficients for each chemical component. No other transport coefficients are present because, in the operator-split form of the component transport equations, no source terms from reactions appear. Each component-transport equation contains only advective, dispersive, and storage terms.

D.4. Well-Source Conditions

A point source (sink) term is used to represent injection (withdrawal) by a well in equations D.1 and D.4. Spatial discretization converts this term to a line source. Although a well is treated as a line source for the flow and transport equations, a well is a finite-radius cylinder for the local well-bore model.

The local well-bore model calculates flow rates between the well bore and each layer of cells (Kipp, 1987). A well can be used for fluid injection or fluid withdrawal with associated solute injection or production. A well also can be used for observation of aquifer conditions. The well bore can communicate with any subset of cells along the Z-coordinate direction at a given X–Y coordinate location. That is, the well may be screened or open-hole over several intervals of its depth.

Only the two simplest options for the well-bore model from HST3D are included in PHAST. A well can be a source or sink with specified flow rate within the aquifer that is allocated by mobility or by mobility times head difference as described by Kipp (1987, p. 32-36). Mobility is the product of hydraulic conductivity and the length of open interval within the cell.

After discretization, the well flow rate to or from each penetrated cell depends on the length of open interval in that cell. The PHASTINPUT preprocessor calculates the length of the open interval above and below the node in each cell. This information on the upper and lower bounds of the open interval enables the use of the appropriate element properties in calculating the well mobility factors for each cell that communicates with a well. Variation in hydraulic conductivity among the eight elements that comprise a full cell is handled by computing the arithmetic average of the conductivity in the four elements in the upper portion and the four elements in the lower portion of the cell. Then, a weighted-average hydraulic conductivity is computed by using the fractional open-interval lengths in the upper and lower portions of the cell as weighting factors.

For a well with flow-rate allocation by mobility only, every cell that communicates with the well has only injection or only production, depending on the total flow rate specified. It is not possible to have a cell with injection and a cell with production in the same well for mobility-only flow-rate allocation. For an injection well, an associated solution composition is specified by the user by using solution indices (**-associated_solution** in the **WELL** data block). For a production well, the total aqueous concentrations of components produced from the well are calculated from a weighted average of concentrations from the producing cells; the concentrations are weighted in proportion to the cell flow rates to the well.

For a well with flow-rate allocation by mobility times head difference, the different cells that communicate with the well may produce or inject water. For a production well, the flow is accumulated from the bottom to the top, with the restriction that the well-bore flow at all points must be upward or the simulator terminates with an error. The total aqueous concentrations of components produced from the well are calculated by accumulating and distributing flow to each cell connected to the well, beginning from the lowermost cell. If water is produced from a cell, it changes the concentration in the well bore by mixing in proportion to the cell flow rate and the flow rate in the well bore; if water is injected into a cell, it is injected with the concentration in the well bore at that point. Similarly, for an injection well, the flow is distributed from the top to the bottom, with the restriction that the well-bore flow at all points must be downward, or the program terminates with an error. The total aqueous concentrations in the well bore are determined by accumulating and distributing flow to each cell connected to the well, beginning at the uppermost cell. If water is injected into a cell, it is injected with the concentration in the well bore at that point; if water is produced from a cell, the production changes the concentration in the well bore by mixing in proportion to the cell flow rate and the flow rate in the well bore. Thus, mixing of water within the well bore is simulated, but no chemical reactions are considered within the well bore.

The pumping or injection rate and the associated solution composition for a well may change with time. Temporal changes in these values occur discontinuously in time; a change in value causes the beginning of a new simulation period. This discontinuity means that the effective parameter value for the first time step of a new simulation period is the new value under backward-in-time differencing and the average of the old and new values under centered-in-time differencing. For compatibility among well-source conditions and boundary conditions, see section D.5.7. Boundary-Condition Compatibility.

D.5. Boundary Conditions

To define a simulation, boundary conditions need to be specified. The types of boundary conditions available in PHAST include (1) specified potentiometric head and associated solution composition (**SPECIFIED_HEAD_BC** data block; **-head** and **-associated_solution**), (2) specified potentiometric head and specified solution composition (**SPECIFIED_HEAD_BC** data block; **-head** and **-fixed_solution**), (3) specified fluid flux and associated solution composition (**FLUX_BC** data block), (4) aquifer leakage and associated solution composition (**LEAKY_BC** data block), (5) river leakage and associated solution composition (**RIVER** data block), and (6) unconfined aquifer with a free surface (water table) (**FREE_SURFACE_BC** data block). For all boundary conditions with associated solution compositions, inflow enters the region with the total aqueous concentrations defined by the associated solution composition, but outflow leaves the region with the ambient total aqueous concentrations at the boundary location. A specified-head boundary condition may have a specified solution composition instead of an associated solution composition. For a specified-head boundary cell with a specified solution composition, the total aqueous concentrations of all components are constant over each simulation period and any specified changes in values cause the beginning of a new simulation period.

Associated and specified solution compositions are designated by solution indices, which refer to solution compositions defined in the chemistry data file. Preliminary geochemical simulations defined in the chemistry data file can be used to establish the solution compositions for boundary conditions. These preliminary simulations may include mineral equilibria, surface complexation, ion exchange, solid-solution equilibria, gas-phase equilibration, and general kinetic reactions, plus mixing, irreversible reactions, and temperature variation (see section D.6. Initial Conditions).

The spatial distribution of all boundary conditions is specified on a cell basis rather than on an element basis. Changes in boundary-condition type or parameters occur at cell boundaries rather than at nodes (fig. D.2). The default boundary condition for the flow and transport equations is no fluid flux and no dispersive or advective flux through the faces of the cells at the boundary of the active grid region. Cells at edges and corners of the active grid have two or three boundary faces, respectively (fig. 4.2).

Some boundary-condition parameters may be changed with time, including head, flux, and solution composition. Temporal changes in these values occur discontinuously in time; a specified change in value causes the beginning of a new simulation period. This discontinuity means that the effective parameter value for the first time step of a new simulation period is the new value under backward-in-time differencing and the average of the old and new values under centered-in-time differencing.

D.5.1. Specified-Head Boundary with Associated Solution Composition

A specified potentiometric-head boundary condition is available for the flow equation (**SPECIFIED_HEAD_BC** data block). The head value is specified relative to the Z coordinate of the simulation region. Specified-head boundary conditions are

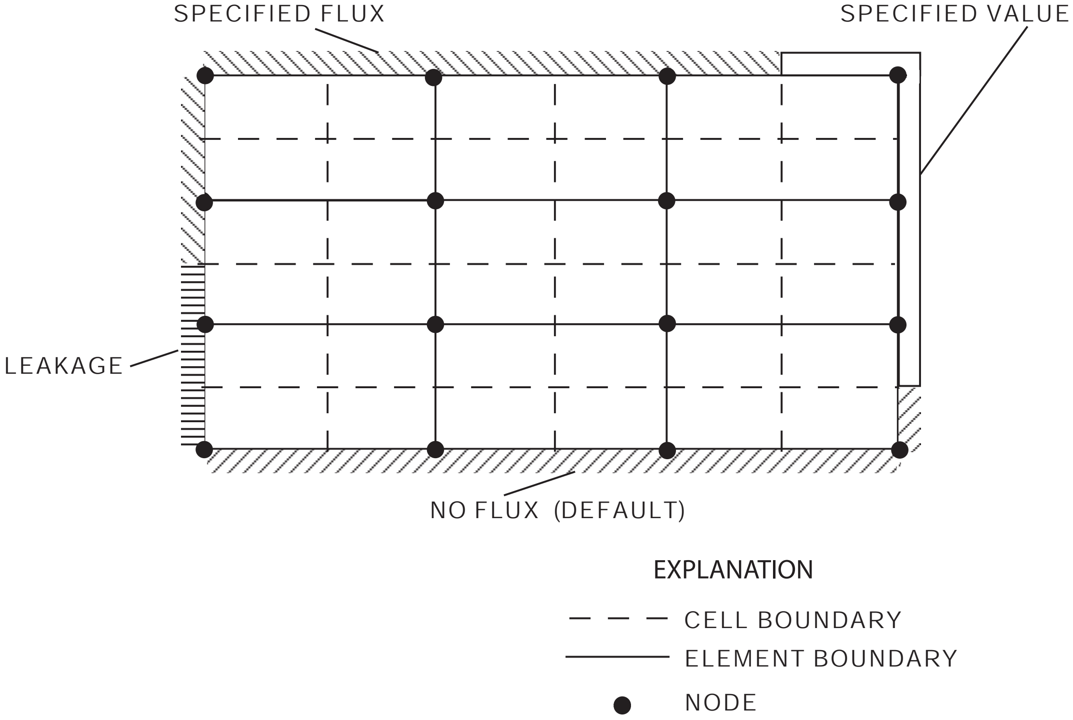


Figure D.2.—Types of boundary-condition cells in a two-dimensional grid.

incorporated into the difference equations by replacing the flow equation for those nodes by equations defining the specified heads. Therefore, it is not possible to define flux, leaky, or river boundary conditions for any face of a cell that is a specified-head cell.

The associated solution-composition option (**-associated_solution**) for a specified-head boundary defines the composition of water flowing into the boundary cells. The solution composition of a specified-head boundary cell with an associated solution composition will vary during the simulation period depending on the flux of water through the boundary relative to the composition currently in the cell and the flux and composition of water from other faces of the cell.

The head and associated solution composition for a specified-head boundary condition can be a function of location and time. Zones are used to define the spatial distribution of the specified-head boundaries. The head and associated solution composition may vary during the simulation; any specified change in value causes the beginning of a new simulation period. For compatibility among well-source conditions and boundary conditions, see section D.5.7. Boundary-Condition Compatibility.

D.5.2. Specified-Head Boundary with Specified Solution Composition

Specified solution compositions are an alternative to associated solution compositions for specified-head boundary conditions (**SPECIFIED_HEAD_BC** data block, **-fixed_solution**). Unlike associated solution-composition boundary cells, the solution composition in a specified solution-composition boundary cell does not vary during a single simulation period. The solution composition of a specified solution-composition boundary cell is constant regardless of inflow or outflow; however, it is not physically realistic to specify concentrations at an outflow boundary. The composition of the fixed solution may vary during a simulation; any specified change in composition (solution index number) causes the beginning of a new simulation period. All of the discussion in the previous section related to the flow equation for a specified-head boundary applies to boundary cells with specified solution composition. For compatibility among well-source conditions and boundary conditions, see section D.5.7. Boundary-Condition Compatibility.

D.5.3. Specified-Flux Boundary

Normal fluxes of fluid can be specified over parts of the boundary of the grid region as functions of location and time (**FLUX_BC** data block). Examples of physical boundary conditions that can be represented as specified-flux boundaries include infiltration from precipitation, known flux through lateral boundaries where head gradients can be estimated, and known recharge-and discharge-boundary flow rates for simple steady-state flow problems. On fluid-inflow boundaries, the associated solution composition of the inflowing fluid needs to be defined (**-associated_solution**), which then determines the advective flux of solute components. At fluid-outflow boundaries, the component concentrations of the outflowing fluid are equal to the ambient total aqueous concentrations. Purely diffusive solute-component flux with no fluid flow cannot be defined as a flux boundary condition.

The flux and associated solution composition for a specified-flux boundary condition can be a function of location and time. Zones are used to define the spatial distribution of the specified-flux boundaries. The flux and associated solution composition may vary during the simulation; any specified change in value causes the beginning of a new simulation period. Note that the fluid flux is specified as a volumetric flux with vector components expressed relative to the coordinate system of the simulation region. Thus, fluxes are signed quantities that express direction relative to cell faces that are perpendicular to a coordinate direction. For example, precipitation recharge would have a negative sign indicating a flux in the negative Z-coordinate direction. For compatibility among well-source conditions and boundary conditions, see section D.5.7. Boundary-Condition Compatibility.

D.5.4. Leakage Boundary

A leakage boundary condition is represented as a fluid flux that is driven by a difference in potentiometric head across a layer of finite thickness (**LEAKY_BC** data block). Flow and transport rates at leakage boundaries are functions of potentiometric head and total aqueous concentrations at a point in time on the region boundary, and do not depend on the histories of these values. Often this boundary condition is used to represent vertical or lateral flow across leaky semi-confining layers, where the hydraulic conductivity of the semi-confining layer is orders of magnitude smaller than the conductivity of the adjacent simulation region. The leakage boundary also can be used to extend the simulation region laterally.

Mathematical representation of leakage boundary conditions is explained in Kipp (1987). The treatment of leakage boundaries is based on the following simplifying assumptions: (1) changes in fluid storage in the leaky layer are neglected, (2) leaky-layer capacitance effects on solute transport are neglected, and (3) flow and solute component transport are affected by the leakage fluxes that enter the region, but flow and solute conditions that exist on the far side of the leaky layer (outside the simulation region) are not affected by fluxes that enter or leave the simulation region. Furthermore, no chemical reactions are simulated within the leaky layer.

For illustration, assume that part of the upper boundary surface is overlain by a semi-confining layer. Another aquifer lies above the semi-confining layer with a head distribution at its contact with the semi-confining layer that is a known function of time. The leakage boundary flux is given by

$$q_L = \frac{K_L}{b_L}(h_e - h_b), \tag{D.24}$$

where q_L is the volumetric flux at a leakage boundary (m^3 m^{-2} s^{-1}); K_L is the hydraulic conductivity of the semi-confining layer (m/s); b_L is the thickness of the semi-confining layer (m); h_e is the potentiometric head at the external boundary of the semi-confining layer (m); and h_b is the potentiometric head at the boundary of the simulation region (m). The associated solute component fluxes are assumed to be purely advective. They are obtained from the total aqueous concentrations of the associated solution composition if water flux is into the simulation region, or from the ambient total aqueous concentrations at the boundary location if the water flux is out of the simulation region. The associated solution compositions are defined by position along the leakage boundary and may vary during the simulation; any specified change in solution composition (solution index number) causes the beginning of a new simulation period. The head at the external boundary may also be specified to vary during the simulation.

Leakage boundary conditions are transformed into source terms in the difference equations on a cell-by-cell basis. The discretized equation for boundary cell, *m*, is

$$Q_{Lm} = \frac{K_{Lm}}{b_{Lm}}(h_{em} - h_m^{(n)})S_{BLm} - \frac{K_{Lm}}{b_{Lm}}S_{BLm}\delta h_m, \tag{D.25}$$

where Q_{Lm} is the volumetric flow rate at a leakage boundary for cell *m* (m^3/s); K_{Lm} is the hydraulic conductivity of the semi-confining layer for cell *m* (m/s); b_{Lm} is the thickness of the semi-confining layer for cell *m* (m); h_{em} is the potentiometric head at the external boundary of the semi-confining layer for cell *m* (m); h_m is the potentiometric head at the boundary of the simulation region for cell *m* (m); (*n*) is the index of the discrete time value when the variable is evaluated; S_{BLm} is the area of the leakage boundary face of cell *m* (m^2); and δh_m is the change in potentiometric head at the boundary of the simulation region over the time step for cell *m* (m). The leakage flow rate given by equation D.25 has an explicit term for the right-hand side of the discretized flow equations and an implicit factor for the left-hand side. For compatibility among well-source conditions and boundary conditions, see section D.5.7. Boundary-Condition Compatibility.

D.5.5. River-Leakage Boundary

The river-leakage boundary condition (**RIVER** data block) is similar to the leakage boundary condition, but has some special characteristics: (1) this boundary condition is appropriate only for unconfined aquifer regions; (2) the river is a land-surface feature that traverses the upper boundary of the simulation region; (3) for a vertical stack of cells, any river leakage enters or leaves the cell that contains the water table; (4) the riverbed-sediment layer is the semi-confining layer through which leakage occurs; and (5) the flux is limited to a maximum value when the head in the aquifer is below the bottom of the river.

The calculation of river-leakage flux is the same as for a leakage boundary condition. The flux depends on the difference in head between the river and the aquifer, the area of the river bottom, the thickness of the riverbed, and the hydraulic conductivity of the riverbed. The maximum flux from the river to the aquifer is limited by not allowing the flux to increase when the aquifer potentiometric head decreases to less than the elevation of the bottom of the riverbed. Physically, this means that if the water table declines below the bottom of the riverbed, the increased resistance to flow of the partially saturated porous medium prevents any further increase in flux from the river to the aquifer. Unsaturated flow is not simulated explicitly. The simplified geometry of a cross section at a river-leakage boundary is shown in figure D.3.

In the numerical implementation, a river is a piece-wise linear feature defined by its trace across the top surface of the grid region (fig. 4.6). A chain of piece-wise linear segments (river segments) represents the river route. The end points of each segment have data defining the river head, width, bottom elevation, riverbed hydraulic conductivity, and associated solution index.

Several steps are involved in translating the data from the sequence of river points that defines a river into boundary conditions for the affected cells of the active grid region. First, each pair of river points that define a river segment and the associated widths are used to define trapezoidal areas as shown in figure D.4A. The collection of trapezoids has gaps between trapezoids and overlapping areas (fig. D.4B). The gap between the end lines of adjacent trapezoids is appended to the trapezoid that was defined first in the sequence of river point definitions to produce a polygon that is associated with a river segment (river polygon). The overlapping area of two trapezoids is assigned to the river segment that was defined first in sequence, and removed from any other

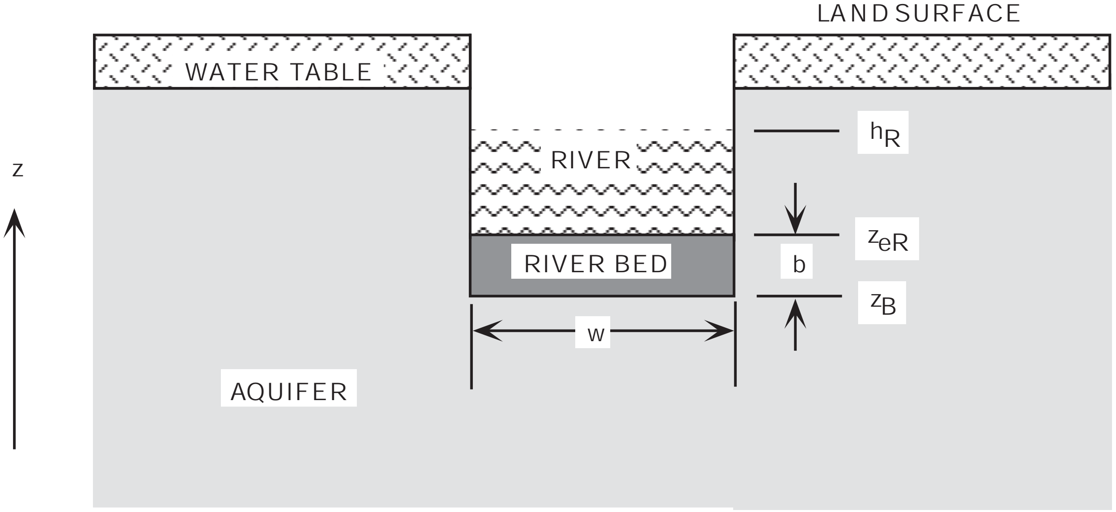


Figure D.3.—Schematic section showing geometry for a river-leakage boundary, where b is thickness of riverbed, h_r is elevation of river surface, w is width of river, z_{eR} is elevation of the top of the riverbed, and z_B is elevation of the bottom of the riverbed.

river polygons (fig. D.4C). This removal process acts over all river polygons, such that overlapping areas between river polygons are assigned to the river that has the smallest integer identification number (**RIVER** *n*). No subarea of the surface of the active grid region is assigned to more than one river polygon. Each river polygon is then intersected with each cell on the surface of the active grid region (fig. D.4D). The intersection divides the area of each river polygon into subpolygons each of which is contained within a single cell (fig. D.4E). Finally, the centroid of each subpolygon is located. Properties for a subpolygon are determined by interpolating from the two nearest river points with the property defined, to the point on the river segment that is closest to the centroid of the subpolygon (fig. D.4F). For reactive-transport simulations, the composition of the solution associated with the river subpolygon is determined by proportional mixing of the end-member solution compositions at the nearest river points where solution compositions are defined.

Any number of river segments may overly a vertical column of cells, thus enabling tributaries or branches to be represented (fig. 4.6). Flow into or out of the aquifer is calculated for each river subpolygon that affects a given column of cells. The flows for

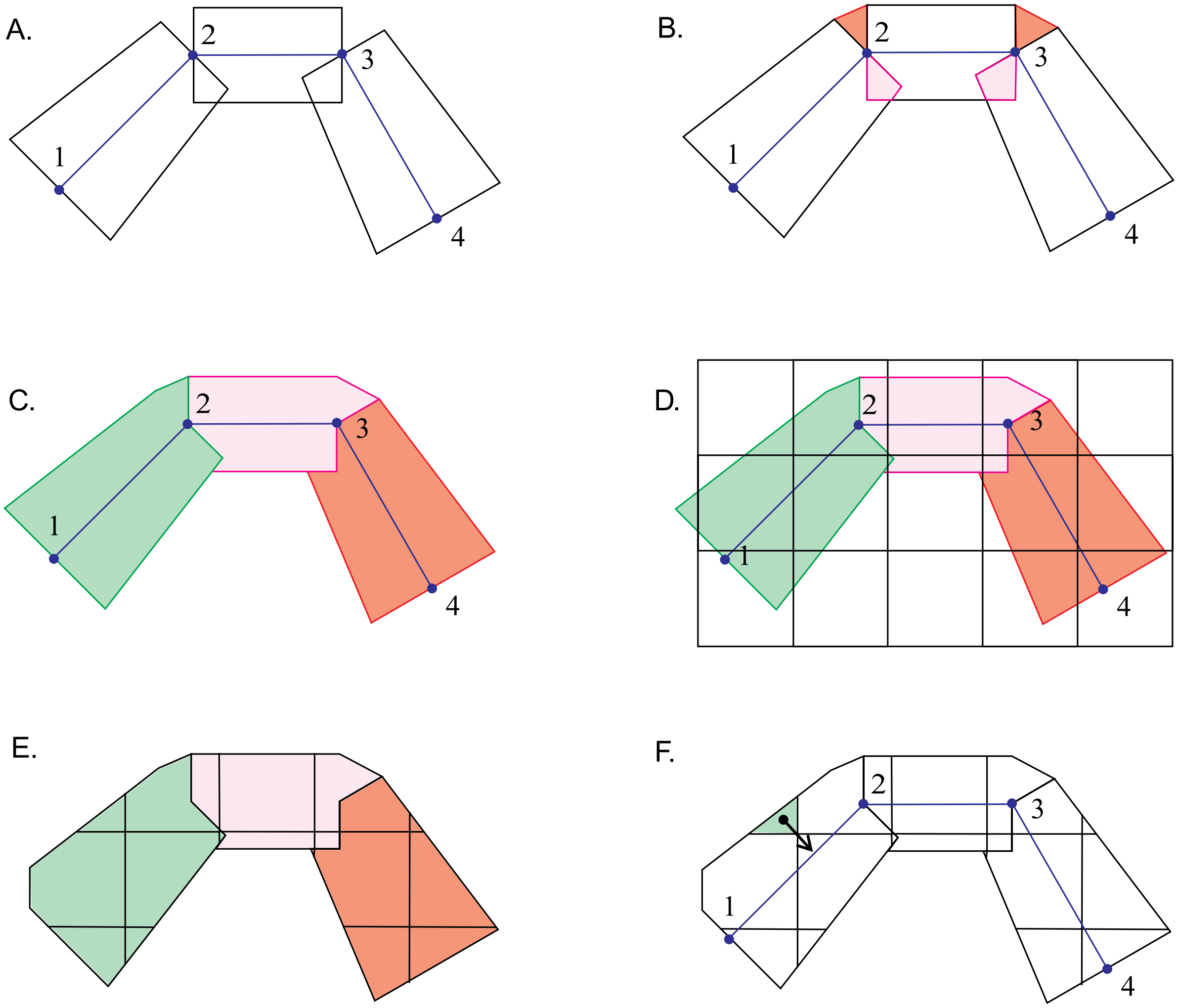


Figure D.4.—Steps by which river subpolygons and river subpolygon properties are determined. (A) Trapezoidal areas are defined by the river points (numbered blue dots) and the river widths at each river point. (B) Gaps are filled and overlaps are eliminated. (C) River polygons are created for each river segment. (D) River polygons are intersected with the grid to form river subpolygons. (E) River subpolygons are ready for final processing, color indicates association with original data. (F) Properties are assigned by finding the point on the river nearest the centroid (black dot) of the river subpolygon and interpolating from values at river points.

all river subpolygons in a vertical column are aggregated, and the aggregated flow is injected into or removed from the cell that contains the water table.

Conceptually, all river leakage into and out of the aquifer occurs through the river bottom; no leakage through the sides of the riverbed is considered. Rivers with extensive vertical sides that contact multiple layers of cells cannot be represented adequately because the river is connected through its riverbed to only one cell in each vertical column. The configuration of a deep riverbed with lateral flow into one or more layers of aquifer cells could be approximated by a set of specified-head cells. However, a fine areal discretization would be required because the river would have an effective width of an entire cell.

When a single river subpolygon recharges an aquifer cell, the incoming water has the chemical composition of the associated solution composition for that subpolygon. When water discharges from the aquifer to the river, the discharging water has the ambient chemical composition of the water-table cell. For the case of multiple branches of rivers traversing a given column of cells, with net recharge into the aquifer, the total aqueous concentrations of the recharge are determined as the flow-rate-weighted averages of the associated solution concentrations of all the river subpolygons recharging to that column of cells. It is possible that some river subpolygons receive discharge from the aquifer; however, the recharge concentrations are calculated by using only associated solution concentrations from river subpolygons that recharge the aquifer.

Hydrostatic extrapolation of the potentiometric head is used exclusively to locate the free surface (water table) under a river. This extrapolation prevents a non-convergent cycle of free-surface adjustment that can occur if other types of extrapolation or interpolation are used. Problems will occur if the elevation of the free surface drops to a cell that is below a cell with a partial or full saturation. This implies perched water and a zone of unsaturated flow, which PHAST cannot simulate.

The head and associated solution composition for each river point may vary during the simulation; any specified change in value causes the beginning of a new simulation period. At each new simulation period, head and solution composition are re-interpolated for each river subpolygon. A river-leakage boundary condition and a specified-flux boundary can be applied to the same upper-boundary face of a given cell because they are different types of boundary conditions. This feature is needed to represent river leakage over part of a cell area and precipitation infiltration over the remaining area. Numerically, the precipitation flux is distributed over the entire area of the cell boundary including the portion that is covered by the riverbed. This is not a serious limitation for riverbeds covering only a small fraction of the upper surface of a cell. A river-leakage boundary condition and a leaky boundary condition cannot be applied to the Z face of the same cell. For compatibility among well-source conditions and boundary conditions, see section D.5.7. Boundary-Condition Compatibility.

D.5.6. Free-Surface Boundary

For an unconfined aquifer, a free-surface (water-table) boundary exists with a position in space and time that is determined as part of the simulation. The boundary condition requires that the absolute pressure on the free surface is atmospheric, which implies that the relative pressure is zero and the potentiometric head equals the free-surface elevation. Fluid- and porous-matrix compressibilities are assumed to be zero for unconfined flow systems, so the specific yield is equivalent to the effective porosity.

The simulation region for saturated, unconfined, ground-water flow varies with time as the free-surface moves. The active grid region is fixed and determines the maximum extent of the simulation region. Flow, transport, and reactions are computed at any given time only for those cells of the active grid region that are fully or partly saturated. These fully and partly saturated cells form the discrete approximation to the current simulation region. Boundary potentiometric heads less than the elevation at the point of measurement (pressures less than atmospheric) imply that the free surface is below the node, whereas boundary heads greater than the elevation (pressures greater than atmospheric) imply that the free surface is above the node.

The quantities of components attached to solid phases—that is, the amounts in exchangers, surface complexes, minerals, and solid solutions—are left behind as the free surface falls below the bottom of a cell; these quantities are then outside the simulation region. Resaturation of the porous medium must occur for these quantities of solid-phase components to be reincluded in the simulation region. In other words, components attached to the porous matrix in a cell that becomes dry remain unreactive until the cell is resaturated.

For saturated media, the amounts of reactive solids are defined to be those that are contained in a volume of aquifer that contains one liter of water. Similarly, for partially saturated media, the amounts of reactive solids are defined to be those that are contained in a volume of aquifer that contains one liter of pore space. Within a cell that contains the free surface, two reservoirs of solid reactants are created, the saturated and unsaturated reservoirs. As the fraction of saturation changes as a result of the flow calculation, moles of solid reactants are transferred from one reservoir to the other. A decrease in fraction of saturation causes moles of solid reactants to be transferred from the saturated reservoir to the unsaturated reservoir. Conversely, an increase in fraction of saturation causes moles of solid reactants to be transferred from the unsaturated reservoir to the saturated reservoir. The number of moles transferred is calculated from the previous fraction of saturation and the current fraction of saturation. When the cell is fully saturated, all solid reactants are in the saturated reservoir; when the cell is dry, all solid reactants are in the unsaturated reservoir.

For chemical reactions, a volume of water (L) equal to the saturated fraction is allowed to react with the contents of the saturated reservoir of solid reactants. The composition of the water before reaction is equal to the water composition following the transport calculations. The amounts of solid reactants in the saturated reservoir before reaction are determined by the amounts in the saturated reservoir following the previous chemical-reaction step adjusted by transfers to or from the unsaturated reservoir to account for changes in fraction of saturation. New amounts of solid reactants are stored in the saturated reservoir after completion of the chemical-reaction calculation.

The amounts of solid reactants in the unsaturated reservoir do not participate in chemical reactions. They are only stored in this reservoir until such time as they are transferred to the saturated reservoir. Both reservoirs are assumed to be well mixed, that is, there is no accounting for individual strata of solid reactants as the water table moves up and down within the cell. A major shortcoming of the unsaturated-zone accounting is that water infiltrating to the water table does not contact the solid reactants in the unsaturated reservoirs. The unsaturated reservoir can be reintroduced to the active simulation only by a rising water table. At this time, it is not possible to print or view the amounts of solid reactants and chemical components in the unsaturated reservoirs.

To allow for the definition of an active grid region with a regional topography, an inactive grid region may be defined above the active cells. That is, the active grid region of an unconfined flow simulation does not have to possess a flat upper surface.

The numerical implementation of the unconfined-aquifer, free-surface boundary condition is done by modifying the pressure-coefficient terms in the discretized equations for flow and solute-component transport and adjusting the fluid volume (saturated thickness) of the cells containing the water table (Kipp, l997). PHAST uses a simpler algorithm than HST3D for locating the free surface. The location is established by linear extrapolation of the nodal pressure from the saturated cell below to calculate the elevation of zero (atmospheric) pressure. The algorithm is based on assumption of hydrostatic pressure and does not account for a flux through the bottom and sides of each cell containing the free surface. This simplification can cause the free surface to approach the incorrect elevation for a steady state, especially if strong vertical ground-water flow is present. Approaching a steady state by draining or wetting will give different results for the free-surface elevation. For example, this algorithm gives the correct elevation for a simple one-dimensional flow problem by draining to steady state, but not by filling to steady state.

The free surface can be located at any elevation within the active grid region, and is allowed to rise above the upper boundary of the grid region. This rise is physically realistic only if additional porous medium exists above the upper boundary of the grid. However, large rises above the top of the grid indicate the need for the grid region to be extended vertically to enclose the entire aquifer saturated thickness. In the case of an inactive grid region above the top of the active grid region, no conversion to confined flow conditions is made if the free surface does rise above the uppermost active cells. (In the case of confined flow, no conversion to unconfined flow conditions is made if the potentiometric head drops below the elevation of the confining boundary.) The grid region can be defined only as entirely unconfined or entirely confined. Mixed confined and unconfined conditions over different subareas of the land surface cannot be represented.

The free surface may fall to any cell below the top of the active grid region. Dry cells are isolated from the ground-water flow region by assigning zero to the fluid and solute-dispersive conductances for their lateral faces. A dry cell is resaturated by flux through its bottom face. Thus, if an entire column of cells becomes totally dewatered, there is no mechanism to resaturate it. A cell that becomes dry is removed from the computational region at the end of the time step. Also, a cell that becomes resaturated is added back into the computational region at the end of the time step. The saturation of a cell that goes dry during a time step is constrained to be a small value for the remainder of that time step. Because the free-surface position is treated explicitly, it is adjusted only at the end of each time step. The conductances and source terms in the flow and transport equations have the saturated fraction parameter included, as necessary, for the cell facial-area terms involving horizontal flow. The appropriate factors are given by Kipp (1997).

The case of a free-surface boundary with accretion of fluid by infiltration (precipitation recharge) is handled in an approximate fashion. The recharge fluid flux is specified at the upper boundary of the active grid region. That flux is applied to the cells that contain the free surface at any given time. The associated solution compositions defined for the flux boundary condition combined with the fluid flux determine the amount of each solute component that enters the saturated region through the free surface. The partially saturated flow from the land surface down to the water table is not simulated in PHAST.

D.5.7. Boundary-Condition Compatibility

The set of boundary conditions and well-source conditions for PHAST is diverse, so care must be taken to avoid incompatible combinations when applying multiple boundary conditions to a given cell or a given cell face. As previously mentioned, a boundary cell can have as many as three boundary faces (sections D.1.2. Spatial Discretization and D.5. Boundary Conditions). Thus, a boundary cell could have as many as three different types of boundary conditions, each applied to a different cell face. Only rivers can have multiple instances of the boundary condition that apply to the same face of a boundary cell.

Table D.1.—Compatibility among boundary conditions and wells for a cell

[Yes, one boundary condition of each kind is allowed for a cell; No, this combination of boundary conditions cannot be defined for a cell; One, only one boundary condition of this type can be defined for a cell; Multiple, multiple river segments can affect a cell; --, information can be found in the upper triangle of the table]

Boundary Condition	Specified head	Specified solution composition	Specified flux	Leakage	River leakage	Free surface	Well
Specified head	One	Yes[1]	No	No	No	Yes	No
Specified solution composition	--	One[1]	No	No	No	Yes[1]	No
Specified flux	--	--	One	Yes[2]	Yes	Yes	Yes
Leakage	--	--	--	One	Yes[2]	Yes[3]	Yes
River leakage[4,5]	--	--	--	--	Multiple	Yes	Yes
Free surface	--	--	--	--	--	One	Yes
Well	--	--	--	--	--	--	One

[1] Specified solution composition can only be defined at a specified-head boundary cell.

[2] Combination is allowed provided each condition is on a different cell face.

[3] A cell may not contain the free surface and have leakage through a Z-face.

[4] River leakage only is allowed for unconfined flow simulations (free-surface boundary condition).

[5] River leakage is always through the Z-face of a cell.

Table D.1 shows which types of boundary conditions are compatible for application to a cell. Specified-head boundary conditions are applied to entire cells and, thus, preclude application of any specified-flux, leakage, river-leakage, or well-source condition. Specified solution compositions can only be applied in conjunction with specified-head boundary conditions. Free-surface boundary conditions and well-source conditions also are applied to entire cells, while specified-flux, leakage, and river-leakage boundary conditions are applied to boundary faces of cells. However, only one boundary condition of each type may be applied to a given cell, with the exception of the river-leakage condition, which my have multiple river segments traversing a cell.

River leakage is compatible only with an unconfined (free surface) flow region and is applied only through a Z-face. river-leakage and specified-flux boundary conditions can be used together on a Z-face with a free-surface boundary condition to represent recharge at the land surface. River leakage may not share a boundary face with a leakage boundary condition. If both river-leakage and leakage boundary conditions are defined for the Z-face of a cell, the river-leakage boundary condition will be ignored. Similarly, if both river-leakage and specified-head boundary conditions are applied to a cell, the river-leakage boundary condition will be ignored.

Leakage and specified-flux boundary conditions must be applied to separate boundary faces. A leakage boundary condition cannot be applied to a Z-face for a cell with a free surface. Well-source conditions can be applied to any cell except those with specified-head boundary conditions.

D.6. Initial Conditions

The PHAST simulator solves only the transient forms of the ground-water flow and the component-transport equations; thus, initial conditions are necessary to begin a simulation. Initial potentiometric-head and total-aqueous-concentration distributions are required. Several options are available for specification of these initial conditions.

For the flow equation, the simplest initial condition is a hydrostatic potentiometric-head condition, with a single value specified (**HEAD_IC** data block, **-head**). A more complex initial condition is a water-table head distribution specified over a horizontal plane of the simulation region (**HEAD_IC** data block, **-water_table**). For a water-table condition, the initial head (pressure) distribution in three dimensions will be calculated from the head distribution in the plane by using the assumption of hydrostatic conditions. The most general initial conditions are zone-by-zone or node-by-node potentiometric head specification.

The simulator has an option to solve the flow equation by itself to achieve a steady-state solution for potentiometric head (pressure) compatible to high accuracy with a set of boundary conditions and sources (**STEADY_FLOW** data block). This option can be useful in establishing a head distribution for steady flow that will not change during a transport simulation. Although an initial condition for the head distribution also is required for this case, the steady-state results do not depend on it. There is no option to specify a velocity field as an initial condition.

For the component-transport equations, the initial total aqueous concentration fields need to be specified. In PHAST, initial-solution concentrations are specified for zones by using solution index numbers (**CHEMISTRY_IC** data block, **-solution**). A uniform solution composition can be defined for a zone by using a single solution index number for a zone, or a solution composition that varies linearly in a coordinate direction can be defined. For the latter, solution index numbers are specified at two locations along a coordinate axis, and solution composition is linearly interpolated from the end-member compositions to each node point between the two locations. Points outside the range of the two end-point locations are assigned the solution composition of the nearest end point. Geochemical calculations that include aqueous complexation, mineral equilibria, surface complexation, ion exchange, solid-solution equilibria, gas-phase equilibration, and general kinetic reactions, plus mixing, irreversible reactions, and temperature variation (see discussion of temperature later in this section) may be defined in the chemistry data file; the results of these calculations can be used to define initial conditions for the solutions in the simulation region.

Initial conditions also are specified for the type and amount of solid-phase reactants that are present in each cell of the active grid region (**CHEMISTRY_IC** data block). The reactants are specified for zones in the grid region by using index numbers to identify reactants that are defined in the chemistry data file. The types of reactants include ion exchange (**-exchange**), surface complexation (**-surface**), sets of phases that react to equilibrium (**-equilibrium_phases**), solid solutions (**-solid_solution**), sets of kinetic reactants (**-kinetics**), and gas phases (**-gas_phase**). As with initial solution compositions, it is possible to use linear interpolation between two compositions to distribute reactants for cells within a zone; interpolation can be done with any type of reactant (exchange, surface, equilibrium phases, solid solutions, kinetics, or gas phase).

The units for input in the **SOLUTION** data block are concentration units; internally, all chemical calculations use molality for the concentration unit of solution. The appropriate unit for input in the data blocks **EXCHANGE**, **KINETICS**, **SURFACE**, **EQUILIBRIUM_PHASES**, **SOLID_SOLUTIONS** is moles of reactant per liter of pore space. For each cell in PHAST, the representative porous-medium volume for chemistry contains one kilogram of water, when saturated. Thus, when defining amounts of solid-phase reactants, the number of moles of reactant is numerically equal to the concentration of the reactant (mol/kgw), assuming a saturated porous medium. Note that the concentration of the solid reactants varies spatially as porosity varies, so that if porosity is defined to vary spatially, then logically, the concentrations of solid-phase reactants also must be defined to vary spatially. Molality (mol/kgw) is assumed to equal molarity (mol/L) for all transport calculations.

The chemical calculations of a reactive-transport simulation occur at a specified temperature for each cell. In most cases, the temperature is specified with the **-temperature** identifier in **SOLUTION** data blocks in the chemistry data file, but the temperature of a solution also may be defined with **REACTION_TEMPERATURE** data block. The solutions defined in the chemistry data file are used to define the initial conditions for reactive-transport simulations. Thus, for the chemical calculations of the reactive-transport simulation, a temperature distribution throughout the active grid region, is defined by the temperatures of the solutions used for initial conditions (**CHEMISTRY_IC**, **-solution** identifiers). This temperature distribution for chemical calculations remains in effect for the duration of the reactive-transport simulation. This temperature distribution has no effect on the flow or transport simulations, which are limited to a constant-density, and implicitly constant-temperature, fluid.

D.7. Method of Solution

The ground-water flow equation, the component transport equations, the equilibrium reaction equations, and the kinetic reaction equations form a coupled set of partial differential, ordinary differential, and algebraic equations. For ground water of uniform density, the flow and transport coupling is through the interstitial velocity terms. The component transport equations are coupled to the equilibrium and kinetic reaction equations through chemical source terms, as shown in equation D.4. The chemical source terms are non-linear functions of the chemical compositions of the aqueous and solid phases.

For ground water of uniform density, the coupling between the flow equation and the transport equations is only in one direction, from the flow equation to the transport equations. The potentiometric head (pressure) solution to the flow equation yields the interstitial velocity field, which goes into the component transport equations through the advective transport terms and the dispersion coefficients. This one-way coupling allows the flow equation to be solved independently from all other equations.

Several methods have been used in the literature to solve the transport and reaction equations as summarized by Steefel and MacQuarrie (1996). In PHAST, the calculations for reactive transport are split into a transport calculation step and a chemical-reaction calculation step. This method is referred to as operator splitting with sequential solution (Press and others, 1989). The transport equations (and flow equation) require a sparse-matrix linear equation solver for the finite-difference equations. The chemical-reaction equations require a combination of numerical methods to solve the nonlinear algebraic equations for equilibrium and the ordinary differential equations for kinetics.

D.7.1. Operator Splitting and Sequential Solution

The flow equation can be solved separately from the transport and reaction equations. The three sets of equations that need to be solved simultaneously are the transport equations, the equilibrium reaction equations, and the kinetic reaction equations. The finite-difference equation for flow has the following form:

$$C_f \frac{h^{(n+1)} - h^{(n)}}{\Delta t} - \theta L_f(h^{(n+1)}) = (1-\theta) L_f(h^{(n)}), \tag{D.26}$$

where C_f is the flow storage factor (m^2), θ is the weighting factor for time differencing, and L_f is the spatial discretization of $\nabla \bullet \boldsymbol{K} \nabla$ (m^{-1} s^{-1}).

In PHAST, operator splitting is used to separate the reactive-transport equations into transport equations and chemical-reaction equations. Each reactive transport equation is split into two equations: a solute-transport equation

$$\frac{\partial}{\partial t}(\varepsilon \rho c_j) = \nabla \bullet \varepsilon \boldsymbol{D} \nabla \rho c_j - \nabla \bullet \varepsilon \boldsymbol{v} \rho c_j + q \varepsilon \rho \hat{c}_j ; \quad j = 1, \ldots, N_c, \text{ and} \tag{D.27}$$

a reaction equation for heterogeneous equilibrium and kinetic equations

$$\frac{\partial}{\partial t}(\varepsilon \rho c_j) = -\sum_{e=1}^{N_E} \upsilon_{j,e}^{E} \frac{\partial}{\partial t}(\varepsilon \rho \bar{c}_e) + \sum_{k=1}^{N_K} \upsilon_{j,k}^{K} \varepsilon \rho R_k ; \quad j = 1, \ldots, N_c . \tag{D.28}$$

Note that no reaction terms appear in the transport equation. Discretization of equations D.27 and conversion to units of mass fraction gives the following finite-difference equation for each component:

$$C_s \frac{w_j^{(*)} - w_j^{(n)}}{\Delta t} - \theta L_s(w_j^{(*)}) = (1-\theta) L_s(w_j^{(n)}) + Q \hat{w}_j ; \quad j = 1, \ldots, N_c , \tag{D.29}$$

where C_s is the component storage factor (m^3); w_j is the mass fraction of component *j* (-); L_s is the spatial discretization of $\rho \nabla \bullet \boldsymbol{D} \nabla - \rho \nabla \bullet \boldsymbol{v}$ (kg m^{-3} s^{-1}); $w_j^{(*)}$ is the intermediate mass fraction of component j (-); Q is the source flow rate for the cell (m^3/s); and $\hat{w}_j$ is the mass fraction of component *j* in the source water (-). Discretization of equations D.28 yields

$$c_j^{(n+1)} - c_j^{(*)} = -\sum_{e=1}^{N_E} \upsilon_{j,e}^{E} (\bar{c}_e^{(n+1)} - \bar{c}_e^{(n)}) + \sum_{k=1}^{N_K} \upsilon_{j,k}^{K} \int_{t^{(n)}}^{t^{(n+1)}} R_k dt ; \ j = 1, \ldots, N_c . \tag{D.30}$$

In PHAST, the set of equations D.26, D.29, and D.30 plus the algebraic equations for chemical equilibria and rate expressions are solved sequentially in three steps. First, the flow equation D.26 is solved for the potentiometric head (pressure) values. The heads are used to calculate the interstitial velocity values that are used in the component transport equations. Second, each transport equation (D.29) is solved individually for the total aqueous concentrations of the component. And third, the equilibrium and kinetic reaction equations (D.30 plus the algebraic equations for chemical equilibria and rate expressions) are solved. This completes the simulation of one time step. The process is repeated for each time step for the duration of simulation.

In PHAST, no iterations are performed between the second and third steps of the sequential solution. This combination of operator splitting without iteration is known as the sequential, noniterative approach (SNIA) as described by Yeh and Tripathi (1989). Some reactive-transport algorithms include reaction terms in the transport equations and iterate between the second and third steps in the sequential solution (Steefel and MacQuarrie, 1996), known as the sequential iterative approach (SIA).

SNIA uses much less memory than a fully coupled, simultaneous solution of all the transport and reaction equations. SNIA also is faster per time step than the fully coupled approach because the size of the matrices that are solved are much smaller. (It is possible that the fully coupled approach may be faster if fewer time steps are needed by that approach.) However, SNIA introduces an operator-splitting error of the order of the time-step length. This error can be reduced by iteration (SIA), Strang splitting (Steefel and MacQuarrie, 1996), or by use of smaller time steps. Only the last option is available in PHAST. In many cases, it is more critical to minimize the discretization error of the finite-difference approximations to the transport equations than to minimize the operator-splitting error.

Analysis of one-dimensional transport with constant coefficients for the special case of one linear, equilibrium sorption reaction, yields an effective dispersion coefficient from operator splitting given by

$$D_{nos} = \frac{v_x^2 \Delta t}{2R}(2\theta R - 1)\text{, with} \tag{D.31}$$

$$R = 1 + \frac{\rho_b K_e}{\varepsilon}, \tag{D.32}$$

where D_{nos} is the numerical dispersion coefficient from operator splitting (m^2/s); R is the retardation factor for linear sorption (-); ρ_b is the dry bulk density of the porous medium (kg/m^3); and K_e is the linear equilibrium sorption coefficient (m^3/kg). The retardation factor also expresses the ratio of the effective transport velocity of the sorbed component relative to the interstitial velocity of the ground water.

For numerical accuracy, the numerical dispersion from operator splitting should be much less than the physical dispersion. Thus, for centered-in-time differencing,

$$CrPe \ll \frac{2R}{R-1}\text{, which reduces to} \tag{D.33}$$

$$CrPe \ll 2, \tag{D.34}$$

for large value of *R*. For backward-in-time differencing,

$$CrPe \ll \frac{2R}{2R-1}\text{, which reduces to} \tag{D.35}$$

$$CrPe \ll 1, \tag{D.36}$$

for large value of *R*. Although these equations are from a very restricted reaction case, they can give guidance for discretization of more complex reaction systems.

In SNIA, the reaction terms have been completely removed from the transport equations. Removing these terms may cause overshoot and undershoot in the intermediate concentrations in the transport solution, especially in cases of strong sorption or strong partitioning of components. The undershoot concentrations can be negative, which will result in mass-balance errors when those concentrations are set to zero in the chemical-reaction calculation. Overshoot concentrations may cause nonconvergence in the solution of the algebraic chemical equations. The remedy available in PHAST for these overshoot and undershoot problems is to use a smaller time step.

D.7.2. Linear-Equation Solvers for Flow and Transport Finite-Difference Equations

The set of linear equations for flow or transport of each component includes one finite-difference equation for each node in the grid region. The set of finite-difference equations are solved in two steps: (1) a reduced matrix is formed by renumbering the nodes and performing a partial block Gaussian elimination, which results in a savings in computation time and computer-storage requirements, and (2) the reduced matrix equation is solved by either a direct or iterative linear-equation solver.

When using the direct solver, the reduced matrix is formed by renumbering the nodes by alternating diagonal planes (Price and Coats, 1974), followed by the partial Gaussian elimination. This reduced matrix is solved by complete Gaussian elimination. This method is referred to as the D4 solution technique described in Kipp (1987). The workload and storage requirements for the direct solver are completely defined by the number of nodes and configuration of the active grid region.

When using the iterative solver, the reduced matrix is formed by renumbering the nodes by alternating diagonal planes in a zig-zag fashion, followed by the partial Gaussian elimination. This reduced matrix is solved by a generalized conjugate gradient algorithm. This method is referred to as the D4 zig-zag solution technique described in Kipp and others (1992, 1994) and Kipp (1997). The workload and storage requirements for the iterative solver depend not only on the number of nodes, but also on the user-specified solver parameters.

The iterative algorithm is a restarted ORTHOMIN method with incomplete lower-upper (ILU) triangular factorization preconditioning as described by Meijerink and van der Vorst (1977), Elman (1982), and Behie and Forsyth (1984). The ORTHOMIN method calculates an approximate solution vector that is based upon a set of orthogonal search directions. For the full ORTHOMIN method, the number of search directions is equal to the iteration count, which leads to workload and storage requirements that increase linearly with iteration count. To limit the workload and storage requirements, a modified algorithm, ORTHOMIN(*m*), is implemented by simply restarting ORTHOMIN every *m* iterations. To restart the ORTHOMIN(*m*) algorithm, the latest solution vector is used as the initial guess for the next cycle. The restart interval *m* is specified by the user (**-save_directions**, **SOLUTION_METHOD** data block). Increasing this interval may accelerate convergence, but also will increase the computer memory requirements.

The convergence criterion of the iterative solver is based on the Euclidean norm of the residual vector being small relative to the norms of the linear-equation matrix, the right-hand-side vector, and the initial solution vector (Barrett and others, 1994, p. 54). Mathematically, this is expressed as

$$\left\|\boldsymbol{r}^{(\nu)}\right\| \le \tau(\|\boldsymbol{A}\| \cdot \left\|\boldsymbol{x}^{(\nu)}\right\| + \|\boldsymbol{b}\|)\,, \tag{D.37}$$

where $\boldsymbol{r}^{(\nu)}$ is the residual vector at iteration ν (appropriate units); τ is the convergence tolerance (-); $\boldsymbol{A}$ is the matrix of coefficients of the difference equations (appropriate units); $\boldsymbol{b}$ is the right hand side vector of the difference equations (appropriate units); $\boldsymbol{x}^{(\nu)}$ is the solution vector at iteration ν (appropriate units); $\|\ \|$ is the Euclidean norm for a vector or the Frobenius norm for a matrix (Stoer and Bulirsch, 1993, p.184).

The criterion of equation D.37 yields the forward error bound of

$$\left\|\boldsymbol{e}^{(\nu)}\right\| \le \tau\left\|\boldsymbol{A}^{-1}\right\| \cdot (\|\boldsymbol{A}\| \cdot \left\|\boldsymbol{x}^{(\nu)}\right\| + \|\boldsymbol{b}\|)\,, \tag{D.38}$$

where $\boldsymbol{e}^{(\nu)}$ is the error in the solution vector at iteration ν (appropriate units) and $\boldsymbol{A}^{-1}$ is the inverse of the coefficient matrix of the difference equations (appropriate units). Thus, the norm of the error in the solution vector is related to the norm of the residual vector under this criterion. Unfortunately, it is usually not feasible to compute the norm of the inverse of the coefficient matrix; therefore, the convergence tolerance must be determined empirically. The convergence tolerance τ is specified by the user (**-tolerance**, **SOLUTION_METHOD** data block). Once the convergence tolerance τ has been specified, it is used in equation D.37 to define the criterion for convergence.

D.7.3. Solving Equilibrium and Kinetic Chemical Equations

Chemical-reaction equations are solved independently from the flow and transport equations. In addition, chemical-reaction equations are solved independently for each node in the active grid region. In the case of only equilibrium reactions, a Newton-Raphson method is used to solve the nonlinear mass-action equations and mass-balance equations that describe equilibrium. See Parkhurst and Appelo (1999) for details of the Newton-Raphson algorithm that is used in PHREEQC and PHAST.

If kinetic reactions are simulated then a set of ordinary differential equations must be integrated over the time step, in addition to solving the equilibrium equations. Two methods are available for integrating the rate equations, a 5th order Runge-Kutta algorithm (Parkhurst and Appelo, 1999) and an implicit algorithm for stiff differential equations based on Gear’s method (Cohen and Hindmarsh, 1996). These methods take the time step defined for the flow and transport equations and divide it into sub time steps. The size of these sub time steps is determined automatically to satisfy a user-specified tolerance on the estimated errors of integration. At each sub time step, the rates of reaction are calculated and the kinetic mole transfers resulting from kinetic reactions for the sub time step are calculated. The kinetic mole transfers are added (or removed) from solution, after which all of the equilibrium equations (homogeneous and heterogeneous) are solved simultaneously. Additional sub time steps are simulated until the integration has completed the entire time interval specified by the time step of the flow and transport simulation.

D.8. Accuracy from Spatial and Temporal Discretization

The finite-difference equations that approximate the flow and transport equations (D.26 and D.29) are unconditionally stable when using both backwards-in-time and upstream-in-space or centered-in-time and centered-in-space difference approximations; numerical solutions will not grow unbounded in time or space. The accuracy of the finite-difference equations depends on the choice of cell size and time step length. Avoiding numerical oscillation is a prime consideration when using centered-in-time and centered-in-space differencing, while avoiding excess numerical dispersion is necessary for accuracy when using backward-in-time and upstream-in-space differencing. Guidelines are given in sections D.1.1 and D.1.2 for limiting cell size and time step length to meet these objectives. The operator-splitting algorithm introduces its own temporal truncation error, but no guidelines for a general reaction system are available. Guidelines from the restricted case of linear equilibrium sorption (section D.7.1) indicate the effect of a retarded solute-transport rate on time-step selection for accuracy. Results for this case show that the effective numerical dispersion from operator splitting is always greater than or equal to the numerical dispersion from backward-in-time differencing. Note that avoiding numerical oscillations does not ensure an accurate numerical solution. The user must test for a sufficiently accurate solution by reducing the cell sizes and time step until no significant changes in the solution occur. For field-scale simulations, it is often not practical to refine the grid enough to avoid oscillations or reduce numerical dispersion sufficiently. Some numerical dispersion must be tolerated. It is usually feasible to reduce the time step to avoid oscillations and sufficiently reduce numerical dispersion from operator splitting.

A free-surface boundary condition introduces non-linearity into the finite-difference equations. Because the free-surface coefficients are calculated explicitly in time, there is a stability limit on the time-step length that can be determined only empirically. The critical regions are where the rate of change of the free surface is large.

The river-leakage boundary condition also introduces non-linearity into the finite-difference equations from the equation of flux as a function of head difference between the river and the aquifer. Although this flux is computed semi-implicitly, large rates of change of head in the river or the aquifer may require small time steps for stability. Note that a stable solution is not necessarily an accurate solution.

D.9. Global Mass-Balance Calculations

The discretized ground-water flow equation and the transport equations for components represent fluid and solute mass balances over each cell. Summing over the cells and integrating over time yields global-balance equations that relate the total change of mass of ground-water and components to the net boundary flow, the net injection by wells, and the net formation by reactions. The temporal integration is done over each time step and the cumulative balance masses are simply the sums of the corresponding masses from each time step. The fluid or component global-balance residual is defined as the imbalance between the change in the mass of fluid or solute in the simulation region and the net inflow of fluid or solute plus the net source of solute from reactions. A positive residual means that there is an excess of fluid or components present over what would be expected on the basis of transport across boundaries and sources from reactions. A fractional residual is defined as the ratio of the residual to the inflow, outflow, or magnitude of accumulation, whichever is larger.

Global balance information is calculated for each time step and for the cumulative duration of the simulation. The information produced by the balance calculation includes change in mass in the system, the mass flows of inflow and outflow, and the mass change due to reactions. Change in mass due to net inflow (inflow minus outflow) is aggregated by each type of boundary condition. Numerical integration of the boundary flow rates over a time step and over the cumulative time is done with an algorithm that depends on the weighting factor used for temporal discretization of the governing flow and component-transport equations. Thus, for centered-in-time differencing the trapezoidal algorithm is used and for backward-in-time differencing the rectangular algorithm is used with integrand evaluation at the end of the time step. Having the integration algorithm depend on the temporal weighting factor makes the global mass-balance calculations consistent with the formulation of the finite-difference equations and thus gives an exact mass-balance for each cell over a time step.

The fluid-flow at specified-head boundary cells is obtained by evaluating the residual of the finite-difference flow equation. The component mass flows at specified concentration boundary cells are obtained by evaluating the residual of the transport equations. These residuals are the masses of fluid and components that are necessary to satisfy the fluid-balance and solute-balance equations. Thus, the fluid mass-balance equation is satisfied exactly for each of the specified-head boundary cells, and the component mass-balance equation is satisfied exactly for each of the specified-concentration boundary cells. To make the global-balance calculations for the components compatible with the operator-splitting algorithm, the boundary mass flows are based on the intermediate concentrations resulting from equation D.29. The change in component mass from reactions is taken to be the difference between the mass of the component before and after the reaction step.

The information produced by the global-balance calculation is printed to the file *prefix*.**O.bal**. The primary use of the global-balance calculations is to identify changes in mass in the system relative to reactions and mass flows (kg) through the various types of boundary conditions. Mass-accumulation and mass-flow rates (kg/s) also are tabulated. The utility of the fractional residuals is not great. It is more informative to compare the residuals with each of the various flows, sources, and accumulations in the region.

Flow and component mass balances with small residuals are a necessary, but not a sufficient, condition for an accurate numerical simulation. Because the system equations are a balance for each cell, and the methods used for integrating boundary flows and reaction sources over a time step are compatible with the finite-difference equations and the operator-splitting method, and because the fluxes between the cells are conservative, errors in the global-balance equations may result from the following remaining causes: (1) explicit treatment of the water-table elevation for unconfined flow, (2) explicit treatment of well flow rates, (3) explicit treatment of the nonlinear limit for river leakage, (4) explicit treatment or neglect of the cross-dispersive flux terms, (5) use of the iterative-matrix equation solver to obtain an approximate numerical solution, and (6) roundoff error in special cases, such as wide variation in parameter magnitudes. The first three items are usually the major cause of mass-balance errors. Errors caused by discretization in time or space will not be revealed by these global-balance calculations. However, the inaccuracies resulting from a time step that is too long under conditions of significant nonlinearity will be evident. Significant nonlinearity can be caused by large changes in the water-table location or by non-linear reaction mechanisms.

D.10. Nodal Velocity Calculation

A secondary dependent variable is the interstitial velocity field, which is obtained from the gradient of the potentiometric head field. The numerical implementation of PHAST uses finite differences of the head values, which give velocity values at the cell boundaries for all cell faces internal to the boundary of the active grid region. For all internal cells, interstitial velocities are calculated at the six cell faces. Vector components of interstitial velocity at internal nodes of the active grid region are interpolated from velocity values at the six cell faces.

For cells with one face at the boundary of the active grid region, the velocity vector component normal to that face is calculated from the boundary flow rate. These flow rates have been calculated previously for each type of boundary condition. Because all boundaries of the active grid region are at planes of nodes, the velocity component normal to the face needs no interpolation.

Only the net boundary flow rate for all boundary faces of a cell are known for a cell with a specified-head boundary condition. For specified-head cells with two or three boundary faces, the net boundary flow rate is apportioned among the faces in proportion to the magnitude of the flow rate through the cell face opposite to the given boundary face.

For nodes along inside edges of the boundary of the active grid region with a no-flux boundary condition, the nodal velocity is taken to be zero. In actuality, the velocity components have discontinuous jumps from zero to finite values at the inside edges of the boundary.

For unconfined flow systems, the cells containing the free-surface boundary are partially filled. No velocity vector components are interpolated to nodes that are located above the free surface, because these nodes are outside the boundary of the saturated computation region. If the node is at or below the free surface, the Z-component of nodal velocity is interpolated from the velocity at the bottom cell face and the vertical velocity of the free surface. The free-surface vertical velocity is calculated from a finite-difference approximation to the rate of movement of the free surface over the time step. The free-surface velocity for a cell is not calculated for the time step during which the node becomes covered by the free-surface boundary.

Finally, interpolation to the node is done for velocity components parallel to the boundary faces of the active grid. This computation of nodal velocity-vector components enables visualization of the interstitial velocity field over the entire simulation region. Although these nodal velocity calculations are not conservative, they are used only for visualization.