

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 possibly the *prefix.xyz.chem*, where it is possible for the user to print concentration in alternative units.

The files named *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 named *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 MODELVIEWER (Appendix A) and data can be extracted to an ASCII file with the program PHASTHDF (Appendix B).

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

File name	Contents
<i>prefix.log</i>	Copies of the <i>prefix.trans.dat</i> and <i>prefix.chem.dat</i> input data files and all error or warning messages
<i>prefix.h5</i>	Grid and boundary-condition information, velocity field, potentiometric head field, and selected chemical data
<i>prefix.head.dat</i>	Potentiometric heads at the final time step of the run in a form that can be read as initial head conditions in subsequent runs
<i>prefix.O.bal</i>	Fluid and chemical component regional mass balances, total balances, and balances listed by boundary condition type
<i>prefix.O.bcf</i>	Boundary-condition fluid and solute flow rates for cells listed by boundary-condition type
<i>prefix.O.chem</i>	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 simulations.
<i>prefix.O.comps</i>	Transient total dissolved concentrations for chemical elements (components)
<i>prefix.O.head</i>	Transient potentiometric heads
<i>prefix.O.kd</i>	Static fluid conductances for X-, Y-, and Z-coordinate directions and solute dispersive conductances at cell faces

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

File name	Contents
<i>prefix.O.probdef</i>	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
<i>prefix.O.vel</i>	Interstitial velocities for X-, Y-, and Z-coordinate directions across cell faces and velocities in the X-, Y-, and Z-coordinate directions at nodes
<i>prefix.O.wel</i>	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
<i>prefix.xyz.chem</i>	Selected initial condition and transient chemical data
<i>prefix.xyz.comps</i>	Initial condition and transient cell concentrations for each chemical element (component)
<i>prefix.xyz.head</i>	Initial condition and transient potentiometric heads
<i>prefix.xyz.vel</i>	Steady-state or transient velocity-vector components interpolated to grid nodes
<i>prefix.xyz.wel</i>	Transient concentration data for wells
<i>selected_output</i>	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 <i>prefix.xyz.chem</i> .)

The types of transient information that may be written include the solution-method information (*prefix.log*), the conductance and dispersion-coefficient 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 used to define initial head conditions for subsequent simulation runs. 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.

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, which converts the flow and transport data file in keyword format into an intermediate input file, *Phast.tmp*, and (2) PHAST, which performs the flow and reactive-transport simulations. The script is invoked from a command line with either one or two arguments. The first argument is the prefix and the optional second argument is the thermodynamic database file name:

```
phast prefix [database_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.

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 logical flow of the calculation is described further in the section D.7.1.

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, 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.

The thermodynamic database file is used to define 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, 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 can also 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 define 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, T.J. 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 found 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 that are used in initial and boundary conditions of the reactive-transport simulation. Initial geochemical calculations can be used to generate the solutions, gas phases, or assemblages that are used in the reactive-transport simulation for initial and boundary conditions. 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).

Initial conditions for the reactive-transport simulation are defined with **SOLUTION**, **EXCHANGE**, **KINETICS**, **SURFACE**, **EQUILIBRIUM_PHASES**, **SOLID_SOLUTION**, 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.

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 <i>index</i>	Defines a mineral assemblage	CHEMISTRY_IC
EXCHANGE <i>index</i>	Defines an assemblage of ion exchangers	CHEMISTRY_IC
GAS_PHASE <i>index</i>	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 <i>index</i>	Defines a set of kinetic reactions	CHEMISTRY_IC
RATES	Defines rate expressions for kinetic reactions	—
REACTION <i>index</i>	Adds specified amounts of reactants	—
SAVE solution <i>index</i>	Saves solution composition that is the result of a reaction simulation	CHEMISTRY_IC , FLUX_BC , LEAKY_BC , SPECIFIED_VALUE_BC , RIVER , WELL
SAVE exchange <i>index</i>	Saves reaction simulation results to define	CHEMISTRY_IC
SAVE equilibrium_phases <i>index</i>	new compositions of exchangers, equilibrium phases, gas phase, kinetic reactants,	
SAVE gas_phase <i>index</i>	solid solutions, and surfaces	
SAVE kinetics <i>index</i>		
SAVE solid_solution <i>index</i>		
SAVE surface <i>index</i>		

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
SELECTED_OUTPUT	Defines data to be written to the <i>prefix.h5</i> and <i>prefix.xyz.chem</i> files	—
SOLID_SOLUTIONS <i>index</i>	Defines an assemblage of solid solutions	CHEMISTRY_IC
SOLUTION <i>index</i>	Defines a solution composition	CHEMISTRY_IC, FLUX_BC, LEAKY_BC, SPECIFIED_VALUE_BC, RIVER, WELL
SURFACE <i>index</i>	Defines sets of surface complexation reactions	CHEMISTRY_IC
TITLE	Defines a character string to annotate the calculations	—
USE <i>keyword index</i>	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 <i>prefix.h5</i> and <i>prefix.xyz.chem</i> files	—

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. 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 calculations 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.

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 is also 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.

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, which 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. 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.

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.

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 for one or more simulation periods. The flow and transport data file is also 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).

It requires only a single change in the flow and transport data file to switch from a flow, 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. Multiple simulation periods allow for time-varying boundary conditions. Although constant for each simulation period, boundary-condition values, such as head, flux, pumping rate, and solution composition, may be changed from simulation period to simulation period to accommodate 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 a simulation period
FLOW_ONLY	Defines the simulation to be only a flow -only simulation or a complete reactive-transport simulation
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 <i>prefix.xyz.chem</i> and <i>prefix.O.chem</i> files
RIVER	Defines a river boundary condition
SOLUTION_METHOD	Defines numerical method for solving the finite-difference equations
SPECIFIED_VALUE_BC	Defines specified-value boundary conditions
STEADY_FLOW	Defines flow field to be steady state as determined from the boundary conditions
TIME_CONTROL	Defines time step and the time of the end of a simulation period
TITLE	Defines a character string describing the simulation
UNITS	Defines units of input data
WELL	Defines location, open intervals, and pumping or injection rate for a well

4.1 Organization of the Flow and Transport Data File

PHAST uses sets of keyword data blocks in the flow and transport data file to define one or more simulation periods. The set of data blocks for each simulation period is concluded with an **END** keyword. In general, data defined in a data block for one simulation period will remain in effect until explicitly changed by another data block in a subsequent simulation period. 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, which may change with simulation period. The **TIME_CONTROL** data block must be included for each simulation period. Data blocks that control printing to output files may be redefined for each simulation period.

Data blocks that define the input units, grid, porous-media properties, initial conditions, numerical methods, steady-state flow, and the free-surface boundary condition apply to the entire simulation. These data blocks and the printing selections for title and initial conditions must be defined in the first simulation period, and can not be redefined in subsequent simulation periods. These data blocks include **CHEMISTRY_IC**, **FLOW_ONLY**, **FREE_SURFACE_BC**, **GRID**, **HEAD_IC**, **MEDIA**, **PRINT_INITIAL**, **SOLUTION_METHOD**, **STEADY_FLOW**, **TITLE**, and **UNITS**.

Data blocks for boundary conditions include both static and transient data. If they are to be used in the simulation, the boundary condition data blocks must be defined in the first simulation period to 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 river bed—and transient data for the first simulation period—head, solution composition, flux, and injection/pumping rate. The transient data may be varied over the course of a simulation by using the data blocks in subsequent simulation periods; however, the static properties can not be changed. The boundary-condition data blocks that must be included in the first simulation period and may be included in subsequent simulation periods are **FLUX_BC**, **LEAKY_BC**, **RIVER**, **SPECIFIED_VALUE_BC**, and **WELL**.

The **TIME_CONTROL** data block must be included in every definition of a simulation period and is used to specify the time step and the time of the end of the simulation period. The data blocks defining print frequencies

and sets of nodes for which data are written to output files—**PRINT_FREQUENCY** and **PRINT_LOCATIONS**—may be included in the set of data blocks that define any simulation period.

With one major exception, the order of keyword data blocks is arbitrary within a set of data blocks that defines a simulation period. 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 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.

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.

Figure 4.1.—Grid region showing boundary nodes and one shaded element.

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 and on the boundary of active grid region. Results are not calculated for any inactive nodes.

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

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 share the node 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.

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 a 1 with each node in the zone. Properties are applied only to those nodes with nonzero 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 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 1s and 0s in the mask must also be changed, which is not a trivial task.

4.2.1.1 Use of zones for defining porous-media properties

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

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.

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

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 up to 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 a cell face conductance is a weighted average of up to four different values (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.

4.2.1.2 Use of zones for defining initial- and boundary-condition properties

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.

Initial conditions and specified-value, 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 are used to define boundary conditions where desired within the grid region. Zones are not used in the definition of well and river boundary conditions.

4.2.1.3 Property definitions

For each zone, spatially distributed properties can be defined by any of four 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 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, or (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. 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 or centroid is outside the coordinate range defined by the two points, the composition from the closest point is applied.

For methods 2 and 3, 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

Rivers are defined by a series of X–Y coordinate points, which 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

explicitly, either by interpolation from other river points, or by default values. Well location is defined by the X–Y coordinates of the well head. Data required for each well include pumping or injection rate, depth or elevation data to define open (or screened) intervals, well diameter, and numerical method for allocating flow to grid layers.

4.3 Transient Data

PHAST allows an unlimited number of simulation periods to represent transient boundary conditions. During a single simulation period, boundary conditions are constant; however, some properties associated with boundary conditions may be redefined for each simulation period. Properties for boundary conditions that may be transient include head, solution composition, flux, and injection or pumping rate. For example, the following can be changed for a simulation period: the head associated with a specified-value or leaky boundary condition, the flux of water through a flux boundary condition, the chemical composition of water associated with a river, or the pumping rate of a well. Many properties of boundary conditions are static, and are fixed for the entire simulation. Examples of static properties include the hydraulic conductivity in a leaky boundary condition, the width of a river, 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; identifiers are bold lowercase and preceded with a hyphen. Words in *italics* are input fields that are variable and depend on user selection of appropriate values. Items in brackets ([]) are optional input fields. Mutually exclusive input fields are enclosed in parentheses and separated by the word “or”. 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 input data file. All line numbers associated with example input in the documentation are for identification purposes only. Input data files for PHAST do not contain any line numbering. 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_VALUE_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.

Units—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 pound character (#) is used to delimit the beginning of a comment. All characters in the line which 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 an end-of-line, causes the input processor to ignore the end-of-line. By ignoring the end-of-line character, 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.

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_SOLUTION**, **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. The **CHEMISTRY_IC** data block is mandatory for all reactive-transport simulations. This keyword data block must be defined for the initial simulation period if transport and chemistry are included in the simulations and can not be redefined in subsequent simulation periods.

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 60*2
Line 5:      -surface  file  surface.fil
Line 6:      -exchange X      1      0.0  5      10.0
Line 7:      -gas_phase 2
Line 8:      -solid_solutions      5
Line 9:      -kinetics  2
Line 10:     -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 are initially present in the cells in the zone. 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 the four 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), (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), or (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. 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 are initially present in the cells of the zone. Optionally, **equilibrium_phases, -eq[uilbrium_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 assem-

blages can be entered for the zone with any of the four 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), (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), or (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. Lines 3 and 4 use method 2.

Line 4: *property continued*

property continued—Continuation of property definition.

Note the use of the repeat factor in line 4.

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

-surface—This identifier is used to specify index numbers that define surface-assemblage compositions that are initially present in the cells of 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 four 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), (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), or (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. Line 5 illustrates method 3.

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

-exchange—This identifier is used to specify index numbers that define exchange-assemblage compositions that are initially present in the cells of 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 four 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), (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), or (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. 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 are initially present in the cells of 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 four 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), (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), or (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. Line 7 illustrates method 1.

Line 8: **-solid_solutions** *property*

-solid_solutions—This identifier is used to specify index numbers that define solid-solution assemblages that are initially present in the cells of 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 four 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), (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), or (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. Line 8 illustrates method 1.

Line 9: **-kinetics** *property*

-kinetics—This identifier is used to specify index numbers that define sets of kinetic reactants that are initially present in the cells of 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 four 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), (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), or (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. Line 9 illustrates method 1.

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 nonzero number) and 0s is used to include and exclude nodes from the zone definition. The array can be defined by any of the 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), (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. Line 10 illustrates method 3.

Notes

CHEMISTRY_IC defines initial conditions and must occur before the first **END** keyword in the flow and transport data file. All 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. 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. A negative index number for an entity is interpreted to mean that entity is absent in the zone. 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 pure-phase assemblages, moles of each mineral phase; for exchangers, moles of exchange sites and moles of exchanged ions; for solid-solutions, moles of each component of each solid solution; for surfaces, moles of surface sites, surface area, and moles of sorbed elements; and for gas phases, moles of each gas component.

Example problems

The **CHEMISTRY_IC** data block is used in the example problems 6.2, 6.3, 6.4, and 6.5.

END

This keyword has no associated data. It ends the data input for a simulation period. Data for additional simulation periods may follow in the input data set; each simulation period is terminated with an **END** keyword.

Example problems

The **END** keyword is used in the example problems 6.2, 6.3, 6.4, and 6.5.

FLOW_ONLY

This keyword data block is a switch used to toggle transport and reaction calculations on or off. If the **FLOW_ONLY** data block is not included, the simulation is assumed to be a reactive-transport simulation. If **FLOW_ONLY** is true, 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 is only pertinent to transport calculations. The diffusivity is a scalar value that is uniform over the entire grid region; it is not spatially variable. This keyword data block can only be defined at the initial simulation period and can not be redefined in subsequent simulation periods.

Example

```
Line 0:  FLOW_ONLY      false
Line 1:  -diffusivity   1e-9      # m^2/s
```

Explanation

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

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

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

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

-diffusivity—This identifier is used to specify molecular diffusivity of all chemical solutes. The diffusivity only applies when **FLOW_ONLY false**. Default is 10^{-9} m²/s. Optionally, **diffusivity** or **-d[iffusivity]**.

diffusivity—Molecular diffusivity, m²/s.

Notes

The diffusivity is used in simulations only if the transport equations are solved, that is, when **FLOW_ONLY** is false. The units for diffusivity are m^2/s ; no option for alternative units is available.

Example problems

The **FLOW_ONLY** data block is used in the example problems 6.2, 6.3, 6.4, and 6.5.

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 is also required. This keyword data block is optional and is only needed if specified flux boundaries are to be included in the simulation. All specified flux boundary conditions must be defined for the first simulation period. In subsequent simulation periods, the flux and associated solution can be redefined for zones where these parameters vary in time.

Example

```

Line 0: FLUX_BC
Line 1:      -zone      0      10      10      10      10      10
Line 2:      -flux      -3.5e-3
Line 3:      -associated_solution  5
Line 4:      -face      Z
Line 5:      -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** *property*

-flux—This identifier is used to specify 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**). Optionally, **flux** or **-fl[ux]**.

property—The volumetric flux for the nodes selected by the zone can be entered with any of the 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), (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, value₁, distance₁, value₂, distance₂**, a value is linearly interpolated from *value₁* and *value₂* for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance₁* and *distance₂* along the specified coordinate direction; for locations outside the range of *distance₁* to *distance₂*, the value at the nearest point (*distance₁* or *distance₂*) is used. Line 2 illustrates 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 3: **-associated_solution** *property*

-associated_solution—This identifier is used to specify index numbers that define the solution composition for the flux of water into the active grid region through the specified flux boundary. Optionally, **associated_solution**, **-a[ssociated_solution]**, **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 the four 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), (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), or (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. Line 3 illustrates method 1.

Line 4: **-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 5: **-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 nonzero number) and 0s is used to include and exclude nodes from the zone definition. The array can be defined by any of the 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), (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. Line 5 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. 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 value 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 can not be applied to the positive Z cell face, and (4) both leaky and flux

conditions can not 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.

Example problems

The **FLUX_BC** data block is used in the example problem 6.5.

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** data block is not included, confined flow is simulated. This data block can only be defined at the first simulation period and can not be redefined in subsequent simulation periods.

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[true]** or **f[false]**.

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.

Example problems

The **FREE_SURFACE** data block is used in the example problems 6.2, 6.3, 6.4, and 6.5.

GRID

This keyword data block is used to define the finite-difference node locations for the simulation. This keyword data block is mandatory for all simulations. It must be defined for the first simulation period and can not be redefined in subsequent simulation periods.

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.

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. 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 8.7 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 1 or 2 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 *prefix.O* files. 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 3 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. However, 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 6.2, 6.3, 6.4, and 6.5.

HEAD_IC

This keyword data block is used to define the initial head conditions in the grid region. This keyword data block is mandatory for all simulations. It must be defined for the first simulation period and can not be redefined in subsequent simulation periods.

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 the 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),

(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 nonzero number) and 0s is used to include and exclude nodes from the zone definition. The array can be defined by any of the 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), (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. 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), 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** defines the initial head conditions and must occur in the definition of the first simulation period (before the first **END** keyword) in the flow and transport data file. 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 topmost X–Y plane of nodes. For each vertical stack of nodes in the grid region, the initial water-table head of the topmost node is assigned to all nodes in the stack.

Example problems

The **HEAD_IC** data block is used in the example problems 6.2, 6.3, 6.4, and 6.5.

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 is also required for each leaky-boundary cell. This keyword data block is optional and is only needed if leaky boundary conditions are included in the simulation. All leaky boundary conditions must be defined for the first simulation period. In subsequent simulation periods, head and associated solution can be redefined for zones where these parameters vary in time.

Example

```

Line 0: LEAKY_BC
Line 1:   -zone      0      10      10      10      10      10
Line 2:   -head          100
Line 3:   -thickness    X      100    0      50      10
Line 4:   -hydraulic_conductivity  file leaky_k.fil
Line 5:   -associated_solution  5
Line 6:   -face          X
Line 7:   -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** *property*

-head—This identifier is used to specify the head on the exterior side of the leaky boundary layer.

Optionally, **head** or **-he[ad]**.

property—The head on the exterior side of the leaky boundary layer can be entered for the zone with any of the 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), (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, value₁, distance₁, value₂, distance₂**, a value is linearly interpolated from *value₁* and *value₂* for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance₁* and *distance₂* along the specified coordinate direction; for locations outside the range of *distance₁* to *distance₂*, the value at the nearest point (*distance₁* or *distance₂*) is used. Line 2 illustrates method 1. Units, L, are defined by the **-head** identifier in the **UNITS** data block.

Line 3: **-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 the 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), (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 mul-

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 4. Units, L, are defined by the **-leaky_thickness** identifier in the **UNITS** data block.

Line 4: **-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 the 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), (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 4 illustrates method 3. Units, L/T, are defined by the **-leaky_hydraulic_conductivity** identifier in the **UNITS** data block.

Line 5: **-associated_solution** *property*

-associated_solution—This identifier is used to specify solution index numbers that associate a solution composition with the flux of water into the active grid region through the leaky boundary. Optionally, **associated_solution**, **-a[ssociated_solution]**, **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 the four 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), (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), or (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. Line 5 illustrates method 1.

Line 6: **-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 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 nonzero number) and 0s is used to include and exclude nodes from the zone definition. The array can be defined by any of the 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), (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, value₁, distance₁, value₂, distance₂**, a value is linearly interpolated from *value₁* and *value₂* for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance₁* and *distance₂* along the specified coordinate direction; for locations outside the range of *distance₁* to *distance₂*, the value at the nearest point (*distance₁* or *distance₂*) is used. Line 7 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 value 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 can not be applied to the positive Z cell face, and (4) both leaky and flux conditions can not be applied to the same cell face.

Example problems

The **LEAKY_BC** data block is used in the example problem 6.5.

MEDIA

This keyword data block is used to define media properties, including hydraulic conductivities, porosities, specific storage, and dispersivities. These spatial properties are applied to elements, not cells (see sections 4.2 and D.1.1). This keyword data block is mandatory for the first simulation period for all simulations. It can not be redefined in subsequent simulation periods.

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 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 1a:     -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 the 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), (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, value₁, distance₁, value₂, distance₂**, a value is linearly interpolated from *value₁* and *value₂* for each element centroid in the zone; interpolation is proportional to the fractional distance of the element centroid between *distance₁* and *distance₂* along the specified coordinate direction; for centroids outside the range of *distance₁* to *distance₂*, the value at the nearest point (*distance₁* or *distance₂*) 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 the 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), (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, value₁, distance₁, value₂, distance₂**, a value is linearly interpolated from *value₁* and *value₂* 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 the 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), (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 the 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), (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 for the zone can be entered for the zone with any of the 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), (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 the 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), (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**, *value₁*, *distance₁*, *value₂*, *distance₂*, a value is linearly interpolated from *value₁* and *value₂* for each element centroid in the zone; interpolation is proportional to the fractional distance of the element centroid between *distance₁* and *distance₂* along the specified coordinate direction; for centroids outside the range of *distance₁* to *distance₂*, the value at the nearest point (*distance₁* or *distance₂*) 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 the 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), (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**, *value₁*, *distance₁*, *value₂*, *distance₂*, a value is linearly interpolated from *value₁* and *value₂* for each element centroid in the zone; interpolation is proportional to the fractional distance of the element centroid between *distance₁* and *distance₂*

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 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 the 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), (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 (active) and 0 (inactive). The active and inactive elements can be defined for the zone by three 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), (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). 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 nonzero number) and 0s is used to include and exclude elements from the zone definition. The array can be defined for the zone by any of the 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), (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, value₁, distance₁, value₂, distance₂**, a value is linearly interpolated from *value₁* and *value₂* for each element centroid in the zone; interpolation is proportional to the fractional distance of the centroid location between *distance₁* and *distance₂* along the specified coordinate direction; for locations outside the range of *distance₁* to *distance₂*, the value at the nearest point (*distance₁* or *distance₂*) is used. 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 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 magnitudes are 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 (**-longitudinal_dispersivity**) is on the order of 10 percent of the longitudinal transport distance, whereas horizontal transverse dispersivity (**-horizontal_dispersivity**) on the order of 1 percent of the longitudinal transport distance and vertical transverse dispersivity is on the order of 0.001 to 0.1 percent of the longitudinal transport distance (**-vertical_dispersivity**).

Example problems

The **MEDIA** data block is used in the example problems 6.2, 6.3, 6.4, and 6.5.

PRINT_FREQUENCY

This keyword data block is used to select the results to be written to output files and the frequency at which the results are written. This data block and all identifiers are optional. The example below indicates the default settings. The default setting is not to write data for identifiers with values of zero or “false” in the example. The default setting is to write data at the end of each simulation period for all identifiers with nonzero values in the example. Explicit definition of an identifier in this data block will override the default setting. This keyword data block may be included in the definition of any simulation period.

Example

```

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

```

Explanation

Line 0: **PRINT_FREQUENCY**

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

Line 1: **-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 either an interval of time or a number of time steps between writing results to the file. 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 2: **-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[true]** or **f[false]**.

Line 3: **-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 either an interval of time or a number of time steps between writing results to the file. 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 4: **-conductances** *frequency*, (*time unit* or **step**)

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

frequency—Frequency at which transient conductances are written. Frequency can be either an interval of time or a number of time steps between writing results to the file. 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 5: **-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 are 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 either an interval of time or a number of time steps between writing results to the file. 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 6: **-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* can also 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 either an interval of time or a number of time steps between writing results to the file. 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 7: **-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 either an interval of time or a number of time steps between writing results to the file. 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 8: **-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 either an interval of time or a number of time steps between writing results to the file. 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 9: **-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 head data are written to the HDF file. Frequency can be either an interval of time or a number of time steps between writing results to the file. 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: **-heads** *frequency*, (*time unit* or **step**)

-heads—This identifier is used to specify the frequency for writing 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 either an interval of time or a number of time steps between writing results to the file. If *frequency* is zero, no transient 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 11: **-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 either an interval of time or a number of time steps between writing results to the file. 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 12: **-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 that 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[true]** or **f[false]**.

Line 13: **-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 interstitial velocities at cell boundaries are written. Frequency can be either an interval of time or a number of time steps between writing results to the file. 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 14: **-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 either an interval of time or a number of time steps between writing results to the file. 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 15: **-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 either an interval of time or a number of time steps between writing results to the file. 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 16: **-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 head data are written. Frequency can be either an interval of time or a number of time steps between writing results to the file. 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 17: **-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 either an interval of time or a number of time steps between writing results to the file. If *frequency* is zero, no transient 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 18: **-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 either an interval of time or a number of time steps between writing results to the file. 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 19: **-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 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 either an interval of time or a number of time steps between writing results to the file. 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 simulation periods (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.

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.

A value of 0 (zero) for *frequency* suppresses writing of the specified data. By the default settings, print frequencies are set such 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 such that writing is suppressed. Once set by the **PRINT_FREQUENCY** data block, options for print frequencies remain in effect until the end of the simulation or until changed in a subsequent **PRINT_FREQUENCY** data block.

All print frequencies in the **PRINT_FREQUENCY** data block may be redefined for each simulation period if desired. When using **steps** for a print frequency, writing to the files is determined by 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 in the first simulation period and every 3 steps in the second, and both simulation periods have 5 time steps, 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 can also be requested in the **PRINT_INITIAL** data block.

The file names containing “.O.” are data formatted to be printed or viewed on a screen. The orientation of the printout of the spatially distributed properties in the “.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 columns to facilitate importing into spreadsheets and writing post-processing programs for graphical display.

Example problems

The **PRINT_FREQUENCY** data block is used in the example problems 6.2, 6.3, 6.4, and 6.5.

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 default setting is not to write data for identifiers with values of “false” and to write data for identifiers with values of “true” in the example. Explicit definition of an identifier in this data block will override the default settings. This data block can be defined only at the first simulation period.

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 data 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[true]** or **f[false]**.

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[true]** or **f[false]**.

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[true]** or **f[false]**.

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 of 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[true]** or **f[false]**.

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[true]** or **f[false]**.

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 storage, 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 these data to the files. If neither **true** nor **false** is entered on the line, **true** is assumed. Optionally, **t[true]** or **f[false]**.

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[true]** or **f[false]**.

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[true]** or **f[false]**.

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[true]** or **f[false]**.

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 component concentrations 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 initial 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[true]** or **f[false]**.

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[true]** or **f[false]**.

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 file names containing “.O.” are data formatted to be printed or viewed on a screen. The orientation of the printout of the spatially distributed properties in the “.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 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 6.2, 6.3, 6.4, and 6.5.

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 may be included in the definition of any simulation period.

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 indicates results will be written to the specified file for nodes selected by the zone; a value of 0 indicates results will not be written. Values for **-print** can be entered with any of three 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), or (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). 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_chemistry]**.

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 nonzero number) and 0s is used to include and exclude nodes from the zone definition. The array can be defined by any of the 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), (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. 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 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. Although, the selection of nodes for which results are written to the file *prefix.xyz.chem* can be changed for any simulation period, normally, the selection of nodes will not be changed because a fixed set of nodes written at each selected time step is easier to postprocess. 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. Printing will be defined by the accumulation of all the **PRINT_LOCATIONS** data blocks from the beginning of the simulation through the current simulation period.

Example problems

The **PRINT_LOCATIONS** data block is used in the example problem 6.5.

RIVER

This keyword data block is used to define a river boundary condition. This data block is optional and is only needed if river boundary conditions are included in the simulation. Rivers are 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 explicitly, either by interpolation from other river points, or by default values. Leakage into and out of the aquifer through the river bed is calculated from these parameters. Multiple **RIVER** data blocks are used to define multiple rivers in the grid region. All rivers must be defined at the first simulation period. In subsequent simulation periods river head and river solution composition may be redefined for river points where these parameters vary in time.

Example

```

Line 0: RIVER 1 Rubicon River
Line 1:     -default_width           150.
Line 2:     -default_depth          4.0
Line 3:     -default_bed_hydraulic_conductivity 1e-2
Line 4:     -default_bed_thickness           1.0
Line 5:     -default_solution            2
Line 6:     155.0      3633. # X, Y location of 1st river point
Line 7:         -head                  275.
Line 8:         -width                  125.
Line 9:         -depth                   3.5
Line 10:        -bed_hydraulic_conductivity 1.5e-2
Line 11:        -bed_thickness           1.6
Line 12:        -solution                 4
Line 5a:    -default_solution            4
Line 6a:    165.0      3663. # X, Y location of 2nd river point
Line 7a:        -head                    274.
Line 13:        -bottom                   271.5
Line 14: END
Line 0a: RIVER 1
Line 6b:    165.0      3663. #Second river point
Line 7b:        -head                    269. #Head for new simulation period

```

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: **-default_width** *default_width*

-default_width—This identifier is used to specify a default width of the river. The default width will be used for each succeeding river point in the data block that does not have an explicit definition of the river width. The default width can be redefined at any point in the definition of the river with another Line 1. Units, L, are defined by the **-horizontal_grid** identifier in the **UNITS** data block. Optionally, **default_width**, or **-default_w[idth]**.

default_width—Width of the river.

Line 2: **-default_depth** *default_depth*

-default_depth—This identifier is used to specify a default depth of the river. The default depth will be used for each succeeding river point in the data block that does not have an explicit definition of the river depth. The river depth is subtracted from the initial head definition for a river point to define the elevation of the bottom of the river. The default depth can be redefined at any point in the definition of the river with another Line 2. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block. Optionally, **default_depth**, or **-default_d[epth]**.

default_depth—Depth of the river.

Line 3: **-default_bed_hydraulic_conductivity** *default_bed_hydraulic_conductivity*

-default_bed_hydraulic_conductivity—This identifier is used to specify the default hydraulic conductivity for the river bed. The default hydraulic conductivity for the river bed will be used for each succeeding river point in the data block that does not have an explicit definition of hydraulic conductivity for the river bed. The default hydraulic conductivity for the river bed can be redefined at any point in the definition of the river with another Line 3. Units, L/T, are defined by the **-river_bed_hydraulic_conductivity** identifier in the **UNITS** data block. Optionally, **default_bed_hydraulic_conductivity**, **default_bed_k**, **default_k**, **-default_bed_h[ydraulic_conductivity]**, **-default_bed_k**, or **-default_k**.

default_bed_hydraulic_conductivity—Hydraulic conductivity of the river bed.

Line 4: **-default_bed_thickness** *default_bed_thickness*

-default_bed_thickness—This identifier is used to specify the default thickness for the river bed. The default thickness for the river bed will be used for each succeeding river point in the data block that does not have an explicit definition of thickness for the river bed. The default thickness for the river bed can be redefined at any point in the definition of the river with another Line 4. Units, L, are defined by the **-river_bed_thickness** identifier in the **UNITS** data block. Optionally, **default_bed_thickness**, **default_thickness**, **-default_bed_t[hickness]**, or **-default_t[hickness]**.

default_bed_thickness—Thickness of the river bed.

Line 5: **-default_solution** *default_solution*

-default_solution—This identifier is used to specify the default solution index number for the river. The solution index number will be used for each succeeding river point in the data block that does not have an explicit definition of a solution index number. Solution index numbers correspond to solution compositions defined in the chemistry data file. The default solution index number can be redefined at any point in the definition of the river data block with another Line 5. Optionally, **default_solution**, or **-default_s[olution]**.

default_solution—Solution index number.

Line 6: *X, Y*

The X–Y coordinate location of a river point is defined. Line 6 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.

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 7: **-head** *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. Optionally, **head**, or **-he[ad]**.

head—Head at the river point.

Line 8: **-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 9: **-depth** *depth*

-depth—This identifier is used to specify the depth of the river at the X–Y river-point location. 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: **-bed_hydraulic_conductivity** *bed_hydraulic_conductivity*

-bed_hydraulic_conductivity—This identifier is used to specify the hydraulic conductivity for the river bed 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 river bed at the river point.

Line 11: **-bed_thickness** *bed_thickness*

-bed_thickness—This identifier is used to specify the thickness of the river bed 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 river bed at the river point.

Line 12: **-solution** *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. Optionally, **solution**, **associated_solution**, **-s[olution]**, or **-a[ssociated_solution]**.

solution—Solution index number for the river point.

Line 13: **-bottom** *bottom*

-bottom—This identifier is used to specify the elevation of the top of the river bed at the X–Y river-point location. 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 river bed at the river point.

Line 14: **END**

END—This keyword is used to mark the end of input for a simulation period.

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 points within a **RIVER** data block define line segments. The head, width, and river-bed elevation, hydraulic conductivity, and thickness must be defined at each point to quantify river leakage to or from the aquifer. For reactive-transport simulation, an associated solution must be defined for each point to specify the composition of water leaking from the river into 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 defining a river depth (**-depth**) that will be subtracted from the head (**-head**) defined for the first simulation period.

Head must be explicitly defined for the first and last point in a river. For a point between the first and last points, the head can be defined explicitly (**-head**) or it can be interpolated from the nearest river points upstream and downstream where river heads have been explicitly defined. Linear interpolation is performed by using distance along the river line segments that lie between the two points where river heads are explicitly defined.

For each river property except head and river-bed elevation—that is, width, depth, associated solution, and river-bed hydraulic conductivity and thickness—it is possible to define default values that will be used if no explicit value is given at a river point (**-default_width**, **-default_depth**, **-default_solution**, **-default_bed_hydraulic_conductivity**, **-default_bed_thickness**). The most recently defined default value will be used for each river point for which a property is not explicitly defined.

Figure 4.6.—Discretization of river reaches 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 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 AB 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). 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 the first simulation period. Implicit definition is by interpolation or by application of default values. Only the head (**-head**) and the solution composition (**-solution**) for a river point can be specified in subsequent simulation periods. When head or solution composition for a river point is specified in a subsequent simulation period, the X and Y location of the river point must have been previously defined in a **RIVER** data block in the first simulation period; both the river identification number and the X–Y location must correspond to the previous **RIVER** definition. The X and Y location is followed by the **-head** and (or) **-solution** identifiers. Lines 0a, 6b, and 7b show the redefinition of head for a river point for a new simulation period. 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, both a river and a leaky boundary condition can not be defined for the positive Z face of a cell. If both a river and a specified value boundary condition are defined for a cell, the cell is a specified-value boundary cell and the river boundary condition has no effect. Rivers can not be used in confined flow simulations.

Example problems

The **RIVER** data block is used in the example problem 6.5.

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. If this data block is not included, the iterative solver with default parameter values is used. This keyword data block can only be defined at the first simulation period and can not be redefined in subsequent simulation periods.

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 employed. 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-8
Line 3:      -save_directions        5
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[true]** or **f[false]**.

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 1×10^{-8} (-). 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 5. 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 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 based on 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 Orthomin solver. The default value of 5 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 10 or 15. 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 forward-in-time (**-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.

If cross-dispersion terms (see section D.3) 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[true]** or **f[false]**.

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 (up to 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 6.2, 6.3, 6.4, and 6.5.

SPECIFIED_VALUE_BC

This keyword data block is used to define specified-value boundary conditions. For flow-only simulations, only the heads are required for each specified-value boundary condition node. For reactive-transport simulations, the index number of a fixed solution composition or an associated solution composition is also required for each specified-value boundary condition node. This keyword data block is optional and is only needed if specified-value boundary conditions are included in the simulation. All specified-value boundary conditions must be defined for the first simulation period. In subsequent simulation periods the head and associated or fixed solution compositions may be redefined for zones where these parameters vary in time.

Example

```

Line 0:  SPECIFIED_VALUE_BC
Line 1:    -zone      0      10      10      10      10      10
Line 2:          -head                100
Line 3:          -associated_solution  6
Line 1a:   -zone      0      0      10      10      10
Line 4:          -fixed_solution      5
Line 5:          -mask                file      mask.file

```

Explanation

Line 0: **SPECIFIED_VALUE_BC**

SPECIFIED_VALUE_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-value 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** *property*

-head—This identifier is used to specify the head at the specified-value boundary. Optionally, **head** or **-he[ad]**.

property—The head at the boundary can be entered with any of the 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), (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, value₁, distance₁, value₂, distance₂**, a value is linearly interpolated from *value₁* and *value₂* for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance₁* and *distance₂* along the specified coordinate direction; for locations outside the range of *distance₁* to *distance₂*, the value at the nearest point (*distance₁* or *distance₂*) is used. Units, L, are defined by the **-head** identifier in the **UNITS** data block.

Line 3: **-associated_solution** *property*

-associated_solution—This identifier is used to specify index numbers that define the associated solution compositions for the zone of specified-value boundary cells. Inflowing water at the boundary will have the specified solution composition, outflowing water will have the same solution composition as the water in the cell. Optionally, **associated_solution**, **associated**, or **-a[ssociated_solution]**.

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 four 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), (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), or (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. Line 3 illustrates method 1.

Line 4: **-fixed_solution** *property*

-fixed_solution—This identifier is used to specify index numbers that define fixed solution compositions for zones of specified-value boundary cells. When this identifier is used, the concentrations at the specified-value boundary cells remain constant over each simulation period and do not vary with the flux of water in or out of the boundary cells. Optionally, **fixed_solution**, **fixed**, or **-f[fixed_solution]**.

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 four 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), (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), or (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. Line 4 illustrates method 1.

Line 5: **-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 nonzero number) and 0s is used to include and exclude nodes from the zone definition. The array can be defined by any of the 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), (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, value₁, distance₁, value₂, distance₂**, a value is linearly interpolated from *value₁* and *value₂* for each node in the zone; interpolation is proportional to the fractional distance of the node location between *distance₁* and *distance₂* along the specified coordinate direction; for locations outside the range of *distance₁* to *distance₂*, the value at the nearest point (*distance₁* or *distance₂*) is used. Line 5 illustrates method 3.

Notes

A choice between two identifiers is used to define boundary conditions for the concentrations of chemical components at specified-value boundaries, **-fixed_solution** and **-associated_solution**. With the **-fixed_solution** identifier, the concentrations of components in the boundary cell are constant for each simulation period and 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. 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-value 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_VALUE_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 value cell, including leaky, flux, and river boundary conditions.

Example problems

The **SPECIFIED_VALUE_BC** data block is used in the example problems 6.2, 6.3, 6.4, and 6.5.

STEADY_FLOW

This keyword data block is used to specify that the flow is steady state and to define criteria for the attainment of a steady-state flow condition. 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. This keyword data block can only be defined at the first simulation period and can not be redefined in subsequent simulation periods.

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
Line 4:      -maximum_time_step        1000
Line 5:      -head_change_target        100.

```

Explanation

Line 0: **STEADY_FLOW**

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 (**FLOW_ONLY false**), 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[true]** or **f[false]**.

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*

-minimum_time_step—This identifier is used to specify the initial time step used in calculating steady-state flow and is also the minimum time step allowed in the steady-state flow calculation. Default is the value for **-time_step** defined in **TIME_CONTROL** divided by 1000. 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 by **-time** identifier in the **UNITS** data block.

Line 4: **-maximum_time_step** *time_step*

-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 1000. Optionally, **maximum**, **maximum_time**, **maximum_time_step**, or **maximum_time_step**.

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

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_change_target**, or **-head_target**.

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

Notes

STEADY_FLOW can be specified with or without transport calculations (**FLOW_ONLY**). 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.1.3 for details of the automatic time-stepping algorithm for the steady-flow calculation. 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 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 can not 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 6.2 and 6.5.

TIME_CONTROL

This keyword data block is used to define the time step and the time at which the current simulation period ends. The time of the end of each simulation period is measured from the beginning of the simulation. For transient flow and (or) transport simulations the time step is constant for the simulation period, except for variations needed to reach times that are targeted for printing results. This keyword data block is mandatory for all simulation periods.

Example

```
Line 0:  TIME_CONTROL
Line 1:      -time_step      10    day
Line 2:      -time_end       1     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** *length*, [*units*]

-time_step—This identifier is used to specify the time-step length. There is no default value. Optionally, **step**, **time_step**, **delta**, **delta_time**, **-s[tep]**, **-time_s[tep]**, or **-d[elta_time]**.

length—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 1: **-time_end** *time*, [*units*]

-time_end—This identifier is used to specify the time at which the simulation period ends. This time can be the end of the simulation or the time at which the next simulation period begins. The time is measured from the beginning of the simulation. Optionally, **time_change**, **change_time**, **end**, **end_time**, **time_end**, **-time_c[hange]**, **-c[hange_time]**, **-e[nd_time]**, or **-time_e[nd]**.

time—Time at which the simulation period ends. *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 required for each simulation period. If a time unit is specified for an identifier (**-time_step** or **-time_end**), then the specified unit overrides the default unit. The units specified for **-time_step** and **-time_end** need not be the same.

Example problems

The **TIME_CONTROL** data block is used in the example problems 6.2, 6.3, 6.4, and 6.5.

TITLE

This keyword data block is used to define two title lines which are written to each of the output files, except the *prefix.xyz* and *prefix.h5* files. This keyword data block only has effect when defined at the first simulation period; the data block is ignored if defined in subsequent simulation periods.

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 in the definition of the first simulation period, 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 6.2, 6.3, 6.4, and 6.5.

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 is mandatory for all simulations and must be defined in the first simulation period. This keyword data block can not be redefined in subsequent simulation periods.

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[eritical_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[hydraulic_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[diameter]**.

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”, “ft³/sec”, or “ft³/sec”), metric (for example, “Liter/minute”, “meters³/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[hydraulic_conductivity]**, **-river_bed_k**, or **-river_k**.

units—Input units for hydraulic conductivity (L/T) of river beds. 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 river bed 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 river bed (L). Units of thickness for the river bed 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 present in the data for the first simulation period for all flow and transport simulations. The definitions in this data block specify the units for input data for all keyword data blocks of in the flow and transport data file. Note that all definitions, except **-time**, apply only 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 all output data.

Example problems

The **UNITS** data block is used in the example problems 6.2, 6.3, 6.4, and 6.5.

WELL

This keyword data block is used to define an injection or pumping well. This data block is optional and is only 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. All wells must be defined for the first simulation period. In subsequent simulation periods the pumping or injection rate and injection solution may be redefined for wells where these parameters vary in time.

Example

```

Line 0: WELL 122 Metropolis Injection Well 122
Line 1:      1766.      2356.
Line 2:      -diameter          12
Line 3:      -injection_rate  4.5
Line 4:      -solution        16
Line 5:      -elevation       101. 107.
Line 5a:     -elevation       143. 153.
Line 5b:     -elevation       175. 183.
Line 0a: WELL 165 Metropolis Supply Well 165
Line 1a:     1833.      2320.
Line 6:      -radius          12
Line 7:      -pumping_rate    4.5
Line 8:      -land_surface_datum 292.
Line 9:      -depth          42. 47.
Line 9a:     -depth          99. 103.
Line 10:     -allocate_by_head_and_mobility      true
Line 11: END
Line 0b: WELL 165 Metropolis Supply Well 165
Line 7a:     -pumping_rate    1.0

```

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: **-diameter** *diameter*

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

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

Line 3: **-injection_rate** *injection_rate*

-injection_rate—This identifier is used to specify the rate of fluid injection into the well. Optionally, **injection_rate**, **injection**, or **-i[njection_rate]**.

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

Line 4: **-solution** *solution*

-solution—This identifier is used to specify a solution index number, which 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. Optionally, **solution**, **associated_solution**, **-s[olution]**, or **-a[ssociated_solution]**.

solution—Solution index number.

Line 5: **-elevation** *elevation₁*, *elevation₂*

-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]**.

elevation₁—Elevation of start of open interval in the well. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block.

elevation₂—Elevation of end of open interval in the well. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block.

Line 6: **-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 7: **-pumping_rate** *pumping_rate*

-pumping_rate—This identifier is used to specify the pumping rate for the well. Optionally, **pumping_rate**, **pumping**, **pumpage**, **-pu[mping_rate]**, or **-pu[mpage]**.

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

Line 8: **-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 9: **-depth** *depth₁*, *depth₂*

-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]**.

depth₁—Depth of start of open interval in the well. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block.

*depth*₂—Depth of end of open interval in the well. Units, L, are defined by the **-vertical_grid** identifier in the **UNITS** data block.

Line 10: **-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[true]** or **f[false]**.

Line 11: **END**

END—This keyword is used to mark the end of input for a simulation period.

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, and open intervals. Open intervals for a well 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 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 different simulation periods.

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.

Complete definitions of all wells are required for the first simulation period. Only the flow rate (**-injection_rate** or **-pumping_rate**) and the solution composition (**-solution**) for a well may be specified in subsequent simulation periods. When flow rate or solution composition for a well are specified in subsequent simulation periods, a **WELL** data block is needed with the corresponding well identification number. The keyword line is followed by an identifier for the flow rate (**-injection_rate** or **-pumping_rate**) and (or) the **-solution** identifier. Lines 0b and 7a in the example illustrate the definition of a new pumping rate for a subsequent simulation period.

Example problems

The **WELL** data block is used in the example problem 6.5.

FIGURES

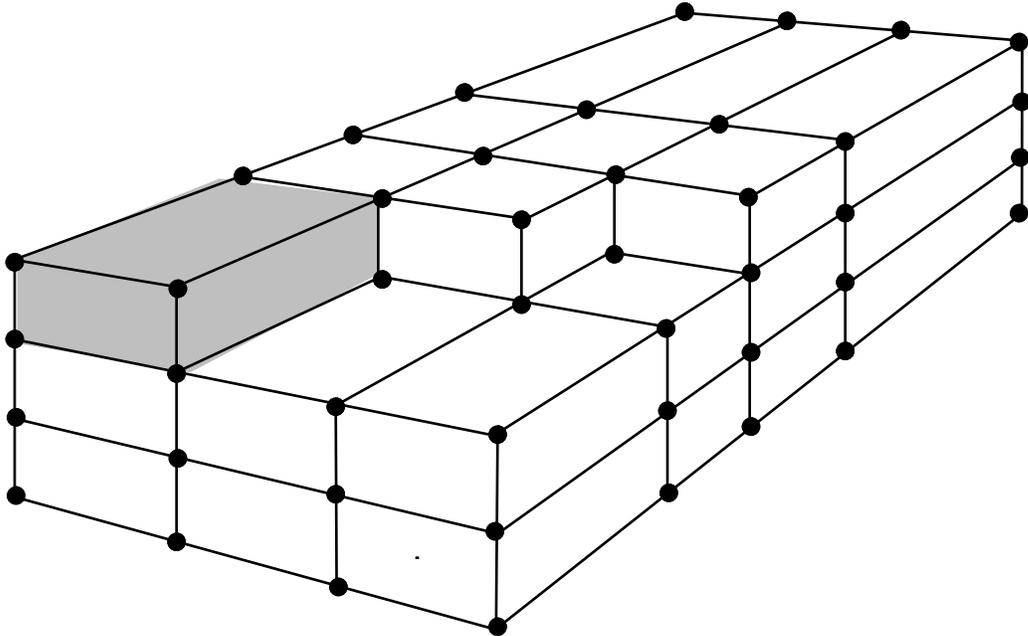


Figure 4.1.—Grid region showing boundary nodes and one shaded element.

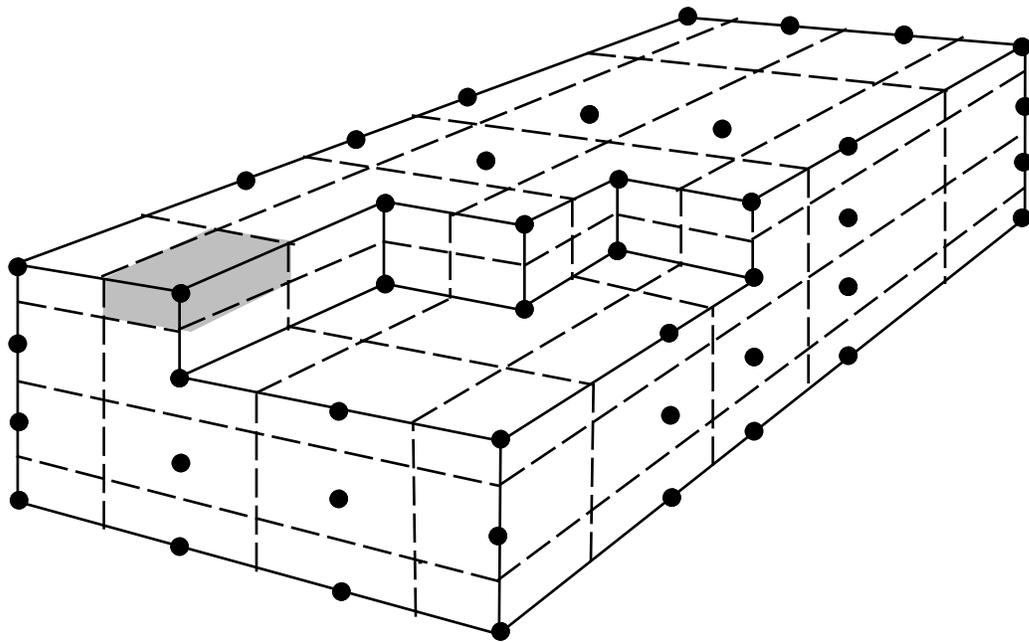


Figure 4.2.—Grid region showing boundary of the active grid region (solid lines), boundary nodes (dots), cell boundaries (dashed lines), and one shaded cell.

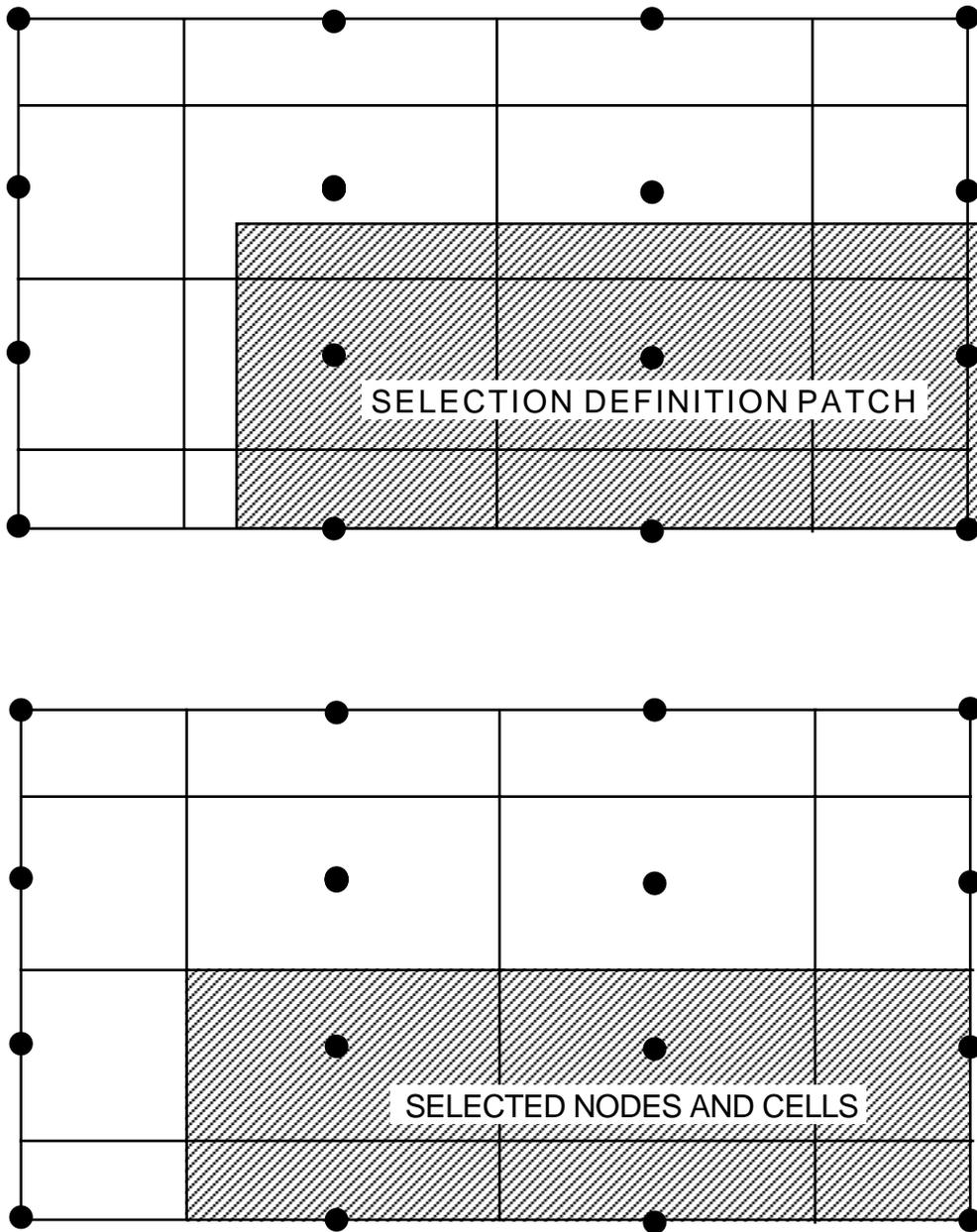
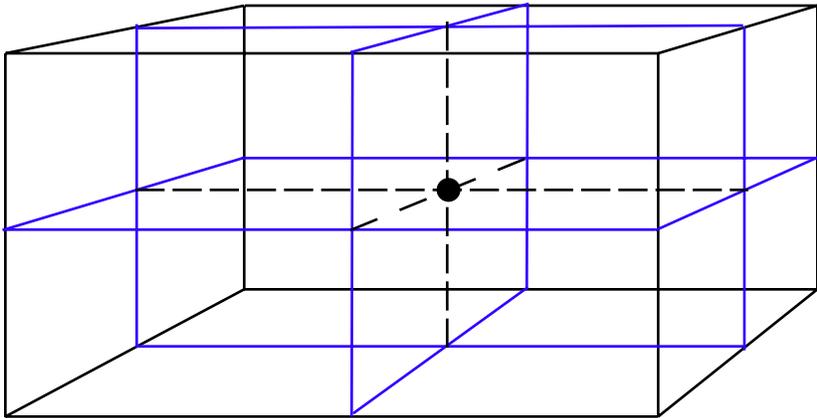


Figure 4.3.—(A) Rectangular zone used to select elements from a two-dimensional grid, and (B) the elements selected by the zone.



- ELEMENT EDGE
- CELL EDGE
- ELEMENT FACE INTERSECTION
WITH CELL FACE

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

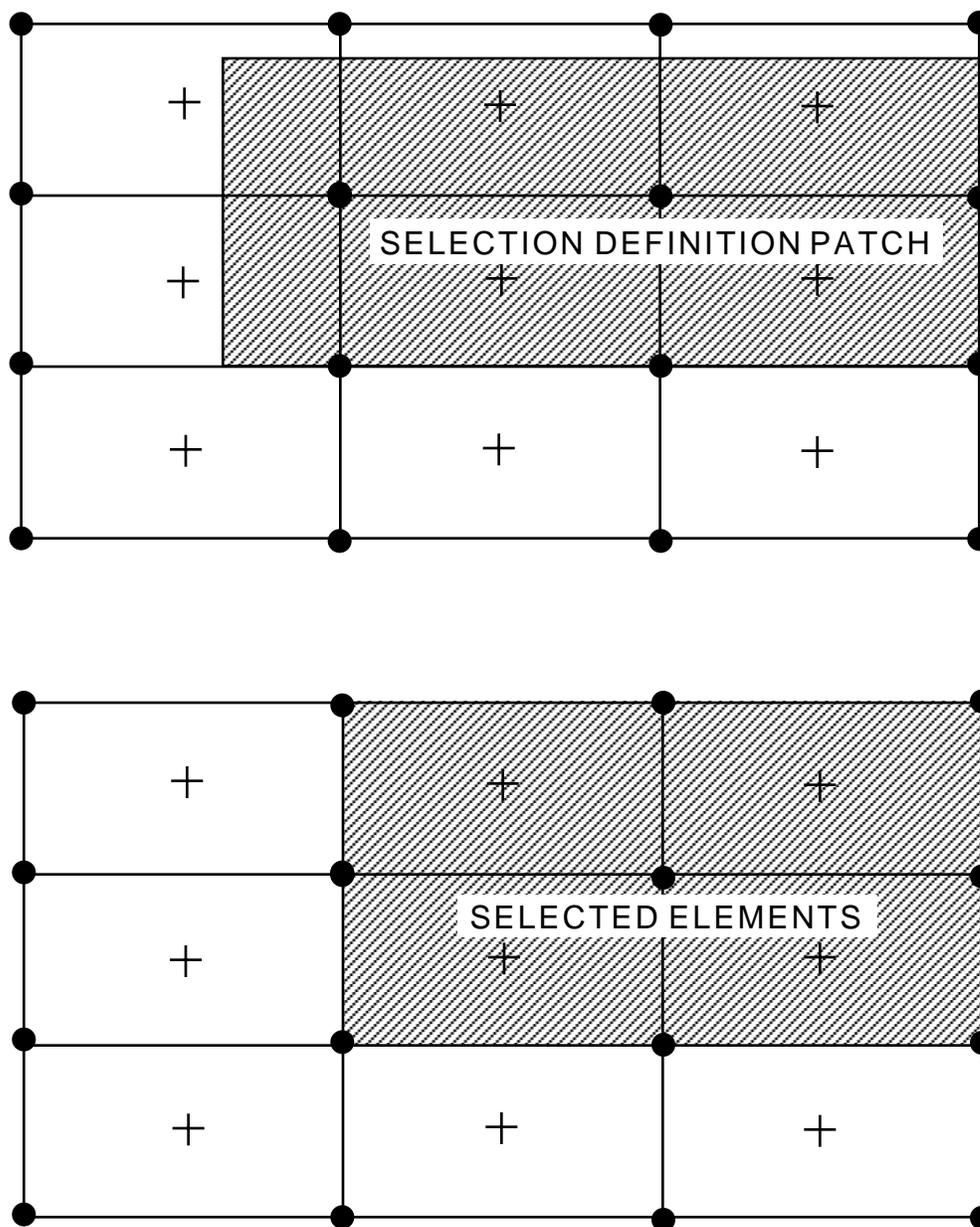


Figure 4.5.—(A) Rectangular zone used to select nodes from a two-dimensional grid and (B) the nodes and cells selected by the zone. Lines represent cell boundaries.

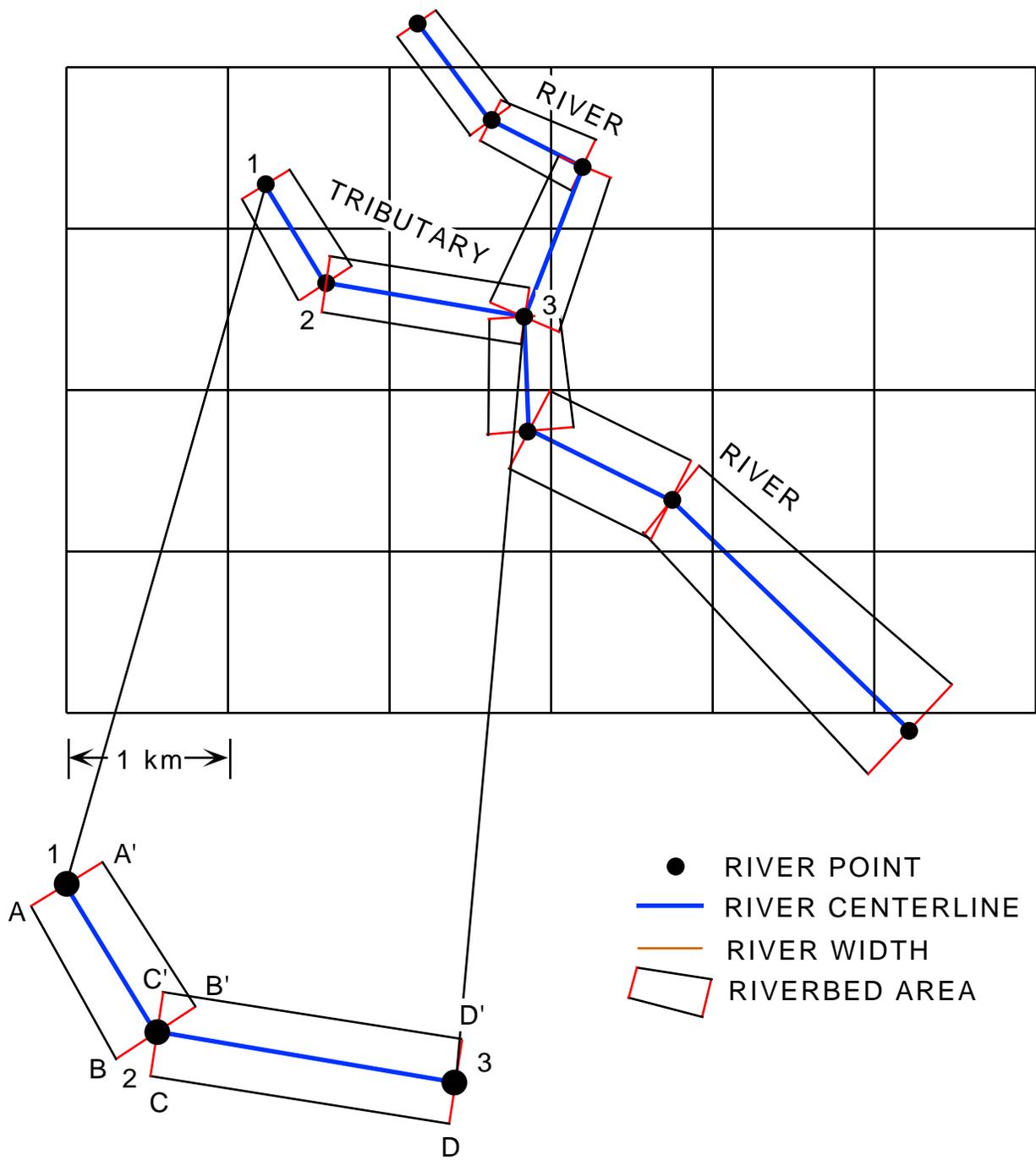


Figure 4.6.—Discretization of river reaches for a river and its tributary; initial river-bed areas are shown before fill-in of gaps and removal of overlapping areas.