xFLO VERSION 1.0β USER'S MANUAL

Prepared for

U.S. Nuclear Regulatory Commission
Contract NRC–02–02–012

Prepared by

Scott Painter

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

July 2006

PREVIOUS REPORTS IN SERIES

Number	Name	Date Issued
Letter Report	Software Requirements Description for *xFlo* An Extensible Flow and Transport Model for Use in U.S. Nuclear Regulatory Commission Performance Confirmation Oversight	May 2003

ABSTRACT

Coupled-process simulators are important tools for understanding the near-field environment of the potential high-level radioactive waste repository at Yucca Mountain, Nevada. Similar tools have a variety of uses in evaluating and interpreting the results of potential performance confirmation activities. The *xFlo* computer code is under development as a general modeling tool to support possible future performance confirmation activities. The *xFlo* software provides a flexible simulation framework that can be easily modified to accommodate new physical models. Each coupled-process model in the *xFlo* code is encapsulated in a set of computational routines known collectively as a "physics package" which is independent of the numerical routines. Version 1.0β of *xFlo* has one physics package which simulates two-phase thermal hydrology processes. This user's manual describes the input format for *xFlo* Version 1.0β, including the thermal hydrology package. A general overview of the computational framework and underlying equations for the thermal hydrology model are also provided.

CONTENTS

Section Page

PREVIOUS REPORTS IN SERIES ii
FIGURES vii
TABLES ix
ACKNOWLEDGMENTS xi

1 INTRODUCTION 1-1

1.1 Overview of the *xFlo* Generic Framework 1-1
1.2 Overview of the Thermal Hydrology Package 1-3
1.3 Validation Status 1-3

2 INSTALLATION AND EXECUTION 2-1

3 GENERIC INPUT 3-1

3.1 Input Data Files for Structured and Unstructured Grids 3-1
3.2 Data Formats in the Preprocessor Input File 3-1
3.3 Specifying the Grid 3-3
3.4 Control of the Linear Solver 3-4
3.5 Time Step and Output Control 3-5
3.6 Dual-Continuum Simulations 3-7

4 THERMAL HYDROLOGY PACKAGE 4-1

4.1 Governing Equations 4-1
4.1.1 Mass Conservation Equations and Equilibrium Constraints 4-1
4.1.2 Capillary Pressure and Relative Permeability Functions 4-2
4.1.3 Energy Balance Equation 4-4
4.1.4 Thermophysical Properties 4-4
4.2 Assignment of Rock Properties to Grid Cells 4-5
4.3 Specifying Initial Conditions 4-6
4.4 Specifying Boundary Conditions 4-7
4.5 Specifying Mass and Heat Sources 4-9
4.6 Thermal Hydrology Package Output 4-11
4.7 Example Input File 4-11

5 PLANS FOR FUTURE DEVELOPMENT 5-1

6 REFERENCES 6-1

FIGURES

Figure		Page
1-1	Large-Scale Data Structure and Data Flows for the *xFlo* System	1-2
3-1	Data Flow for the *xFlo* System	3-2
3-2	Example of the GRID Keyword Block	3-4
3-3	Example of the SOLVER Keyword Block	3-6
3-4	Example of the CONTROL Keyword Block	3-7
3-5	Example of the DCM Keyword Block	3-8
4-1	Example `*.flo` File for a One-dimensional Simulation Using the Thermal Hydrology Package	4-12

TABLES

Table Page

3-1 Parameters for the GRID Keyword Block 3-3
3-2 Parameters in the SOLVER Keyword Block 3-5
3-3 Parameters in the CONTROL Keyword Block 3-6

ACKNOWLEDGMENTS

This user's manual documents work performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for the U.S. Nuclear Regulatory Commission (NRC) under Contract No. NRC–02–02–012. The activities reported here were performed on behalf of the NRC Office of Nuclear Material Safety and Safeguards, Division of High-Level Waste Repository Safety. This manual is an independent product of the CNWRA and does not necessarily reflect the view or regulatory position of the NRC.

The authors acknowledge the technical review of K. Das, the editorial review of L. Mulverhill, the programmatic review of S. Mohanty, and the assistance of J. Gonzalez in preparing this report.

QUALITY OF DATA, ANALYSES, AND CODE DEVELOPMENT

DATA: No data is presented in this report.

ANALYSES AND CODES: The *xFlo* computer code was developed under the software development procedures described in Geosciences and Engineering Division Technical Operating Procedure TOP-018.

1 INTRODUCTION

Coupled-process simulators are important tools for understanding the near-field environment of the potential high-level waste repository at Yucca Mountain, Nevada. Coupled-process simulators that can be easily modified to accommodate new process models are needed to support U.S. Nuclear Regulatory Analyses (NRC) regulatory activities related to potential U.S. Department of Energy performance confirmation activities. The *xFlo* code is a flexible computational system that can be used to simulate various heat and mass transport processes in fractured porous media. The *xFlo* code is specifically designed to be adaptable and is intended to eventually replace the existing coupled-process simulator MULTIFLO (Lichtner, 1996; Painter and Seth, 2003). The software requirements description for *xFlo* was transmitted to NRC in May 2003.

The design philosophy behind *xFlo* is significantly different from that of MULTIFLO. Specifically, *xFlo* is designed as a general computational framework for solving conservation equations for heat and mass in fractured porous media. The *xFlo* design enforces a strict separation between numerical routines and physical models. In particular, physical models are encapsulated in "packages" which contain all secondary variables and physics-specific calculations. This design allows new physical models to be added without modifying the underlying numerical infrastructure. Version 1.0β of *xFlo* includes one physics package, the thermal hydrology package.

This User's Manual's primary purpose is to provide detailed data format and instructions on preparing input data for *xFlo* Version 1.0β, including the thermal hydrology package. Overviews of the mathematical models solved by the thermal hydrology package and the *xFlo* internal structure are also provided. Detailed data structures are not addressed here.

In this manual, the `courier` font is used for filenames, the contents of input files, commands given at the command prompt, and screen output. When specifying input formats, a line consisting of a single colon in the input block means that lines are skipped. An ellipsis (…) on an input line indicates that items on that line are skipped. Any text following an exclamation point in an input block is to be regarded as a comment or explanation.

1.1 Overview of the *xFlo* Generic Framework

The generic framework of the *xFlo* code solves coupled sets of conservation equations using the integral finite-difference method (Narasimhan and Witherspoon, 1976). Fully unstructured grids are supported without restriction on the computational cell shape or intercell connectivity. Dual-continuum systems, which model the fracture and matrix systems as distinct but interacting flow systems, are also supported. A convenient keyword driven preprocessor option is available for use with regular structured grids. Dynamic allocation is used throughout the code, thus avoiding the need to recompile the code when using large computational grids.

Time stepping is fully implicit. Newton-Raphson iteration is used to resolve any non-linear terms. The Jacobian matrix required for Newton-Raphson method is developed by the perturbation method. A numerical Jacobian was chosen instead of an analytical one to facilitate the introduction of new physical models; with a numerical Jacobian, developers of physics packages are relieved of the time-consuming and error-prone task of developing analytical Jacobian models.

The linear system resulting from each Newton-Raphson iteration is solved by a sparse preconditioned conjugate gradient solver. The solver is a Fortran 95 adaptation of the WATSOLV package (van der Kwaak, et al., 1995). Incomplete lower-upper decomposition with fixed level of fill is used for the preconditioning step. Acceleration is achieved by the generalized minimum residual method or biconjugate gradient stabilized method.

The *xFlo* data structure is shown in simplified schematic form in Figure 1-1. There are three main components in the data structure: STEPPER, PHYSICS, and Generic System Description.

The STEPPER module is responsible for advancing the numerical solution in time using fully-implicit time discretization and Newton-Raphson iteration. The STEPPER module assembles the residual mass/energy vectors and the Jacobian matrices. The SOLVER, a sub-module of the STEPPER, solves the linear equations resulting from the discretization processes.

All information about the physics is encapsulated in the PHYSICS module. PHYSICS is implemented as an abstract class. That is, there may be several flavors of PHYSICS modules to choose from, but only one can be active at any time and all must present the same interface to the rest of the code. Version 1 of *xFlo* has only one PHYSICS option: thermal hydrology.

Data are not passed directly between the STEPPER and PHYSICS modules. Instead each has access to the Generic System Description data structure. The Generic System Description data structure contains static data such as grid information, boundary conditions, and mass/energy

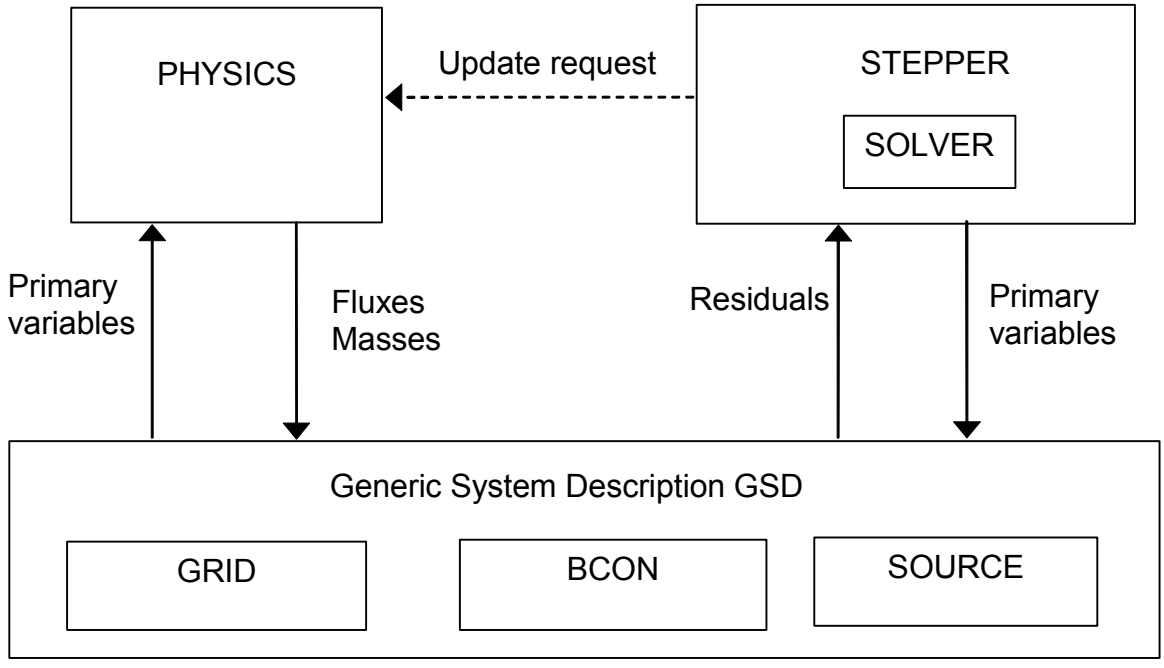

Figure 1-1. Large-Scale Data Structure and Data Flows for the *xFlo* System. Note that the Physics Module Is Implemented as an Abstract Data Class. Specific Examples from the Physics Class Will, in General, Have Additional Internal Structure.

sources, which are encapsulated in sub-modules. The Generic System Description also contains primary variables, intercell fluxes for each conserved quantity (masses and energy), and the mass/energy in each cell. The STEPPER is responsible for updating primary variables in the Generic System Description, and the PHYSICS module is responsible for updating fluxes, masses, and energy. The only direct communication between the STEPPER and PHYSICS module is in the form of an update request which is passed from STEPPER to PHYSICS after STEPPER has updated a primary variable. The update request instructs the PHYSICS module to update fluxes and masses for specific nodes.

1.2 Overview of the Thermal Hydrology Package

The thermal hydrology package solves conservation equations for air, water and energy. Air may be in the gaseous phase or dissolved in liquid. Water may be in the liquid or gas (vapor) phase.

In a two-phase model, there are three possible phase states: two-phase, all liquid, or all gas. The primary variables used in the thermal hydrology package are different for different phase states. In the two-phase state, the primary variables are liquid saturation, gas pressure, and temperature. In the all-liquid state, the primary variables are molar density of dissolved air, liquid pressure, and temperature. In the all-gas state, the primary variables are air pressure, gas pressure, and temperature. Different computational cells may be in different phase states, and the phase states may change in the course of the simulation.

The thermal hydrology package treats mass transport by multiphase Darcy's law with capillary, gravity, and viscous forces. Binary diffusion of air/vapor and diffusion of air in the dissolved phase are also included. Energy transport is by conduction and convection. Interpolation of tabulated values is used for equations-of-state and thermophysical properties. Capillary pressure/relative permeability options include the van Genuchten model and the active fracture model. More details of the thermal hydrology package are given in Section 4.

1.3 Validation Status

xFlo Version 1.0β has been tested on realistic two-dimensional example simulations, but has not undergone validation. Current plans are to continue evaluating *xFlo* with respect to ease-of-use and computational performance before initiating validation activities.

2 INSTALLATION AND EXECUTION

Included with the Version 1.0β release of *xFlo* are source code, a makefile, data files for physical properties of water, and input/output files for two acceptance tests.

The *xFlo* software is written in Fortran 95 and should compile unmodified with most modern Fortran compilers. The only known potential compatibility issue is associated with the retrieval of command-line arguments. The Fortran 95 standard does not provide an intrinsic function for retrieving command-line arguments. The *xFlo* software uses the *getarg* intrinsic, which is typically available in Unix environments and provided by many compilers that work on the Windows operating system. The *xFlo* software was developed and tested in the SunOS/Solaris and Windows XP environments. The code has also been successfully installed on an Apple iMac computer running Mac OS X.

The makefile provided with Version 1.0β can be used to compile *xFlo*. The makefile is customized for the Sun Microsystems compilers. The definitions of the compile and link commands must be changed to compile on other systems.

The thermal hydrology module requires data files for the pressure-volume-temperature properties of water. The pressure-volume-temperature input files must be placed in the same directory, but that directory may be located anywhere in the local file system provided it is accessible at run time. The location of the pressure-volume-temperature data directory is part of the thermal hydrology input, as described in Section 4.2 of this document.

To execute *xFlo*, the location of the directory containing the executable file should be in the user's search path. To run *xFlo*, type

```
xFlo FILE
```

at the command line, where FILE is the root-name of the input file, which must be in the current working directory. The extension `.flo` will be added to form the complete file name.

3 GENERIC INPUT

The input for *xFlo* includes generic parameters that are independent of the physics model and model-specific parameters. This section describes the generic input.

3.1 Input Data Files for Structured and Unstructured Grids

The internal data storage in *xFlo* is based on an unstructured grid with arbitrary intercell connectivity. Thus, the grid is defined by specifying a connectivity list for each node, a volume for each node, and surface area and intercell distance for each connection between nodes. This unstructured grid format provides a large degree of modeling flexibility.

Each of the computational modules in *xFlo* is responsible for its own input/output. Data defining the grid, physical properties, sources, initial conditions, and boundary conditions are read from input files in a format that closely matches the internal data structures. These unstructured grid input files are required if the simulation uses an unstructured grid. If structured Cartesian grids are to be used, an optional preprocessing capability is provided. The preprocessor reads data from a `*.flo` input file and writes the intermediate data files. The data flow during initialization is shown in Figure 3-1.

Note that the `*.dat` files contain input data that will be used without processing and may have different units from these in the `*.flo` file. For example, temperatures may be specified in °C in the `*.flo` file and Kelvin in the `*.dat` files.

3.2 Data Formats in the Preprocessor Input File

The input to the *xFlo* preprocessor is in a flexible keyword-driven format. Input data are divided into keyword blocks. The beginning of a keyword block is indicated by the presence of an allowed keyword as the first word in a line. Keyword blocks are terminated by the word END as the first word in a line. Keywords and the END terminator must be uppercase. Leading blanks are ignored. Keyword blocks may come in any order. Any characters that are not contained in a keyword block are ignored. Thus, any number of comment lines may be placed between keyword blocks. Comment lines are not allowed within a keyword block. However, partial lines of comment may be placed after the input parameters on a given line.

Within a keyword block, data take one of two formats. Input data that apply globally take a namelist-like format. Data that represent the assignment of a physical parameter to individual grid cells must appear in a more traditional fixed order. All data are free-format.

The namelist-like format for a global input parameter is

PARAMETER = VALUE

where PARAMETER is an allowed parameter name and VALUE is the value to be assigned. The equal sign must appear. Parameters can appear in any order. Parameter values may be scalars or arrays, depending on the parameter.

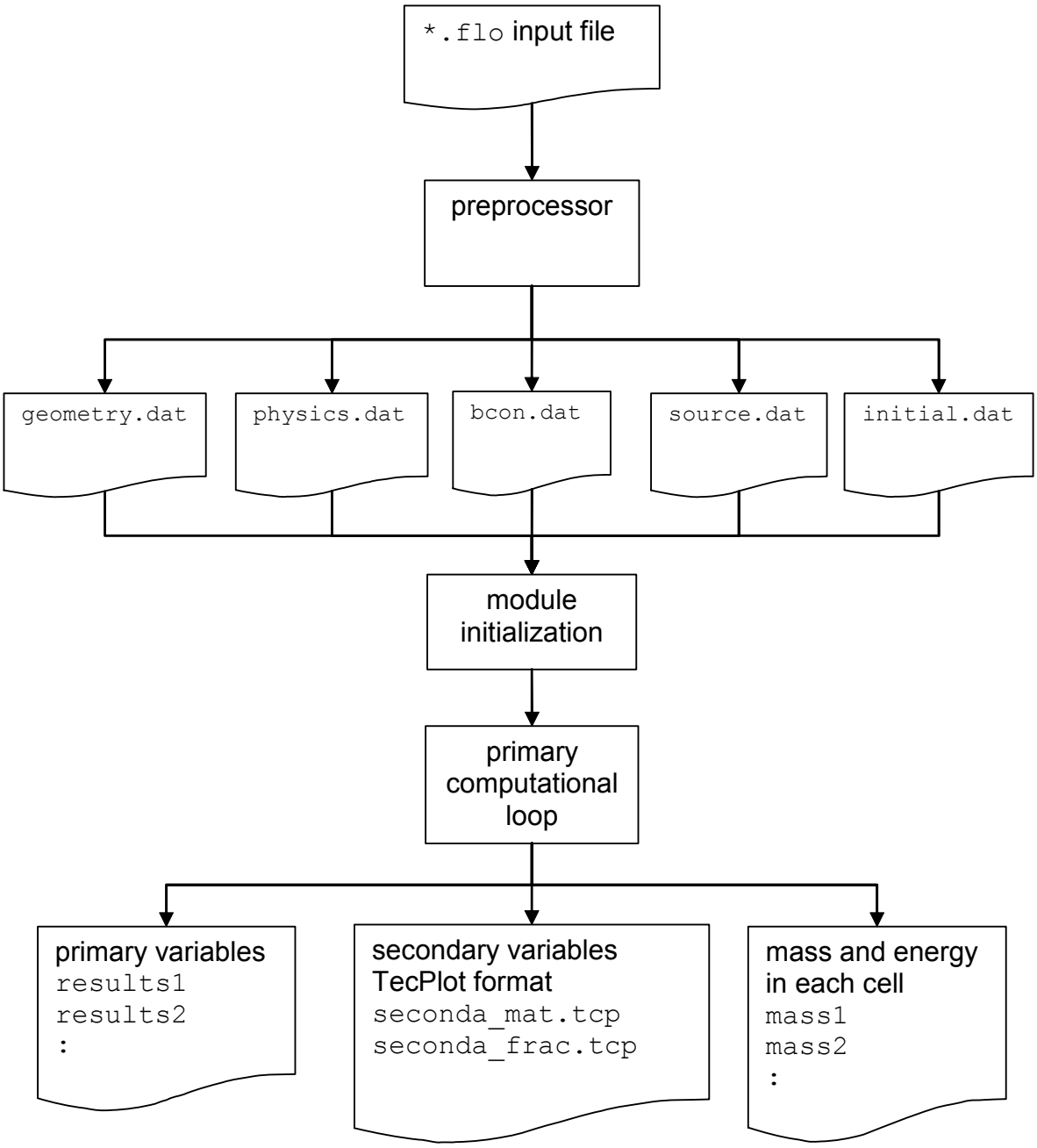

Figure 3-1. Data Flow for the *xFlo* System. For Unstructured Grids, the Preprocessor Step Is Skipped and Data Are Read Directly from the `*.dat` Files.

3.3 Specifying the Grid

The computational grid is specified in the GRID keyword block in the `*.flo` file. If the GRID keyword is not present, the grid information will be read in unstructured format from the `geometry.dat` file.

The format for the GRID keyword block is

```
GRID GRIDTYPE SPACING
:
END
```

Here GRIDTYPE is the type of the grid (XYZ, indicating Cartesian grids, is the only allowed value in Version 1.0β) and SPACING (VARIABLE or CONSTANT) specifies where the grid is uniform (constant spacing) or nonuniform (variable spacing). Parameter names in the namelist-type input are given in Table 3-1. If SPACING is set to VARIABLE, then the parameters DX, DY, and DZ are required. If SPACING is set to uniform, then XMIN, XMAX, YMIN, YMAX, ZMIN, and ZMAX are required. NX, NY, and NZ are required in all cases.
An example GRID keyword block is shown in Figure 3-2.

If the GRID keyword is not in the `*.flo` file, then grid information for each cell and each connection is read directly from the `geometry.dat` file. The first line of the `geometry.dat` file contains five integer parameters that set the overall dimensions of the various arrays containing grid information. The remainder of the file is divided into two sections. Data for each intercell connection is given in the first section, followed by data for each grid cell.

Table 3-1. Parameters for the GRID Keyword Block

Parameter Name	Type	Description
NX	integer scalar	number of grid cells in x-direction
NY	integer scalar	number of grid cells in y-direction
NZ	integer scalar	number of grid cells in z-direction
XMIN	real scalar	minimum x-coordinate [m]
XMAX	real scalar	maximum x-coordinate [m]
YMIN	real scalar	minimum y-coordinate [m]
YMAX	real scalar	maximum y-coordinate [m]
ZMIN	real scalar	minimum z-coordinate [m]
ZMAX	real scalar	maximum z-coordinate [m]
DX	real array of size NX	cell widths in x-direction [m]
DY	real array of size NY	cell widths in y-direction [m]
DZ	real array of size NZ	cell widths in z-direction [m]

```
GRID XYZ VARIABLE
  nx =20   ny=1 nz= 100
  dx = 20*1.
  dz= 45*10.  10*1 45*10.
  dy=1.0
END GRID
```

Figure 3-2. Example of the GRID Keyword Block

The format for the `geometry.dat` file is

```
NCELL NBC NCONN MXNBRS IDCM
N1 N2 D1 D2 AREA IFM                    ! data for connection 1
:                                       ! repeat for a total of NCONN
VOL Z Y X                               ! data for cell 1
:                                       ! repeat for a total of NCELL+NBC
```

The parameters appearing in the first line are as follows.

NCELL	Number of grid cells
NBC	Number of boundary conditions
NCONN	Number of intercell connections
MXNBRS	Maximum number of neighbors for a single cell
IDCM	1 for single continuum, 2 for DCM

The parameters in the connections section are as follows.

N1	Node number on one side of the connection
N2	Node number on the other side of the connection
D1	Distance from node N1 to the cell boundary [m]
D2	Distance from node N2 to the cell boundary [m]
AREA	Surface area of the connection [m^2]
IFM	2 for fracture-to-matrix connection, 1 otherwise

The parameters in the cell definition section are as follows.

VOL	Volume for the cell	[m^3]
Z	z-coordinate for the cell	[m]
Y	y-coordinate for the cell	[m]
Z	z-coordinate for the cell	[m]

3.4 Control of the Linear Solver

Version 1.0β of *xFlo* uses a pre-conditioned conjugate gradient solver with two accelerator options: biconjugate gradient stabilized and generalized minimum residual. Preconditioning is based on an incomplete lower-upper decomposition with fixed level of infill. The solver is an adaptation of the WATSOLV solver (van der Kwaak, et al., 1995).

Parameters controlling the linear solver are specified with the SOLVER keyword block. Input for the SOLVER keyword block is in namelist format. The SOLVER block is optional; default parameters will be used if the SOLVER keyword is missing. The parameters are summarized in Table 3-2.

Table 3-2. Parameters in the SOLVER Keyword Block

Parameter Name	Type	Description
ACCELERATOR	integer scalar	= 3 GMRES* accelerator = 4 CGSTAB† accelerator
NUM_VECTORS	integer scalar	maximum number of vectors in GMRES orthogonalization
FACTOR_LEVEL	integer scalar	level of infill in incomplete lower-upper preconditioning
ITERMAX	integer scalar	maximum number of solver iterations
DETAIL	integer scalar	debugging options < 1 no solver debugging output = 2–4 print convergence summary > 4 full debug output (produces large output)
MAX_RESIDUAL	real scalar	convergence criterion based on maximum residual
MAX_UPDATE	real scalar	convergence criterion based on maximum solution update
SCALED_RESIDUAL	real scalar	convergence criterion based on Euclidean norm of residual scaled by initial value

*GMRES—generalized minimum residual
†CGSTAB—biconjugate gradient stabilized

An example of the SOLVER keyword block is shown in Figure 3-3.

3.5 Time Step and Output Control

Time step and output control are specified by the CONTROL keyword. Input data within this keyword block are in the namelist format. The parameters are summarized in Table 3-3. Convergence of the Newton-Raphson iteration during a time step is achieved when all residuals satisfy the convergence criteria for residuals (RTOLER) or when all changes in primary variables satisfy the convergence criteria for primary variables.

Output is written at the times specified in the RPTIMES array. The maximum allowed time-step size (DTMAX array) may also change at each of the RPTIMES.

```
SOLVER
 accelerator_type = 4
 factor_level = 2
 itermax = 200
 detail = 2
 max_residual = 1.0d-6
 max_update = 1.0d-9
 scaled_residual = 1.0d-6
END SOLVER
```

Figure 3-3. Example of the SOLVER Keyword Block

Table 3-3. Parameters in the CONTROL Keyword Block

Parameter Name	Type	Description
ITERMAX	integer scalar	maximum number of Newton-Raphson iterations
DELMAX	real array*	maximum allowed changes in primary variables
TARGET	real array*	target changes in primary variables
VTOL	real array*	convergence tolerances for primary variables
RRTOL	real array*	relative convergence tolerances for residuals
RATOL	integer scalar	absolute convergence tolerances for residuals
NREPORTS	integer scalar	number of times at which output is desired
RPTIMES	integer array of length NREPORTS	list of times at which output is desired [yr]
DTMAX	integer array of length NREPORTS	maximum allowed time step for various time periods [yr]

*Size of the array is equal to the number of unknowns per grid cell (e.g., 3 for the thermal hydrology module). The number of unknowns per grid cell is determined by the active PHYSICS module.

The time-stepping algorithm will attempt to adjust the size of the time step to match a target change in the primary variables as specified in the TARGET array. The time-step size may not exceed the time-dependent maximum specified by the RPTIMES and DTMAX arrays.

Time-step failure occurs if the number of iterations exceeds ITERMAX or if a calculated change in a primary variable exceeds the maximum allowed value specified in DELMAX.

An example of the CONTROL keyword block is shown in Figure 3-4.

```
CONTROL
 itermax = 20
 delmax= 0.7 0.3 0.02
 target= 0.05 0.05 0.005
 vtoler= 1.0e-5 1.0e-6 1.0e-6
 rtoler= 1.0e-4 1.0e-4 1.0e-4
 nreports= 10
 rptime= 0.1 0.2  1.   2   5  10  50. 100. 400.  1000. 2000.
100000.
 dtmax =    0.01 0.01 0.5 0.5  2  5.   10. 10.  20. 100. 100.
10000.
END CONTROL
```

Figure 3-4. Example of the CONTROL Keyword Block

3.6 Dual-Continuum Simulations

The *xFlo* system accommodates an optional dual-continuum mode wherein the rock matrix and interconnected fracture system are modeled as separate but interacting continua. The internal implementation of the dual-continuum mode in *xFlo* is through the grid definition. In discussing multiple-continua models, the grid resulting from the geometrical partitioning of the physical domain into grid cells is referred to as the primary grid. To generate the grid for a dual-continuum model, the primary grid is cloned so that there are two grid cells at every physical location. Volumes associated with fracture cells are calculated as the volume of the physical grid cell times the fracture volume fraction (bulk fracture porosity). Similarly, the surface area for each connection between fracture grid cells is calculated as the product of the surface area for the corresponding primary grid cell and the bulk fracture porosity. Surface areas for connections between fracture and matrix grid cells are required inputs and can be calculated from fracture spacing information.

In the internal data structures of *xFlo* there is no distinction between single- and dual-continuum simulations. The choice of single- or dual-continuum is handled purely by grid definition. Thus, for dual-continuum simulations involving an unstructured primary grid, the user must specify the dual-continuum grid in the `geometry.dat` file.

The structured grid preprocessing system will convert the primary grid to the dual-continuum grid if a structured-grid dual continuum simulation is requested. Parameters for the dual

continuum model are specified in the DCM keyword block. If the DCM keyword is not present, the simulation will be single continuum.

The DCM keyword block is of the grid-assignment type and is thus in fixed-order format. The format for the DCM keyword block is

```
DCM
I1 I2 J1 J2 K1 K2 VOLF BLKSIZE AREAFM
:
END DCM
```

Each line applies to a rectilinear subregion bounded by cell number I1 and I2 in the x-direction, J1 and J2 in the y-direction, and K1 and K2 in the z-direction. The cell numbers refer to the primary grid. The user is responsible for ensuring that all primary grid cells are assigned. If a grid cell is assigned to more than one of the sub-regions, then the last assignment is used.

VOLF	Fracture volume fraction (bulk fracture porosity) [-]
BLKSIZE	Fracture block size [m^2]
AREAMF	Fracture/matrix specific surface area [m^2/m^3]

An example of the DCM keyword block is shown in Figure 3-5.

```
DCM
 1 20 1 1 1 100 0.02  1.0 0.4
 1 20 1 1 1  20 0.01  1.0 0.2
END DCM
```

Figure 3-5. Example of the DCM Keyword Block

4 THERMAL HYDROLOGY PACKAGE

The thermal hydrology package is invoked by the presence of the THERMALHYDROLOGY keyword, which may be placed anywhere in the `*.flo` file.

4.1 Governing Equations

4.1.1 Mass Conservation Equations and Equilibrium Constraints

The conservation equations for the water and air components are given by Eq. (4-1)

$$\frac{\partial}{\partial t}\left[\phi \sum_{p=l,g} X_p^j \rho_p s_p\right] = -\nabla \cdot \sum_{p=l,g}\left[X_p^j \rho_p \mathbf{q}_p + \rho_p D_p^j \nabla X_p^j\right] + S^j \tag{4-1}$$

where

superscript j	—	a for air or w for water
subscripts l and g	—	liquid and gas phases
$\mathbf{q}$	—	Darcy flux [m/s]
s	—	is phase saturation
t	—	time [s]
ϕ	—	porosity
ρ	—	phase density on a molar basis [mol/m^3]
X	—	molar mass fraction
D	—	effective diffusion coefficient [m^2/s]
S	—	mass source rate [mol/m^3-s]

The Darcy flux for phase p is

$$q_p = -\frac{k k_{rp}}{\mu_p}\nabla\left(P_p + \rho_p g z\right) \tag{4-2}$$

where

P	—	phase pressure [Pa]
k	—	absolute (intrinsic) permeability [m^2]
k_r	—	relative permeability [-]
μ	—	dynamic viscosity [Pa·s]
z	—	vertical coordinate [m]
g	—	acceleration due to gravity [m/s^2]

The phase saturations and the mole fractions in each phase must sum to unity

$$s_l + s_g = 1 \tag{4-3}$$

$$X_p^g + X_p^w = 1 \qquad for \quad p = l, g \tag{4-4}$$

If the phase state is two-phase, then some additional equilibrium relationships are required. First, Henry's law is used to relate dissolved air fraction to the air pressure

$$X_l^a = HP_a \tag{4-5}$$

where

H — Henry's constant [Pa^{-1}]
P_a — partial pressure of air [Pa]

Second, water vapor is assumed to be in equilibrium with the liquid. That is, the vapor pressure follows the saturated pressure curve $P_{sat}(T)$, where T is temperature [K]. The total gas pressure is then

$$P_g = P_a + P_{sat}(T) \tag{4-6}$$

In addition, the liquid and gas pressures are related through the capillary pressure.

$$P_l = P_g - P_c(s_l) \tag{4-7}$$

It is computationally convenient to replace one of the conservation equations in Eq. (4-1) with a conservation equation for the total mass (air plus water). Summing the two equations eliminates the diffusive terms and results in

$$\frac{\partial}{\partial t}\left[\phi \sum_{p=l,g} \rho_p s_p\right] = -\nabla \cdot \sum_{p=l,g} \left[\rho_p \mathbf{q}_p\right] + S^a + S^w \tag{4-8}$$

The thermal hydrology package solves this equation for total mass coupled with the air equation from Eq. (4-1).

4.1.2 Capillary Pressure and Relative Permeability Functions

The thermal hydrological package provides two options for relating capillary pressure and relative permeability to liquid saturation: the van Genuchten (1980) phenomenological relation, and the active fracture model (Liu, et al., 1998).

Capillary pressure is a function of liquid saturation in van Genuchten's relation

$$P_c\left(s_{l,eff}\right) = \frac{1}{\alpha}\left[s_{l,eff}^{-1/\lambda} - 1\right]^{1/m} \tag{4-9}$$

where $\lambda = 1 - 1/m$,

$$s_{l,eff} = \frac{s_l - s_r}{1 - s_r} \tag{4-10}$$

is the effective saturation and s_r is the residual liquid saturation. The parameters α [Pa^{-1}] and m [-] are fitting parameters in the van Genuchten relation.

Relative permeability is related to saturation by combining van Genuchten's relation with Mualem's (1976) model

$$k_{rl} = \sqrt{s_{l,eff}}\left[1 - \left(1 - s_{l,eff}^{1/\lambda}\right)^{\lambda}\right]^2 \tag{4-11}$$

Gas-phase relative permeability is assumed to be a linear function of saturation.

The active fracture model (Liu, et al., 1998) is a modification to the van Genuchten/Mualem relation that is applicable to the fracture system in a dual-continuum model. The active fracture model introduces another parameter γ, which is in the range zero (inclusive) to one (exclusive). In the special case $\gamma = 0$, the van Genuchten model corresponds to the active fracture model.

In the active fracture model, the capillary pressure and relative permeability in the fracture system are modified to become

$$P_c\left(s_{l,eff}\right) = \frac{1}{\alpha}\left[s_{l,eff}^{(\gamma-1)/m} - 1\right]^{1/m} \tag{4-12}$$

and

$$k_{rl} = s_{l,eff}^{(1+\gamma)/2}\left[1 - \left(1 - s_{l,eff}^{(1-\gamma)/\lambda}\right)^{\lambda}\right]^2 \tag{4-13}$$

In addition, the active fracture model introduces a new relative permeability function that applies only to flow from the fracture continuum to the matrix continuum

$$k_{rl}^{f\to m} = s_{l,eff}^{\left(1-\gamma^2\right)/2}\left[1 - \left(1 - s_{l,eff}^{(1-\gamma)/\lambda}\right)^{\lambda}\right]^2 \tag{4-14}$$

4.1.3 Energy Balance Equation

The energy balance equation, assuming thermal equilibrium between rock and fluid, is given by where

$$\frac{\partial}{\partial t}\left[\sum_{p=l,g}\left(\phi\rho_p s_p u_p\right)+\left(1-\phi\right)\rho_s C_p^s T\right] = \quad -\sum_{p=l,g}\nabla\cdot\left(\rho_p h_p q_p\right) \; + \; \nabla\cdot\left[k_e\nabla T\right] + S_E \tag{4-15}$$

u_p and h_p — internal energy and enthalpy of phase *p* defined on a molar bases [J/mol]
ρ_s and C_p^s — density [kg/m^3] and heat capacity [J/K·kg] of the mineral solid
k_e — effective thermal conductivity [W]
S_E — energy source rate [J/m·K]

4.1.4 Thermophysical Properties

The effective diffusion coefficients in the gas phase are

$$D_g^a = D_g^w = \phi s_g \tau \frac{P_0}{P_g}\left(\frac{T}{T_0}\right)^{3/2} D_{g0} \tag{4-16}$$

where a weak dependence on molecular weight has been neglected and

P_0 — pressure [Pa]
T_0 — temperature [K]
τ — tortuosity
D_{g0} — binary diffusion coefficient [m^2/s] at reference conditions

Effective thermal conductivity is modeled as

$$k_e = k_{dry} + \sqrt{s_l}\left(k_{wet} - k_{dry}\right) \tag{4-17}$$

where

k_{dry} — dry thermal conductivities
k_{wet} — wet thermal conductivities

The saturated vapor pressure as a function of temperature is calculated in the thermal hydrology model by linear interpolation of tabulated values. Thermophysical properties of water as a function of pressure and temperature are calculated by bilinear interpolation of two-dimensional tables. The thermophysical properties tables span the temperature range 273.16–647 K [32–705 °F] and the pressure range 611 Pa–22 MPa [6.11 mBar–220 Bar].

4.2 Assignment of Rock Properties to Grid Cells

Rock properties are assigned to individual grid cells in the PHYSICS keyword block. All properties can be assigned cell by cell except for capillary pressure and relative permeability relationships. Capillary pressure and relative permeability relationships are assigned to a particular rock type. Each grid cell is then associated with a particular rock type.

The PHYSICS keyword block has two sub-blocks. Capillary pressure and relative permeability relationships are first defined in the PCKR sub-block. Grid blocks are assigned properties and associated capillary pressure and relative permeability curves in the PROPERTIES sub-block.

The format for the PHYSICS keyword block is

```
PHYSICS
DIRPATH
PCKR NMODELS
TYPE PARAMETERS
:
PROPERTIES
I1 I2 J1 J2 K1 K2 PERM POR KDRY KWET DIFFLIQ CPR RHO IMODEL ...
:
END PHYSICS
```

Here DIRPATH is the file system path to the directory containing the data files for thermophysical properties at the water (see Chapter 2).

The parameters appearing in the PCKR sub-block are as follows.

NMODELS	Number of PCKR models to be defined
TYPE	Keyword defining the capillary pressure/relative permeability model
PARAMETERS	Set of parameters defining the curve

The allowed values for TYPE are “VG” for the van Genuchten model and “AF” for the active fracture model. If VG is specified, parameters α, λ, and s_r are read; if AF is specified, parameters α, λ, s_r, and γ are read. These parameters are defined in Section 4.1.1.

Each line in the PROPERTIES sub-block applies to a rectilinear subregion bounded by cell numbers I1 and I2 in the x-direction, J1 and J2 in the y-direction, and K1 and K2 in the z-direction. The cell numbers refer to the primary grid. The user is responsible for ensuring that all primary grid cells are assigned. If a grid cell belongs to more than one of the subregions, then the last property assignment is used.

The other parameters appearing on each line of the PROPERTIES keyword block are as follows.

PERM	Permeability [m^2]
POR	Porosity [-]

KDRY	Thermal conductivity of dry rock [W/m-K]
KWET	Thermal conductivity of fully saturated rock [W/m-K]
DIFFLIQ	Diffusion coefficient of air in the liquid phase [m^2/s]
CPR	Specific heat of the rock [J/kg-K]
RHO	Rock density [kg/m^3]
IMODEL	Model from the list defined in the PCKR sub-block.

The IMODEL parameter is an integer index that associates a model defined in the PCKR sub-block to the cells in the subregion.

If the DCM option is active, then 16 parameters are read for each line. The eight fracture parameters are read first, followed by eight matrix parameters. Fracture parameters are defined on an intrinsic (not bulk) basis.

If the PHYSICS keyword is not present, the physical parameters are read directly from the `physics.dat` file. The format of the `physics.dat` file is the same as the PHYSICS keyword block in the `*.flo` file, except that the PHYSICS and PROPERTIES keywords are not included and the assignment is on a cell-by-cell basis with one line of input for each cell. Note that the assignment in the `physics.dat` file is to the DCM grid, not the primary grid. A fracture property value in cell 1 of the primary grid is assigned to cell 1 of the DCM grid; a matrix value in cell 1 of the primary grid is assigned to cell 2 of the DCM grid, and so on.

4.3 Specifying Initial Conditions

Initial conditions are specified in the INITIAL keyword block of the `*.flo` file. If the INITIAL keyword is missing, the initial conditions are read from the `initial.dat` file.

The format for the INITIAL keyword block is as follows.

```
INITIAL
I1 I2 J1 J2 K1 K2  PHASE U  P T …
:
END INITIAL
```

The data read on each line are as follows.

PHASE	Integer flag that determines the phase state 0 implies two-phase 1 implies all gas −1 implies no gas
U	First primary variable, which is dependent on phase state For two-phase conditions, U1 is gas saturation [-] For all-gas conditions, U1 is air partial pressure [Pa]

For no-gas conditions, U1 is mole fraction of air [-]

P — Phase pressure [Pa]
For an all-gas and two-phase conditions, P is gas pressure
For no-gas conditions, P is liquid pressure

T — Temperature [°C]

If the DCM option is active, then eight values are read on each line: four for the fractures and four for the matrix.

If the INITIAL keyword is missing from the `*.flo` file, then initial conditions are read from the `initial.dat` file. One line of data should be provided for each computational cell, in order, in the `initial.dat` file. For DCM simulations, the `initial.dat` file populates each cell in the DCM grid, not the primary grid. The data for each line are identical to that of the INITIAL keyword except that temperature should be in K and pressure should be in MPa in the `initial.dat` file.

4.4 Specifying Boundary Conditions

Boundary conditions are read from the BCON keyword block in the `*.flo` file. If the BCON keyword block is missing, boundary conditions are read from the `bcon.dat` file.

Version 1.0β of the thermal hydrology model supports two types of boundary conditions: Dirchlet and infiltration. If the Dirchlet option is chosen, then the user must specify the primary variables and the phase state at the boundary. If a given boundary is an infiltration type, then the user must specify the infiltration rate, the temperature, and the gas pressure. Note that infiltration boundaries are, by definition, two-phase.

The format for the BCON keyword block is as follows.

```
BCON
FACE BCTYPE M1 M2 N1 N2
TIME PHASE U P T …
:       ! repeat for multiple times
:       !  repeat above block for other faces, as needed
END BCON
```

Here FACE signals the beginning of a new boundary condition, and must be one of the strings “TOP,” “BOTTOM,” “LEFT,” “RIGHT,” “BACK,” or “FRONT.” The TOP boundary sets k=NZ, BOTTOM means k=1, LEFT means i=1, RIGHT means i=NX, FRONT means j=NY, and BACK means j=1, where i, j, k are the cell indices in the x-, y-, and z-direction, respectively.

The data for a given boundary condition apply to the subregion defined by cell numbers M1, M2, N1, and N2 on the specified face. For example, if LEFT is specified, then the boundary condtion applies to the cells with i=1, $M1 \le j \le M2$, $N1 \le k \le N2$.

The BCTYPE parameter determines the type of boundary condition and must be either "DIRCHLET" or "INFILTRATION."

Boundary conditions in *xFlo* may be time dependent. Thus, for each boundary condition, multiple lines of data may be entered. Each line of data applies at a specified time. The data entry for a given boundary condition terminates when a new FACE or the END statement is encoutered. The data to be entered for each line are:

TIME	Time [yr]
PHASE	Integer flag that determines the phase state For INFILTRATION type, not used For DIRCHLET type, same as in the INITIAL keyword
U	For INFILTRATION type, the infiltration rate For DIRCHLET type, same as in the INITIAL keyword
P	Pressure [Pa]
T	Temperature [°C]

If the DCM option is active, then eight values are read for each time: four for the fractures and four for the matrix.

If the BCON keyword block is missing, boundary conditions are read from the `bcon.dat` file. The data in the `bcon.dat` file are assigned directly to individual boundary cells. If the DCM option is specified, the boundary conditions are assigned to cells in the DCM grid, not the primary grid. The format for the `bcon.dat` file is as follows.

```
NBCELLS
MEXT MINT TYPE NTIMES
  TIME PHASE U P  T  FLUX1 FLUX2 FLUX3
   :   ! repeat for a total of NTIMES
:   ! repeat above block for a total of NBCELLS
```

where NBCELLS is the total number of cells to be assigned boundary conditions.

To set the boundary conditions in an unstructured grid, it is necessary to understand the cell numbering scheme in *xFlo*. The algorithm for incorporating boundary conditions in *xFlo* uses fictitious external cells. The external cells are assigned values for the primary variables, but are never updated. The cell numbering scheme places the external cells at the end of the cell list. Thus, if a simulation has a total of NCELLS internal cells and NBCELLS of these are associated with boundary conditions, then the total number of cells would be NCELLS+NBCELLS and the last NBCELLS listed would be external cells.

The data read for each of the boundary cells are as follows.

MEXT	Node number for the external cell

MINT	Node number for the internal cell connected to the boundary
TYPE	Boundary condition type (INFILTRATION or DIRCHLET)
NTIMES	Number of times used to specify the boundary time history
PHASE	For INFILTRATION type, must be set to 0 For DIRCHLET type, same as in the INITIAL keyword
U	For INFILTRATION type, not used For DIRCHLET type, same as in the INITIAL keyword
P	Pressure [MPa]
T	Temperature [K]
FLUXA	For DIRCHLET type, not used For INFILTRATION type, flux of dissolved air [mol/m^2-yr]
FLUXM	For DIRCHLET type, not used For INFILTRATION type, air plus water flux [mol/m^2-yr]
FLUXE	For DIRCHLET type, not used For INFILTRATION, energy flux into the system [MJ/m^2-yr]

Note that the user is responsible for correctly setting the incoming fluxes for unstructured grids and INFILTRATION boundary conditions. The FLUXA is the flux of air dissolved in the infiltrating water, the FLUXM is the total mass flux into the system including the infiltrating water and the dissolved air, and FLUXE is the enthalpy of the infiltrating water.

4.5 Specifying Mass and Heat Sources

Boundary conditions are read from the SOURCE keyword block in the `*.flo` file. If the SOURCE keyword block is missing, boundary conditions are read from the `source.dat` file.

Version 1.0β of the thermal hydrology package supports three types of sources: heat sources with no mass, water sources (including dissolved air), and air sources. For heat sources, only the input power is required. For air sources, the air mass input rate and the temperature and pressure of the incoming air are required. For water sources, the water input rate, the temperature and pressure of the incoming water, and the mole fraction of air dissolved in the water are required.

For DCM simulations, twice as many items of data are read for each time: the set of fracture values are read first, followed by the set of matrix values.

The format of the SOURCE keyword block is as follows.

```
SOURCE
 STYPE BASIS   I1 I2 J1 J2 K1 K2
```

```
    TIME DATA ...
     :   ! repeat for multiple times
   :     ! repeat sub-block for multiple sources
END SOURCE
```

Source data are specified by subregions. The integers I1, I2, J1, J2, K1, and K2 define a subregion.

The character string BASIS defines whether the source is specified on a volumetric basis or defined for the entire subregion. The allowed values are “VOLUMETRIC” or “TOTAL” for a volumetric or total rate, respectively.

The character string STYPE defines the source type. Allowed values are “AIR,” “WATER,” and “HEAT.”

Transient sources are allowed for each subregion. One line of data is read for each time in the tabulated source. The variable TIME is the time in years.

If STYPE is set to “HEAT,” then one item of data is read per each time (two for a dual-continuum simulation). That item is the input power. Units are either W/m^3 or W, depending on whether volumetric or total basis was chosen.

If STYPE is set to “AIR,” then the data are air source rate [kg/m^3-yr or kg/yr], pressure of the air [Pa], and temperature of the air [°C].

If STYPE is set to “WATER,” then the required data are water source rate [kg/m^3-yr or kg/yr], pressure of the incoming water [Pa], temperature of the incoming water [°C], and mole fraction of air in the water [-].

If the SOURCE keyword block is missing, boundary conditions are read from the `source.dat` file. The data in the `source.dat` file are assigned directly to individual boundary cells. If the DCM option is specified, the boundary conditions are assigned to cells in the DCM grid, not the primary grid. The format for the `source.dat` file is as follows.

```
NSCELLS
M  TYPE NTIMES
  TIME RATEA RATEM RATEH
   :   ! repeat for a total of NTIMES
:   ! repeat above block for a total of NSCELLS
```

where NSCELLS is the total number of cells to be assigned boundary conditions.

The data read for each of the source cells are as follows.

M	Node number for the cell
TYPE	Not used in Version 1.0β
NTIMES	Number of times used to specify the source time history

For each of the tabulated times in the source history, the following are read.

TIME	Time [yr]
RATEA	Source rate for air [mol/yr]
RATEM	Source rate for total mass (air plus water) [mol/yr]
RATEH	Source rate for energy [MJ/yr]

If the option of reading directly from the `source.dat` file is chosen, the user is responsible for ensuring that the rates are correctly specified. In particular, if the mass injection options are used, then the associated enthalpy of the incoming fluid must be correctly specified in the RATEH field.

4.6 Thermal Hydrology Package Output

As described in Section 3.5, *xFlo* produces output files at user-specified report times. Version 1.0β produces three types of output files: primary variable files, secondary variable files, and mass files.

Primary variable files have the root name `results` and are numbered sequentially. Thus, the primary variables at the first reporting time would have the name `results1`. The header for each primary variable file contains the report time. The contents of the primary variable files are in the same format as the `initial.dat` file, except for the header. Thus, upon removing the header, a primary variable file may be used as an `initial.dat` file in a restart run.

The secondary variable file at the i^{th} report time has the name `secondai`. The following secondary variable files are written: liquid saturation, gas pressure [Pa], temperature [K], and molar fraction of air in gas [-].

The mass files contain the energy content and the total masses for each component in each cell. Each line contains total air mass, total water mass, and energy content for a cell.

4.7 Example Input File

An example `*.flo` file is shown in Figure 4-1. This one-dimensional repository example uses a dual-continuum grid and the active fracture model for the fracture continuum. The upper boundary is an infiltration boundary and the lower boundary is a Dirchlet condition. The lower boundary is below the water table. The INITIAL keyword block is missing, so initial conditions will be read from an `initial.dat` file. At t = 0, a heat source of 60 W is applied to grid cell 50. The heat source is applied to the matrix continuum. No heat is applied to the fracture continuum. The heat source is constant for 50 years and then ramps down to 0 at 100 years.

```
 GRID XYZ VARIABLE
  nx =1   ny=1 nz= 100
  xmin=0. xmax=1.0
  ymin=0. ymax=1.0
  zmin=0. zmax=1000.
  dx = 1.
  dz= 45*10.  10*1. 45*10.
  dy=1.0
END Grid

THERMALHYDROLOGY

PHYSICS
/home/spainter/xFlo/xFlo1.0/data/
 PCKR 3
VG 1.e-5 0.5 0.01 0.02   ! this is a comment
AF 5.e-4 0.6 0.00 0.05 0.4  ! active fracture
VG 5.e-4 0.6 0.01 0.05
 PROPERTIES
 1 1 1 1 1 100   5.0e-11 1.0 0.015 0.03 1.0e-11  800. 2000.  2  1.0e-16 0.2  1.5
       3.0  1.0e-9   800. 2000.  1       ! Continuation of above line
END PHYSICS

DCM
 1 1 1 1 1 100 0.01  0.5 10.
END DCM

BCON
 TOP INFILTRATION 1 1 1 1
 0.0 0  0.05  90000. 10.  0 0.001 90000. 10.
 BOTTOM DIRCHLET 1 1 1 1
 0.0 1 1.0d-5 1000000. 20.0  1 1.0d-5 1000000. 20.0
END BCON

CONTROL
 itermax = 30
 delmax= 0.4 0.3 0.02
 target= 0.05 0.05 0.005
 vtol= 1.0e-4 1.0e-5 1.0e-5
 rrtol= 1.0e-4 1.0e-4 1.0e-4
 nreports= 14
 rptime= 0.1  0.2  1.  1.9   5  10  35. 50.  100. 400.  1000. 2000. 100000. 500000.
 dtmax = 0.01 0.01 0.01 0.01  1  1.   1. 5.  5.  10.    20. 20. 10000.  20000.
END CONTROL

SOURCE
 HEAT TOTAL  1 1 1 1 50 50
 0.0  0.  60.
 50. 0.  60.
 100. 0.  0.
END SOURCE
```

Figure 4-1. Example `*.flo` File for a One-dimensional Simulation Using the Thermal Hydrology Package

5 PLANS FOR FUTURE DEVELOPMENT

The original software requirements description for *xFlo* specified a staged release of the code with thermal-hydrological capabilities in Version 1.0 and thermal-hydrological-chemical capabilities in Version 2.0. Consistent with this plan, future work on the *xFlo* code will eventually shift toward reactive-transport capabilities. However, immediate plans for the *xFlo* code are to test and validate Version 1.0. Minor revisions will be released as needed if coding errors are identified or if minor enhancements are necessary.

6 REFERENCES

Lichtner, P.C. "Continuum Formulation of Multicomponent-Multiphase Reactive Transport." *Reviews of Mineralogy Volume 34: Reactive Transport in Porous Media*. P.C. Lichtner, C.I. Steefel, and E.H. Oelkers, eds. Washington, DC: Mineralogical Society of America. 1996.

Liu, H.H., C. Doughty, and G.S. Bodvarsson. "An Active Fracture Model for Unsaturated Flow and Transport in Fractured Rocks." *Water Resources Research*. Vol. 34. pp. 2,633–2,646. 1998.

Mualem, Y. "A New Model for Predicting the Hydraulic Conductivity of Unsaturated Porous Media." *Water Resources Research*. Vol. 12. pp. 513–522. 1976.

Narasimhan, T.N. and P.A. Witherspoon. "An Integrated Finite Difference Method for Analyzing Fluid Flow in Porous Media." *Water Resources Research*. Vol. 12. pp. 57–64. 1976.

Painter, S. and M. Seth. "MULTIFLO User's Manual." Two-Phase Nonisothermal Coupled Thermal-Hydrologic-Chemical Flow Simulator. Version 2.0. San Antonio, Texas: CNWRA. 2003.

van der Kwaak, J.E., P.A. Forsyth, and E.A. Sudicky. " WATSOLV, Sparse Matrix Iterative Solver Package, User's Guide." Waterloo, Ontario, Canada: Waterloo Centre for Groundwater Research, University of Waterloo. 1995.

van Genuchten, M.T. "A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils." *Soil Science Society of America Journal*. Vol. 44. pp. 892–898. 1980.