

# **xFLO VERSION 1.1 $\beta$ USER'S MANUAL DECOVALEX-2015 VERSION**

*Prepared for*

**U.S. Nuclear Regulatory Commission  
Contract NRC-02-02-012**

*Prepared by*

**Stuart Stothoff<sup>1</sup>  
Scott Painter<sup>2</sup>**

**<sup>1</sup>Center for Nuclear Waste Regulatory Analyses  
San Antonio, Texas**

**<sup>2</sup>Oak Ridge National Laboratory  
Oak Ridge, Tennessee**

**April 2016**

## ABSTRACT

Coupled-process simulators are important tools for understanding the near-field environment of a potential high-level radioactive waste repository. The xFlo code is under development as a general modeling tool for simulating coupled processes in porous media, providing a flexible simulation framework that can be modified to accommodate various coupled-process models. The simulation framework consists of a set of generic numerical routines that assemble and solve coupled balance equations, linked to a set of physics packages. Each physics package encapsulates a specific set of coupled processes independent of the numerical routines. Version 1.0 $\beta$  of xFlo (Painter, 2006) was intended to support possible performance confirmation activities, and included one physics package (the thermal hydrology package) that considers nonisothermal two-phase (water and air) thermal hydrology processes in single- and dual-continuum (fractured) porous media. Associated with the U.S. Nuclear Regulatory Commission/Center for Nuclear Waste Regulatory Analyses participation in DECOVALEX-2015, the thermal hydrology package was adapted to perform modeling tasks associated with swelling buffers. This User's Manual for Version 1.1 $\beta$  of xFlo documents these adaptations and other updates to Version 1.0 $\beta$  that occurred prior to participation in DECOVALEX-2015. This User's Manual also describes aspects of an additional physics package independently developed for the MarsFlo simulator (Painter, 2011) that extend the thermal hydrology package to represent nonisothermal variably frozen water/air/solid systems. This User's Manual provides a general overview of the computational framework and underlying equations, and describes the input format, focusing on aspects of the model used for DECOVALEX-2015. The variably frozen package was not used or tested during DECOVALEX-2015 activities; thus, only high-level aspects of this package are described.

## REFERENCES

Painter, S. "xFLO Version 1.0 $\beta$  User's Manual." San Antonio, Texas: Center for Nuclear Waste Regulatory Analyses. 2006.

Painter, S. "Three-Phase Numerical Model of Water Migration in Partially Frozen Geological Media: Model Formulation, Validation, and Applications." *Computational Geoscience*. Vol. 15 Issue 1. pp. 69–85. DOI 10.1007/s10596-010-9197-z. 2011.

# CONTENTS

Section	Page
ABSTRACT .....	ii
FIGURES .....	v
TABLES .....	vi
ACKNOWLEDGMENTS .....	vii
 1 INTRODUCTION .....	 1-1
1.1 Overview of the xFlo Generic Framework .....	1-2
1.2 Overview of the Physics Packages .....	1-3
1.2.1 Thermal Hydrology Package .....	1-3
1.2.2 Variably Frozen Fresh Water Package .....	1-3
1.3 Validation Status .....	1-4
 2 INSTALLATION AND EXECUTION .....	 2-1
 3 THEORETICAL BASIS FOR PHYSICS PACKAGES .....	 3-1
3.1 Governing Equations .....	3-1
3.1.1 Mass Conservation Equations and Equilibrium Constraints .....	3-1
3.1.2 Composite Mass Balance Equations .....	3-2
3.1.3 Phase Fluxes .....	3-2
3.1.4 Energy Balance Equation .....	3-3
3.2 Constitutive Relationships .....	3-3
3.2.1 Equilibrium Between Phases .....	3-3
3.2.2 Capillary Pressure and Relative Permeability Functions .....	3-4
3.3 Thermal Hydrology Package .....	3-8
3.3.1 Component Properties .....	3-8
3.3.2 Gas Phase Properties .....	3-8
3.3.3 Liquid Phase Properties .....	3-10
3.3.4 Solid Phase Properties .....	3-11
3.3.5 Properties for Fluxes in the Gas Phase .....	3-11
3.3.6 Properties for Fluxes in the Liquid Phase .....	3-12
3.3.7 Properties for Energy Fluxes .....	3-13
3.4 Variably Frozen Fresh Water Package .....	3-13
3.4.1 Constitutive Relationships Above the Freezing Point for Water .....	3-14
3.4.2 Constitutive Relationships Below the Freezing Point for Water .....	3-15
 4 GENERIC INPUT .....	 4-1
4.1 Input Data Files for Structured and Unstructured Grids .....	4-1
4.2 Data Formats in the Preprocessor Input File .....	4-1
4.3 Specifying the Grid .....	4-3
4.4 Control of the Linear Solver .....	4-4
4.5 Time Step and Output Control .....	4-6
4.6 Dual-Continuum Simulations .....	4-7
 5 PHYSICS PACKAGES .....	 5-1
5.1 Assignment of Rock Properties to Grid Cells .....	5-1
5.1.1 File Format for physics.dat .....	5-1
5.1.2 File Format for Structured Grids in *.flo .....	5-3

5.1.3	Thermal Hydrology Package Options and Properties.....	5-4
5.1.4	Variably Frozen Fresh Water Package Options and Properties .....	5-5
5.2	Specifying Initial Conditions .....	5-5
5.2.1	Thermal Hydrology Initial Conditions .....	5-6
5.2.2	Variably Frozen Fresh Water Initial Conditions .....	5-6
5.3	Specifying Boundary Conditions .....	5-6
5.3.1	Thermal Hydrology Boundary Conditions .....	5-7
5.3.2	Variably Frozen Fresh Water Boundary Conditions .....	5-9
5.4	Specifying Mass and Heat Sources.....	5-9
5.4.1	Thermal Hydrology Sources .....	5-10
5.4.2	Variably Frozen Fresh Water Sources.....	5-11
5.5	Package Output.....	5-11
5.5.1	Thermal Hydrology Package Output.....	5-11
6	REFERENCES.....	6-1

## FIGURES

Figure	Page
1-1	High-Level Data Structure and Data Flows for the xFlo System ..... 1-3
4-1	Data Flow for the xFlo System ..... 4-2
4-2	Example of the GRID Keyword Block ..... 4-4
4-3	Example of the SOLVER Keyword Block ..... 4-5
4-4	Example of the CONTROL Keyword Block ..... 4-6
4-5	Example of the DCM Keyword Block ..... 4-8

## TABLES

Table		Page
4-1	Parameters in the GRID Keyword Block .....	4-4
4-2	Parameters in the SOLVER Keyword Block .....	4-5
4-3	Parameters in the CONTROL Keyword Block.....	4-6

## 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-HQ-12-C-02-0089. The activities reported here were performed on behalf of the NRC Office of Nuclear Material Safety and Safeguards, Division of Spent Fuel Management. This manual is an independent product of the CNWRA and does not necessarily reflect the view or regulatory position of the NRC.

xFlo Version 1.0β was coded by S. Painter, who also wrote its user's manual. S. Painter is no longer at CNWRA and did not participate in updating the code or manual. He remains a co-author on the manual to recognize his contributions to the xFlo model, but any errors in the user's manual are the responsibility of the first author.

The authors acknowledge the technical review of G. Walter, the editorial review and programmatic review of David Pickett, and the assistance of L. Naukam and A. Ramos in preparing this report.

## QUALITY OF DATA, ANALYSES, AND CODE DEVELOPMENT

**DATA:** No data are 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, Control, Development, and Modification of Scientific and Engineering Software. The Software Release Notice, Software Requirements Description, Software Development Plan,, Software Validation Test Plan, and Software Validation Report are currently being revised to reflect the updates to xFlo Version 1.1β. The revisions will be completed before the public release of this xFlo version.

# 1 INTRODUCTION

Coupled-process simulators are important tools for understanding the near-field environment of a potential high-level waste repository. Coupled-process simulators that are easily modified to accommodate new process models are useful for examining different aspects of a repository system. The xFlo code is a flexible computational system that can be used and adapted to simulate various heat and mass transport processes in porous media. The xFlo code is intended to be an evolutionary development from the MULTIFLO family of codes (e.g., Lichtner, 1996; Painter and Seth, 2003). Version 1.0 $\beta$  of xFlo (Painter, 2006) was developed to consider potential performance confirmation activities associated with the proposed Yucca Mountain repository.

The design philosophy behind xFlo differs from the MULTIFLO family. Specifically, xFlo is designed as a general computational framework for solving conservation equations in 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 1.1 $\beta$  of xFlo includes two physics packages: (i) thermal hydrology and (ii) variably frozen fresh water.

The thermal hydrology package was developed to aid in understanding perturbations to the near-field environment that might occur at a high-level nuclear waste geologic disposal facility following emplacement of waste. This package was used to perform modeling tasks associated with the U.S. Nuclear Regulatory Commission/Center for Nuclear Waste Regulatory Analyses (NRC/CNWRA<sup>®</sup>) participation in the **DE**velopment of **CO**upled Models and their **VAL**idation Against **EX**periments (DECOVALEX)–2015 project (Stothoff et al., 2015; Manepally et al., 2015). The thermal hydrology package was tested extensively during the DECOVALEX–2015 cycle, and some of the routines were modified from Version 1.0 $\beta$ . This User’s Manual release is intended to document the functionality of the thermal hydrology package as used for DECOVALEX–2015.

The variably frozen fresh water (or frozen soil) package reproduces functionality of the MarsFlo code (Painter, 2011), which extended the thermal hydrology package from xFlo Version 1.0 $\beta$  using the same generic computational approach. The thermal hydrology package is limited to above-freezing conditions, whereas the frozen soil package focuses on below-boiling conditions. Aside from some differences in constitutive properties assumed for water, the thermal hydrology package in Version 1.0 $\beta$  and the variably frozen fresh water package are almost identical between the freezing and boiling points. The variably frozen fresh water package was not evaluated or updated during the DECOVALEX–2015 activities.

This User’s Manual’s primary purpose is to provide detailed data format and instructions on preparing input data for xFlo Version 1.1 $\beta$ . Overviews of the mathematical models solved by the physics packages 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 nonlinear terms. The Jacobian matrix required for the 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.

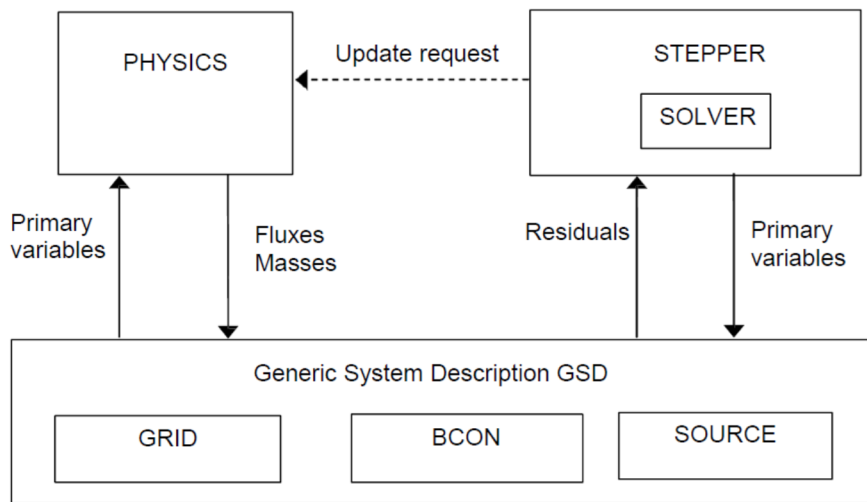
The linear system of equations resulting from each Newton-Raphson iteration is solved by a sparse preconditioned conjugate gradient solver. The solver is a Fortran 95 adaptation of the WATSOLV package (van der Kwaak et al., 1995). Incomplete lower-upper decomposition with fixed level of fill is used for the preconditioning step. Acceleration is achieved by the bi-conjugate 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: (i) STEPPER, (ii) PHYSICS, and (ii) 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. SOLVER, a submodule of STEPPER, solves the linear equations resulting from the discretization processes.

All information about the physics is encapsulated in the PHYSICS module, which is implemented as an abstract class. The PHYSICS module provides access to a library of submodules, each implementing an internally consistent and complete set of physics. Each submodule must present the same interface to the rest of the code. Currently only one submodule can be active at any time. Version 1.1 of xFlo has two PHYSICS submodules: (i) thermal hydrology and (ii) variably frozen fresh water.

Data are not passed directly between the STEPPER and PHYSICS modules. Instead, each module 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 sources, which are encapsulated in submodules. The Generic System Description also contains primary variables, inter-cell fluxes for each conserved quantity (masses and energy), and the mass/energy in each cell. The STEPPER module is responsible for updating primary variables in the Generic System Description, and the PHYSICS module is responsible for updating fluxes, masses, and energy given the current values of the primary variables. The only direct communication between the STEPPER and PHYSICS modules is in the form of an update request that 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.



**Figure 1-1. High-Level 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.**

## 1.2 Overview of the Physics Packages

### 1.2.1 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. Liquid-to-soil phase changes (freezing and melting) are not considered, but above-boiling conditions are considered.

In a two-fluid-phase model, there are three possible fluid phase states: (i) two-phase, (ii) all liquid, or (iii) 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.

### 1.2.2 Variably Frozen Fresh Water Package

The variably frozen fresh water package is based on the MarsFlo program (Painter, 2011). The considered physics are closely analogous to the thermal hydrology package, except that liquid-to-solid phase changes (freezing and melting) are included, but not above-boiling conditions. Salinity effects are not considered.

The variably frozen fresh water 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, ice, or gas (vapor) phase. The strategy for primary variables is identical to the thermal hydrology package. Ice content is determined from the primary state variables, assuming that the retention relations valid for water are also valid for ice once surface tension effects are accounted for.

The variably frozen fresh water package can be set to consider Earth or Mars. Selecting one of these options determines the acceleration due to gravity and the composition of the dry gas phase.

### **1.3 Validation Status**

The thermal hydrology package of xFlo Version 1.1 $\beta$  has been tested on analytical problems<sup>1</sup> and two-dimensional example simulations of laboratory and field tests (Stothoff et al., 2015; Manepally et al., 2015), but has not undergone validation. The variably frozen fresh water package inherits the testing performed for MarsFlo (Painter, 2011).

---

<sup>1</sup>Stothoff, S. "xFlo Development and Analyses." Scientific Notebook No. 1177. San Antonio, Texas: Center for Nuclear Waste Regulatory Analyses. 2015.

## 2 INSTALLATION AND EXECUTION

Included with the Version 1.1 $\beta$  release of xFlo are source code, a makefile and compilation batch file, 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/Linux environments and provided by many compilers that work on the Microsoft® Windows® operating system. Version 1.0 $\beta$  of the xFlo software was developed and tested in the SunOS®/Solaris® and Windows XP environments. The Version 1.1 $\beta$  release was developed in a Linux environment using the gfortran compiler and ported to Windows 7 using the Lahey lgf compiler.

The makefile provided with Version 1.1 $\beta$  can be used to compile xFlo for the gfortran compiler under Linux. The definitions of the `compile` and `link` commands may need to be changed to compile on other systems. A batch file is provided to compile xFlo for the Lahey lgf compiler under the Windows 7 operating system.

The physics modules require data files describing the pressure-volume-temperature properties of water. The pressure-volume-temperature input files must all be placed in a single 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 physics input, as described in Section 5.1 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 internally to xFlo to form the complete file name.

### 3 THEORETICAL BASIS FOR PHYSICS PACKAGES

Both physics packages are based on the same generic solution approach and balance equations. The particular packages specialize the balance equations to represent the particular physics of interest.

#### 3.1 Governing Equations

All of the physics packages are based on a set of mass and energy balance equations for individual components in a various phases, augmented by a set of equilibrium constraints. With the assumption that each component is locally in equilibrium across all phases, individual mass balance equations are assembled into component balance equations across all phases present. With the assumption of local thermal equilibrium across all phases, the individual energy balance equations are assembled into an overall energy balance equation.

##### 3.1.1 Mass Conservation Equations and Equilibrium Constraints

The mass conservation equation for a generic component  $j$  in generic phase  $\alpha$  is given by

$$\frac{\partial}{\partial t} (\omega_{\alpha}^j \eta_{\alpha} \theta_{\alpha}) = -\nabla \cdot [(\omega_{\alpha}^j \eta_{\alpha} \mathbf{q}_{\alpha} - \eta_{\alpha} D_{\alpha}^j \nabla \omega_{\alpha}^j)] + \Gamma_{\beta\alpha}^j + \Upsilon_{\alpha}^j \quad (3-1)$$

where

$\omega_{\alpha}^j$	–	Molar mass fraction of species $j$ in phase $\alpha$ ( $0 \leq \omega_{\alpha}^j \leq 1$ )
$\eta_{\alpha}$	–	Phase density on a molar basis [mol/m <sup>3</sup> ]
$\theta_{\alpha}$	–	Phase volume fraction ( $0 \leq \theta \leq 1$ )
$t$	–	Time [s]
$\mathbf{q}_{\alpha}$	–	Phase volumetric flux [m <sup>3</sup> /s/m <sup>2</sup> ]
$D_{\alpha}^j$	–	Effective diffusion coefficient [m <sup>2</sup> /s]
$\Gamma_{\alpha\beta}^j$	–	Mass transfer rate of component $j$ from phase $\beta$ to phase $\alpha$ [mol/m <sup>3</sup> -s]
$\Upsilon_{\alpha}^j$	–	External mass source rate [mol/m <sup>3</sup> -s]

The generic mass balance equation includes unknowns  $\omega_{\alpha}^j$  and  $\theta_{\alpha}$ , thus there are more unknowns than equations. Constraint equations require that molar mass fractions sum to unity for each phase and the phase volume fractions sum to unity, providing additional equations to close the system.

$$\sum_j \omega_{\alpha}^j = 1 \quad \text{for each phase } \alpha \quad (3-2)$$

$$\sum_{\alpha} \theta_{\alpha} = 1 \quad (3-3)$$

The set of fluid phases is assumed to completely occupy the pore space, so the phase volume fraction for fluid phase  $\alpha$  can also be described using

$$\theta_{\alpha} = \phi S_{\alpha} \quad (3-4)$$

where

- $\phi$  – Porosity (volume fraction not occupied by solids)
- $S$  – Saturation

The phase saturations must sum to unity across the fluid phases. Note that the frozen soil packages consider ice to be an immobile fluid phase rather than a solid phase.

$$\sum_{\alpha} S_{\alpha} = 1 \quad (3-5)$$

### 3.1.2 Composite Mass Balance Equations

It can be computationally advantageous to replace a component balance equation with a composite mass balance equation. Composite equations may be advantageous when (i) the equation has a simpler form because terms in the equations cancel or (ii) the mass in the composite equation is dominated by the balance equation that it replaces.

A component balance equation is one type of composite mass balance equation. The component balance equation for component  $j$  obtained by summing component balance equations across all phases eliminates the interphase mass transfer term and avoids the need to specify the particular phase that externally supplied mass partitions into. A component balance approach is appropriate when component equilibrium is maintained across phases, because the equilibrium conditions mean that the individual balance equations are not independent.

$$\frac{\partial}{\partial t} \left( \sum_{\alpha} \omega_{\alpha}^j \eta_{\alpha} \theta_{\alpha} \right) = - \nabla \cdot \left[ \sum_{\alpha} (\omega_{\alpha}^j \eta_{\alpha} \mathbf{q}_{\alpha} - \eta_{\alpha} D_{\alpha}^j \nabla \omega_{\alpha}^j) \right] + \sum_{\alpha} Y_{\alpha}^j \quad (3-6)$$

A total mass balance equation is another type of composite balance equation. A total mass balance equation can be obtained by summing component balance equations across all components and phases:

$$\frac{\partial}{\partial t} \left( \sum_{\alpha} \eta_{\alpha} \theta_{\alpha} \right) = - \nabla \cdot \left[ \sum_{\alpha} \left( \eta_{\alpha} \mathbf{q}_{\alpha} - \sum_j (\eta_{\alpha} D_{\alpha}^j \nabla \omega_{\alpha}^j) \right) \right] + \sum_{\alpha} \sum_j (Y_{\alpha}^j) \quad (3-7)$$

The thermal hydrology package uses a component balance equation for the air component and a total mass balance equation for the air and water components. The variably frozen fresh water package uses component balance equations for both the air component and the water component.

### 3.1.3 Phase Fluxes

Each of the physics packages represents the volumetric phase flux for a fluid as a Darcy flux, which describes the volumetric flux with respect to a coordinate system moving with the solid matrix. In all physics packages, the solid matrix is assumed to have negligibly small velocity.

$$\mathbf{q}_\alpha = - \frac{k k_{r\alpha}}{\mu_\alpha} \nabla (P_\alpha + \rho_\alpha g z) \quad (3-8)$$

where

$P$	–	Pressure [Pa]
$k$	–	Absolute (intrinsic) permeability [m <sup>2</sup> ]
$k_r$	–	Relative permeability [-]
$\rho$	–	Mass density [kg/m <sup>3</sup> ]
$\mu$	–	Dynamic viscosity [Pa·s]
$z$	–	Vertical coordinate [m]
$g$	–	Acceleration due to gravity [m/s <sup>2</sup> ]

Defining the volumetric phase flux in terms of gradients in phase pressures adds an additional state variable in order to describe each mobile phase.

### 3.1.4 Energy Balance Equation

The energy balance equation describes the change in internal energy from a reference state. The energy balance equation, assuming thermal equilibrium between rock and fluid, is given by

$$\frac{\partial}{\partial t} \left[ \sum_\alpha (\eta_\alpha \theta_\alpha u_\alpha) \right] = - \nabla \cdot \left( \sum_\alpha \eta_\alpha h_\alpha \mathbf{q}_\alpha \right) + \nabla \cdot [k_e \nabla T] + S_E \quad (3-9)$$

where

$u_\alpha$	–	Internal energy of phase $\alpha$ defined on a molar basis [J/mol]
$h_\alpha$	–	Enthalpy of phase $\alpha$ defined on a molar basis [J/mol]
$k_e$	–	Effective thermal conductivity [W/m <sup>2</sup> ·K]
$T$	–	Temperature [K]
$S_E$	–	Energy source rate [J/m <sup>3</sup> ·K]

## 3.2 Constitutive Relationships

Additional equations are needed to provide the additional equations needed to account for these state variables. The additional equations are typically provided by interphase equilibrium constraints and by constitutive relationships relating phase volume fractions to differences in phase pressures across phase/phase interfaces.

### 3.2.1 Equilibrium Between Phases

Henry's Law relates the partial pressure of a slightly soluble gas species to the dissolved molar mass fraction in a solute:

$$x_s^j = H P_g^j \quad (3-10)$$

where

$x_s^j$	–	mole fraction of component $j$ in the solute [–]
$P_g^j$	–	partial pressure of component $j$ [Pa]
$H$	–	Henry's coefficient [Pa <sup>-1</sup> ]

Henry's coefficient generally depends on temperature, but the physics packages currently assume that the Henry's coefficient is constant.

The partial pressure of the solute in the gas phase is described in terms of a saturated partial pressure curve, modified by gas/solute interfacial curvature effects. The saturated partial pressure is defined for conditions with a flat interface. Only water is considered as a solute in the physics packages. The partial pressure is also referred to as vapor pressure for water. For water, the saturated vapor pressure is almost entirely determined by temperature:

$$P_{\text{sat}}^w = P_{\text{sat}}^w(T) \quad (3-11)$$

where

$P_{\text{sat}}^w$	–	Saturated vapor pressure for water [Pa]
$T$	–	Temperature [K]

Different functions calculate  $P_{\text{sat}}^w$  for the different physics packages.

Curvature of the gas/solute interface reduces the vapor pressure adjacent to the interface (called vapor pressure lowering). Considering vapor phase lowering is optional, and the different physics packages use different functions to account for vapor phase lowering. This relationship is expressed by

$$\text{RH} = \frac{P_g^w}{P_{\text{sat}}^w} = \exp\left(-\frac{\sigma}{\sigma_0} \frac{P_c}{\eta_\alpha RT}\right) \quad (3-12)$$

where

RH	–	Relative humidity [–]
$\sigma$	–	Surface tension [J/m <sup>2</sup> ]
$\sigma_0$	–	Surface tension under reference conditions [J/m <sup>2</sup> ]
$P_c$	–	Capillary pressure (difference between nonwetting- and wetting-phase pressures) [Pa]
$R$	–	Ideal gas constant [8.3143 J/mol-K]

By default, the surface tension ratio is assumed to be unity. The thermal hydrology package allows an option for calculating the surface tension ratio.

### 3.2.2 Capillary Pressure and Relative Permeability Functions

Capillary pressure is typically characterized in terms of the mix of phases and the pore size distribution. Interfacial curvatures depend on a variety of factors, such as the mix of phases, differences between phase pressures, interfacial surface tension, and pore characteristics.



Empirical relationships accounting for such factors provide additional equations that link pressure differences (expressed as capillary pressures) and fluid saturations, providing additional equations to account for the number of degrees of freedom in the balance equations. Often the capillary pressure is expressed in terms of phase saturations and parameters accounting for the range of pore sizes and shapes in the porous medium. The relative permeability of each fluid phase also depends on the configuration of fluid phases in the porous medium, thus empirical relationships are typically developed using the same parameters.

The available relationships differ among the physics packages, depending on the number of fluid phases present. The available relationships are based on two models, the van Genuchten (1980) phenomenological relation, and the active fracture model (Liu et al., 1998).

### 3.2.2.1 van Genuchten/Mualem Model

Capillary pressure is a function of liquid saturation in the van Genuchten (1980) relation

$$P_c(S_{l,\text{eff}}) = \frac{1}{\alpha_{vG}} \left[ S_{l,\text{eff}}^{-1/m} - 1 \right]^{1/n} \quad (3-12)$$

$$S_{l,\text{eff}} = \frac{S_l - S_r}{1 - S_r} \quad (3-13)$$

where

$S_{l,\text{eff}}$	–	Effective liquid saturation [–]
$S_l$	–	Liquid saturation [–]
$S_r$	–	Residual liquid saturation [–]
$\alpha_{vG}$	–	Characteristic capillary pressure parameter [ $\text{Pa}^{-1}$ ]
$n$	–	Pore size distribution parameter [–]

The parameters  $\alpha_{vG}$  [ $\text{Pa}^{-1}$ ] and  $n$  [–] are fitting parameters in the van Genuchten relation. Using the Mualem (1976) pore distribution model,

$$m = 1 - \frac{1}{n} \quad (3-14)$$

In the van Genuchten/Mualem model, liquid relative permeability is expressed as

$$k_{ri} = (S_{l,\text{eff}})^{1/2} \left[ 1 - \left( 1 - S_{l,\text{eff}}^{1/m} \right)^m \right]^2 \quad (3-15)$$

Internally, xFlo requires that  $0 \leq k_{ri} \leq 1$  regardless of the values of  $S_{l,\text{eff}}$ . The van Genuchten/Mualem model is accessed when a VG input flag is provided in the PCKR section of the physics input file.

### 3.2.2.2 Active Fracture Model

The active fracture model (Liu et al., 1998) is a modification to the van Genuchten/Mualem model that is applicable to the fracture system in a dual-continuum model. The active fracture model introduces another parameter  $0 \leq \gamma < 1$ . In the special case where  $\gamma = 0$ , the active fracture model reduces to the van Genuchten model. The  $\gamma$  parameter accounts for the effects

of a limited flow area in an unsaturated fracture system, which restricts exchange from the fractures to the matrix (but not from the matrix to the fractures). The capillary pressure and relative permeability relationships for the matrix, including relative permeability governing flow from the matrix to the fracture system, remain the same as in Section 3.2.2.1.

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

$$P_c(S_{l,\text{eff}}) = \frac{1}{\alpha_{vG}} \left[ S_{l,\text{eff}}^{(\gamma-1)/m} - 1 \right]^{1/n} \quad (3-16)$$

$$k_{ri} = S_{l,\text{eff}}^{(1+\gamma)/2} \left[ 1 - \left( 1 - S_{l,\text{eff}}^{(1-\gamma)/m} \right)^m \right]^2 \quad (3-17)$$

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

$$k_{ri}^{f \rightarrow m} = S_{l,\text{eff}}^{(1-\gamma^2)/2} \left[ 1 - \left( 1 - S_{l,\text{eff}}^{(1-\gamma)/m} \right)^m \right]^2 \quad (3-18)$$

The active fracture model is accessed when an AF input flag is provided in the PCKR section of the physics input file. The current xFlo implementation appears to apply the fracture-matrix term,  $k_{ri}^{f \rightarrow m}$ , for both fracture-fracture and fracture-matrix connections. This model was not used for DECOVALEX work.

### 3.2.2.3 Smoothing Adjustments

The  $P_c$  and  $k_r$  functions in xFlo Version 1.0 $\beta$  were adjusted after the xFlo Version 1.0 $\beta$  documentation by Painter (2006). The  $P_c$  and  $k_r$  relationships are adjusted near the extremes for liquid saturation. These adjustments are intended to improve convergence. For some combinations of parameters, the Newton-Raphson approach can exhibit poor convergence when capillary pressures are close to zero (i.e., at the transition from partially to fully water saturated). The Newton-Raphson approach may diverge when the solution is not sufficiently smooth (i.e., Jacobians are discontinuous). Some combinations of parameters create very sharp inflections at  $P_c = 0$ , so that very small time steps may be necessary to prevent successive Newton-Raphson updates from diverging when capillary pressures are near this transition.

The xFlo models use various strategies to smooth the relationship between capillary pressure and saturation at extreme saturations. Smoothing is only used when calculating capillary pressure from saturation (i.e., calculations for saturation from capillary pressure do not include adjustments). The thermal hydrology package currently does not calculate saturation from capillary pressure, but the variably frozen fresh water package does. Note that the active fracture package does not provide a function to calculate saturation from capillary pressure, thus the variably frozen fresh water package cannot be applied with the active fracture model.

The van Genuchten relationship requires that capillary pressure approaches infinity as  $S_l$  approaches zero. One extension to xFlo Version 1.0 $\beta$  linearizes the relationship below a user-specified minimum effective saturation,  $S_{\min}$ .

Another extension adjusted the van Genuchten/Mualem capillary pressure function in a range of effective saturation of  $S^* \leq S_{l,\text{eff}} < 1$ . The revised relation for this range is

$$P_c(S_{l,\text{eff}}) = \frac{1}{\alpha_{\text{VG}}} [a_{Pc} + b_{Pc}(1 - S_{l,\text{eff}})](1 - S_{l,\text{eff}}) \quad (3-19)$$

where

$a_{Pc}$	–	User-provided fitting parameter [–]
$b_{Pc}$	–	User-provided fitting parameter [–]
$S^*$	–	User-provided threshold effective liquid saturation [–]

The  $a_{Pc}$  and  $b_{Pc}$  parameters are required when a VG input flag is provided in the PCKR section of the physics input file. Due to lack of documentation on selecting parameters during DECOVALEX modeling, xFlo Version 1.1 $\beta$  was revised to ignore the user-supplied values and to use fixed values of  $a_{Pc}$  and  $b_{Pc}$ . The option to use user-supplied parameters or internally calculated values could be reinstated in the future.

The  $a_{Pc}$  and  $b_{Pc}$  parameters should be selected such that  $P_c(S_{l,\text{eff}})$  is continuous at  $S^*$ .

A linear option would be to set  $b_{Pc} = 0$ , in which case  $a_{Pc} = \alpha_{\text{VG}} P_c(S^*) / (1 - S^*)$  and  $P_c$  decreases linearly from  $P_c(S^*)$  to zero. This option introduces a discontinuity in the derivative of  $P_c$  with respect to  $S_{l,\text{eff}}$  at both  $S_{l,\text{eff}} = S^*$  and  $S_{l,\text{eff}} = 1$ , which may affect convergence rates near the discontinuities. Due to lack of documentation on selecting parameters, xFlo Version 1.1 $\beta$  was revised to follow this option, ignoring the user-supplied values. The xFlo Version 1.0 $\beta$  approach for the active fracture model automatically uses linear interpolation in the same range, and does not expect user values for the  $a_{Pc}$  and  $b_{Pc}$  parameters.

A quadratic option preserves the slope at  $S_{l,\text{eff}} = S^*$ . In this case,

$$a_{Pc} = \frac{2P_c(S^*)}{(1 - S^*)} - P'_c(S^*) \quad (3-20)$$

$$b_{Pc} = \frac{P'_c(S^*)}{(1 - S^*)} - \frac{P_c(S^*)}{(1 - S^*)^2} \quad (3-21)$$

where  $P'_c = dP_c/dS_{l,\text{eff}}$ . It would be preferable to hard code this relationship internally to xFlo if the quadratic option were reinstated, rather than have the user supply  $a_{Pc}$  and  $b_{Pc}$ . Compared to the linear option, the quadratic approach better follows the shape of the  $P_c(S_{l,\text{eff}})$  relationship but has a larger discontinuity in the derivative of  $P_c$  with respect to  $S_{l,\text{eff}}$  at  $S_{l,\text{eff}} = 1$ , which may adversely affect convergence.

The updated relative permeability function also includes smoothing for the van Genuchten/Mualem model (VG flag) and active fracture model (AF flag).

A quadratic relationship for relative permeability for  $S^* \leq S_{l,\text{eff}} < 1$ , exactly analogous to the quadratic relationship for capillary pressure, was added to xFlo after version 1.0 $\beta$ . The relationship requires two additional parameters, called  $a_{kr}$  and  $b_{kr}$ , that are exactly analogous to the  $a_{Pc}$  and  $b_{Pc}$  parameters. xFlo Version 1.1 $\beta$  was revised during DECOVALEX activities to retain the original van Genuchten/Mualem formulation and ignore the user-supplied values,

because (i) discontinuous derivatives in the relative permeability function have little influence on Newton-Raphson convergence and (ii) guidance for selecting the parameters was not available. Two placeholder values are still required to preserve backward compatibility, corresponding to  $a_{kr}$  and  $b_{kr}$ , whenever a VG input flag is provided in the PCKR section of the physics input file, but the parameter values are ignored in xFlo Version 1.1 $\beta$ .

The active-fracture version of the relative permeability function (associated with the AF flag in the physics file) linearly interpolates  $k_{ri}$  from  $k_{ri}(S^*)$  to one as a function of  $S_{l,eff}$ . In the active-fracture function,  $S^*$  is hard-coded to a value of 0.95.

### 3.2.2.4 Gas Relative Permeability

xFlo uses package-specific models to represent gas relative permeability, including Knudsen diffusion and enhanced gas permeability due to Klinkenberg effects (Klinkenberg, 1941).

## 3.3 Thermal Hydrology Package

The thermal hydrology package considers mass balance of two components (air and water) in two phases (gas and liquid) and energy balance of three phases (gas, liquid, and solid) on Earth. The solid phase is assumed immobile but compressible. Some of the constitutive relationships become undefined when  $T < 273.15$  K. The thermal hydrology package is invoked by the presence of the THERMALHYDROLOGY keyword, which may be placed anywhere in the \*.flo file.

The acceleration due to gravity is internally set to 9.81 m/s<sup>2</sup>. The energy balance equation is based on a reference state at the triple point of water.

### 3.3.1 Component Properties

The air component represents a composite of atmospheric gases. It is assigned a molar mass of 28.964 kg/kmol. Following the EOS3 module of TOUGH2 (Pruess et al., 2012), the air component is assumed to have a constant Henry's coefficient of  $10^{-10}$  Pa<sup>-1</sup> (air solubility at 60 °C). Pruess et al. (2012) justify neglecting temperature dependence based on the weak temperature dependence and small solubility of air in water.

The water component is assigned a molar mass of 18.02 kg/kmol.

### 3.3.2 Gas Phase Properties

The gas phase pressure and molar density are linked using the ideal gas law

$$P_g = P_g^a + P_g^w = \eta_g RT \quad (3-22)$$

The vapor pressure  $P_g^w$  is calculated using the formula

$$P_g^w = P_{sat}^w(T) RH(P_c, T) \quad (3-23)$$

where  $P_{sat}^w$  is calculated by linear interpolation of tabulated values provided in an input file. The default table spans the temperature range 273.16 to 647 K. The calculated relative humidity,

RH, depends on whether vapor-phase lowering and surface-tension scaling are enabled. Surface-tension scaling accounts for the temperature-dependent curvature of the interface between liquid and gas phases.

If vapor-phase lowering (VPL present in the `physics.dat` file) and surface-tension scaling (RHSURTEN present in the `physics.dat` file) are enabled,

$$RH(P_c, T) = \exp\left(-\frac{\sigma}{\sigma_0} \frac{P_c}{\eta_\alpha RT}\right) \quad (3-24)$$

$$\frac{\sigma}{\sigma_0} = \frac{(a_{stw} - b_{stw}T)}{(a_{stw} - b_{stw}T_{20})} \quad (3-25)$$

where

- $a_{stw}$  – Coefficient for surface tension (internally set to 0.118)
- $b_{stw}$  – Coefficient for surface tension (internally set to  $1.54 \times 10^{-4} \text{ K}^{-1}$ )
- $T_{20}$  – Reference temperature for surface tension (internally set to 293.15 K)

If surface-tension scaling is not enabled (the default condition),  $\sigma/\sigma_0$  is set to 1. If vapor-phase lowering is not enabled (the default condition), RH is set to 1.

The gas-phase internal energy is defined as the sum of component internal energies

$$u_g = x_g^a u_g^a + x_g^w u_g^w \quad (3-26)$$

$$u_g^a = C_{vg}^a (T - T_0) \quad (3-27)$$

$$u_g^w = a + b (T - T_0) + c(T - T_0)^2 \quad (3-28)$$

where  $C_{vg}^a$  is the heat capacity at constant volume for the air component in gas. xFlo internally sets  $C_{vg}^a$  to 21.12 J/mol-K based on values at 100 °C from CRC (1986), and internally sets  $a$  to 42,669.1 J/mol,  $b$  to 31.745 J/mol-K, and  $c$  to  $-0.0060797 \text{ J/mol-K}^2$ . The source for the vapor internal energy equation is internally undocumented, but is similar to tabulated values based on the International Association for the Properties of Water and Steam (IAPWS) formulation from 1995, which has a reference state of zero internal energy at the triple-point state for water ( $T = 273.16 \text{ K}$  and  $P_g^w = 611.73 \text{ Pa}$ ).

The gas-phase enthalpy is defined as

$$h_g = u_g + \frac{P_g}{\eta_g} \quad (3-29)$$

The gas-phase viscosity is calculated as a mixture of viscosities for pure gases

$$\mu_g = \sum_\alpha \frac{x_\alpha \mu_\alpha}{\sum_\beta x_\beta \Phi_{\alpha\beta}} \quad (3-30)$$

$$\Phi_{\alpha\beta} = \frac{1}{\sqrt{8}} \left( 1 + \frac{M_\alpha}{M_\beta} \right)^{-1/2} \left[ 1 + \left( \frac{\mu_\alpha}{\mu_\beta} \right)^{1/2} \left( \frac{M_\beta}{M_\alpha} \right)^{1/4} \right]^2 \quad (3-31)$$

$$\mu_\alpha = \frac{C_\alpha T^{3/2}}{T + T_{r\alpha}} \quad (3-32)$$

where

$\mu_\alpha$	–	Viscosity of gas for pure component $\alpha$ [Pa·s]
$M_\alpha$	–	Molar mass of component $\alpha$ [kg/mol]
$C_\alpha$	–	Fitting coefficient [kg/(s·m·K <sup>1/2</sup> )]
$T_{r\alpha}$	–	Fitting coefficient [K]

The equation for gas viscosity was proposed by Sutherland (1893). The  $C_\alpha$  coefficients are hard-coded at values of  $1.51175 \times 10^{-6}$  and  $1.75528 \times 10^{-6}$  kg/(s·m·K<sup>1/2</sup>) for air and water vapor, respectively. The  $T_{r\alpha}$  coefficients are hard-coded at values of 125.262 and 620.992 K for air and water vapor, respectively.

### 3.3.3 Liquid Phase Properties

Both liquid phase molar density and liquid phase internal energy,  $u_l$ , are calculated using bilinear interpolation of two-dimensional tables based on temperature and pressure. The tables are provided in input files. The tables define the mass density and specific internal energy of pure liquid water for selected temperatures and pressures, with the default input files spanning the temperature range 273.16 to 647 K and the pressure range 611 Pa to 22 MPa. The interpolated mass density is internally converted to molar density, and the interpolated specific internal energy is internally converted to a molar basis.

The liquid-phase enthalpy is defined as

$$h_l = u_l + \frac{P_g}{\eta_l} \quad (3-33)$$

The liquid phase molar density, internal energy, and enthalpy calculations all use the gas-phase pressure instead of the liquid-phase pressure. The gas pressure is internally set to the liquid pressure when no gas is present. Using gas pressure instead of liquid pressure is an excellent approximation for evaluating nearly incompressible liquid water under two-phase conditions. The liquid pressure is less useful, because the liquid pressure also accounts for surface effects that have little influence on water density.

The liquid-phase viscosity of pure water is defined using an empirical formula (CRC, 1986). Different relationships are used for temperatures above and below  $T_{\text{cross}} = 20$  °C. Defining  $T_{\text{cross}} = T - T_{\text{ref}}$  and  $\Delta = T - T_{\text{ref}} - T_{\text{cross}}$ , where  $T_{\text{ref}} = 273.15$  K, for  $\Delta < 0$

$$\log(\mu_l) = 1301 \left[ \frac{1}{998.333 + \Delta(8.1855 + 0.00585\Delta)} - \frac{1}{998.333} \right] - 3 \quad (3-34)$$

For  $\Delta > 0$

$$\log(\mu_l) = \frac{\Delta(1.3272 - 0.001053\Delta)}{T_{\text{cross}} + 105} - 3 \quad (3-35)$$

The expression yields  $\mu_l$  in units of Pa·s.

### 3.3.4 Solid Phase Properties

Solid phase properties are used for the energy balance equation. The internal energy is calculated assuming that change in internal energy is proportional to temperature in the form

$$u_s = u_{s0} + C_{PS}\rho_s(T - T_{\text{ref}}) \quad (3-36)$$

where

$C_{PS}$	–	Solid heat capacity [J/kg]
$\rho_s$	–	Solid density [kg/m <sup>3</sup> ]
$u_{s0}$	–	Internal energy at the reference state (assumed zero) [J/mol]

The  $C_{PS}$  and  $\rho_s$  properties are provided as user input. Solid phase enthalpy is not calculated, because the solid is assumed to be unmoving.

The mass terms in the balance equations use a corrected porosity in the form

$$\phi = \phi_0[1 + C_{PP}(P_g - P_{\text{ref}})] \quad (3-37)$$

where  $\phi_0$  is the user-supplied input porosity,  $C_{PP}$  (solid matrix compressibility) has a hard-coded value of  $10^{-9}$  Pa<sup>-1</sup>, and  $P_{\text{ref}}$  has a hard-coded value of 1,000 Pa. The equation and its values are not internally documented. The corrected porosity is not used to calculate fluxes.

### 3.3.5 Properties for Fluxes in the Gas Phase

Advective (Darcy) fluxes rely on the concept of relative permeability. In release 1.0β, the gas phase relative permeability  $k_{rg}$  was set to  $S_g$  (i.e., scaled linearly from 0 to 1 as gas saturation varied from 0 to 1). Painter (2006) based this simple relationship on the rationale that the gas phase is typically so mobile that the particular form of the relationship is usually not important.

During DECOVALEX activities exploring granular bentonite (Stothoff et al., 2015), the granular bentonite pellets appeared to be more permeable to gas under dry conditions than the linear scaling suggested. Accordingly, the relationship was revised to  $k_{rg} = 1 - k_{rl}$  for Version 1.1β.

Even with this adjustment to gas-phase relative permeability, gas permeability for the granular bentonite pellets under dry conditions appeared orders of magnitude larger than could be accounted for with the saturated permeability. This larger permeability is due to the large pores around the individual pellets, which are not present under saturated conditions. An optional method was added to account for this interpellet pore space using the Klinkenberg scaling

model (Klinkenberg, 1941), which increases the bulk gas permeability for porous media with very low permeability. The Klinkenberg model has the form

$$k_g = k_{rg} k \left( 1 + \frac{b}{P_g} \right) \quad (3-38)$$

where  $k_g$  is the gas permeability [m<sup>2</sup>] and  $b$  is the Klinkenberg scaling parameter [Pa]. This model is implemented in xFlo by scaling the gas relative permeability.

Binary diffusion of a component in a phase,  $J_\alpha^j$ , has the form

$$J_\alpha^j = -\eta_\alpha D_\alpha^j \nabla \omega_\alpha^j \quad (3-39)$$

The binary diffusion coefficients for the components in the gas phase have the form

$$D_g^a = D_g^w = \phi S_g \tau \frac{P_0}{P_g} \left( \frac{T}{T_0} \right)^{1.8} D_{g0} \quad (3-40)$$

where  $D_{g0}$  is the diffusion coefficient under reference conditions [m<sup>2</sup>/s] and  $\tau$  is the gas phase tortuosity [–]. xFlo currently internally sets  $D_{g0}$  to  $2.13 \times 10^{-5}$  m<sup>2</sup>/s,  $P_0$  to 10<sup>5</sup> Pa, and  $T_0$  to 273.15 K.

Tortuosity is implemented in the form

$$\tau = \tau_0 \phi^{1/3} S_g^{7/3} \quad (3-41)$$

where  $\tau_0$  is an optional user-specified tortuosity factor [–] set to 1 by default.

### 3.3.6 Properties for Fluxes in the Liquid Phase

The binary diffusion coefficient for the air component in the liquid phase has the form

$$D_l^a = \phi \eta_l S_l D_{l0} \quad (3-42)$$

where

$D_l^a$	–	Diffusion coefficient [m <sup>2</sup> /s]
$n_l$	–	Liquid molar density [mol/m <sup>3</sup> ]
$D_{l0}$	–	Diffusion coefficient under reference conditions [m <sup>2</sup> /s]

The value for  $D_{l0}$  is provided in the PHYSICS input files. Tortuosity in the liquid phase is not considered in the thermal hydrology package.



### 3.3.7 Properties for Energy Fluxes

Thermal conductivity is described using

$$\kappa_e = \kappa_{\text{dry}} + S_l^{\alpha_u} (\kappa_{\text{wet}} - \kappa_{\text{dry}}) \quad (3-43)$$

where

$\kappa_e$	–	Bulk thermal conductivity [J/m·s·K]
$\kappa_{\text{dry}}$	–	Thermal conductivity at $S_l = 0$ [J/m·s·K]
$\kappa_{\text{wet}}$	–	Thermal conductivity at $S_l = 1$ [J/m·s·K]
$\alpha_u$	–	Fitting parameter ( $\alpha_u > 0$ ) [–]

The  $\alpha_u$ ,  $\kappa_{\text{dry}}$ , and  $\kappa_{\text{wet}}$  parameters are user input values supplied for each grid cell. The relationship is concave down when  $\alpha_u < 1$  and concave up when  $\alpha_u > 1$ . Somerton et al. (1974) indicate that a value of 0.5 for  $\alpha_u$  is appropriate for sandy materials.

## 3.4 Variably Frozen Fresh Water Package

The variably frozen fresh water package is an implementation of the MarsFlo model, and the description by Painter (2011) should be consulted for the basis of the model. The variably frozen fresh water package was not used or tested during DECOVALEX activities.

The variably frozen fresh water package considers mass balance of two components (air and water) in three phases (gas, liquid, and ice) and energy balance of four phases (gas, liquid, ice, and solid) on Earth or on Mars. The air component is assumed to be present only in the gas and liquid phases. On Earth, the air component is the same as in the thermal hydrology package, while on Mars the air component is considered non-condensable carbon dioxide (the air component doesn't form a liquid or solid). The ice and solid phases are assumed immobile but compressible. The variably frozen fresh water package is invoked by the presence of the FROZENSOIL keyword, which may be placed anywhere in the \*.flo file.

The acceleration due to gravity is internally set to 9.81 m/s<sup>2</sup> when the planet is set to EARTH and is internally set to 3.6 m/s<sup>2</sup> when the planet is set to MARS. The energy balance equation is based on a reference state at the triple point of water.

The variably frozen fresh water package considers three balance equations: (i) air mass, (ii) water mass, and (iii) internal energy. This equation set is slightly different from the thermal hydrology package, in which the water mass balance equation is replaced by a balance equation for total air and water mass.

The variably frozen fresh water package uses similar constitutive relationships as the thermal hydrology package for temperatures above the freezing point of water, but some of the constitutive relationships become inaccurate at temperatures approaching the boiling point. Some of the constitutive relationships may become inaccurate or undefined at temperatures less than approximately 170 K.

The variably frozen fresh water package differs from the thermal hydrology package for temperatures below the freezing point of water, because the package explicitly considers the formation of ice and mixtures of ice, water, and gas in the pore space.

### 3.4.1 Constitutive Relationships Above the Freezing Point for Water

Above the freezing point for water, most constitutive relationships are the same for the thermal hydrology package and the variably frozen fresh water package. Only the constitutive relationships that are different above the freezing point are described in this section.

The saturated vapor pressure switches at the freezing point, calculated using

$$\log(P_{\text{sat}}^w) = a_0 + \frac{a_1}{T} + \frac{a_2}{T^2} \quad (3-44)$$

for  $P_{\text{sat}}^w$  and  $T$  in units of Pa and K, respectively, where

- $a_0$  – Internally set to 10.047763 ( $T \geq T_0$ ) or 12.534522 ( $T < T_0$ )
- $a_1$  – Internally set to  $-1606.6 \text{ K}$  ( $T \geq T_0$ ) or  $-2662.9 \text{ K}$  ( $T < T_0$ )
- $a_2$  – Internally set to  $-102989 \text{ K}^2$  ( $T \geq T_0$ ) or 0 ( $T < T_0$ )

Liquid density is calculated using

$$\rho_l = \rho_0 [1 + \kappa_T (P_g - P_0)] \quad (3-45)$$

for  $\rho_l$  and  $P_g$  in units of  $\text{kg/m}^3$  and Pa, respectively, where

- $\rho_0$  – Internally set to  $1000 \text{ kg/m}^3$
- $\kappa_T$  – Internally set to  $5 \times 10^{-10} \text{ Pa}^{-1}$
- $P_0$  – Internally set to  $10^5 \text{ Pa}$

Liquid internal energy is calculated using

$$u_l = u_{l0} (T - T_0) \quad (3-46)$$

where  $u_{l0}$  is internally set to  $76 \text{ J/mol} \cdot \text{K}$  and  $T$  is in units of K.

The binary diffusion coefficients for the components in the gas phase have the form

$$D_g^a = D_g^w = \frac{D_{12} D_{1K}}{D_{12} + D_{1K}} \quad (3-47)$$

$$D_{1K} = A_K r T^{1/2} \quad (3-48)$$

where

$D_{12}$	–	Binary diffusion coefficient calculated by the thermal hydrology package [m <sup>2</sup> /s]
$D_{1K}$	–	Knudsen diffusion coefficient [m <sup>2</sup> /s]
$A_K$	–	Knudsen coefficient (internally set to 22.86 [m/s·K <sup>1/2</sup> ])
$r$	–	Pore radius internally set to $5 \times 10^{-6}$ m

The tortuosity for components in a phase are considered in the form

$$\tau = \tau_0 \phi^{\beta_1 - 1} S_l^{\beta_2} \quad (3-49)$$

where

$\tau_0$	–	User-supplied reference tortuosity [–]
$\beta_1$	–	User-supplied fitting coefficient [–]
$\beta_2$	–	User-supplied fitting coefficient [–]

### 3.4.2 Constitutive Relationships Below the Freezing Point for Water

#### 3.4.2.1 Ice Properties

Ice molar density is calculated using

$$\eta_i = \eta_0 [1 + \alpha_i (P_g - P_0) + \beta_i (T - T_0)] \quad (3-50)$$

where

$\eta_i$	–	Ice molar density [mol/m <sup>3</sup> ]
$\eta_0$	–	Reference molar density (internally set to 50942.4 mol/m <sup>3</sup> )
$\alpha_i$	–	Fitting coefficient (internally set to $3.3 \times 10^{-10}$ Pa <sup>-1</sup> )
$\beta_i$	–	Fitting coefficient (internally set to $1.53 \times 10^{-4}$ K <sup>-1</sup> )
$P_0$	–	Reference pressure (internally set to 10 <sup>5</sup> Pa)

Ice molar internal energy is calculated using

$$u_i = a(T - T_0) + b(T^2 - T_0^2) + c\left(\frac{1}{T_0} - \frac{1}{T}\right) - h_{iw} M_w \quad (3-51)$$

where

$u_i$	–	Ice molar internal energy [J/mol]
$a$	–	Fitting coefficient (internally set to –10.6644 J/K)
$b$	–	Fitting coefficient (internally set to 0.0849 J/K <sup>2</sup> )

$c$	–	Fitting coefficient (internally set to $1.98148 \times 10^5 \text{ J}\cdot\text{K}$ )
$h_{iw}$	–	Heat of fusion (internally set to $3.34 \times 10^5 \text{ J/kg}$ )
$M_w$	–	Molar mass of water (internally set to $0.01802 \text{ kg/mol}$ )

### 3.4.2.2 Retention Relationships

The retention relationships for phase pairs are related to the retention relationship for the two-phase liquid water/gas system calculated using

$$S_{uf} = \frac{S_l}{S_l + S_i} = S_* \left( \frac{\sigma_{gl}}{\sigma_{il}} P_{cil} \right) \quad (3-52)$$

$$P_{cgl} = S_*^{-1} \left( \frac{S_l}{S_l + S_g} \right) \quad (3-53)$$

where

$S_*$	–	Retention relationship for the two-phase liquid water/gas system
$S_*^{-1}$	–	Inverse retention relationship for the two-phase liquid water/gas system
$S_{uf}$	–	Unfrozen water fraction [–]
$\sigma_{gl}$	–	Gas-liquid surface tension [N/m]
$\sigma_{il}$	–	Ice-liquid surface tension [N/m]
$P_{cil}$	–	Capillary pressure between ice and liquid [Pa]
$P_{cgl}$	–	Capillary pressure between gas and liquid [Pa]

For this calculation,  $\sigma_{gl}$  and  $\sigma_{il}$  are hard-coded to 0.077 and 0.033 N/m, respectively.

The value of  $S_g$  is always known, because it is used as a primary variable in cells with the two-phase state and is known to be 0 for the all-liquid state and 1 for the all-gas state. The unfrozen water fraction is used to calculate  $S_l$  and  $S_i$  using

$$S_l = S_{uf}(1 - S_g) \quad (3-54)$$

$$S_i = 1 - S_g - S_l \quad (3-55)$$

Most porous media are sufficiently coarse that  $S_{uf}$  changes from 1 to nearly 0 over a very small temperature range, which requires small time steps to resolve with the Newton-Raphson approach. In order to improve convergence in the Newton-Raphson iterations near the freezing point,  $S_{uf}$  is linearly interpolated over the range of  $-3$  to  $0$  °C.

The Loch (1977) model for the capillary pressure between ice and liquid

$$P_{cil} = -P_\Omega - \rho_i h_{iw}^0 \ln \left( \frac{T}{T_0} \right) \quad (3-56)$$

is approximated by

$$P_{cil} = -P_\Omega - \rho_i h_{iw}^0 \left( \frac{T - T_0}{T_0} \right) \quad (3-57)$$

where

$P_{\Omega}$	–	Osmotic pressure [Pa]
$h_{iw}^0$	–	Heat of fusion of water at $T_0$ [J/kg]
$\rho_i$	–	Ice density [kg/m <sup>3</sup> ]

For this fresh water calculation, internally  $P_{\Omega}$  is set to zero and  $\rho_i h_{iw}^0$  is set to  $3.348 \times 10^8$  J/m<sup>3</sup>.

The variably frozen fresh water package uses both the  $S_*$  retention relationship and its inverse  $S_*^{-1}$ . The active fracture model (flagged with AF) does not provide the inverse relationship, thus cannot be used with the variably frozen fresh water package. The van Genuchten/Mualem model (flagged with VG) provides an inverse relationship. As discussed in Section 3.2.2.3, however, the inverse relationship does not include the undocumented smoothing adjustment included in the retention relationship. This may lead to a slightly inconsistent formulation.

### 3.4.2.3 Thermal Conductivity

Thermal conductivity is described using an extension of the thermal hydrology model for unfrozen conditions, extended to fully frozen conditions

$$\kappa_e = \kappa_{\text{dry}} + \text{Ke}_u (\kappa_{\text{sat},l} - \kappa_{\text{dry}}) \quad \text{unfrozen} \quad (3-58)$$

$$\kappa_e = \kappa_{\text{dry}} + \text{Ke}_f (\kappa_{\text{sat},i} - \kappa_{\text{dry}}) \quad \text{frozen} \quad (3-59)$$

$$\text{Ke}_u = (S_l + \varepsilon)^{\alpha_u} \quad (3-60)$$

$$\text{Ke}_f = (S_i + \varepsilon)^{\alpha_f} \quad (3-61)$$

where

$\kappa_e$	–	Bulk thermal conductivity [J/m·s·K]
$\text{Ke}_u$	–	Kersten number for unfrozen conditions [–]
$\text{Ke}_f$	–	Kersten number for frozen conditions [–]
$\kappa_{\text{dry}}$	–	Thermal conductivity at $S_g = 1$ [J/m·s·K]
$\kappa_{\text{sat},l}$	–	Thermal conductivity at $S_l = 1$ [J/m·s·K]
$\kappa_{\text{sat},i}$	–	Thermal conductivity at $S_i = 1$ [J/m·s·K]
$\alpha_u$	–	Fitting parameter for unfrozen conditions ( $\alpha_u > 0$ ) [–]
$\alpha_f$	–	Fitting parameter for frozen conditions ( $\alpha_f > 0$ ) [–]
$\varepsilon$	–	Regularization parameter to prevent unbounded derivatives ( $\varepsilon \ll 1$ ) [–]

For variably frozen conditions, trilinear interpolation in the Kersten numbers is used

$$\kappa_e = \text{Ke}_f \kappa_{\text{sat},i} + \text{Ke}_u \kappa_{\text{sat},l} + (1 - \text{Ke}_f - \text{Ke}_u) \kappa_{\text{dry}} \quad (3-62)$$

Two approaches are available for calculating endpoint thermal conductivities, using an approach accounting for grain conductivity and porosity or based on input empirical parameters.

One approach uses  $\kappa_{\text{dry}}$  and  $\kappa_{\text{sat,l}}$  as input parameters. The other approach uses the mineral grain thermal conductivity as input and builds endpoint thermal conductivities using internal representations for ice, water, and air thermal conductivities.

Both rely on the gas-free endpoint formula

$$\kappa_{\text{sat,l}} = \kappa_{\text{mg}}^{1-\phi} \kappa_{\text{w}}^{\phi} \quad (3-63)$$

$$\kappa_{\text{sat,i}} = \kappa_{\text{mg}}^{1-\phi} \kappa_{\text{i}}^{\phi} \quad (3-64)$$

$$\kappa_{\text{i}} = \kappa_{\text{i}}^0 \left( \frac{T_0}{T} \right)^{p_i} \quad (3-65)$$

leading to

$$\kappa_{\text{sat,i}} = \kappa_{\text{sat,l}} \left( \frac{\kappa_{\text{i}}}{\kappa_{\text{w}}} \right)^{\phi} \quad (3-66)$$

If  $\kappa_{\text{mg}}$  is provided as an input parameter

$$\kappa_{\text{sat,l}} = \kappa_{\text{mg}}^{1-\phi} \kappa_{\text{w}}^{\phi} \quad (3-67)$$

$$\kappa_{\text{sat,i}} = \kappa_{\text{mg}}^{1-\phi} \kappa_{\text{i}}^{\phi} \quad (3-68)$$

$$\kappa_{\text{dry}} = \frac{d(1-\phi)\kappa_{\text{mg}} + \phi\kappa_{\text{a}}}{d(1-\phi) + \phi} \quad (3-69)$$

where

$\kappa_{\text{mg}}$	–	Mineral grain thermal conductivity [J/m·s·K]
$\kappa_{\text{w}}$	–	Water thermal conductivity [J/m·s·K]
$\kappa_{\text{a}}$	–	Air thermal conductivity [J/m·s·K]
$\kappa_{\text{i}}$	–	Ice thermal conductivity [J/m·s·K]
$d$	–	Fitting parameter (internally set to 0.053) [–]
$\kappa_{\text{i}}^0$	–	Reference ice thermal conductivity (internally set to 2.2335 J/m·s·K) [J/m·s·K]
$p_i$	–	Fitting parameter (internally set to 1.0552) [–]

The values for  $\kappa_{\text{w}}$  and  $\kappa_{\text{a}}$  are internally set to 0.5611 and 0.024 J/m·s·K, respectively.

The user input provides four parameters for each grid cell:  $\alpha_u$ ,  $\alpha_f$ ,  $\kappa_1$ , and  $\kappa_2$ . If  $\kappa_2 > 0$ , then  $\kappa_1$  and  $\kappa_2$  represent  $\kappa_{\text{dry}}$  and  $\kappa_{\text{sat,l}}$ , respectively, otherwise  $\kappa_{\text{mg}}$  is set to the absolute value of  $\kappa_2$  and  $\kappa_1$  is disregarded.

## 4 GENERIC INPUT

The input for xFlo includes some generic parameters that are independent of the physics model and some parameters that are model specific. This section describes the generic input defining the gridding, solver, and control parameters, which is minimally dependent on the particular physics packages. Section 5 describes input that is specific to the physics packages, including physics parameters, state variables, initial conditions, and boundary conditions.

The generic input format is essentially identical to the format defined for xFlo Version 1.0 $\beta$  (Painter, 2006).

Modeling work for DECOVALEX-2015 did not use the structured grid or dual continuum input, and the input associated with the GRID and DCM keywords has not been retested since their original installation. The modeling work for DECOVALEX-2015 did use the SOLVER and CONTROL keywords.

### 4.1 Input Data Files for Structured and Unstructured Grids

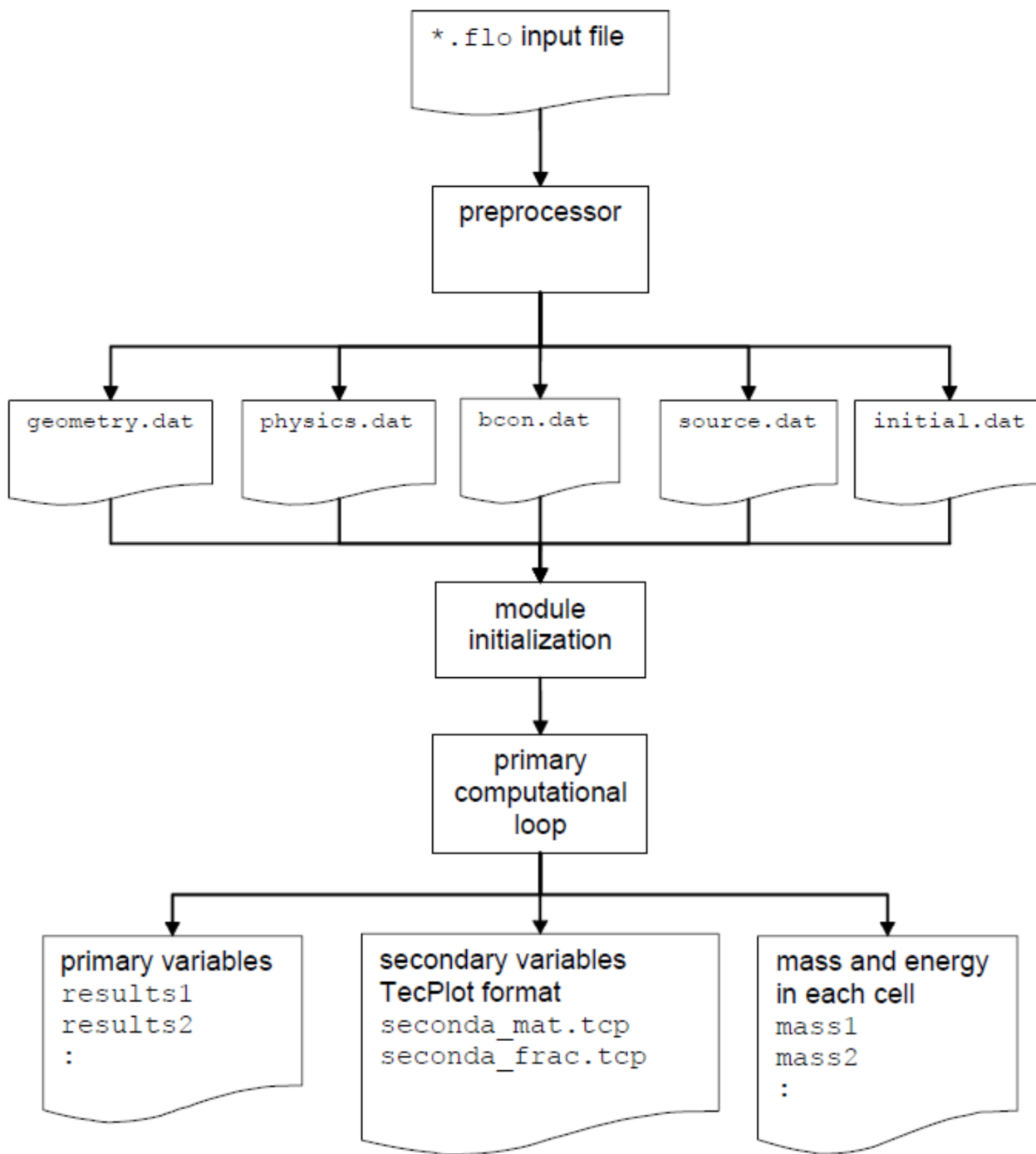
The internal data storage in xFlo is based on an unstructured grid with arbitrary intercell connectivity. Thus, the grid is defined by specifying a connectivity list for each node, a volume for each node, and surface area and inter-cell distance for each connection between nodes. This unstructured grid 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 for all simulations. An optional preprocessing capability is provided to generate the unstructured input files corresponding to structured Cartesian grids. The preprocessor reads data from a \*.flo input file and creates the unstructured data files, overwriting any existing files. The preprocessor can be used to translate a structured-grid description of the grid, rock properties, initial conditions, boundary conditions, and source fluxes. The data flow during initialization is shown in Figure 4-1.

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.

### 4.2 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.



**Figure 4-1. Data Flow for the xFlo System. For Unstructured Grids, the Preprocessor Step Is Skipped and Data Are Read Directly from the \*.dat Files.**

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.

### 4.3 Specifying the Grid

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 inter-cell connection is given in the first section, followed by data for each grid cell.

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 <sup>2</sup> ]
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 <sup>3</sup> ]
Z	z-coordinate for the cell [m]
Y	y-coordinate for the cell [m]
X	x-coordinate for the cell [m]

The Z coordinate is assumed to be vertically up, and is used to calculate gravity-dependent fluxes. The X and Y coordinates are stored but are not used for calculations.

A structured grid can be generated using the GRID keyword block in the \*.flo file. The preprocessor overwrites the geometry.dat file during execution.

The format for the GRID keyword block is

```

GRID GRIDTYPE SPACING

GRIDPARAMS (repeat as needed)

END GRID

```

Here GRIDTYPE is the type of the grid (XYZ, indicating Cartesian grids, is the only allowed value in Version 1.1 $\beta$ ) and SPACING (VARIABLE or CONSTANT) specifies where the grid is uniform (constant spacing) or nonuniform (variable spacing). GRIDPARAMS refer to parameter names in the namelist-type input, as given in Table 4-1. 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 4-2.

Table 4-1. Parameters in the GRID Keyword Block			
Parameter Name	Type	Units	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	m	Minimum x-coordinate
XMAX	real scalar	m	Maximum x-coordinate
YMIN	real scalar	m	Minimum y-coordinate
YMAX	real scalar	m	Maximum y-coordinate
ZMIN	real scalar	m	Minimum z-coordinate
ZMAX	real scalar	m	Maximum z-coordinate
DX	real array of size NX	m	Cell widths in x-direction
DY	real array of size NY	m	Cell widths in y-direction
DZ	real array of size NZ	m	Cell widths in z-direction

```

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

```

**Figure 4-2. Example of the GRID Keyword Block**

## 4.4 Control of the Linear Solver

Version 1.0 $\beta$  of xFlo used an adaptation of the WATSOLV solver (van der Kwaak et al., 1995), including a preconditioned conjugate gradient solver with two accelerator options provided: biconjugate gradient stabilized and generalized minimum residual, with preconditioning based on an incomplete lower-upper decomposition with fixed level of infill. The generalized minimum

residual option was removed from the xFlo code, but Version 1.1 $\beta$  still recognizes the corresponding keywords.

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 4-2. An example of the SOLVER keyword block is shown in Figure 4-3.

<b>Table 4-2. Parameters in the SOLVER Keyword Block</b>		
<b>Parameter Name</b>	<b>Type</b>	<b>Description</b>
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 (not provided in Version 1.1 $\beta$ )		
†CGSTAB—bi-conjugate gradient stabilized		

```

SOLVER
  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 4-3. Example of the SOLVER Keyword Block**

## 4.5 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 4-3.

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 DTMAX array defines the maximum allowed time-step size for the period following each output.

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 4-4.

Table 4-3. Parameters in the CONTROL Keyword Block			
Parameter Name	Type	Units	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	real array*	–	Absolute convergence tolerances for residuals†
NREPORTS	integer scalar	–	Number of times at which output is desired
RPTIMES	real array of length NREPORTS	yr	List of times at which output is desired
DTMAX	real array of length NREPORTS	yr	Maximum allowed time step for various time periods
*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.			
†Units must match the primary variables.			

```

CONTROL
  itermax = 20
  delmax = 0.7 0.3 0.02
  target = 0.05 0.05 0.005
  vtol = 1.0e-5 1.0e-6 1.0e-6
  rrtol = 1.0e-4 1.0e-4 1.0e-4
  nreports = 12
  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

```

**Figure 4-4. Example of the CONTROL Keyword Block**

## 4.6 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.

There is no distinction between single- and dual-continuum simulations in the internal xFlo data structures. The choice of single- or dual-continuum simulations is handled purely by grid definition, including the IFM code indicating whether connections are fracture to matrix. Thus, for dual-continuum simulations involving an unstructured primary grid, the user must specify the dual-continuum grid in the `geometry.dat` file.

In discussing multiple-continua models, the primary grid refers to the grid that results from the geometrical partitioning of the physical domain into grid cells. The preprocessor routines generate the grid for a dual-continuum model by first cloning the primary grid so that there are two grid cells at every physical location, then updating the geometric parameters to account for differences between the matrix and fracture cells. Volumes associated with fracture cells are calculated as the volume of the physical grid cell times the fracture volume fraction (bulk fracture porosity), and volumes associated with matrix cells are assigned the remaining physical cell volume. 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.

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 <sup>2</sup> ]
AREAFM	Fracture/matrix specific surface area [m <sup>2</sup> /m <sup>3</sup> ]

An example of the DCM keyword block is shown in Figure 4-5.



```
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 4-5. Example of the DCM Keyword Block**

## 5 PHYSICS PACKAGES

The physics package for a given model is selected with a keyword, which may be placed anywhere in the \*.flo file. A single physics package is used for the entire model domain.

Both physics packages are based on the same generic solution approach and balance equations. The particular packages specialize the balance equations to represent the particular physics of interest.

The input for xFlo includes some generic parameters that are independent of the physics model and some parameters that are model-specific. Section 4 described the generic input defining the gridding, solver, and control parameters, which is minimally dependent on the particular physics packages. Section 5 describes input that is specific to the physics packages, including physics parameters, initial conditions, boundary conditions, and sources.

The physics packages assume that the grid is unstructured and use input defined in a set of files including the physics.dat, initial.dat, bcon.dat, and source.dat files. An internal preprocessor using keyword blocks in the \*.flo file creates the corresponding \*.dat files for structured grids, overwriting any existing \*.dat file each time xFlo is executed.

Modeling work for DECOVALEX-2015 was limited to single continuum thermal hydrology modeling using the van Genuchten/Mualem retention approach. The variably frozen fresh water package and active fracture model were not used.

### 5.1 Assignment of Rock Properties to Grid Cells

Rock properties are assigned to individual grid cells in the physics.dat file. The file structure is similar for each physics module. The first section provides preliminary information specific to a physics package. The second section provides a list of data to define retention and relative permeability relationships, which are defined for rock types rather than by grid cell. The last section provides package-dependent parameters for each grid cell, including a pointer to the rock type for the grid cell.

#### 5.1.1 File Format for physics.dat

The format for the physics.dat file is

```
! first sub block (file information and package-specific options)
PACKAGESPEC (repeat as needed)

! second sub block (capillary pressure and relative permeability models)
PCKR NMODELS
TYPE PARAMETERS (repeat NMODELS times)

! third sub-block (package-specific parameters)
ICELL PROPERTIES IMODEL (repeat until all blocks are defined)
```

In the first sub-block, PACKAGESPEC represents one or more lines of package-specific information. All physics modules expect that the first line is a character string defining the file system path to the directory containing the data files for thermophysical properties of water



(see Chapter 2), but the variably frozen package does not use this information. Any remaining package-specific options can be omitted or specified in a package-specific order.

The parameters appearing in the second (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 numerical parameters defining the capillary pressure/relative permeability model

The allowed values for the PCKR TYPE are “VG” for the van Genuchten/Mualem model, “AF” for the active fracture model, and “TAB” for a tabulated set of relationships.

For each PCKR model, the numerical parameters are specified in a model-dependent order (i.e., not in the namelist format).

If VG is specified, the parameters from Sections 3.2.2.1 and 3.2.2.3 are provided as follows:

ALPHA	van Genuchten alpha ( $\alpha_{VG}$ ) [ $\text{Pa}^{-1}$ ]
VGM	van Genuchten m
SRESIDUAL	Residual liquid saturation ( $S_r$ )
SMIN	Transition effective liquid saturation for linearization ( $S_{\min}$ )
SSTAR	Transition effective liquid saturation for smoothing ( $S^*$ )
AKR	Parameter for smoothing relative permeability ( $a_{kr}$ ) (not used)
BKR	Parameter for smoothing relative permeability ( $b_{kr}$ ) (not used)
APC	Parameter for smoothing capillary pressure ( $a_{pc}$ ) (not used)
BPC	Parameter for smoothing capillary pressure ( $b_{pc}$ ) (not used)

If AF is specified, the parameters from Section 3.2.2.2 are provided as follows:

ALPHA	van Genuchten alpha ( $\alpha_{VG}$ ) [ $\text{Pa}^{-1}$ ]
LAMBDA	van Genuchten m
SRESIDUAL	Residual liquid saturation ( $S_r$ )
SMIN	Transition effective liquid saturation for linearization ( $S_{\min}$ )
GAMMA	Active fracture parameter ( $\gamma$ )

If TAB is specified, one-dimensional interpolation is used to determine three relationships:  $P_c(S_{l,\text{eff}})$ ,  $k_{ri}(S_{l,\text{eff}})$ , and  $S_{l,\text{eff}}(P_c)$ . Each relationship is provided in a separate file. Capillary pressure is specified in units of Pa. The file names are provided as follows:

FILE1	Relationship for $P_c(S_{l,\text{eff}})$
FILE2	Relationship for $k_{ri}(S_{l,\text{eff}})$
FILE3	Relationship for $S_{l,\text{eff}}(P_c)$

The third sub-block assigns package-specific properties (e.g., porosity and permeability) to each grid cell and includes a pointer to the PCKR model used for the cell. The parameters appearing in the third (PROPERTIES) sub-block are as follows:

ICELL	Grid cell number
PROPERTIES	Package-specific parameters for the grid cell
IMODEL	Index for the capillary pressure/relative permeability model

The number and type of parameters appearing on each line of the PROPERTIES keyword block depend on the particular physics model.

### 5.1.2 File Format for Structured Grids in \*.flo

When structured grids are used, the preprocessor expects a similar structure in the \*.flo file with additional information. The preprocessor creates a `physics.dat` file from definitions in the PHYSICS keyword block for the \*.flo file. The PHYSICS keyword block has three sub-blocks, following the structure of the `physics.dat` file. The preprocessor copies the first two sub-blocks verbatim to the `physics.dat` file. The third sub-block, signaled by the PROPERTIES keyword, generates definitions for each grid cell by applying properties to all grid cells within rectilinear sections of the domain.

If the DCM option is used, the assignment in the `physics.dat` file is to the DCM grid (separate grid cells for fractures and matrix), not the primary grid (one grid cell partitioned into fractures and matrix). 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.

The format for the PHYSICS keyword block is

```

PHYSICS
! first sub-block
PACKAGESPEC (repeat as needed)

! second sub-block
PCKR NMODELS
TYPE PARAMETERS (repeat NMODELS times)

! third sub-block
PROPERTIES
I1 I2 J1 J2 K1 K2 PROPERTIES IMODEL (repeat until all blocks are defined)
END PHYSICS

```

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. Each subregion specification overwrites any previous assignment to a grid cell.

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

### 5.1.3 Thermal Hydrology Package Options and Properties

The thermal hydrology package accepts a series of inputs for the first sub-block. All of the inputs are optional except the DATADIR parameter. Each of the optional inputs can be commented out by adding a '!' to the keyword (e.g., 'VPL' becomes '!VPL'). The thermal hydrology package expects that inputs for the first sub-block are in the following sequence:

DATADIR	File system path to the directory containing the data files for thermophysical properties of water
VPL	Enable vapor pressure lowering (default off)
RHSURTEN	Enable surface tension scaling is enabled in relative humidity relationships (default off)
KLINK	The first value on the same line is the Klinkenberg $b$ parameter [Pa], assigned to all grid cells
VKLINK	Enable cell-by-cell assignment of the Klinkenberg $b$ parameter [Pa] in the PROPERTIES block
GASTORT	The first value on the same line is a tortuosity parameter $\tau$ for the gas phase [-], assigned to all grid cells
VGASTORT	Enable cell-by-cell assignment of a tortuosity parameter $\tau$ for the gas phase [-] in the PROPERTIES block

The VKLINK and VGASTORT options are not available when using the structured grid preprocessor.

The third (PROPERTIES) sub-block reads parameters for each grid cell in the domain and in ghost (boundary) cells. The grid cells can be supplied in arbitrary order, but each cell must be specified exactly once. The parameters have the following sequence on each line:

ICELL	Grid cell number
PERM	Permeability [m <sup>2</sup> ]
POR	Porosity [-]
KDRY	Thermal conductivity of dry rock [W/m-K]
KWET	Thermal conductivity of fully saturated rock [W/m-K]
TCAL	Exponent on saturation scaling for thermal conductivity [-]
CPROCK	Specific heat of the rock [J/kg-K]
RHOROCK	Rock grain density [kg/m <sup>3</sup> ]
DIFFLIQ	Diffusion coefficient of air in the liquid phase [m <sup>2</sup> /s]
IMODEL	Model from the list defined in the PCKR sub-block.

If the VKLINK option is enabled, another sequence of parameters is read for each grid cell with the following sequence on each line:

ICELL	Grid cell number
KLINK	The Klinkenberg $b$ parameter [Pa]

If the VGASTORT option is enabled, another sequence of parameters is read for each grid cell with the following sequence on each line:

ICELL	Grid cell number
GASTORT	Tortuosity $\tau$ for the gas phase [–]

### 5.1.4 Variably Frozen Fresh Water Package Options and Properties

The variably frozen fresh water package accepts one or two inputs for the first sub-block. The second input is optional. The variably frozen package expects that inputs for the first sub-block are in the following sequence:

DATADIR	File system path to the directory containing the data files for thermophysical properties of water (not used)
VPL	Enable vapor pressure lowering (default off)

The third (PROPERTIES) sub-block reads parameters for each grid cell in the domain and in ghost (boundary) cells. The grid cells can be supplied in arbitrary order, but each cell must be specified exactly once. The parameters have the following sequence on each line:

ICELL	Grid cell number
PERM	Permeability [m <sup>2</sup> ]
POR	Porosity [–]
KDRY	Thermal conductivity of dry rock [W/m-K]
K2	If positive, K2 is thermal conductivity of fully liquid saturated rock [W/m-K]; If negative, abs(K2) is thermal conductivity of mineral grains [W/m-K]
TCAL	Kersten number exponent for liquid thermal conductivity [–]
TCAL	Kersten number exponent for frozen thermal conductivity [–]
CPROCK	Specific heat of the rock [J/kg-K]
RHOROCK	Rock grain density [kg/m <sup>3</sup> ]
DIFFLIQ	Diffusion coefficient of air in the liquid phase [m <sup>2</sup> /s]
IMODEL	Model from the list defined in the PCKR sub-block.

## 5.2 Specifying Initial Conditions

Initial conditions (initial values for state variables) are read from the `initial.dat` file, with one line of input per grid cell. The number of lines must match the number of domain cells plus the number of boundary condition cells. The generic format is as follows:

ICELL PHASE SV1 SV2 SV3 ...

If the INITIAL keyword is included in the \*.flo file, initial conditions are specified in the INITIAL keyword block of the \*.flo file. The preprocessor overwrites the initial.dat file at the start of the run.

The format for the INITIAL keyword block is as follows.

```
INITIAL
I1 I2 J1 J2 K1 K2 PHASE SV1 SV2 SV3 ... (repeat until all blocks are defined)
END INITIAL
```

If the DCM option is active, the initial.dat file populates each cell in the DCM grid, not the primary grid. In the INITIAL keyword block, two sets of PHASE and SV values are provided. The first set is for the fractures and the second for the matrix.

### 5.2.1 Thermal Hydrology Initial Conditions

The data read on each line are as follows.

PHASE	Integer flag that determines the phase state 0 implies two-phase 1 implies no gas -1 implies all gas
SV1	First primary variable, which is dependent on phase state For two-phase conditions, SV1 is gas saturation [-] For all-gas conditions, SV1 is air partial pressure [MPa] For no-gas conditions, SV1 is mole fraction of air [-]
SV2	Phase pressure [MPa] For an all-gas and two-phase conditions, P is gas pressure For no-gas conditions, P is liquid pressure
SV3	Temperature [K]

Initial conditions specified in the INITIAL keyword block of the \*.flo file undergo unit transformations before being transferred to the initial.dat file. In the INITIAL keyword block, SV2 and SV3 are provided in units of Pa and °C, respectively.

### 5.2.2 Variably Frozen Fresh Water Initial Conditions

The initial.dat file and INITIAL keyword block of the \*.flo file use exactly the same protocols as the thermal hydrology package.

## 5.3 Specifying Boundary Conditions

Boundary conditions are read from the bcon.dat file. For a structured grid, the preprocessor creates the bcon.dat file using the specifications in the BCON keyword block in the \*.flo file.

Version 1.1β of the thermal hydrology model supports two types of boundary conditions: Dirichlet and infiltration (a Cauchy or mixed condition). A second-type (flux) boundary condition is not specifically offered, but it can be defined using internal sources or by setting appropriate

ghost cell parameters to zero. If the Dirichlet 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 flux rates and the state variables describing the boundary fluid. Note that infiltration boundaries are, by definition, two-phase.

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 VAL1 VAL2 VAL3 RATE1 RATE2 RATE3
  : ! 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 (ghost) cells. The external cells are assigned values for the primary variables based on the time history provided in the `bcon.dat` file, but are never updated based on the solution in the internal cells. 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.

### 5.3.1 Thermal Hydrology Boundary Conditions

The data read in the `bcon.dat` file 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
BCTYPE	Boundary condition type (INFILTRATION or DIRICHLET)
NTIMES	Number of times used to specify the boundary time history
PHASE	For INFILTRATION type, must be set to 0 For DIRICHLET type, same as in the INITIAL keyword
VAL1	Same as in the INITIAL keyword
VAL2	Pressure [MPa]
VAL3	Temperature [K]
RATE1	For DIRICHLET type, not used For INFILTRATION type, flux of dissolved air [mol/m <sup>2</sup> -yr]

RATE2	For DIRICHLET type, not used For INFILTRATION type, air plus water flux [mol/m <sup>2</sup> -yr]
RATE3	For DIRICHLET type, not used For INFILTRATION, energy flux into the system [MJ/m <sup>2</sup> -yr]

Note that the user is responsible for correctly setting the incoming fluxes for unstructured grids and INFILTRATION boundary conditions. The RATE1 is the flux of air dissolved in the infiltrating water, the RATE2 is the total mass flux into the system including the infiltrating water and the dissolved air, and RATE3 is the enthalpy of the infiltrating water.

The DIRICHLET boundary condition sets the ghost cell primary values to the specified values. Fluxes are calculated normally, allowing the domain cells to respond, but the ghost cells are only updated based on the specified time history.

The INFILTRATION boundary condition translates the RATE1, RATE2, and RATE3 fluxes from boundary fluxes into source terms added to the interior domain cell adjacent to the boundary. The translation internally multiplies each RATE by the interfacial area divided by the volume of the interior cell. Again the ghost cells are only updated based on the specified time history. The INFILTRATION boundary condition also sets the ghost cell primary values to the specified values, allowing fluxes to be calculated normally. The additional flux created by setting VAL1, VAL2, and VAL3 can be turned off by setting to zero the parameter properties affecting mass and energy transport in the ghost cell, such as PERM, KDRY, KWET, and DIFFLIQ. The same effect could be achieved using the SOURCE condition, with areal fluxes manually converted into the corresponding volumetric flux.

The format for the BCON keyword block in the \*.flo file requires that the domain is a rectilinear block with six faces described with a structured grid. Boundary conditions are only specified on the faces. The format is as follows.

```
BCON
FACE BCTYPE M1 M2 N1 N2
TIME PHASE VAL1 VAL2 VAL3 ...
:      ! 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 condition applies to the cells with i=1, M1 ≤ j ≤ M2, N1 ≤ k ≤ N2.

The BCTYPE parameter determines the type of boundary condition and must be either "DIRICHLET" 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 encountered. 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 DIRICHLET type, same as in the INITIAL keyword
VAL1	For INFILTRATION type, the infiltration rate For DIRICHLET type, same as in the INITIAL keyword
VAL2	Pressure [Pa]
VAL3	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.

### 5.3.2 Variably Frozen Fresh Water Boundary Conditions

The `bcon.dat` file and BCON keyword block of the `*.flo` file use exactly the same protocols as the thermal hydrology package.

## 5.4 Specifying Mass and Heat Sources

Internal sources are read from the `source.dat` file. For a structured grid on a rectilinear domain, the preprocessor can create the `source.dat` file using the specifications in the SOURCE keyword block in the `*.flo` file.

Source rates are read from the `source.dat` file, and each row of the data in the `source.dat` file is assigned directly to individual boundary cells. If the DCM option is specified, the source rates 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 RATE1 RATE2 RATE3
  : ! 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 1.1β
NTIMES	Number of times used to specify the source time history



### 5.4.1 Thermal Hydrology Sources

For each of the tabulated times in the source history, the data read for each of the source cells are as follows:

TIME	Time [yr]
RATE1	Source rate for air [mol/yr]
RATE2	Source rate for total mass (air plus water) [mol/yr]
RATE3	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 internally consistent. In particular, if the mass injection options are used, then the associated enthalpy of the incoming fluid must be consistently specified in the RATE3 field.

The area-integrated flux crossing an exterior boundary into a grid cell adjacent to the boundary is mathematically indistinguishable from a volumetric source in the grid cell. Accordingly, sources in cells can be used to represent area-integrated boundary fluxes.

Version 1.1 $\beta$  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.

The format of the SOURCE keyword block in the `*.flo` file, which requires that the domain is rectilinear and the grid is structured, 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
```

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.

Source data in the `*.flo` file are specified by subregions. The integers I1, I2, J1, J2, K1, and K2 define indices in the structured grid defining the boundaries of 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  $\text{W/m}^3$  or  $\text{W}$ , depending on whether volumetric or total basis was chosen.

If STYPE is set to "AIR," then the data are air source rate [ $\text{kg/m}^3\text{-yr}$  or  $\text{kg/yr}$ ], air pressure [Pa], and air temperature [ $^{\circ}\text{C}$ ].

If STYPE is set to "WATER," then the required data are water source rate [ $\text{kg/m}^3\text{-yr}$  or  $\text{kg/yr}$ ], pressure of the incoming water [Pa], temperature of the incoming water [ $^{\circ}\text{C}$ ], and mole fraction of air in the water [–].

#### **5.4.2 Variably Frozen Fresh Water Sources**

The `source.dat` file and SOURCE keyword block of the `*.flo` file use exactly the same protocols as the thermal hydrology package.

### **5.5 Package Output**

As described in Section 4.5, xFlo produces output files at user-specified report times. Version 1.1 $\beta$  produces several types of output files: primary variable files, secondary variable files, mass files, and flux files. All output files are created at each report time. Each output file reports results for one grid cell per output file line.

The output files share a naming convention, with a root name that has a report number appended. The report number starts at one and increments by one each report time. For example, primary variable files have the root name `results` and file containing the primary variables at the first reporting time would have the name `results1`.

Almost all output files contain a two-line header providing a time stamp. Without this header, the contents of the primary variable files are in the same format as the `initial.dat` file. The thermal hydrology package in Version 1.1 $\beta$  omits the header for the primary variable files, allowing a primary variable file to be used as an `initial.dat` file in a restart run. This change was made to allow output from one phase of an extended DECOVALEX-2015 simulation to be used as input to the next phase without manual editing of the file.

Each line of a flux output file reports fluxes from all neighboring cells (sign convention positive inward). The fluxes are ordered by the unstructured grid number of the neighboring grid cells. Each line may have a different number of reported fluxes, because each cell may have a different number of neighbors.

#### **5.5.1 Thermal Hydrology Package Output**

A secondary variable file contains the following secondary variables in order: liquid saturation, gas pressure [Pa], liquid pressure [Pa], temperature [K], molar fraction of air in liquid [–], molar fraction of air in gas [–], molar fraction of water in liquid [–], molar fraction of water in gas [–], relative humidity [–], porosity [–], bulk thermal conductivity [ $\text{MJ/yr-m-K}$ ], and a variable called HYCOND [ $\text{mol-yr/kg}$ ].

The HYCOND variable, a measure of effective permeability, is defined as

$$\text{HYCOND} = \frac{\eta_l k_{rl} k}{\mu_l}$$

A mass file contains the total masses for each component and the energy content in each cell. Each line contains total air mass [mol/m<sup>3</sup>], total water mass [mol/m<sup>3</sup>], and energy content [MJ/m<sup>3</sup>] for a cell.

Six different flux files are output each report time. The fluxes associated with the root names are as follows:

AFLUX	Air flux [mol/m <sup>2</sup> -yr]
WFLUX	Water flux [mol/m <sup>2</sup> -yr]
HFLUX	Energy flux [MJ/m <sup>2</sup> -yr]
TOTAFLUX	Areal-integrated air flux [mol/yr]
TOTWFLUX	Areal-integrated water flux [mol/yr]
TOTHFLUX	Areal-integrated energy flux [MJ/yr]

## 6 REFERENCES

- CRC. *Handbook of Chemistry and Physics*. 67<sup>th</sup> Edition. R.C. Weast and D.R. Lide, eds. Boca Raton, Florida: CRC Press, Inc. 1986.
- Klinkenberg, L.J. "The Permeability of Porous Media to Liquids and Gases." *Drilling and Production Practice*. New York, New York: American Petroleum Institute. pp. 200–213. 1941.
- 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.
- Loch, J.P.G. "Thermodynamic Equilibrium Between Ice and Water in Porous Media." *Soil Science*. Vol. 126. pp. 77–80. 1977.
- Manepally C., S. Stothoff, G. Ofoegbu, B. Dasgupta, and R. Fedors. "Modeling and Analyses of the Thermohydrological-Mechanical Behavior of Bentonite Buffer and Clay Host Rock in the HE-E Test." San Antonio, Texas: Center for Nuclear Waste Regulatory Analyses. 2015 [in review].
- 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. "Three-Phase Numerical Model of Water Migration in Partially Frozen Geological Media: Model Formulation, Validation, and Applications." *Computational Geoscience*. Vol. 15 Issue 1. pp. 69–85. DOI 10.1007/s10596-010-9197-z. 2011.
- Painter, S. "xFLO Version 1.0 $\beta$  User's Manual." San Antonio, Texas: Center for Nuclear Waste Regulatory Analyses. 2006.
- Painter, S. and M. Seth. "MULTIFLO User's Manual: Two-Phase Nonisothermal Coupled Thermal-Hydrologic-Chemical Flow Simulator." Version 2.0. San Antonio, Texas: Center for Nuclear Waste Regulatory Analyses. 2003.
- Pruess, K., C. Oldenburg, and G. Mordis. "TOUGH2 User's Guide, Version 2." LBNL–43134. Berkeley, California: Lawrence Berkeley National Laboratory. 2012.
- Somerton, W.H, J.A. Keese, and S.C. Chu. "Thermal Behaviour of Unconsolidated Oil Sands." *Society of Petroleum Engineers Journal*. Vol. 14. pp. 513–521. 1974.

Stothoff, S., C. Manepally, G. Ofoegbu, B. Dasgupta, and R. Fedors. "Modeling Thermohydrological-Mechanical Behavior of Granular Bentonite in a Laboratory Column Test." ML15234A128. San Antonio, Texas: Center for Nuclear Waste Regulatory Analyses. 2015.

Sutherland, W. "LII. The Viscosity of Gases and Molecular Force." *Philosophical Magazine Series 5*. 36:223. pp. 507–531. DOI: 10.1080/14786449308620508. 1893.

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.