PFLOTRAN User Manual

A Massively Parallel Reactive Flow and Transport Model for Describing Surface and Subsurface Processes

Peter C. Lichtner$^\alpha$, Glenn E. Hammond$^\beta$, Chuan Lu$^\sigma$,
Satish Karra$^\delta$, Gautam Bisht$^{\gamma_1}$, Benjamin Andre$^{\gamma_2}$,
Richard Mills$^{\epsilon_1}$, Jitu Kumar$^{\epsilon_2}$

$^\alpha$ OFM Research  peter.lichtner@gmail.com
$^\beta$ SNL  gehammo@sandia.gov
$^\sigma$ INL  chuan.lu@inl.gov
$^\delta$ LANL  satkarra@lanl.gov
$^{\gamma_1}$ LBNL  gbrisht@lbl.gov
$^{\gamma_2}$ LBNL  bandre@lbl.gov
$^{\epsilon_1}$ Intel  rtm@eecs.utk.edu
$^{\epsilon_2}$ ORNL  jitu1503@gmail.com

January 20, 2015
# TABLE OF CONTENTS

1 Introduction .................................................. 9

2 Quick Start ..................................................... 10
   2.1 Installing PFLOTRAN ........................................... 10
   2.2 Running on Big Iron Parallel Architectures ..................... 13

3 Installation ..................................................... 14
   3.1 Compilers ..................................................... 14
   3.2 Building PETSc ................................................. 14
      3.2.1 Mac OS X ..................................................... 15
         3.2.1.1 Lion, Mavericks (≤ Mac OS X 10.9.x) ............... 15
         3.2.1.2 Yosemite .................................................. 15
      3.2.2 Windows .................................................... 16
      3.2.3 ORNL’s Jaguar XT4/5 ...................................... 16
   3.3 Building PFLOTRAN ............................................ 17
   3.4 Updating PFLOTRAN ........................................... 17
   3.5 Parallel I/O using Sarat Sreepathi’s SCORPIO library with PFLOTRAN ........................................... 18
   3.6 Running PFLOTRAN .............................................. 21
   3.7 Multiple Realization Simulation Mode ........................... 22
   3.8 Multiple Simulation Mode ....................................... 22

4 Creating the Input File: PFLOTRAN Keywords ....................... 23
   4.1 Conventions and Notation ....................................... 24
   4.2 Example Input File ............................................. 26
   4.3 Keyword: BOUNDARY_CONDITION ............................... 31
   4.4 Keyword: BRINE ................................................ 33
   4.5 Keyword: CHARACTERISTIC_CURVES ......................... 34
   4.6 Keyword: CHECKPOINT ......................................... 37
4.7   Keyword: CHEMISTRY .............................................. 38
4.8   Keyword: COMPUTE_STATISTICS ................................. 47
4.9   Keyword: CO2_DATABASE ......................................... 48
4.10  Keyword: CONSTRAINT .......................................... 49
4.11  Keyword: DATASET ................................................ 51
4.12  Keyword: DBASE_FILENAME ..................................... 53
4.13  Keyword: DEBUG .................................................. 55
4.14  Keyword: EOS ....................................................... 56
4.15  Keyword: FLOW_CONDITION ..................................... 57
4.16  Keyword: FLUID_PROPERTY ........................................ 61
4.17  Keyword: GEOMECHANICS ........................................ 62
4.18  Keyword: GEOMECHANICS_GRID ............................... 63
4.19  Keyword: GEOMECHANICS_OUTPUT ............................ 64
4.20  Keyword: GEOMECHANICS_MATERIAL_PROPERTY ............. 65
4.21  Keyword: GEOMECHANICS_REGION ............................ 66
4.22  Keyword: GEOMECHANICS_CONDITION ...................... 67
4.23  Keyword: GEOMECHANICS_BOUNDARY_CONDITION .......... 69
4.24  Keyword: GEOMECHANICS_STRATA ........................... 71
4.25  Keyword: GRID Required ........................................ 72
4.26  Keyword: INITIAL_CONDITION .................................. 75
4.27  Keyword: INTEGRAL_FLUX ....................................... 76
4.28  Keyword: LINEAR_SOLVER ...................................... 77
4.29  Keyword: MATERIAL_PROPERTY .................................. 80
4.30  Keyword: MODE ..................................................... 84
4.31  Keyword: MULTIPLE_CONTINUUM ............................... 85
4.32  Keyword: NEWTON_SOLVER ...................................... 86
4.33  Keyword: NONUNIFORM_VELOCITY ............................ 88
4.34  Keyword: NUMERICAL_JACOBIAN_FLOW ....................... 90
4.35 Keyword: NUMERICAL_JACOBIAN_RXN ........................................ 91
4.36 Keyword: NUMERICAL_JACOBIAN_MULTI_COUPLE ...................... 92
4.37 Keyword: OBSERVATION ...................................................... 93
4.38 Keyword: ORIGIN (ORIG) .................................................... 95
4.39 Keyword: OUTPUT ........................................................... 96
4.40 Keyword: OVERWRITE_RESTART_TRANSPORT .......................... 101
4.41 Keyword: PROC .......................................................... 102
4.42 Keyword: REGION ......................................................... 103
4.43 Keyword: RESTART ......................................................... 105
4.44 Keyword: SATURATION_FUNCTION ......................................... 106
4.45 Keyword: SOURCE_SINK .................................................. 108
4.46 Keyword: STRATIGRAPHY (STRATA) ..................................... 109
4.47 Keyword: TIME .......................................................... 110
4.48 Keyword: TIMESTEPPER .................................................... 111
4.49 Keyword: TRANSPORT_CONDITION ...................................... 113
4.50 Keyword: UNIFORM_VELOCITY Optional ................................... 116
4.51 Keyword: USE_TOUCH_OPTIONS ......................................... 117
4.52 Keyword: VELOCITY_DATASET ............................................. 118
4.53 Keyword: WALLCLOCK_STOP .............................................. 119

5 Example Input Files ......................................................... 120
5.1 Richards Equation .......................................................... 120
5.2 Carbon Sequestration Problem with Reaction with Calcite .......... 125

6 PFLOTRAN Regression Test Manager ..................................... 130
6.1 Running the Test Manager ................................................ 130
6.2 Test output ............................................................... 131
6.3 Configuration Files ....................................................... 132
6.4 Creating New Tests ....................................................... 133
6.5 Updating Test Results ................................................... 134
7 Visualization

8 Benchmark Problems

8.1 Ion Exchange ................................................................. 136

8.2 SX-115 Hanford Tank Farm ............................................. 144
  8.2.1 Problem Description ................................................. 144
  8.2.2 Governing Equations ................................................ 144
    8.2.2.1 Semi-Analytical Solution for Steady-State Conditions ... 145
    8.2.2.2 Watertable ....................................................... 145
  8.2.3 Model Parameters ................................................... 145
  8.2.4 Simulation Results .................................................. 146
  8.2.5 PFLOTRAN Input Files ............................................. 149

9 References ................................................................. 159

10 Appendix: Governing Equations ....................................... 161

10.1 Mode: RICHARDS ....................................................... 161

10.2 Capillary Pressure Relations ........................................ 161
  10.2.1 Brooks-Corey Saturation Function .............................. 162
  10.2.2 Relative Permeability ............................................. 162
  10.2.3 Smoothing ......................................................... 163

10.3 Mode: MPHASE .......................................................... 163
  10.3.1 Source/Sink Terms ................................................ 164
  10.3.2 Variable Switching ................................................ 165
    10.3.2.1 Gas: \( p_g, T, x_{CO_2}^g \) \( \rightarrow \) Two-Phase: \( p_g, T, s_g \) .... 165
    10.3.2.2 Liquid: \( p_l, T, x_{CO_2}^l \) \( \rightarrow \) Two-phase: \( p_g, T, s_g \) .... 165
    10.3.2.3 Two-Phase: \( p_g, T, s_g \) \( \rightarrow \) Liquid \( p_l, T, x_{CO_2}^l \) or Gas \( p_g, T, x_{CO_2}^g \) 166

10.4 Mode: IMMIS ............................................................ 167

10.5 Mode: MISCIBLE ....................................................... 167

10.6 Mode: Air-Water ........................................................ 169
10.7 Mode: TH (Thermal-Hydrologic) ............................................. 169
  10.7.1 Ice Model ................................................................. 170
10.8 Thermal Conduction Multiple Continuum Model ............................ 172
10.9 Mode: Reactive Transport (Keyword CHEMISTRY) ...................... 176
  10.9.1 Mineral Precipitation and Dissolution ............................... 177
    10.9.1.1 Rate Limiter ...................................................... 178
    10.9.1.2 Changes in Material Properties ............................... 178
    10.9.1.3 Affinity Threshold .............................................. 180
    10.9.1.4 Surface Armoring .............................................. 180
  10.9.2 Sorption ............................................................... 180
    10.9.2.1 Ion Exchange ..................................................... 180
    10.9.2.2 Surface Complexation ......................................... 182
    10.9.2.3 Multirate Sorption ............................................. 183
  10.9.3 Sorption Isotherm <Under Revision> .................................. 184
  10.9.4 Colloid-Facilitated Transport ....................................... 185
10.10 Tracer Mean Age .................................................................. 187
10.11 Thermodynamic Database .................................................... 188
10.12 Eh, pe ............................................................................. 188
10.13 Geomechanics ................................................................. 189

11 Appendix: Method of Solution .................................................. 190
  11.1 Integrated Finite Volume Discretization ................................... 190
  11.2 Fully Implicit ..................................................................... 191
    11.2.1 Multirate Sorption .................................................... 192
  11.3 Operator Splitting ............................................................. 192
    11.3.1 Constant $K_d$ .......................................................... 193

12 Appendix: Reaction Sandbox .................................................... 194
  12.1 Background ...................................................................... 194
  12.2 Implementation ............................................................... 194
LIST OF FIGURES

1 Breakthrough curves for Ca$^{2+}$, Mg$^{2+}$ and Na$^+$ compared with experimental results from Voegelin et al. (2000). .............................................................. 137

2 Steady-state saturation and pressure profiles for infiltration rates of 0, 8 and 80 mm/y. The water table is located at 6 m from the bottom of the computational domain. .............................................................. 147

3 Simulation of a tank leak with a duration of two weeks showing the saturation profile for different times indicated in the figure. .............................................................. 148

4 The solute concentration profile corresponding to Figure 7 for different times indicated in the figure. .............................................................. 148

5 Control volumes in DCDM multiple continuum model with fracture aperture $2\delta$ and matrix block size $d$. .............................................................. 174

LIST OF TABLES

1 Options for configuring petsc-dev. .............................................................. 11

2 Compile options for PFLOTRAN. .............................................................. 12

3 Available modes in PFLOTRAN .............................................................. 12

5 Units .............................................................. 25

6 Stratigraphic sequence used in the calculations, after Ward et al. (1996). ............. 145

7 Parameters for material and thermal properties for intrinsic rock density $\rho_s$, heat capacity $c$, thermal conductivity $\kappa$, porosity $\varphi$, residual water saturation $s_r$, van Genuchten parameters $\alpha$ and $\lambda$, and vertical water saturated permeability $k_{sat}$. Data taken from Khaleel and Freeman (1995), Khaleel et al. (2001), and Pruess et al. (2002). .............................................................. 146

8 Choice of primary variables. .............................................................. 165

9 DCDM geometric parameters. .............................................................. 175

10 Independent and dependent nested cube parameters. .............................................................. 175

11 Format of thermodynamic database. .............................................................. 189

12 Fit coefficients for log $K$ of reaction 10-193. .............................................................. 189
1 Introduction

PFLOTRAN solves a system of generally nonlinear partial differential equations describing multi-phase, multicomponent and multiscale reactive flow and transport in porous materials. The code is designed to run on massively parallel computing architectures as well as workstations and laptops (e.g. Hammond et al., 2011). Parallelization is achieved through domain decomposition using the PETSc (Portable Extensible Toolkit for Scientific Computation) libraries for the parallelization framework (Balay et al., 1997).

PFLOTRAN has been developed from the ground up for parallel scalability and has been run on up to $2^{18}$ processor cores with problem sizes up to 2 billion degrees of freedom. Written in object oriented Fortran 90, the code requires the latest compilers compatible with Fortran 2003. At the time of this writing this requires gcc 4.7.x, Intel 12.1.x and PGC compilers. As a requirement of running problems with a large number of degrees of freedom, PFLOTRAN allows reading input data that is too large to fit into memory allotted to a single processor core. The current limitation to the problem size PFLOTRAN can handle is the limitation of the HDF5 file format used for parallel IO to 32 bit integers. Noting that $2^{32} = 4,294,967,296$, this gives an estimate of the maximum problem size that can be currently run with PFLOTRAN. Hopefully this limitation will be remedied in the near future.

Currently PFLOTRAN can handle a number of surface and subsurface processes including Richards equation, two-phase flow involving supercritical CO$_2$, and multicomponent reactive transport including aqueous complexing, sorption and mineral precipitation and dissolution. The reactive transport equations can be solved using either a fully implicit Newton-Raphson algorithm or the less robust operator splitting method. In addition, a novel approach is used to solve equations resulting from a multiple interacting continuum method for modeling flow and transport in fractured media. This implementation is still under development.

A novel feature of the code is its ability to run multiple input files and multiple realizations of permeability and porosity fields simultaneously on one or more processor cores per run. This can be extremely useful when conducting sensitivity analyses and quantifying model uncertainties. When running on machines with many cores this means that hundreds of simulations can be conducted in the amount of time needed for a single realization.

Additional instructions can be found on the PFLOTRAN wiki home page. Questions regarding the PFLOTRAN installation, and bug reports may be directed to: pflotran-dev at googlegroups dot com. For questions regarding running PFLOTRAN contact: pflotran-users at googlegroups dot com.
2 Quick Start

For those adverse to reading manuals the following is a quick start guide to getting up and running with PFLOTRAN in four easy steps.

Required Software Packages

- Compilers (compatible with Fortran 2003: gcc 4.7.x, Intel 12.1.x, PGI)
  For MacOSX 10.8.2 the latest Fortran compiler binaries (gcc/gfortran 4.8 at the time of this writing) may be downloaded from the web site HPC for MacOSX.
- Mercurial (hg) (version control system)
- CMAKE (version 2.8.x)
- Developer version of PETSc: petsc-dev
  - MPI (for running in parallel)
  - BLAS/LAPACK libraries
  - HDF5 (parallel output)
  - METIS/ParMETIS (unstructured grids)
- PFLOTRAN

2.1 Installing PFLOTRAN

Step 1: Installing PETSc
The first step is to download the developer version of PETSc, petsc-dev. To do this follow the directions at the PETSc Developer web site. The recommended approach is to use git.

To compile PETSc it is first necessary to configure the makefile. To this end define the environmental variables PETSC_DIR and PETSC_ARCH pointing to the directory where petsc-dev is installed and an identifying name for the installation, respectively. E.g. if using the t-shell on MacOSX 10.8.2 and gcc-4.8 enter into the .tcshrc file:

```tcsh
setenv PETSC_DIR 'path to petsc-dev'
setenv PETSC_ARCH MacOSX-10.8.2-gcc4.8.
```

Finally, set the environmental variable MPI_HOME and add to the PATH variable:

```bash
setenv MPI_HOME ${PETSC_DIR}/${PETSC_ARCH}
setenv PATH $MPI_HOME/bin:$PATH
```

Decide which options are needed for running PFLOTRAN: structured grids are the simplest; if unstructured grids are needed then install METIS and ParMETIS. See Table 1 for several of the different configure options possible.

For example, to configure PETSC to run in production mode and install openmpi, HDF5, METIS and ParMETIS use:
cd ./petsc-dev

./config/configure.py --with-debugging=0 --with-shared-libraries=0 --with-x=0
--download-openmpi=1 --download-hdf5=1 --download-metis=1 --download-parmetis=1

Check to make sure mpicc and mpif90 are working and are recent versions that are compatible with Fortran 2003: e.g. GNU gfortran version 4.7.x, Intel version 12.x or PGI.

Table 1: Options for configuring petsc-dev.

<table>
<thead>
<tr>
<th>Option</th>
<th>Package</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI</td>
<td>Parallel</td>
<td>OpenMPI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPICH</td>
</tr>
<tr>
<td>Grid</td>
<td>Structured</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Unstructured</td>
<td>METIS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ParMETIS*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>Output</td>
<td>TecPlot, VTK</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>HDF5</td>
<td>Parallel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HDF5</td>
</tr>
<tr>
<td></td>
<td>Solvers</td>
<td>Parallel</td>
</tr>
<tr>
<td></td>
<td>Direct</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Iterative (GMRES)</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Preconditioners</td>
<td>Hypre</td>
</tr>
</tbody>
</table>

* Need CMAKE

Step 2: Downloading PFLOTRAN  To obtain the latest version of the PFLOTRAN source code use Mercurial. Mercurial (hg) is the version control system used by both PETSc and PFLOTRAN. Often one can find executables for particular platforms and there is no need to compile the application from scratch. On a UNIX system you can check to see if Mercurial is already installed by issuing the command: which hg.

PFLOTRAN can be downloaded from bitbucket.org by issuing the command line:

hg clone https://bitbucket.org/pflotran/pflotran-dev.

Step 3: Compiling PFLOTRAN  Once PETSc and associated third party software has been installed, it is easy to install PFLOTRAN. To compile PFLOTRAN first decide on the options needed to run your problem. The most common of these are listed in Table 2.
Table 2: Compile options for PFLOTRAN.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scco2=1</td>
<td>MPHASE mode for supercritical CO$_2$</td>
</tr>
<tr>
<td>scorpio=1</td>
<td>SCORPIO* parallel IO</td>
</tr>
</tbody>
</table>

* See §3.5 for installing SCORPIO.

```
cd PLOTTRAN_DIR/src/pflotran
make [options] ./pflotran
```

For example: `make scco2=1 ./ppflotran`

**Step 4: Running PFLOTRAN**  PFLOTRAN is run in parallel on $N$ cores using the command:

```
mpirun -np N ./pflotran -pflotranin input_file_name.in
```

with input file `input_file_name.in`. The default input file name is simply `pflotran.in`.

To get the most out of PFLOTRAN subscribe to the PFLOTRAN User Group:

```
pflotran-users@googlegroups.com.
```

PFLOTRAN developers monitor this site and will as quickly as possible respond to your queries.

The final challenge is setting up an input file. Various modes available are listed in Table 3 and example input decks are stored in the `./pflotran/example_problems` and `./pflotran/shortcourse` directories.

Table 3: Available modes in PFLOTRAN

<table>
<thead>
<tr>
<th>MODE</th>
<th>Description</th>
<th>Required Databases$^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RICHARDS</td>
<td>Solves Richards equation</td>
<td>—</td>
</tr>
<tr>
<td>MPHASE</td>
<td>Supercritical CO$_2$</td>
<td>co2data0.dat</td>
</tr>
<tr>
<td>TH</td>
<td>Thermal-Hydrology mode</td>
<td>—</td>
</tr>
<tr>
<td>CHEMISTRY</td>
<td>Reactive Transport</td>
<td>hanford.dat$^+$</td>
</tr>
</tbody>
</table>

$^+$ Provided with the PFLOTRAN distribution in directory `./pflotran/database`.

$^+$ An equivalent user supplied thermodynamic database for chemical reactions may also be used (see §10.11 for a description of the database format).

An example input file is listed in §4.2 for a coupled problem using Richards and Chemistry modes.
2.2 Running on Big Iron Parallel Architectures

Generally these machines use module to load the computing environment. Make sure the correct compilers are loaded that are compatible with Fortran 2003. The following instructions apply to Yellowstone running Red Hat linux using the Intel compiler. As MPI is already installed it is not necessary to reinstall it through PETSc.

Use a recent version of CMAKE: module load cmake/2.8.10.2.

Set the environmental variable for the BLAS/LAPACK libraries to use MKL:

```
setenv BLAS_LAPACK_LIB_DIR /ncar/opt/intel/12.1.0.233/composer_xe_2013.1.117/mkl
```

Configure PETSc:

```
./config/configure.py --with-cc=mpicc --with-fc=mpif90 --with-cxx=mpicxx
--with-clanguage=c --with-blas-lapack-dir=$BLAS_LAPACK_LIB_DIR
--with-shared-libraries=0 --with-debugging=0 --download-hdf5=yes
--download-parmetis=yes --download-metis=yes
```
3 Installation

The source code for PFLOTRAN can be downloaded from the bitbucket.org web site using Mercurial (hg):

```
hg clone https://bitbucket.org/pflotran/pflotran-dev.
```

This requires first creating a free account on bitbucket.

It should be possible to build and run PFLOTRAN on essentially any system with modern C and Fortran (2003 standard or later) compilers and an available implementation of the Message Passing Interface (MPI) system that has been built with Fortran bindings. Besides these requirements, the major third-party library required is the open-source library PETSc—the Portable, Extensible Toolkit for Scientific Computation—that provides the parallel framework on which PFLOTRAN is built. Most of the work involved in building PFLOTRAN lies in building PETSc. PETSc uses a sophisticated Python-based build tool, BuildSystem, to perform extensive platform discovery and configuration as well as automatic download and build for any of the open-source third-party libraries that PETSc can use. The PFLOTRAN makefiles use the information generated by BuildSystem as part of the PETSc build process; once PETSc is built, building PFLOTRAN is straightforward.

Besides PETSc, third party libraries commonly installed are

- MPI, message passing interface (OpenMPI or MPICH)
- HDF5, required for parallel I/O and reading HDF5-formatted input files
- Metis and ParMetis, graph partitioning libraries required for unstructured grids
- Hypre, which provides a variety of preconditioners and multilevel solves

PETSc can be used to download, compile and install all of these third party libraries during compilation of PETSc as explained below, so that the user does not have to do this themselves individually for each library.

3.1 Compilers

The installation of PFLOTRAN on MacOSX requires compiler versions 4.7 or later for gfortran and gcc to be compatible with Fortran 2003.

For MacOSX compilers can be obtained from several sources including MacPorts and the web site High Performance Computing for MacOSX.

3.2 Building PETSc

The first step to building PFLOTRAN is to configure and build the PETSc toolkit. This requires, at minimum, working installations of C and Fortran 95-compliant compilers. For users looking for
an open-source compiler, we recommend the gcc and gfortran compilers that are part of the GNU Compiler Collection (GCC), version 4.7.x or later. Users may also wish to install MPI and other libraries from source or via a package manager, but the PETSc ./configure script can be used not only to install PETSc but also MPI, HDF5, Par METIS/METIS, and various solver libraries such as MUMPS for sparse direct solvers, Hypre for a variety of preconditioners and multi-level solvers, and the Trilinos multilevel solver ML. For systems that do not provide specially optimized versions of these libraries, we recommend using PETSc’s configure to install these third-party libraries. If you do wish to install any of these third-party libraries yourself, you will need to do so before installing PETSc to that the necessary PETSc interfaces to these packages can be built.

The development branch of PFLOTRAN tracks the main development branch of PETSc and hence requires this “petsc-dev” version, which can be either downloaded from the PETSc web page petsc-dev, or installed using Mercurial following instructions on the PETSc developer web page. We recommend that PETSc be obtained using the version control system (Mercurial).

Define environment variables PETSC_DIR and PETSC_ARCH giving the location of the petsc-dev source and the directory where the libraries for the particular architecture are stored after compiling and installing PETSc.

3.2.1 Mac OS X

3.2.1.1 Lion, Mavericks (≤ Mac OS X 10.9.x) To install ParMETIS/METIS on MacOSX it is necessary to first install the latest version (3.0.x) of CMAKE from e.g. Homebrew, MacPorts or Fink.

To install PETSc, MPI using openmpi, HDF5, and ParMETIS/METIS with debugging turned off configure PETSc using:

```bash
./config/configure.py --with-debugging=0 --with-shared-libraries=0 --with-x=0 --download-openmpi --download-hdf5=1 --download-metis=1 --download-parmetis=1
```

followed by

```
make all
```

and

```
make test
```

Note that ParMETIS/METIS is only needed for using the unstructured grid capability in PFLOTRAN. HDF5 is recommended for large problems for use with Visit for (parallel) visualization.

3.2.1.2 Yosemite Special considerations are required to install PETSc and PFLOTRAN on Yosemite (MacOSX 10.10). It is necessary to install gcc-4.9 from Homebrew [other installations
such as HPC MacOS X are not bug free at the time of this writing (10.23.14)]. The following steps are required:

1. install Homebrew (standard location /usr/local must be clean to avoid collisions)
2. install gcc (need version 4.9.x): brew install gcc
3. install cmake (need version 3.0.x): brew install cmake
4. configure PETSc using the Mac c and c++ compilers clang, clang++ and gfortran installed from Homebrew:
   cd petsc-dev
   ./configure --with-cc=clang --with-cxx=clang++
   --with-fc=gfortran --download-mpich=1
   --download-hdf5=1 --download-metis=1
   --download-parmetis=1 --with-debugging=0
5. make all
   make test

Notes: At the time of this writing (10.23.14) openmpi did not compile. HDF5 did not compile using gcc-4.9 installed from Homebrew.

### 3.2.2 Windows

To install PETSc and PFLOTRAN on Windows see instructions on the PFLOTRAN wiki (https://bitbucket.org/pflotran/pflotran/pflotran-dev/wiki/Home).

### 3.2.3 ORNL's Jaguar XT4/5

./config/configure.py PETSC_ARCH=cray-xt4-pgi_fast \ 
--configModules=PETSc.Configure \ 
--optionsModule=PETSc.compilerOptions \ 
--known-level1-dcache-size=65536 \ 
--known-level1-dcache-linesize=64 \ 
--known-level1-dcache-assoc=2 \ 
--known-memcmp-ok=1 \ 
--known-sizeof-char=1 \ 
--known-sizeof-void-p=8 \ 
--known-sizeof-short=2 \ 
--known-sizeof-int=4 \ 
--known-sizeof-long=8 \ 
--known-sizeof-long-long=8 \ 
--known-sizeof-float=4 \ 
--known-sizeof-double=8 \

< 15 >
The user will need to load the following HDF5 module beforehand: `module load hdf5-parallel`.

### 3.3 Building PFLOTRAN

PFLOTRAN is compiled with the command:

```
make [options] pflotran
```

where several possible options are:

<table>
<thead>
<tr>
<th>Compile Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>scorpio=1</code></td>
<td>–enable parallel IO</td>
</tr>
<tr>
<td><code>coll=1</code></td>
<td>–enable colloid-facilitated transport option</td>
</tr>
<tr>
<td><code>mfd=1</code></td>
<td>–enable MFD full permeability tensor capability (not operational)</td>
</tr>
</tbody>
</table>

Thus for compiling with the supercritical CO\textsubscript{2} option use: `make sasco2=1 pflotran`. See the PFLOTRAN `makefile` for additional options.

### 3.4 Updating PFLOTRAN

To update the PFLOTRAN source code type:

```
hg pull -u
```
from within the PFLOTRAN source repository. Recompile PFLOTRAN using:

```make
make clean
make [options] pflotran
```

### 3.5 Parallel I/O using Sarat Sreepathi’s SCORPIO library with PFLOTRAN

The SCORPIO (parallel I/O library) enables a scalable general purpose parallel I/O capability for HPC by taking advantage of existing parallel I/O libraries, such as HDF5 which are being widely used by scientific applications, and modifying these algorithms to better scale to larger number of processors. The library has been tested with MPICH-3.0.2 and OpenMPI-1.6.

It is recommended that values for the variables `HDF5_READ_GROUP_SIZE` and `HDF5_WRITE_GROUP_SIZE` be set in the input file. If they are unset, `HDF5_READ_GROUP_SIZE` is set to total MPI ranks and `HDF5_WRITE_GROUP_SIZE` is set to 1 by default.

Typical values are to set the write group size equal to the number of processes on a compute node (typically 16 or 32). A much higher read group size is preferred, e.g. use 512 when running on 512 cores so that one process reads the input file and broadcasts relevant sections. Put `HDF5_WRITE_GROUP_SIZE` under the `OUTPUT` keyword:

```plaintext
:=========================== output options ===============================
OUTPUT
  TIMES y 5. 10. 15. 20.
  FORMAT HDF5
  HDF5_WRITE_GROUP_SIZE 16
END
```

and `HDF5_READ_GROUP_SIZE` in the main body of the input file:

```
HDF5_READ_GROUP_SIZE 1024
```

For more details on the SCORPIO library, please see Appendix A in Sarat Sreepathi’s <admin@sarats.com> dissertation.

Instructions for downloading and installing the SCORPIO library for use with PFLOTRAN is provided below. Note that this software is separate from PFLOTRAN and under a LGPL.

1. Download source code for building the SCORPIO library:

```
svn co http://ascem-io.secure-water.org/ascem-io/scorpio DIRNAME
```

where `DIRNAME` is the installation directory (Default: scorpio).

The username and password are:

- **username:** `pflotran_dev`
- **password:** `gr0undw@t3r`

< 17 >
2. Compile SCORPIO library: First, set the environment variable SCORPIO_DIR to a directory where you wish to install the library files. Please make sure that you have the correct permissions to write to that location. For example, you can use something like ${HOME}/parallelIO/scorpio. Depending on the shell you are using you may use:

```bash
export SCORPIO_DIR=<your-iolib-install-dir>
```

or

```bash
setenv SCORPIO_DIR <your-iolib-install-dir>
```

To compile the library, check to make sure that the Makefile has the right settings for your machine. Typically, the current configuration suffices. So you can just follow the instructions below. For advanced users, please edit the section for pflotran_machine in the Makefile as desired.

This assumes that you have let PETSc build MPI and mpicc and mpif90 are located in ${PETSC_DIR}/${PETSC_ARCH}/bin. If not, you may need to alter the Makefile to provide the correct path to the location of the MPI compilers.

```bash
cd DIRNAME/src
make MACHINE=pflotran_machine
make install
```

This will build the library libscorpio.a and copy corresponding files to SCORPIO_DIR/lib and SCORPIO_DIR/include directories.

On machines with MPI installed modify the makefile to use the native mpi compilers:

```bash
ifeq ($(MACHINE),pflotran_machine)
CC=mpicc
FC=mpif90
LINKER=${FC}
CFLAGS+= -I${PETSC_DIR}/${PETSC_ARCH}/include -O3
FFLAGS+= -I${PETSC_DIR}/${PETSC_ARCH}/include -O3
LDFLAGS+= -Wl,-L${PETSC_DIR}/${PETSC_ARCH}/lib -lhdf5 -lz
endif
```

3. Compile PFLOTRAN:

Please ensure that environmental variable: SCORPIO_DIR is pointed to <your-iolib-install-dir>:

```bash
cd PFLOTRAN_DIR/src/pflotran
make scorpio=1 pflotran
```

---

**SCORPIO**

Scalable Parallel I/O module for Environmental Management Applications

---

This library provides software that read/write data sets from/to parallel file systems in an efficient and scalable manner.

In this context, scalable means that the simulators read/write performance does not degrade significantly as the number of cores grows.

---

**COPYRIGHT AND LICENSE**

---

SCORPIO is distrubuted under the terms of the GNU Lesser General Public
License (LGPL). The copyright is held jointly by North Carolina State University and Pacific Northwest National Laboratory.

The copyright and license information is specified in the included file COPYRIGHT.

Repository Access

Please request write access by contacting Kumar Mahinthakumar (gmkumar@ncsu.edu). Use the following command to access repository:
svn co http://ascem-io.secure-water.org/ascem-io

Building Library

The current stable release is 2.2.

Prerequisites:

MPI
C compiler
HDF5 libraries (preferably with parallel(MPI) support)
Optional: Fortran (for Fortran example)

After downloading SCORPIO and gathering details of HDF5 installation, the following commands can be used to build and install SCORPIO:

cd <SCORPIO check out directory>/src
make CC=<C-compiler> HDF5_INCLUDE_DIR=<location of the HDF5 include directory>
make SCORPIO_INSTALL_DIR=<user defined install location> install

In this case, CC refers to C compiler with MPI support, e.g., mpicc.
3.6 Running PFLOTRAN

PFLOTRAN can be run from the command line as

```
mpirun -np 10 pflotran [options]
```

A number of command line options are available:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-pflotranin &lt;string&gt;</td>
<td>specify input file [Default: pflotran.in]</td>
</tr>
<tr>
<td>-input_prefix &lt;string&gt;</td>
<td>specify input (and output) file prefix [default: pflotran]</td>
</tr>
<tr>
<td>-output_prefix &lt;string&gt;</td>
<td>specify output file prefix [default: pflotran]</td>
</tr>
<tr>
<td>-screen_output off</td>
<td>turn off screen output</td>
</tr>
<tr>
<td>-realization_id &lt;integer&gt;</td>
<td>run specified realization ID</td>
</tr>
<tr>
<td>-multisimulation</td>
<td>run multiple input files in one run</td>
</tr>
<tr>
<td>-stochastic</td>
<td>Monte Carlo multiple realization run</td>
</tr>
</tbody>
</table>

PETSc

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-help</td>
<td>print list of command line options</td>
</tr>
<tr>
<td>-log_summary</td>
<td>print out run performance</td>
</tr>
<tr>
<td>-snes_converged_reason</td>
<td>print the reason for convergence/divergence after each solve</td>
</tr>
<tr>
<td>-on_error_abort</td>
<td>aborts run on hitting NaNs</td>
</tr>
<tr>
<td>-v</td>
<td>&lt;int&gt; verbose</td>
</tr>
<tr>
<td>-options_left</td>
<td>for debugging: lists any PETSc objects that have not been freed</td>
</tr>
<tr>
<td>-snes_monitor</td>
<td>print to screen function norm with each iteration</td>
</tr>
<tr>
<td>-snes_linesearch_monitor</td>
<td>print to screen line search information</td>
</tr>
<tr>
<td>-snes_view</td>
<td>print line search method</td>
</tr>
<tr>
<td>-snes_ls</td>
<td>options: [cubic, quadratic, basic, basic-norms]</td>
</tr>
<tr>
<td>-snes_type newtonls</td>
<td></td>
</tr>
<tr>
<td>-snes_type newtontr</td>
<td></td>
</tr>
<tr>
<td>-snes_tr_delta0</td>
<td>&lt;delta0&gt;</td>
</tr>
<tr>
<td>-snes_monitor_lg_residualnumnorm</td>
<td>draws line graph of the residual norm convergence using X11</td>
</tr>
<tr>
<td>-ksp_monitor_lg_residualnumnorm</td>
<td></td>
</tr>
</tbody>
</table>

< 20 >
3.7 Multiple Realization Simulation Mode

To launch 1000 realizations on 100 processor groups using 10,000 processor cores:

```bash
mpirun -np 10000 pflotran -stochastic -num_realizations 1000
   -num_groups 100
```

Each processor group will utilize 100 processor cores and run 10 realizations apiece
(num_realizations/num_groups), one after another. Thus, 100 realizations are executed
simultaneously with each processor group simulating a single realization on 100 processor cores at
a time. Each processor group continues to run realizations until its allocation of 10 is completed.

To simulate a specific realization without running in multi-realization stochastic mode use:

```bash
mpirun -np 10000 pflotran -realization_id <integer>
```

where <integer> specifies the realization id.

3.8 Multiple Simulation Mode

To run multiple input decks simultaneously, create a file e.g. filenames.in, containing the list
of filenames that are to be run:

```
sim1.in
sim2.in
...
sim100.in
```

where simn.in is the usual PFLOTRAN input file. The names may be arbitrarily chosen. Then,
launch the run as:

```bash
mpirun -n XXX pflotran -pflotranin filenames.in -multisimulation
```

Note that all simulations run at once. The same logic used to allow a processor group to run mul-
tiple simulations with multirealization is not implemented. Choose the number of cores
to be a multiple of the number of input filenames listed in the input file (i.e. filenames.in). All
output files have Gn appended to the file name, e.g. sim10G10-001.tec.
4 Creating the Input File: PFLOTRAN Keywords

The PFLOTRAN input file construction is based on keywords. Lines beginning with # are treated as comments. Each entry to the input file must begin in the first column. Keywords SKIP and NOSKIP are used to skip over sections of the input file. Blank lines may occur in input file. Alternate keyword spelling is indicated in round brackets ( ). Input options are indicated in square brackets [ ], as well as default values. Curly brackets { } indicate the result of invoking the corresponding keyword. Always refer to source code when in doubt!

Initial and boundary conditions and material properties are assigned to spatial regions using a novel coupler approach. In this approach, initial and boundary conditions (keyword CONDITION) are assigned to regions (keyword REGION) using keywords INITIAL_CONDITION and BOUNDARY_CONDITION. Material properties (keyword MATERIAL) are assigned to regions using the keyword STRATIGRAPHY.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOUNDARY_CONDITION</td>
<td>transport (optional)</td>
</tr>
<tr>
<td>BRINE</td>
<td></td>
</tr>
<tr>
<td>CHARACTERISTIC_CURVES</td>
<td></td>
</tr>
<tr>
<td>CHECKPOINT</td>
<td></td>
</tr>
<tr>
<td>CHEMISTRY</td>
<td></td>
</tr>
<tr>
<td>COMPUTE_STATISTICS</td>
<td></td>
</tr>
<tr>
<td>CO2_DATABASE</td>
<td></td>
</tr>
<tr>
<td>CONSTRAINT</td>
<td></td>
</tr>
<tr>
<td>DATASET</td>
<td></td>
</tr>
<tr>
<td>DBASE_DATASET</td>
<td></td>
</tr>
<tr>
<td>DEBUG</td>
<td></td>
</tr>
<tr>
<td>EOS</td>
<td></td>
</tr>
<tr>
<td>FLOW_CONDITION</td>
<td></td>
</tr>
<tr>
<td>FLUID_PROPERTY</td>
<td></td>
</tr>
<tr>
<td>GEOMECHANICS</td>
<td></td>
</tr>
<tr>
<td>GEOMECHANICS_GRID</td>
<td></td>
</tr>
<tr>
<td>GEOMECHANICS_OUTPUT</td>
<td></td>
</tr>
<tr>
<td>GEOMECHANICS_MATERIAL_PROPERTY</td>
<td></td>
</tr>
<tr>
<td>GEOMECHANICS_REGION</td>
<td></td>
</tr>
<tr>
<td>GEOMECHANICS_CONDITION</td>
<td></td>
</tr>
<tr>
<td>GEOMECHANICS_BOUNDARY_CONDITION</td>
<td></td>
</tr>
<tr>
<td>GEOMECHANICS_STRATA</td>
<td>(required)</td>
</tr>
</tbody>
</table>
4.1 Conventions and Notation

Keywords are in boldface with optional modifying keywords in square brackets [...] , and user entries in typewriter font. Unless otherwise specified, units in the input file are assumed to be as listed in Table 5.
### Table 5: Units

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure:</td>
<td>Pascal [Pa] (absolute)</td>
</tr>
<tr>
<td>Temperature:</td>
<td>Celcius [C]</td>
</tr>
<tr>
<td>Distance:</td>
<td>meter [m]</td>
</tr>
<tr>
<td>Volume:</td>
<td>meter³ [m³]</td>
</tr>
<tr>
<td>Time:</td>
<td>second [s]</td>
</tr>
<tr>
<td>Velocity:</td>
<td>meter/second [m/s]</td>
</tr>
<tr>
<td>Concentration:</td>
<td>molarity [M] or molality [m] (see MOLAL keyword)</td>
</tr>
<tr>
<td>Enthalpy:</td>
<td>kiloJoule/mole [kJ/mol]</td>
</tr>
<tr>
<td>Mass:</td>
<td>kilogram [kg]</td>
</tr>
<tr>
<td>Rate:</td>
<td>kilogram/second [kg/s] or cubic meter/second [m³/s]</td>
</tr>
<tr>
<td>Surface Site Density:</td>
<td>mole/meter³ [mol/m³]</td>
</tr>
</tbody>
</table>
4.2 Example Input File

#Description: 3D infiltration problem with calcite dissolution (# denotes a comment line)

# == debugging
#MATVIEW_JACOBIAN
#VECVIEW_RESIDUAL
#VECVIEW_SOLUTION
#/ 

# == mode
MODE RICHARDS

# == chemistry
CHEMISTRY
OPERATOR_SPLIT
PRIMARY_SPECIES
Ca++
H+
CO2(aq)
Tracer
/
SECONDARY_SPECIES
OH−
HCO3−
CO3−−
CaHCO3+
CaCO3(aq)
/
GAS_SPECIES
CO2(g)
/
MINERALS
Calcite
/
# 
MINERAL_KINETICS
Calcite
RATE_CONSTANT 1.e-8 ! [mol/m^2/s]
/
/
# 
DATABASE /Users/lichtner/flotran/database/hanford.dat
LOG_FORMULATION
ACTIVITY_COEFFICIENTS !NEWTON_ITERATION
MOLAL
OUTPUT
All
/
/

# == reference variables
REFERENCE_POROSITY 0.25d0
# == time stepping ============================================================
TIMESTEPPER
TS_ACCELERATION 8
MAX_STEPS 100000
/

# == discretization ===========================================================
GRID
TYPE structured
NXYZ 6 6 6
BOUNDS
0.d0 0.d0 0.d0
1.d0 1.d0 1.d0
/

# == flow solvers =============================================================
NEWTON_SOLVER FLOW
PRECONDITIONER_MATRIX_TYPE AIJ
RTOL 1.d-8
ATOL 1.d-8
STOL 1.d-30
ITOL_UPDATE 1.d0
#NO_INFINITY_NORM
#NO_PRINT_CONVERGENCE
#PRINT_DETAILED_CONVERGENCE
/

LINEAR_SOLVER FLOW
#KSP_TYPE PREONLY
#PC_TYPE LU
#KSP_TYPE FGMRES !samrai
#PC_TYPE SHELL !samrai
/

# == transport solvers ========================================================
NEWTON_SOLVER TRANSPORT
PRECONDITIONER_MATRIX_TYPE AIJ
RTOL 1.d-12
ATOL 1.d-12
STOL 1.d-30
#NO_INFINITY_NORM
#NO_PRINT_CONVERGENCE
#PRINT_DETAILED_CONVERGENCE
/

LINEAR_SOLVER TRANSPORT
#PC_TYPE LU
#KSP_TYPE PREONLY
#KSP_TYPE FGMRES ! samrai
#PC_TYPE SHELL !samrai
/

< 26 >
# == fluid properties =========================================================
FLUID_PROPERTY
DIFFUSION_COEFFICIENT 1.d-9
/

#UNIFORM_VELOCITY 3.84259d-6 0.d0 0.d0 ! 1.38333 cm/h

# == material properties ======================================================
MATERIAL_PROPERTY HD
ID 1
SATURATION_FUNCTION HD
POROSITY 0.262
TORTUOSITY 1.0
PERMEABILITY
PERM_ISO 5.43d-13
/

# == saturation / permeability functions =======================================
SATURATION_FUNCTION HD
SATURATION_FUNCTION_TYPE VAN_GENUCHTEN
RESIDUAL_SATURATION 0.115
LAMBDA 0.286
ALPHA 1.9401d-4
/

# == output =================================================================
OUTPUT
#PERIODIC TIMESTEP 1
PERIODIC TIME 0.1 y
FORMAT HDF5
FORMAT TECPlot BLOCK
VELOCITIES
/

# == times ==================================================================
TIME
FINAL_TIME 1.d0 y
INITIAL 1.e-6 y
MAXIMUM_TIMESTEP_SIZE 1.e-2 y
/

# == regions ================================================================
REGION all
COORDINATES
0.d0 0.d0 0.d0
6.d0 6.d0 6.d0
/

REGION Top
FACE TOP
COORDINATES
0.d0 0.d0 6.d0
6.d0 6.d0 6.d0
/

< 27 >
REGION Inlet
FACE TOP
COORDINATES
2.d0 2.d0 6.d0
4.d0 4.d0 6.d0
/
#BLOCK 3 4 3 4 6 6
/

REGION Bottom
FACE BOTTOM
COORDINATES
0.d0 0.d0 0.d0
6.d0 6.d0 0.d0
/
/

# == flow conditions ==========================================================
FLOW_CONDITION Inlet
TYPE
FLUX neumann
/
FLUX 0.317098d-6 ! 10 m/y
/

FLOW_CONDITION Initial
TYPE
PRESSURE hydrostatic
/
DATUM 0.d0 0.d0 6.d0
PRESSURE 101325.d0
/
/

# == transport conditions ====================================================
TRANSPORT_CONDITION Inlet
TYPE dirichlet
CONSTRAINT_LIST
0.d0 Inlet
/
/

TRANSPORT_CONDITION Initial
TYPE dirichlet
CONSTRAINT_LIST
0.d0 Initial
/
/

TRANSPORT_CONDITION Outlet
TYPE zero_gradient
CONSTRAINT_LIST
0.d0 Initial
/
/

# == couplers ==============================================================
BOUNDARY_CONDITION Inlet
FLOW_CONDITION Inlet
TRANSPORT_CONDITION Inlet
REGION Inlet
/

BOUNDARY_CONDITION Outlet
FLOW_CONDITION Initial
TRANSPORT_CONDITION Outlet
REGION Bottom
/

INITIAL_CONDITION Initial
FLOW_CONDITION Initial
TRANSPORT_CONDITION Initial
REGION all
/

# == stratigraphy ==============================================================
STRATA
MATERIAL HD
REGION all
/

# == transport constraints ==============================================================
CONSTRAINT Initial
CONCENTRATIONS
Ca++ 1.d-6 M Calcite
H+ 3.d0 pH
CO2(aq) 1.d-3 G CO2(g)
Tracer 1.d-0 T
/
MINERALS ! vol. frac. area
Calcite 0.75 1.d2
/
/

CONSTRAINT Inlet
CONCENTRATIONS
Ca++ 1.d-6 T
H+ 3.d0 pH
CO2(aq) 1.d-3 G CO2(g)
Tracer 1.d-0 T
/
/
4.3 Keyword: BOUNDARY_CONDITION

Description: The BOUNDARY_CONDITION keyword couples conditions specified under the FLOW_CONDITION and/or TRANSPORT_CONDITION keywords to a REGION in the problem domain. The use of this keyword enables the use/reuse of flow and transport conditions and regions within multiple boundary and initial conditions and source/sinks in the input deck.

Input:

BOUNDARY_CONDITION boundary_condition_name

    FLOW_CONDITION flow_condition_name
    TRANSPORT_CONDITION transport_condition_name
    REGION region_name

END [/ .]

Explanation:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOUNDARY_CONDITION</td>
<td>Defines the beginning of a boundary condition entry and the name of the boundary condition.</td>
</tr>
<tr>
<td>FLOW_CONDITION</td>
<td>Defines the name of the flow condition to be linked to this boundary condition.</td>
</tr>
<tr>
<td>TRANSPORT_CONDITION</td>
<td>Defines the name of the transport condition to be linked to this boundary condition.</td>
</tr>
<tr>
<td>REGION</td>
<td>Defines the name of the region to which the conditions are linked.</td>
</tr>
<tr>
<td>END</td>
<td>Terminates the boundary condition entry.</td>
</tr>
</tbody>
</table>
Examples:

BOUNDARY_CONDITION river
    FLOW_CONDITION river_stage
    TRANSPORT_CONDITION river_chemistry
    REGION river_bank
END

BOUNDARY_CONDITION recharge
    FLOW_CONDITION infiltration_flux
    TRANSPORT_CONDITION infiltration_chemistry
    REGION ground_surface
END

↩ Return to List of Keywords
4.4 Keyword: BRINE

BRINE <float> [MOLAL, MASS, MOLE]

Description: Units refer to concentration of an NaCl brine in molality, mass fraction or mole fraction with equal concentration of Na\(^+\) and Cl\(^-\).

Example:

BRINE 4.\text{d}0 \ MOLAL
4.5 Keyword: CHARACTERISTIC_CURVES

Description: Specifies relative permeability and saturation functions and parameters to be associated with a material property.

Input:

CHARACTERISTIC_CURVES <string>

SATURATION_FUNCTION [VAN_GENUCHTEN, BROOKS_COREY]
  ALPHA <float> [Pa⁻¹]
  M <float> [—]
  LIQUID_RESIDUAL_SATURATION <float> [—]
  MAX_CAPILLARY_PRESSURE <float> [Pa] [Default 10⁹ Pa]
END [/ .]

PERMEABILITY_FUNCTION MUALEM
  M
  LIQUID_RESIDUAL_SATURATION
END [/ .]

PERMEABILITY_FUNCTION BURDINE
  LAMBDA
  LIQUID_RESIDUAL_SATURATION
END [/ .]

PERMEABILITY_FUNCTION MUALEM_VG_GAS
  M
  LIQUID_RESIDUAL_SATURATION
  GAS_RESIDUAL_SATURATION
END [/ .]

PERMEABILITY_FUNCTION BURDINE_BC_GAS
  LAMBDA
  LIQUID_RESIDUAL_SATURATION
  GAS_RESIDUAL_SATURATION
END [/ .]

END [/ .]

Parameter Description:

Required cards within block:
• SATURATION_FUNCTION: Opens a saturation function block with the string indicating the type of function (options include VAN_GENUCHTEN, BROOKS_COREY).

• PERMEABILITY_FUNCTION: Opens a relative permeability function block with the string indicating the type of function (options include MUALEM, BURDINE, MUALEM_VG_GAS, BURDINE_BC_GAS).

• ALPHA <float> Inverse of the air entry pressure for the saturation function [Pa\(^{-1}\)].

• GAS_RESIDUAL_SATURATION <float> Residual saturation for gas phase [—].

• LAMBDA <float> Brooks-Corey lambda [—].

• LIQUID_RESIDUAL_SATURATION <float> Residual saturation for liquid phase [—].

• M <float> van Genuchten \(m\) as in \((m = 1 - 1/n)\) [—].

• SMOOTH Applies polynomial smoothing to relative permeability or saturation function. Strongly recommended for the Brooks-Corey saturation function if cells in the domain transition from saturated to variably-saturated conditions.

Optional Cards:

• PHASE <string> Phase to which the permeability function applies [LIQUID, GAS]

• MAX_CAPILLARY_PRESSURE <float> Cut off for maximum capillary pressure (default = \(10^9\) [Pa]).

• POWER <float> Placeholder. Currently not used.

Examples:

```
CHARACTERISTIC_CURVES cc1
SATURATION_FUNCTION VAN_GENUCHTEN
  LIQUID_RESIDUAL_SATURATION 0.d0
  M 0.5d0
  ALPHA 1.d-4
  MAX_CAPILLARY_PRESSURE 1.d6
/
PERMEABILITY_FUNCTION MUALEM
  PHASE LIQUID
  LIQUID_RESIDUAL_SATURATION 0.d0
  M 0.5d0
```

PERMEABILITY_FUNCTION MUALEM_VG_GAS
PHASE GAS
LIQUID_RESIDUAL_SATURATION 0.d0
GAS_RESIDUAL_SATURATION 1.d-40
M 0.5d0
/
/

CHARACTERISTIC_CURVES cc2
SATURATION_FUNCTION BROOKS_COREY
LIQUID_RESIDUAL_SATURATION 0.2d0
LAMBDA 0.7d0
ALPHA 9.869d-6
MAX_CAPILLARY_PRESSURE 1.d8
SMOOTH
/
PERMEABILITY_FUNCTION BURDINE
PHASE LIQUID
LIQUID_RESIDUAL_SATURATION 0.2d0
LAMBDA 0.7d0
/
PERMEABILITY_FUNCTION BURDINE_BC_GAS
PHASE GAS
LIQUID_RESIDUAL_SATURATION 0.2d0
GAS_RESIDUAL_SATURATION 1.d-5
LAMBDA 0.7d0
/
4.6 Keyword: CHECKPOINT

Description: Checkpoint files enable the restart of a simulation at any discrete point in simulation where a checkpoint file has been printed. When the CHECKPOINT card is included in the input deck, checkpoint files are printed every N time steps, where N is the checkpoint frequency, and at the end of the simulation, should the simulation finish or the be shut down properly mid-simulation using the WALL_CLOCK_STOP card. Checkpoint files are named pflotran.chkN, where N is the number of the timestep when the checkpoint file was printed. A file named restart.chk will also be written when PFLOTRAN properly terminates execution. One use this file to pick up from where the simulation stopped by increasing the final time.

Checkpointing can be used to start from an initial steady-state solution, but note that porosity and permeability are checkpointed as there are scenarios where they can change over time. To override this behavior add: OVERWRITE_RESTART_FLOW_PARAMS to the input file to set porosity/permeability to their read-in values.

Input:

CHECKPOINT <checkpoint_frequency>

Explanation:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECKPOINT</td>
<td>toggles on checkpointing</td>
</tr>
<tr>
<td>checkpoint_frequency</td>
<td>frequency at which checkpoint files are printed &lt;integer&gt;</td>
</tr>
</tbody>
</table>

Examples:

CHECKPOINT 1000

CHECKPOINT 5

↩ Return to List of Keywords
4.7 Keyword: CHEMISTRY

Description: The CHEMISTRY keyword invokes the reactive transport mode and provides input for primary species, secondary species, minerals, gases, colloids and colloid-facilitated transport, and sorption including ion exchange and surface complexation. Mineral reactions are described through a kinetic rate law based on transition state theory and surface complexation reactions may involve equilibrium, kinetic (reversible or irreversible) or a multirate formulation.

Input:

CHEMISTRY

PRIMARY_SPECIES
   Species Name
END [/ .]

SECONDARY_SPECIES
   Species Name
END [/ .]

REDOX_SPECIES
   Species Name
END [/ .]

GAS_SPECIES
   Species Name
END [/ .]

MINERALS
   Mineral Name
END [/ .]

COLLOIDS
   Colloid Name Mobile_Fraction[—]
END [/ .]

MINERAL_KINETICS
   Mineral Name <Char>
   RATE_CONSTANT <float> [mol/m²/s], If <float> < 0, <float> =

< 37 >
ACTIVATION_ENERGY  <float> [kJ/mol/K],  Referenced to rate constant at 25°C.

AFFINITY_THRESHOLD  <float> [—]
AFFINITY_POWER  <float> [—]

TEMPKINS_CONSTANT  <float> [—]

SURFACE_AREA_POROSITY_POWER  <float> [—]  [see Eqn.(10-123)]
SURFACE_AREA_VOL_FRAC_POWER  <float> [—]  [see Eqn.(10-123)]

RATE_LIMITER  <float> [mol/m²/s]  [see Eqn.(10-113)]

IRREVERSIBLE

ARMOR_MINERAL  <Char>  [See Eqn.(10-131)]
ARMOR_PWR  <float> [—]  [See Eqn.(10-131)]
ARMOR_CRIT_VOL_FRAC  <float> [—]  [See Eqn.(10-131)]

PREFACTOR

RATE_CONSTANT  <float> [mol/m²/s],  If <float> < 0, <float> = 10<float>

ACTIVATION_ENERGY  <float> [kJ/mol/K]
PREFACTOR_SPECIES  <Char>

ALPHA  <float> [—]
BETA  <float> [—]
ATTENUATION_COEF  <float>

END [/ .]
END [/ .]
RATE, RATES <float> [1/s] (Continuation line `\`)  
MULTIRATE_SCALE_FACTOR <float> [--]  

MINERAL Mineral Name  
SITE Name Site Density [mol/m³]  
COMPLEXES  
Complex Name  
END [/ .]  

COMPLEX_KINETICS  
COMPLEX name  
FORWARD_RATE_CONSTANT <float> [1/s]  
BACKWARD_RATE_CONSTANT <float> [1/s] If value < −999 calculate backward rate constant from expression: 
\[ k_b = K_{eq}k_f, \]
where \( K_{eq} \) is the corresponding equilibrium constant of the reaction.  
END [/ .]  
END [/ .]  
END [/ .]  

COLLOID Name  
SITE Name Site Density [mol/m³]  
COMPLEXES  
Surface_Complex Name  
END [/ .]  
END [/ .]  
END [/ .]  

ION_EXCHANGE_RXN  
MINERAL Mineral Name  
CEC <float> [mol/m³]  
CATIONS  
Name Selectivity_Coefficient  
END [/ .]  
END [/ .]  
END [/ .]  

ISOTHERM_REACTIONS  
Species_Name

< 39 >
TYPE LINEAR, LANGMUIR, FREUNDLICH

DISTRIBUTION_COEFF, KD <float> [kg water/m³ bulk] [see Eqn.(10-177)]
LANGMUIR_B <float> [—] [see Eqn.(10-178)]
FREUNDLICH_N <float> [—] [see Eqn.(10-179)]

END [/ .]
END [/ .]

JUMPSTART_KINETIC_SORPTION
NO_CHECKPOINT_KINETIC_SORPTION
NO_RESTART_KINETIC_SORPTION

END [/ .]

OPERATOR_SPLITTING  Toggles operator-splitting mode (Default implicit)

GEOTHERMAL_HPT  Use high pressure and temperature thermodynamic database

DATABASE  Path/Database_Name

LOG_FORMULATION
NO_CHECKPOINT_ACT_COEFS
ACTIVITY_COEFFICIENTS [LAG, NEWTON, TIMESTEP, NEWTON_ITERATION]
ACTIVITY_H2O, ACTIVITY_WATER
MOLAL, MOLALITY
NO_BDOT
UPDATE_POROSITY  [see Eqn.(10-118)]
UPDATE_TORTUOSITY  [see Eqn.(10-122)]
UPDATE_PERMEABILITY  (Must activate update_porosity, see Eqn.(10-119).)
UPDATE_MINERAL_SURFACE_AREA  [see Eqn.(10-123)]
(Must set SURFACE_AREA_VOL_FRAC_POWER)
UPDATE_MNRL_SURF_AREA_WITH_POR  [see Eqn.(10-123)]
(Must set SURFACE_AREA_POROSITY_POWER)
MINIMUM_POROSITY <float>
UPDATE_ARMOR_MINERAL_SURFACE  [See Eqn.(10-131)]
MAX_DLNC  (Default 5)

OUTPUT

MOLALITY
MOLARITY

< 40 >
ACTIVITY_COEFFICIENTS

ALL Output all primary species
OFF Turn off printout
Species Name Primary or secondary species
FREE_ION Output free-ion primary species concentrations
TOTAL Output total primary species concentrations
PRIMARY_SPECIES Output primary species including pH
SECONDARY_SPECIES Output all secondary species concentrations
GASES Output gas species
MINERALS Output all kinetic mineral volume fractions and reaction rates
Mineral Name Output mineral saturation index
IMMOBILE
pH Output pH
pe Output pe
Eh Output Eh [V]
O2 Output log $f_{O_2}$ [$f_{O_2}$ in bars]
TOTAL_SORBED
TOTAL_SORBED_MOBILE
COLLOIDS
KD
TOTAL_SORBED
TOTAL_BULK
TOTAL_SORBED_MOBILE
AGE Output mean solute or water age
SITE_DENSITY

END [/ .]

MAX_RELATIVE_CHANGE_TOLERANCE Relative speciation tolerance (Default 1.d-12)

MAX_RESIDUAL_TOLERANCE Speciation tolerance (Default 1.d-12)

END [/ .]

Example: Output total and free-ion primary species concentrations.

```
OUTPUT
```
... 
ALL 
FREE_ION 
TOTAL 
... 
/

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary_Species</td>
<td>List of primary species that fully describe the chemical composition of the fluid. The set of primary species must form an independent set of species in terms of which all homogeneous aqueous equilibrium reactions can be expressed.</td>
</tr>
<tr>
<td>Secondary_Species</td>
<td>List of aqueous species in equilibrium with primary species.</td>
</tr>
<tr>
<td>Gas_Species</td>
<td>List of gas species.</td>
</tr>
<tr>
<td>Sorption</td>
<td>Surface complexation, ion exchange and specified isotherm.</td>
</tr>
</tbody>
</table>

```
SORPTION 
  ISOTHERM_REACTIONS 
    Tracer 
      TYPE LINEAR 
      DISTRIBUTION_COEFFICIENT 500. ! kg water/m^3 bulk 
    / 
    Tracer_Age 
      TYPE LINEAR 
      DISTRIBUTION_COEFFICIENT 500. ! kg water/m^3 bulk 
    / 
    / 
```

**Output** To print secondary aqueous complex concentrations, either add the names of the secondary species of interest or the keyword “SECONDARY_SPECIES” for all secondary species to the CHEMISTRY OUTPUT card:
CHEMISTRY
...
OUTPUT
...
CO2(aq) ! where co2(aq) is a secondary species
SECONDARY_SPECIES
...
/
...
END

Deactivate redox equilibrium reactions for the listed redox pairs:

REDOX_SPECIES
   Fe++
   Fe+++
   SO4--
   HS-
   Acetate-
   Ethanol(aq)
/

By default, if ALL or MINERALS are listed under CHEMISTRY OUTPUT, the volume fractions and rates of kinetic minerals are printed. To print out the saturation indices of minerals listed under the MINERAL keyword, add the name of the mineral to the OUTPUT specification. In other words, if the following appears in the input deck:

CHEMISTRY
...
MINERALS
   Quartz
   Calcite
   Gibbsite
/
MINERAL_KINETICS
   Calcite
       RATE_CONSTANT 1.d-12 mol/cm^2-sec
/

<43>
volume fraction and reaction rate are printed for both calcite and gibbsite and saturation indices are printed for gibbsite. Saturation indices are not printed when only ALL or MINERALS are specified since an input deck. Outputting tens to hundreds of minerals, most of which are simply considered, but not modeled through a kinetic precipitation-dissolution reaction, would overwhelm the output.

Examples:

**PREFACCTOR (see Eqn. (10-112))**:

```
MINERAL_KINETICS
Quartz
   RATE_CONSTANT -17.99d0 mol/cm^2-sec
   ACTIVATION_ENERGY 87.7d0
/
Albite
   PREFACCTOR
   RATE_CONSTANT -16.56d0 mol/cm^2-sec
   ACTIVATION_ENERGY 69.8d0
/
PREFACCTOR
   RATE_CONSTANT -14.16d0 mol/cm^2-sec
   ACTIVATION_ENERGY 65.0d0
   PREFACCTOR_SPECIES H+
   ALPHA 0.457d0
   ! BETA
   ! ATTENUATION_COEF
/
/PREFACCTOR
```
RATE_CONSTANT -19.6d0 mol/cm^2-sec
ACTIVATION_ENERGY 71.0d0
PREFACTOR_SPECIES H+
   ALPHA -0.572d0
! BETA
! ATTENUATION_COEF
/
/
/
END
4.8 Keyword: COMPUTE_STATISTICS

Description: COMPUTE_STATISTICS enables the calculation statistical analysis of flow velocities during a simulation. The average, maximum, minimum, and standard deviations velocities are computed.

Input:

COMPUTE_STATISTICS {compute_statistics = .true.}

Explanation:

Example:

COMPUTE_STATISTICS

↩ Return to List of Keywords
4.9 Keyword: CO2_DATABASE

Description: The keyword CO2_DATABASE is for specifying the path to CO2 database file providing a lookup table for CO2 fluid properties density, viscosity, fugacity and fugacity coefficient derived from the Span-Wagner (1996) EOS. The temperature-pressure range in the default lookup table is \( T : 0 \rightarrow 375^\circ C \) and \( P : 0.01 \rightarrow 1250 \text{ bars} \). This keyword is required for MPHASE, FLASH2 and IMMIS modes.

Input:

\[
\text{CO2_DATABASE \{Path/Database_Name\}} \quad \text{The default database file is} \quad ./pflotran-dev/database/co2_data0.dat
\]

↩ Return to List of Keywords
4.10 Keyword: CONSTRAINT

Description: The keyword CONSTRAINT sets up fluid compositions based on various constraint conditions chosen by the user. Use SECONDARY_CONSTRAINT for constraining secondary continua initial concentrations.

Input:

CONSTRAINT (SECONDARY_CONSTRAINT) constraint_name

CONC, CONCENTRATIONS

Primary Species Name, Concentration_Value, Constraint, Name (mineral, gas)

END [/ .]

MNRL, MINERALS

mineral_name, volume_fraction [—], surface_area [m\(^{-1}\)]

END [/ .]

END [/ .]

Explanation:

The variable Constraint is chosen from the following list:

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F, FREE</td>
<td>Free ion/species concentration</td>
</tr>
<tr>
<td>T, TOTAL</td>
<td>Total aqueous concentration</td>
</tr>
<tr>
<td>TOTAL_SORB</td>
<td>Total aqueous and sorbed concentration</td>
</tr>
<tr>
<td>P, pH</td>
<td>pH</td>
</tr>
<tr>
<td>L, LOG</td>
<td>Log base 10 of free ion concentration</td>
</tr>
<tr>
<td>M, MINERAL, MNRL</td>
<td>Mineral equilibrium constraint</td>
</tr>
<tr>
<td>G, GAS</td>
<td>Gaseous species constraint [bars]</td>
</tr>
<tr>
<td>SC, CONSTRAINT_SUPERCRIT_CO2</td>
<td>Supercritical CO(_2) EOS</td>
</tr>
<tr>
<td>Z, CHG</td>
<td>Charge balance</td>
</tr>
</tbody>
</table>

Example:

CONSTRAINT initial

CONCENTRATIONS

H+ 7.3 P
<table>
<thead>
<tr>
<th>Species</th>
<th>Concentration</th>
<th>Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>O2(aq)</td>
<td>1.78132e-4 T</td>
<td></td>
</tr>
<tr>
<td>Al+++</td>
<td>1.e-9 M</td>
<td>K-Feldspar</td>
</tr>
<tr>
<td>Ca++</td>
<td>1.20644e-3 M</td>
<td>Calcite</td>
</tr>
<tr>
<td>Cu++</td>
<td>1.e-6 T</td>
<td></td>
</tr>
<tr>
<td>Fe++</td>
<td>1.e-9 M</td>
<td>Ferrihydrite</td>
</tr>
<tr>
<td>Mg++</td>
<td>5.09772e-4 T</td>
<td></td>
</tr>
<tr>
<td>UO2++</td>
<td>2.34845e-7 T</td>
<td></td>
</tr>
<tr>
<td>K+</td>
<td>1.54789e-4 T</td>
<td></td>
</tr>
<tr>
<td>Na+</td>
<td>2.03498e-3 T</td>
<td></td>
</tr>
<tr>
<td>HCO3-</td>
<td>2.57305e-3 T</td>
<td></td>
</tr>
<tr>
<td>Cl-</td>
<td>6.97741e-4 T</td>
<td></td>
</tr>
<tr>
<td>F-</td>
<td>2.09491e-5 T</td>
<td></td>
</tr>
<tr>
<td>HPO4--</td>
<td>1.e-6 T</td>
<td></td>
</tr>
<tr>
<td>NO3-</td>
<td>4.69979e-3 T</td>
<td></td>
</tr>
<tr>
<td>SO4--</td>
<td>6.37961e-4 T</td>
<td></td>
</tr>
<tr>
<td>SiO2(aq)</td>
<td>5.36989e-4 T</td>
<td></td>
</tr>
<tr>
<td>Tracer</td>
<td>2.34845e-7 F</td>
<td></td>
</tr>
</tbody>
</table>

MINERALS

<table>
<thead>
<tr>
<th>Mineral</th>
<th>Volume Fraction</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quartz</td>
<td>0.35</td>
<td>1. cm^2/cm^3</td>
</tr>
<tr>
<td>Calcite</td>
<td>0</td>
<td>1. cm^2/cm^3</td>
</tr>
<tr>
<td>Metatorbernite</td>
<td>0</td>
<td>1. cm^2/cm^3</td>
</tr>
</tbody>
</table>

BEGIN

Second Constraint

SECONDARY_CONSTRAINT sec

SECONDARY_CONSTRAINT sec

CONCENTRATIONS

<table>
<thead>
<tr>
<th>Species</th>
<th>Concentration</th>
<th>Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ca++</td>
<td>1.e-3 M</td>
<td>Calcite</td>
</tr>
<tr>
<td>H+</td>
<td>7. pH</td>
<td></td>
</tr>
<tr>
<td>SO4--</td>
<td>1.e-3 M</td>
<td>Gypsum</td>
</tr>
<tr>
<td>HCO3-</td>
<td>-3.5 G</td>
<td>CO2(g)</td>
</tr>
<tr>
<td>Cl-</td>
<td>1.e-3 Z</td>
<td></td>
</tr>
<tr>
<td>SiO2(aq)</td>
<td>1.e-4 M</td>
<td>Quartz</td>
</tr>
<tr>
<td>Tracer</td>
<td>1.d-8 F</td>
<td></td>
</tr>
</tbody>
</table>

MINERALS

<table>
<thead>
<tr>
<th>Mineral</th>
<th>Volume Fraction</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quartz</td>
<td>0.019175</td>
<td>1.d2</td>
</tr>
<tr>
<td>Calcite</td>
<td>0.730825</td>
<td>1.d2</td>
</tr>
<tr>
<td>Gypsum</td>
<td>0.0</td>
<td>1.d2</td>
</tr>
</tbody>
</table>
4.11 Keyword: DATASET

Description: Specifies a data set to be associated with parameters sets in the model.

DATASET <string>: Opens the card block with the name of the data set in the string.

NAME <string>: Name of the data set if not included with DATASET card. Note: this string overwrites the name specified with the DATASET.

FILENAME <string>: Name of file containing data.

TYPE <string>: Reserved for future application where the data set can be a single scalar or vector value or a functional relationship. The TYPE is currently fixed at HETEROGENEOUS by default. Other types report an unsupported error message.

REALIZATION_DEPENDENT: Toggle that causes PFLOTRAN to load the data set based on the realization ID. For instance, if the data set is tied to PERMEABILITY within a MATERIAL_PROPERTY and the realization ID is 99, PFLOTRAN searches for an HDF5 data set labeled "PERMEABILITY99". For POROSITY, "POROSITY99".

Examples:

Reading heterogeneous permeability and porosity for the Hanford unit for realization ID = 99. The name of the data sets within the HDF5 file are PERMEABILITY99 and POROSITY99, respectively.

```
DATASET perm
  FILENAME hanford_unit.h5
  REALIZATION_DEPENDENT
END

DATASET poros
  FILENAME hanford_unit.h5
  REALIZATION_DEPENDENT
END

MATERIAL_PROPERTY hanford_unit
  ...
  POROSITY DATASET poros
  PERMEABILITY
  ...
  DATASET perm
  ...
```
/  
...  
END
4.12 Keyword: DBASE_FILENAME

Keyword DBASE_FILENAME allows the user to define input parameters through an external database stored in an ASCII text or HDF5 file. These parameters can be realization dependent where an array of values is provided and indexed by the realization id.

**DBASE_FILENAME** <string> The path/filename of the external database to be employed.

**DBASE_VALUE** <string1> [:: <string2>] The names of the parameters to be read from the database file. This card combination may be entered anywhere a double precision value is read from the input file. The second string <string2> must be preceded by a double colon without spaces and is optional. See example below.

**Examples:**

```plaintext
DBASE_FILENAME 543_dbase.h5
...
MATERIAL_PROPERTY soil1
...
  POROSITY DBASE_VALUE POROSITY::SOIL1
END
MATERIAL_PROPERTY soil4
...
  PERM_X DBASE_VALUE PERMEABILITY_X::SOIL4
END
...
CHARACTERISTIC_CURVES sf3
...
  ALPHA DBASE_VALUE ALPHA-sf3
END
...
REGION north
  FACE NORTH
  COORDINATES
    0.d0 46.d0 0.d0
    60.d0 46.d0 DBASE_VALUE NORTH_MAX_Z
  /
END
...
CONSTRAINT river_water
  CONCENTRATIONS
    Tracer 1.e-3 F
    Tracer2 DBASE_VALUE RIVER_TRACER2 F
  /
END
```

< 52 >
Values are assigned based on the database stored in `543_dbase.h5` and the command line argument `-realization_id 5` (note the zero-based index of 4 in the HDF5 file is actually the 5th realization). An example script for creating the database file may be found in `PFLOTRAN_DIR/src/python/dbase_creator.py`. 
4.13 Keyword: DEBUG

Input:

DEBUG

PRINT_SOLUTION [VECVIEW_SOLUTION, VIEW_SOLUTION]
PRINT_RESIDUAL [VECVIEW_RESIDUAL, VIEW_RESIDUAL]
PRINT_JACOBIAN [MATVIEW_JACOBIAN, VIEW_JACOBIAN]
PRINT_JACOBIAN_NORM [NORM_JACOBIAN]
PRINT_COUPLERS [PRINT_COUPLER]
PRINT_JACOBIAN_DETAILED [MATVIEW_JACOBIAN_DETAILED,
                           VIEW_JACOBIAN_DETAILED]
PRINT_NUMERICAL_DERIVATIVES [VIEW_NUMERICAL_DERIVATIVES]

END [/ .]

Explanation:

Examples:

DEBUG
    PRINT_RESIDUAL
    PRINT_JACOBIAN
END

⇦ Return to List of Keywords
4.14  Keyword: EOS

The EOS keyword defines an equation of state for a simulated fluid.

EOS  <string> [WATER, GAS]  Specifies the fluid for which EOS applies.

Optional Input:

  DENSITY  <string> <optional parameters>

    DENSITY CONSTANT  <float>

    DENSITY EXPONENTIAL  <float> <float> <float>  (ref. density [rho0], ref. pressure [p0], compressibility) \( \rho = \rho_0 e^{\kappa(p-p_0)} \).

    DENSITY DEFAULT  Default water EOS based on International Formulation Committee of the Sixth International Conference on Properties of Steam (1967)

  ENTHALPY  <string> <optional parameters>

  VISCOSITY  CONSTANT  <float>

END  [/ .]

Examples:

EOS WATER
  DENSITY EXPONENTIAL 997.16d0 101325.d0 1.d-8
END
EOS WATER
  DENSITY CONSTANT 997.16d0
  ENTHALPY CONSTANT 1.8890d0
  VISCOSITY CONSTANT 8.904156d-4
END

⇦ Return to List of Keywords
4.15 Keyword: FLOW_CONDITION

Description: The FLOW_CONDITION keyword specifies scalar or vector data sets to be associated with a given boundary or initial condition. For instance, to specify a hydrostatic boundary condition, the use would specify a condition with a pressure associated with a point in space (i.e. datum) in space and a gradient, both vector quantities. Note that in the case of a hydrostatic boundary condition, the vertical gradient specified in the input deck must be zero in order to enable the hydrostatic pressure calculation. Otherwise, the specified vertical gradient overrides the hydrostatic pressure. Transient pressures, temperatures, concentrations, datums, gradients, etc. are specified using the FILE filename combination for the name of the data set.

Input:

FLOW_CONDITION  flow_condition_name

UNITs <char> (not currently supported)
<char> is one of the following entries:
  s, min, h (hr), d, day, w, week, mo, month, y (yr) (time)
  mm, cm, m, dm, km (length)
  kg/s, kg/yr (rate)
  Pa, KPa (pressure)
  m/s, m/yr (velocity)
  C, K (temperature)
  M, mol/L (concentration)
  KJ/mol (enthalpy)

CYCLIC
INTERPOLATION
  step
  linear

SYNC_TIMESTEP_WITH_UPDATE

TYPE
  PRESSURE [dirichlet, hydrostatic, zero_gradient, conductance, seepage]
  RATE [mass_rate, volumetric_rate, scaled_volumetric_rate]: specifies an injection/extraction rate in mass [kg/s], volume [m³/s], and a volumetric injection/extraction rate [m³/s] that is scaled across a well screen, weighted as a function of the interfacial area and permeability of neighboring cells (in x, y).
  FLUX [dirichlet, neumann, mass_rate, hydrostatic, conductance, zero_gradient, production_well, seepage, volumetric, volumetric_rate, equilibrium]
  TEMPERATURE [dirichlet, hydrostatic, zero_gradient]
  CONCENTRATION [dirichlet, hydrostatic, zero_gradient]
SATURATION [dirichlet]

ENTHALPY (H) [dirichlet, hydrostatic, zero_gradient]

END [/ .]

TIME (not currently supported)

IPHASE <int>

DATUM

\[ x \ y \ z \]

FILE file_name

GRADIENT, GRAD

PRES, PRESS, PRESSURE

\[ d_x \ d_y \ d_z \]

FILE file_name

FLUX

TEMP, TEMPERATURE

CONC, CONCENTRATION

H, ENTHALPY

END [/ .]

TEMPERATURE, TEMP <float>

ENTHALPY, H <float>

PRESSURE, PRES, PRESS <float>

RATE <float>

FLUX, VELOCITY, VEL <float>

CONC, CONCENTRATION <float>

SAT, SATURATION <float>

CONDUCTANCE <float>

END [/ .]

Explanation:

<table>
<thead>
<tr>
<th>IPHASE</th>
<th>Phases present</th>
<th>CO\textsubscript{2} concentration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H\textsubscript{2}O</td>
<td>(X\textsubscript{H\textsubscript{2}O}^{\text{CO}_2})</td>
</tr>
<tr>
<td>2</td>
<td>SC CO\textsubscript{2}</td>
<td>(X\textsubscript{SCCO}_2^{\text{CO}_2})</td>
</tr>
<tr>
<td>3</td>
<td>H\textsubscript{2}O – SC CO\textsubscript{2}</td>
<td>((X\textsubscript{H\textsubscript{2}O}^{\text{SCCO}<em>2})</em>{eq})</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>FLOW/TRANSPORT_CONDITION</td>
<td>Initiates a condition entry and defines its name</td>
<td></td>
</tr>
<tr>
<td>CYCLIC</td>
<td>Instructs PFLOTRAN to cycle the transient data set should the simulation time exceed the last time in the data set</td>
<td></td>
</tr>
<tr>
<td>INTERPOLATION</td>
<td>Defines the method for interpolating between data set times</td>
<td></td>
</tr>
<tr>
<td>SYNC_TIMESTEP_WITH_UPDATE</td>
<td>Synchronizes time step with waypoints</td>
<td></td>
</tr>
<tr>
<td>DATUM</td>
<td>Location is space where prescribed scalar (e.g. pressure, temperature concentration, etc.) is defined</td>
<td></td>
</tr>
<tr>
<td>TYPE</td>
<td>Specifies condition type</td>
<td></td>
</tr>
<tr>
<td>PRESSURE</td>
<td>Specifies pressure condition type</td>
<td></td>
</tr>
<tr>
<td>TEMPERATURE</td>
<td>Specifies temperature condition type</td>
<td></td>
</tr>
<tr>
<td>CONCENTRATION</td>
<td>Specifies the type of concentration condition</td>
<td></td>
</tr>
<tr>
<td>SATURATION</td>
<td>Specifies saturation condition type</td>
<td></td>
</tr>
<tr>
<td>ENTHALPY</td>
<td>Specifies enthalpy condition type</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td>Terminates condition entry</td>
<td></td>
</tr>
<tr>
<td>GRADIENT</td>
<td>Gradient of the scalar field in 3D space</td>
<td></td>
</tr>
<tr>
<td>PRESSURE</td>
<td>Pressure gradient in ( x ), ( y ), and ( z )-directions</td>
<td></td>
</tr>
<tr>
<td>TEMPERATURE</td>
<td>Temperature gradient in ( x ), ( y ), and ( z )-directions</td>
<td></td>
</tr>
<tr>
<td>CONCENTRATION</td>
<td>Concentration gradient in ( x ), ( y ), and ( z )-directions</td>
<td></td>
</tr>
<tr>
<td>ENTHALPY</td>
<td>Enthalpy gradient in ( x ), ( y ), and ( z )-directions</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td>Terminates gradient entry</td>
<td></td>
</tr>
<tr>
<td>PRESSURE</td>
<td>Absolute fluid pressure at the datum</td>
<td></td>
</tr>
<tr>
<td>FLUX</td>
<td>Darcy velocity of fluid defining flux across specified boundary</td>
<td></td>
</tr>
<tr>
<td>TEMPERATURE</td>
<td>Temperature in °C at datum</td>
<td></td>
</tr>
<tr>
<td>CONCENTRATION</td>
<td>Solute concentration at datum</td>
<td></td>
</tr>
<tr>
<td>SATURATION</td>
<td>Solute saturation at datum</td>
<td></td>
</tr>
<tr>
<td>ENTHALPY</td>
<td>Enthalpy at datum</td>
<td></td>
</tr>
<tr>
<td>CONSTRAINT_LIST</td>
<td>Specifies list of concentration constraints for solute transport</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td>Terminates condition entry</td>
<td></td>
</tr>
</tbody>
</table>
Examples:

FLOW_CONDITION initial
    TYPE
        PRESSURE hydrostatic
    /
        PRESSURE 1956741.84 ! 200 meter piezometric head (200*997.32*9.81)
    END

FLOW_CONDITION source
    TYPE
        RATE volumetric_rate
    /
        RATE 10. m^3/hr
    END

TRANSPORT_CONDITION initial
    TYPE zero_gradient
    CONSTRAINT_LIST
        0.d0 initial
    /
    END

TRANSPORT_CONDITION source
    TYPE dirichlet
    CONSTRAINT_LIST
        0.d0 well
    /
    END

FLOW_CONDITION East
    TYPE
        :PRESSURE seepage
        PRESSURE conductance
    /
    CYCLIC
    DATUM file ../../river_scope3.datum
    GRADIENT
        PRESSURE file ../../river_scope3.gradient
    /
    CONDUCTANCE 1.d-12
    PRESSURE 101325.d0
    END
4.16  Keyword: FLUID_PROPERTY

Description: Assign diffusion coefficients and temperature dependence.

Input:

FLUID_PROPERTY

    PHASE  <name>  (LIQUID_PHASE, GAS_PHASE) [Default: LIQUID_PHASE]
    DIFFUSION_COEFFICIENT  <float>  [m²/s] [Default: 0 × 10⁻⁹ m²/s]
    DIFFUSION_ACTIVATION_ENERGY  <float>  [kJ/mol] [Default: 0 kJ/mol]

END [/ .]

Explanation: Read in reference diffusion coefficient $D_m^0$ and diffusion activation energy $A_D$. Temperature dependence of diffusion coefficient is calculated from the expression:

$$D_m(T) = D_m^0 \exp \left[ \frac{A_D}{R} \left( \frac{1}{T_0} - \frac{1}{T} \right) \right],$$

(4-1)

where $D_m^0$ is the reference diffusion coefficient at temperature $T_0 = 25°C$ and $A_D$ denotes the diffusion activation energy.

Example:

```
FLUID_PROPERTY
    DIFFUSION_COEFFICIENT 1.d-9  ! m^2/s
    DIFFUSION_ACTIVATION_ENERGY 12.6  ! kJ/K/mol
/
```

$\leftrightarrow$ Return to List of Keywords
4.17 Keyword: GEOMECHANICS

Description: This keyword is required when using geomechanics. All the geomechanics part should go in this card.

GEOMECHANICS

Required Input Parameters: All the geomechanics keywords should go here.

END [/ .]
4.18 Keyword: GEOMECHANICS_GRID

Description: The grid type and the format for geomechanics is specified here. Only unstructured grids can be read. If you need to use a structured grid, generate the grid as an unstructured grid in the implicit format (see GRID keyword for details of the format below).

GEOMECHANICS_GRID

Required Input Parameters:

- **TYPE <type>:** Grid type (unstructured)
- **FILE <filename>:** Name of file containing grid information (unstructured only)

Optional Input Parameters:

- **GRAVITY <# # #>:** Specifies gravity vector for geomechanics calculations (or specific body force) [Default: 0 0 -9.81 m/s²]

END [/ .]
4.19 Keyword: GEOMECHANICS_OUTPUT

Description: This keyword is required for output of geomechanics data. The geomechanics data is saved in separate set of files (unlike flow and transport). The filenames of these files have the word -geomech- included in them. Uses the same keywords as OUTPUT.
4.20  Keyword: GEOMECHANICS_MATERIAL_PROPERTY

Description:  Specifies geomechanics material properties to be associated with a geomechanics region in the problem domain.

Input:

GEOMECHANICS_MATERIAL_PROPERTY  <char>

   ID  <int>
   YOUNGS_MODULUS  <float> [Pa]
   POISSONS_RATIO  <float>
   ROCK_DENSITY  <float> [kg/m^3]
   BIOT_COEFFICIENT  <float>
   THERMAL_EXPANSION_COEFFICIENT  <float> [Pa/K]

END [/ .]

Explanation:  The Young’s modulus and Poisson’s ratio (for the linear elastic model), rock density (used in body force calculation), Biot’s coefficient and thermal expansion coefficient can be set here.

Example:

MATERIALPROPERTY rock
   ID 1
   YOUNGS_MODULUS 1.d10
   POISSONS_RATIO 0.3
   ROCK_DENSITY 2200
   BIOT_COEFFICIENT 1.0
   THERMAL_EXPANSION_COEFFICIENT 1.d-5
END
4.21 Keyword: GEOMECHANICS_REGION

**Description:** The GEOMECHANICS_REGION keyword defines a set of geomechanics finite element grid vertices. The GEOMECHANICS_REGION name can then be used to link this set of vertices to geomechanics material properties, strata and boundary conditions. The list of vertices can be read from an ASCII file by using the keyword FILE under the GEOMECHANICS_REGION card.

**Examples:**

```
GEOMECHANICS_REGION top
   FILE top.vset
END
```

↩ Return to List of Keywords
4.22  Keyword: GEOMECHANICS_CONDITION

Description: Condition coupler between regions and geomechanics boundary conditions. Since the geomechanics is solved in a quasi-steady manner, initial conditions are not needed.

Input:

GEOMECHANICS_CONDITION  geomechanics_condition_name

    TYPE
        DISPLACEMENT_X [dirichlet]
        DISPLACEMENT_Y [dirichlet]
        DISPLACEMENT_Z [dirichlet]
        FORCE_X [dirichlet]
        FORCE_Y [dirichlet]
        FORCE_Z [dirichlet]
    END [/ .]

    DISPLACEMENT_X  <float>
    DISPLACEMENT_Y  <float>
    DISPLACEMENT_Z  <float>
    FORCE_X  <float>
    FORCE_Y  <float>
    FORCE_Z  <float>

    END [/ .]

Explanation:
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEOMECHANICS_CONDITION</td>
<td>Initiates a geomechanics condition entry and defines its name</td>
</tr>
<tr>
<td>TYPE</td>
<td>Specifies condition type</td>
</tr>
<tr>
<td>DISPLACEMENT_X</td>
<td>Specifies ( x )-displacement condition type</td>
</tr>
<tr>
<td>DISPLACEMENT_Y</td>
<td>Specifies ( y )-displacement condition type</td>
</tr>
<tr>
<td>DISPLACEMENT_Z</td>
<td>Specifies ( z )-displacement condition type</td>
</tr>
<tr>
<td>FORCE_X</td>
<td>Specifies ( x )-force condition type</td>
</tr>
<tr>
<td>FORCE_Y</td>
<td>Specifies ( y )-force condition type</td>
</tr>
<tr>
<td>FORCE_Z</td>
<td>Specifies ( z )-force condition type</td>
</tr>
<tr>
<td>END</td>
<td>Terminates type entry</td>
</tr>
<tr>
<td>DISPLACEMENT_X</td>
<td>Displacement in the ( x )-direction</td>
</tr>
<tr>
<td>DISPLACEMENT_Y</td>
<td>Displacement in the ( y )-direction</td>
</tr>
<tr>
<td>DISPLACEMENT_Z</td>
<td>Displacement in the ( z )-direction</td>
</tr>
<tr>
<td>FORCE_X</td>
<td>Force in the ( x )-direction</td>
</tr>
<tr>
<td>FORCE_Y</td>
<td>Force in the ( y )-direction</td>
</tr>
<tr>
<td>FORCE_Z</td>
<td>Force in the ( z )-direction</td>
</tr>
<tr>
<td>END</td>
<td>Terminates condition entry</td>
</tr>
</tbody>
</table>

**Example:**

```plaintext
GEOMECHANICS_CONDITION
   TYPE
       DISPLACEMENT_Z dirichlet
   /
   DISPLACEMENT_Z 0.d0
END
```

↩ Return to List of Keywords
4.23  Keyword: GEOMECHANICS_BOUNDARY_CONDITION

Description: The GEOMECHANICS_BOUNDARY_CONDITION keyword couples condition specified under the GEOMECHANICS_CONDITION keyword to a REGION in the problem domain. The use of this keyword enables the use/reuse of geomechanics conditions and regions within multiple geomechanics boundary conditions the input deck.

Input:

GEOMECHANICS_BOUNDARY_CONDITION geomechanics_boundary_condition_name

   GEOMECHANICS_CONDITION GEOMECHANICS_condition_name
   REGION region_name

END [/ .]

Explanation:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEOMECHANICS_BOUNDARY_CONDITION</td>
<td>Defines the beginning of a geomechanics boundary condition entry and the name of the geomechanics boundary condition.</td>
</tr>
<tr>
<td>GEOMECHANICS_CONDITION</td>
<td>Defines the name of the geomechanics condition to be linked to this geomechanics boundary condition.</td>
</tr>
<tr>
<td>REGION</td>
<td>Defines the name of the region to which the conditions are linked.</td>
</tr>
<tr>
<td>END</td>
<td>Terminates the geomechanics boundary condition entry.</td>
</tr>
</tbody>
</table>

Example:

GEOMECHANICS_BOUNDARY_CONDITION bottom
GEOMECHANICS_CONDITION z_disp_zero
GEOMECHANICS_REGION bottom
END

← Return to List of Keywords
4.24 Keyword: GEOMECHANICS_STRATA

Description: Couples geomechanics material IDs and/or properties with a geomechanics region in the problem domain.

GEOMECHANICS_STRATA

GEOMECHANICS_MATERIAL <string> name of the geomechanics material property to be associated with a geomechanics region

GEOMECHANICS_REGION <string> name of geomechanics region associated with a geomechanics material property

END [/ .]

Example:

GEOMECHANICS_STRATA
   GEOMECHANICS_MATERIAL granite
   GEOMECHANICS_REGION all
END
4.25 Keyword: GRID

Description: this keyword defines the discretization scheme, the type of grid and resolution, and the geometry employed in the simulation.

GRID

Required Input Parameters:

TYPE <type> <symmetry>:
Grid type (structured, structured_mimetic, unstructured, amr)
Symmetry type (cartesian [default], cylindrical, spherical)

NXYZ <# # #>: # of grid cells in x, y, z directions (structured only)

FILE <filename>: Name of file containing grid information (unstructured only)

BOUNDS:

<x_min, y_min, z_min>
<x_max, y_max, z_max>

END [/ .]

DXYZ: Specifies grid spacing of structured cartesian grid (see examples below)

<dx>
<dy>
<dz>

END [/ .]

Optional Input Parameters:

GRAVITY <# # #>: Specifies gravity vector [Default: 0 0 −9.8068 m/s²]

ORIGIN <# # #>: Coordinate of grid origin [Default: 0 0 0]

INVERT_Z: Inverts the z-axis [Default: positive z points downward]

Format of unstructured grid file:
Implicitly defined grid:

! type: H=hexahedron, T=tetrahedron, W=wedge, P=pyramid
! vertn(H) = 8
! vertn(T) = 4
! vertn(W) = 6
! vertn(P) = 5
! num_cells num_vertices (integers)
! type vert1 vert2 vert3 ... vertn ! for cell 1 (integers)
! type vert1 vert2 vert3 ... vertn ! for cell 2
! ...
! ...
! type vert1 vert2 vert3 ... vertn ! for cell num_cells
! xcoord ycoord zcoord ! coordinates of vertex 1 (real)
! xcoord ycoord zcoord ! coordinates of vertex 2 (real)
! ...
! xcoord ycoord zcoord ! coordinates of vertex num_vertices (real)

Explicitly defined grid:

! Format of explicit unstructured grid file
! id_, id_up_, id_dn_ = integer
! x_, y_, z_, area_, volume_ = real
! definitions
! id_ = id of grid cell
! id_up_ = id of upwind grid cell in connection
! id_dn_ = id of downwind grid cell in connection
! x_ = x coordinate of cell center
! y_ = y coordinate of cell center
! z_ = z coordinate of cell center
! volume_ = volume of grid cell
! CELLS <integer> integer = # cells (N)
! id_1 x_1 y_1 z_1 volume_1
! id_2 x_2 y_2 z_2 volume_2
! ...
! ...
! id_N x_N y_N z_N volume_N
! CONNECTIONS <integer> integer = # connections (M)
! id_up_1 id_dn_1 x_1 y_1 z_1 area_1
! id_up_2 id_dn_2 x_2 y_2 z_2 area_2
Examples:

GRID
   TYPE structured cylindrical
   NXYZ 512 1 32
   DXYZ
      2.d0
      1.d0
      2.d0
   /
   END

GRID
   TYPE structured
   NXYZ 512 1 32
   BOUNDS
      0. 0. 0.
      1024. 1. 64.
   /
   END

By using the BOUNDS keyword, the model domain is specified in a grid-independent fashion and, as a result, the grid spacing may be changed by modifying the keyword NXYZ only.

⇒ Return to List of Keywords

↗ Return to List of Keywords
4.26 Keyword: INITIAL_CONDITION

Description: Condition coupler between regions and flow and transport conditions.

Input:

INITIAL_CONDITION [Name]

REGION region_name
FLOW_CONDITION condition_name
TRANSPORT_CONDITION condition_name

END [/ .]

Explanation:

Example:

:========= condition couplers ==========
: initial condition
INITIAL_CONDITION
  FLOW_CONDITION gradient-north
  TRANSPORT_CONDITION Initial
  REGION all
END
4.27 Keyword: INTEGRAL_FLUX

Description: Sets up a surface through which fluxes of all primary dependent variables can be calculated.

Note: add keyword PERIODIC_OBSERVATION to OUTPUT keyword to toggle printing of integral fluxes to a file with the suffix ‘*-int.dat’.

Input:

INTEGRAL_FLUX <string [Optional]> Opens input block and associates a name with INTEGRAL_FLUX. [Required]

NAME <string> Specifies a name that is associated with the integral fluxes in the ‘*-int.dat’ file. This name will overwrite any name specified with the INTEGRAL_FLUX keyword. [Optional]

COORDINATES Opens a block listing coordinates \((x_1, y_1, z_1), (x_2, y_2, z_2)\), defining the rectangle over which the flux is to be calculated. The coordinates must form a rectangular plane that is aligned with the coordinate axes. [Required]

INVERT_DIRECTION Inverts the sign of the flux. For fluxes at upwind boundaries, influx will be negative. This has no impact on the actual flux values other than to change the sign of the flux. [Optional]

END [/ .]

Examples:

```
INTEGRAL_FLUX flux_up_shaft
  COORDINATES
    25.d0 15.d0 300.d0
    35.d0 20.d0 300.d0
/
/
INTEGRAL_FLUX
  NAME inflow
  COORDINATES
    0.d0 0.d0 0.d0
    0.d0 10.d0 5.d0
/
  INVERT_DIRECTION
/
```

↩ Return to List of Keywords
4.28 Keyword: LINEAR_SOLVER

Description:

Input:

LINEAR_SOLVER [TRAN, TRANSPORT / FLOW]

SOLVER_TYPE (SOLVER, KRYLOV_TYPE, KRYLOV, KSP, KSP_TYPE)

NONE (PREONLY)
DIRECT (LU decomposition)
ITERATIVE (Bi-CGStab (BCGS) and block Jacobi preconditioning with ILU[0] in each block)
GMRES
FGMRES
BCGS (BICGSTAB, BI-CGSTAB)
IBCGS (IBICGSTAB, IBI-CGSTAB) (Improved BCGS)
RICHARDSON
CG

PRECONDITIONER_TYPE (PRECONDITIONER, PC, PC_TYPE)

NONE (PCNONE)
ILU (PCILU)
LU (PCLU)
BJACOBI (BLOCK_JACOBI)
ASM (ADDITIVE_SCHWARTZ)
PCASM
HYPRE
SHELL

HYPRE_OPTIONS TYPE [pilut, parasails, boomeramg, euclid]

BOOMERAMG_CYCLE_TYPE <char> [V, W]
BOOMERAMG_MAX_LEVELS <int>
BOOMERAMG_MAX_ITER <int>
BOOMERAMG_TOL <float>
BOOMERAMG_TRUNCFACTOR <float>
BOOMERAMG_AGGR_NL <float>
BOOMERAMG_AGGR_NUM_PATHS <int>
BOOMERAMG_STRONG_THRESHOLD <float>
BOOMERAMG_GRID_SWEEPS_ALL <float>
BOOMERAMG_GRID_SWEEPS_DOWN <float>
BOOMERAMG_GRID_SWEEPS_UP <float>
BOOMERAMG_GRID_SWEEPS_COARSE <float>
BOOMERAMG_RELAX_TYPE_ALL <Value>
BOOMERAMG_RELAX_TYPE_DOWN <Value>
BOOMERAMG_RELAX_TYPE_UP <Value>
BOOMERAMG_RELAX_TYPE_COARSE <Value>
BOOMERAMG_RELAX_WEIGHT_ALL <Value>
BOOMERAMG_RELAX_WEIGHT_LEVEL <Value>
BOOMERAMG_OUTER_RELAX_WEIGHT_ALL <Value>
BOOMERAMG_OUTER_RELAX_WEIGHT_LEVEL <Value>
BOOMERAMG_NO_CF <Value>
BOOMERAMG_MEASURE_TYPE <Value>
BOOMERAMG_COARSE_TYPE <Value>
BOOMERAMG_INTERPOLATION_TYPE, BOOMERAMG_INTERP_TYPE <Value>
BOOMERAMG_NODAL_COARSE <Value>
BOOMERAMG_NODAL_RELAXATION <Value>

ATOL <float> (Absolute tolerance: \(|b - Ax_n| < \epsilon\))
RTOL <float> (Relative tolerance: \(|b - Ax_n|/|b - Ax_0| < \epsilon\))
DTOL <float> (Divergence tolerance: \(|b - Ax_n|/|b - Ax_0| > \epsilon\))
MAXIT <int> (Maximum number of linear solver iterations)

LU_ZERO_PIVOT_TOL <float> Specifies zero pivot tolerance for ILU/LU preconditioners

END [ / .]

Explanation:

Examples:

LINEAR_SOLVER FLOW
   SOLVER DIRECT
/

LINEAR_SOLVER TRANSPORT
   SOLVER ITERATIVE
/

LINEAR_SOLVER FLOW
   SOLVER GMRES
   PRECONDITIONER ILU
/
Advanced PETSc options

LINEAR_SOLVER FLOW
  KSP_TYPE IBCGS
  PC_TYPE ASM
/

LINEAR_SOLVER TRANSPORT
  KSP_TYPE PCNONE
  PC_TYPE LU
  LU_ZERO_PIVOT_TOL 1d-15
/
4.29  Keyword: MATERIAL_PROPERTY

Description: Specifies material properties to be associated with a region in the problem domain.

Input:

MATERIAL_PROPERTY  <char>

   ID  <int>
   SATURATION_FUNCTION  <char>
   ROCK_DENSITY  <float> [kg/m$^3$]
   SPECIFIC_HEAT  <float> [J/kg/K]
   LONGITUDINAL_DISPERSIVITY  <float> [m]
   TRANSVERSE_DISPERSIVITY  <float> (not implemented) [m]
   THERMAL_CONDUCTIVITY_DRY  <float> [W/m/K]
   THERMAL_CONDUCTIVITY_WET  <float> [W/m/K]
   PORE_COMPRESSIBILITY  <float> (not implemented) [bar$^{-1}$]
   THERMAL_EXPANSITIVITY  <float> (not implemented) [C$^{-1}$]
   POROSITY  <float> [—], porosity_filename
   TORTUOSITY  <float> [—]

PERMEABILITY

   ISOTROPIC  Toggles on isotropy [Default]
   ANISOTROPIC  Toggles on anisotropy
   VERTICAL_ANISOTROPY_RATIO  <float>
   PERM_X  <float> Diagonal permeability $k_{xx}$ [m$^2$]
   PERM_Y  <float> Diagonal permeability $k_{yy}$ [m$^2$]
   PERM_Z  <float> Diagonal permeability $k_{zz}$ [m$^2$]
   PERM_ISO  <float> Isotropic permeability values [m$^2$]
   PERM_XY  <float> Off-diagonal permeability $k_{xy}$ for use with MFD
                     (mimetic_unstructured grid) [m$^2$] (not currently supported)
   PERM_XZ  <float> [m$^2$] Off-diagonal permeability $k_{xz}$
   PERM_YZ  <float> [m$^2$] Off-diagonal permeability $k_{yz}$

END  [/ .]

PERMEABILITY_POWER  <float> (see Eqn.(10-119))
PERMEABILITY_CRIT_POR  <float> (see Eqn.(10-120))
PERMEABILITY_MIN_SCALE_FAC  <float> (see Eqn.(10-121))
TORTUOSITY_POWER  <float>

MINERAL_SURFACE_AREA_POWER  toggle to update mineral surface area

(see MINERAL_KINETICS keyword for setting porosity or volume fraction power)
SECONDARY_CONTINUUM  Activate with MULTIPLE_CONTINUUM keyword

**TYPE**  <char> (SLAB, NESTED_CUBES, NESTED_SPHERES)

**NUM_CELLS**  <int> Number of secondary continuum grid cells

**LOG_GRID_SPACING**  Toggle to use logarithmic grid spacing in secondary continua
  (applies to nested spheres and cubes only)

**OUTER_SPACING**  <float> [m] Outer matrix node grid spacing for logarithmic grid
  (see Eqns.(10-85))

**LENGTH**  <float> [m] length of SLAB type

**AREA**  <float> [m²] cross-section area for SLAB type

**MATRIX_BLOCK_SIZE**  <float> [m] matrix block size for NESTED_CUBES type

**FRACTURE_SPACING**  <float> [m] fracture spacing for NESTED_CUBES type

**RADIUS**  <float> [m] radius for NESTED_SPHERES type

**EPSILON**  <float> Volume fraction of the primary continuum (fracture) in REV

**APERTURE**  <float> Fracture aperture (if specified overrides epsilon—applicable
  for nested cubes only)

**AREA_SCALING_FACTOR**  <float> Factor multiplying primary-secondary continua
  coupling term (default set to 1)

END [/ .]

END [/ .]

Explanation:

Example:

```
MATERIAL_PROPERTY  Hanford
  ID 1
  SATURATION_FUNCTION sf1
  POROSITY  0.332
  TORTUOSITY 1.
  PERMEABILITY
    PERM_X 1.d-12
    PERM_Y 1.d-12
    PERM_Z 1.d-12
  /
END

MATERIAL_PROPERTY soil1
  ID 1
  SATURATION_FUNCTION sf1
  POROSITY 0.1
```

< 80 >
Example: reading from a file for multiple realization simulation

```
mpirun -np 10000 ./pflotran -stochastic -num_realizations 1000 -num_groups 100
```

PFLOTRAN input file:

```
:=================== material properties ===================
MATERIAL_PROPERTY rock
ID 1
POROSITY 0.381d0
TORTUOSITY 1.d-1
ROCK_DENSITY 2.65d3
SPECIFIC_HEAT 1.d3
THERMAL_CONDUCTIVITY_DRY 2.5
THERMAL_CONDUCTIVITY_WET 2.5
SATURATION_FUNCTION sf2
PERMEABILITY
   ANISOTROPIC ! default is ISOTROPIC
   DATASET Permeability
/
/
DATASET Permeability
   FILENAME ./permeability_with_white_noise_multi.h5
   REALIZATION_DEPENDENT
/

Python script fragment to generate permeability field:

```
dataset_name = 'PermeabilityX'
h5dset = h5file.create_dataset(dataset_name, data=rarray1)
```
print 'done with ', dataset_name

dataset_name = 'PermeabilityY'
h5dset = h5file.create_dataset(dataset_name, data=rarray2)
print 'done with ', dataset_name

dataset_name = 'PermeabilityZ'
h5dset = h5file.create_dataset(dataset_name, data=rarray3)
print 'done with ', dataset_name

For the full permeability tensor the dataset naming convention is:

PermeabilityX
PermeabilityY
PermeabilityZ
PermeabilityXY
PermeabilityXZ
PermeabilityYZ
4.30 Keyword: MODE

Description: determines the flow mode: Richards (variably saturated porous media); MPH, MPHASE, FLASH2 (CO\textsubscript{2} + H\textsubscript{2}O); TH (Thermal-Hydrologic); IMMIS, THS (Immiscible).

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERAL</td>
<td>Two-phase air-water, nonisothermal, variably saturated groundwater flow</td>
</tr>
<tr>
<td>RICHARDS</td>
<td>Single-phase, isothermal, variably saturated groundwater flow using Richards equation</td>
</tr>
<tr>
<td>MPHASE (MPH)</td>
<td>Two-phase supercritical CO\textsubscript{2}-brine-energy based on variable switching for phase changes</td>
</tr>
<tr>
<td>FLASH2</td>
<td>Two-phase supercritical CO\textsubscript{2}-brine-energy based on the flash method for phase changes with a persistent set of unknowns</td>
</tr>
<tr>
<td>TH</td>
<td>Thermal-Hydrologic coupled groundwater flow</td>
</tr>
<tr>
<td>IMMIS (IMS, THS)</td>
<td>Immiscible CO\textsubscript{2}-water-energy</td>
</tr>
<tr>
<td>MIS</td>
<td>Missible H\textsubscript{2}O-glycol fluid</td>
</tr>
</tbody>
</table>

Example:

```
MODE TH
```

\(\leftrightarrow\) Return to List of Keywords
4.31 Keyword: MULTIPLE_CONTINUUM

Description: This keyword initiates the multiple continuum formulation (implemented currently only for heat equation). The properties of the secondary continuum can be specified under the MATERIAL_PROPERTY keyword.

| MULTIPLE_CONTINUUM | Activate multiple continuum model for heat and single solute transport equation. |

↩ Return to List of Keywords
4.32 Keyword: NEWTON_SOLVER

Description:

Input:

NEWTON_SOLVER

TRAN, TRANSPORT (tran_solver) / DEFAULT (flow_solver)
INEXACT_NEWTON
NO_PRINT_CONVERGENCE
NO_INF_NORM (NO_INFINITY_NORM)
NO_FORCE_ITERATION
PRINT_DETAILED_CONVERGENCE
ATOL <Value>
RTOL <Value>
STOL <Value>
DTOL <Value>
ITOL (INF_TOL, ITOL_RES, INF_TOL_RES) <Value>
ITOL_UPDATE (INF_TOL_UPDATE) <Value>
ITOL_SEC (ITOL_RES_SEC, INF_TOL_SEC) <Value>
MAXIT <Value> Cuts time step if the number of iterations exceed this value
MAXF <Value>
MAX_NORM <Value> Cuts time step if the convergence norm exceeds this value.

END [/ .]

Explanation:

typedef enum {  /* converged */
SNES_CONVERGED_FNORM_ABS = 2,  /* ||F|| < atol */
SNES_CONVERGED_FNORM_RELATIVE = 3,  /* ||F|| < rtol*||F_initial|| */
SNES_CONVERGED_SNORM_RELATIVE = 4,  /* Newton computed step size small;
          || delta x || < stol || x || */
SNES_CONVERGED_ITS = 5,  /* maximum iterations reached */
SNES_CONVERGED_TR_DELTA = 7,
/* diverged */
SNES_CONVERGED_FUNCTION_DOMAIN = -1,  /* the new x location passed the function is not in the domain of F */
SNES_CONVERGED_FUNCTION_COUNT = -2,
SNES_CONVERGED_LINEAR_SOLVE = -3,  /* the linear solve failed */
SNES_CONVERGED_FNORM_NAN = -4,
SNES_CONVERGED_MAX_IT = -5,
SNES_CONVERGED_LINE_SEARCH = -6,  /* the line search failed */
SNES_CONVERGED_INNER = -7,  /* inner solve failed */
SNES_CONVERGED_LOCAL_MIN = -8,  /* || J^T b || is small, implies converged to local minimum of F */
SNES_CONVERGED_ITERATING = 0} SNESConvergedReason;

< 85 >
Example:

-> Return to List of Keywords
4.33 Keyword: NONUNIFORM_VELOCITY

NONUNIFORM_VELOCITY <file name>

**Explanation:** Name of HDF5 file for specifying nonuniform velocities including boundary velocities. Only applies to structured grids. The velocity must be specified in SI units (m/s).

Internal velocities are specified for each cell corresponding to the downwind face and read in separately for \( v_x \), \( v_y \), and \( v_z \). Boundary velocities are read in at all cells for each boundary condition (cells not in boundary condition region are ignored, but need a value to be read nonetheless).

A boundary condition may not wrap around a corner. Corner cells should be mapped to multiple boundary conditions with their boundary faces residing in different regions. Below is an example python script for generating an HDF5 velocity file.

```python
# Python script for generating an HDF5 velocity file read by PFLOTRAN.
import sys
import math
from h5py import *
import numpy
import random

filename = 'velocity.h5'
h5file = File(filename,mode='w')

nx = 100
ny = 2
nz = 2
nxXny = nx*ny
n = nx*ny*nz
iarray = numpy.zeros((n),'=i4')
# add cell ids to file
for i in range(n):
iarray[i] = i+1
dataset_name = 'Cell Ids'
h5dset = h5file.create_dataset(dataset_name, data=iarray)

rarray = numpy.zeros((n),'=f8')
# x velocity
for i in range(n):
rarray[i] = 3.171e-8
dataset_name = 'Internal Velocity X'
h5dset = h5file.create_dataset(dataset_name, data=rarray)

# y velocity
for i in range(n):
rarray[i] = 0.
```

< 87 >
dataset_name = 'Internal Velocity Y'
h5dset = h5file.create_dataset(dataset_name, data=rarray)

# z velocity
dataset_name = 'Internal Velocity Z'
h5dset = h5file.create_dataset(dataset_name, data=rarray)

# west boundary velocity
i = 0
for index in range(n):
    rarray[index] = 0.
for k in range(nz):
    for j in range(ny):
        index = k*(nx*ny) + j*nx + i
        rarray[index] = 3.171e-8
dataset_name = 'West'
h5dset = h5file.create_dataset(dataset_name, data=rarray)

# east boundary
i = nx-1
for index in range(n):
    rarray[index] = 0.
for k in range(nz):
    for j in range(ny):
        index = k*(nx*ny) + j*nx + i
        rarray[index] = -3.171e-8
dataset_name = 'East'
h5dset = h5file.create_dataset(dataset_name, data=rarray)

h5file.close()
print('done with everything')

Example:

NONUNIFORM\_VELOCITY velocity.h5
4.34 Keyword: NUMERICAL_JACOBIAN_FLOW

NUMERICAL_JACOBIAN_FLOW Uses numerically evaluated Jacobian for flow.
4.35 Keyword: NUMERICAL_JACOBIAN_RXN

NUMERICAL_JACOBIAN_RXN Uses numerically evaluated Jacobian for reactions.
4.36  Keyword: NUMERICAL_JACOBIAN_MULTI_COUPLE

NUMERICAL_JACOBIAN_MULTI_COUPLE  The contribution to the primary continuum Jacobian due to primary-secondary continua coupling term is numerically evaluated.

↩️ Return to List of Keywords
4.37 **Keyword: OBSERVATION**

**Description:** The OBSERVATION card specifies a location (REGION) at which flow and transport results (e.g. pressure, saturation, flow velocities, solute concentrations, etc.) will be monitored in the output. The user must specify either a region or boundary condition to which the observation object is linked. The velocity keyword toggles on the printing of velocities at a point in space.

**Input:**

```
OBSERVATION

  BOUNDARY_CONDITION  boundary condition name
  REGION  region name
  VELOCITY
    AT_CELL_CENTER
    SECONDARY_TEMPERATURE
    SECONDARY_CONCENTRATION
    SECONDARY_MINERAL_VOLFRAC

END  [/ .]
```

**Explanation:**

**Keyword OBSERVATION** initiates an observation point entry.

**Keyword REGION** (optional) defines the name of the region (usually a point in space) to which the observation point is linked.

**Keyword BOUNDARY_CONDITION** (optional) specifies the name of a boundary condition to which the observation point is tied (e.g. to monitor fluxes across a boundary face).

**Keyword VELOCITY** (optional) toggles on the printing of Darcy velocities at the observation point.

**Keyword SECONDARY_TEMPERATURE** (optional) toggles on the printing of the secondary continuum (matrix nodes) temperatures.

**Keyword SECONDARY_CONCENTRATION** (optional) toggles on the printing of the secondary continuum (matrix nodes) concentration.

**Keyword SECONDARY_MINERAL_VOLFRAC** (optional) toggles on the printing of the secondary continuum (matrix nodes) mineral volume fraction.
Examples:

OBSERVATION
   REGION well1
   VELOCITY
END

OBSERVATION
  BOUNDARY_CONDITION river
END
4.38   **Keyword: ORIGIN (ORIG)**

**ORIGIN (ORIG) X_DIRECTION Y_DIRECTION Z_DIRECTION**

→ Return to List of Keywords
4.39  Keyword: OUTPUT

Description: The **OUTPUT** keyword controls formatting and time of output.

Input:

**OUTPUT**

- **TIMES** Unit (s, min, h (hr), d, w, mo, y (yr)) <float>
- **SCREEN OFF** suppress screen output
- **SCREEN PERIODIC** <int>: print to screen every <integer> time steps.
- **PERIODIC TIME** <float> Unit
- **PERIODIC TIMESTEP** <float> Unit
- **PERIODIC_OBSERVATION TIME** <float> <unit>: output the results at observation points and mass balance output at specified output times
- **PERIODIC_OBSERVATION TIMESTEP** <integer>: output the results at observation points and mass balance output at specified time steps
- **NO_PRINT_INITIAL** the initial state of the system will not be printed to the output file if this card is activated
- **NO_PRINT_FINAL** the final state of the system will not be printed to the output file if this card is activated
- **PRINT_COLUMN_IDS** print column numbers in observation and mass balance output files
- **FORMAT** <file format>: specify the snapshot in time file type. File formats available are:
  - **TECPLOT**
    - **POINT** - TecPlot point format (requires a single processor)
    - **BLOCK** - TecPlot block format
    - **FEBRICK** - TecPlot finite element
  - **HDF5**
    - **SINGLE_FILE** - produces single HDF5 file `pflotran.h5`
    - **MULTIPLE_FILES** [TIMES_PER_FILE] - produces a separate HDF5 file for number of times specified by TIMES_PER_FILE [default 1]
    - **MAD** - (not supported)
    - **VTK** - VTK format
- **VOLUME** - Output cell volume
- **PERMEABILITY** - Output cell permeability
- **POROSITY** - Output cell porosity
- **FLUXES** - Output interface fluxes
- **VELOCITY_AT_FACE** - Output interface velocities
  - Structured grid: velocity outputted at internal faces only. Visualization support is available via XDMF.
- Unstructured grid: velocity outputted for internal and boundary faces. Data is outputted in HDF5 format only with no visualization support via XDMF.

**VELOCITY_AT_CENTER** - Output cell-centered velocities (supported for structured and unstructured grids, applies to TecPlot and HDF5 output formats)

**MASS_BALANCE:** output the mass balance of the system if this card is activated. It includes global mass balance as well as fluxes at all boundaries for water and chemical species specified for output in the CHEMISTRY card. For the MPHASE mode only global mass balances are provided including supercritical CO\textsubscript{2}. Output times are controlled by \texttt{PERIODIC_OBSERVATION TIMESTEP} and \texttt{TIME}, and printout times.

END [/ .]

Explanation:
<table>
<thead>
<tr>
<th>OUTPUT:</th>
<th>keyword to control output.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMES:</td>
<td>list of output times.</td>
</tr>
<tr>
<td>SCREEN OFF:</td>
<td>turns off screen output</td>
</tr>
<tr>
<td>SCREEN PERIODIC:</td>
<td>controls screen output frequency.</td>
</tr>
<tr>
<td>PERIODIC TIME:</td>
<td>controls frequency of output times.</td>
</tr>
<tr>
<td>PERIODIC TIMESTEP:</td>
<td>controls frequency of output time steps.</td>
</tr>
<tr>
<td>PERIODIC_OBSERVATION TIME:</td>
<td>frequency of output time.</td>
</tr>
<tr>
<td>PERIODIC_OBSERVATION TIMESTEP:</td>
<td>frequency of output time step.</td>
</tr>
<tr>
<td>NO_FINAL, NO_PRINT_FINAL:</td>
<td></td>
</tr>
<tr>
<td>FORMAT TECPLOT POINT:</td>
<td>Tecplot POINT output, valid for 1D and 2D problems on a single processor core.</td>
</tr>
<tr>
<td>FORMAT TECPLOT BLOCK:</td>
<td>Tecplot BLOCK output for multi-processor core runs.</td>
</tr>
<tr>
<td>FORMAT HDF5:</td>
<td>HDF5 output format written to a .h5 file which can be read by VisIt and ParaView.</td>
</tr>
<tr>
<td>FORMAT MAD:</td>
<td>MAD (Method of Anchored Distributions) format.</td>
</tr>
<tr>
<td>FORMAT VTK:</td>
<td>VTK format which can be read by VisIt and ParaView.</td>
</tr>
<tr>
<td>UNIT:</td>
<td>time units of seconds (s), minutes (min), hours (h, hr), days (d), weeks (w), months (mo), and years (y, yr).</td>
</tr>
<tr>
<td>PERMEABILITY:</td>
<td></td>
</tr>
<tr>
<td>POROSITY:</td>
<td></td>
</tr>
<tr>
<td>FLUXES:</td>
<td>keyword to output velocities.</td>
</tr>
<tr>
<td>VELOCITIES:</td>
<td>keyword to output velocities.</td>
</tr>
<tr>
<td>MASS_BALANCE:</td>
<td>keyword to output global mass balances and boundary fluxes, both cumulative and instantaneous.</td>
</tr>
</tbody>
</table>

The output in the mass balance file refers to global mass conservation. This output is described in detail for the MPHASE mode. The total number of moles of the \( i \)th component in phase \( \alpha \) is given as the integral over the entire computational domain (see Eqn.(10-19a))

\[
N_i^\alpha = \int_V \varphi s_{\alpha} \eta_{\alpha} x_i^\alpha \, dV. \tag{4-2}
\]

From the governing equations, Eqn.(10-19a), the time rate of change of the total number of moles of the \( i \)th component

\[
N_i = \sum_\alpha N_i^\alpha, \tag{4-3}
\]
is given by
\[
\frac{dN_i}{dt} = \sum_\alpha \frac{dN_i^\alpha}{dt} = -\sum_\alpha \int_{\partial V} \mathbf{F}_i^\alpha \cdot d\mathbf{S} + Q_i,
\]
where the surface integral on the right hand side is over the flux flowing through the boundaries of the domain. The surface averaged flux \( \mathcal{F}_i \) is defined as
\[
\mathcal{F}_i^\alpha = -\int_{\partial V} \mathbf{F}_i^\alpha \cdot d\mathbf{S},
\]
defined as positive for flow into and negative for flow out of the domain \( V \). Integrating the flux over time gives the cumulative flux \( \mathcal{F}_i^\alpha \)
\[
\mathcal{F}_i^\alpha = \int_0^t \mathcal{F}_i^\alpha dt',
\]
and time integrated source/sink \( Q_i \)
\[
Q_i = \int_0^t Q_i dt'.
\]
Global conservation of mass implies the equality
\[
N_i(t) = N_i(0) + \sum_\alpha \mathcal{F}_i^\alpha + Q_i.
\]

The header for the MPHASE mass balance file reads (for time in years, \( l = \) aqueous liquid phase, \( sc = \) supercritical CO\(_2\)):
\[
\begin{align*}
t, & \quad \Delta t, \quad [y] \\
N_{H_2O}^l, N_{CO_2}^l, & \quad \tilde{N}_{H_2O}^l, N_{CO_2}^{sc}, \quad \tilde{N}_{CO_2}^{sc}, \quad [\text{kmol}] \\
\mathcal{F}_{H_2O}^l, \mathcal{F}_{CO_2}^l, & \quad \mathcal{F}_{H_2O}^{sc}, \mathcal{F}_{CO_2}^{sc}, \quad [\text{kmol}] \\
\mathcal{F}_{H_2O}, \mathcal{F}_{CO_2}, & \quad \mathcal{F}_{H_2O}^{sc}, \mathcal{F}_{CO_2}^{sc}, \quad [\text{kmol}/y] \\
\overline{Q}_{H_2O}, \overline{Q}_{CO_2}, & \quad \overline{Q}_{H_2O}, \overline{Q}_{CO_2}, \quad [\text{kmol}/y]
\end{align*}
\]

The quantity \( \tilde{N}_i^\alpha \) denotes the trapped component in phase \( \alpha \) defined by
\[
\tilde{N}_i^\alpha = \int_{V, s_\alpha < s_\alpha^0} \varphi s_\alpha \eta_\alpha x_i^\alpha dV,
\]
where \( s_\alpha^0 \) denotes the residual saturation. There are as many rows of cumulative and instantaneous fluxes as there are boundaries and source/sinks.
Examples:

```
OUTPUT
  !SCREEN PERIODIC 10
  !PERIODIC TIME 10 h
  PERIODIC_OBSERVATION TIMESTEP 1
  !times h 1.
  !PERIODIC_OBSERVATION TIME 50 h
  FORMAT TECPLOT POINT ! or BLOCK
  FORMAT HDF5
  VELOCITIES
  MASS_BALANCE
END
```
4.40  Keyword: OVERWRITE_RESTART_TRANSPORT

OVERWRITE_RESTART_TRANSPORT  \{ overwrite_restart_transport = .true. \}

Overwrites checkpointing values with values read in from the input file on restart.

→ Return to List of Keywords
4.41 Keyword: PROC

PROC <int int int>

Description: The number of processor to be employed in each direction \( x, y, \) and \( z \) (structured grids only). Default: let PETSc decide. Warning: the product of the integers must be equal to the number of processor employed.

Examples:

- PROC 100 10 10  \! 10,000 processes
- PROC 2 2 2  \! 2×2×2 decomposition
- PROC 1 1 8  \! Force decomposition in z direction only (1×1×8 decomposition).

↩ Return to List of Keywords
4.42 Keyword: REGION

Description: The REGION keyword defines a set of grid cells encompassed by a volume or intersected by a plane or point, or a list of grid cell ids. The REGION name can then be used to link this set of grid cells to material properties, strata, boundary and initial conditions, source sinks, observation points, etc. Although a region may be defined though the use of (I, J, K) indices using the BLOCK keyword, the user is encouraged to define regions either through COORDINATES or lists read in from an HDF5 file in order to minimize the dependence of the input file on grid resolution. In the case of the FILE keyword, a list of grid cell ids is read from an HDF5 file where the region_name defines the HDF5 data set. It should be noted that given a region defined by a plane or point shared by two grid cells (e.g. a plane defining the surface between two grid cells), PFLOTRAN will select the upwind cell(s) as the region.

Input:

REGION region_name
   FILE file_name
   LIST (to be implemented)
   FACE face_name
   BLOCK i1 i2 j1 j2 k1 k2
   COORDINATE x y z
   COORDINATES
       x1 y1 z1
       x2 y2 z2
END [/ .]

END [/ .]

Explanation:

Keyword REGION begins a region entry with name region_name.

Keyword BLOCK defines a volumetric, planar, or point region through IJK indices: i1 i2 j1 j2 k1 k2.

Keyword COORDINATE defines a point region through coordinates in 3D space.

Keyword COORDINATES Defines a volumetric, planar, or point region between two points in space.

Keyword FILE Defines an HDF5 file within which a dataset named region_name contains a list of grid cells corresponding to a region.

Keyword FACE Defines the face of the grid cell to which boundary conditions are connected where face_name is one of WEST, EAST, NORTH, SOUTH, BOTTOM, TOP (structured grids only).

Keyword END Ends the region entry (can be one of . END).
Examples:

REGION source_zone
   BLOCK 3 5 15 16 2 3
END

REGION source_zone
   BLOCK 3 5 15 16 2 3
END

REGION west_boundary
   BLOCK 1 1 1 30 1 50
   FACE WEST
END

REGION source_zone
   COORDINATES
      50. 10. 10.
      60. 15. 15.
   /
END

REGION river_boundary
   FILE ./regions.h5
   FACE EAST
END

REGION well
   COORDINATE 50. 10. 10.
END

REGION well
   COORDINATE 50. 10. 10.
END

REGION west_boundary
   COORDINATES
      0. 0. 0.
      0. 10. 10.
   /
   FACE WEST
END
4.43 Keyword: RESTART

Description

The RESTART card defines a checkpoint file from which the current simulation should be
restarted. If a time is specified after the file name, the initial simulation time is set to that time.

Input:

```
RESTART <restart_file_name> <restart_time> <time_units>
```

Explanation:

Keyword RESTART defines the checkpoint filename to be read in to restart a simulation at the
specified time.

Examples:

: restart the simulation from the end of the previous /
simulation, but set the time back to 0.

RESTART restart.chk 0.d0 y

: restart the program running from where it left off /
when the file pflotran.chk3000 was printed

RESTART pflotran.chk3000
4.44 Keyword: SATURATION_FUNCTION

Description: Currently being revised to use a common set of saturation and relative permeability functions across all modes. Check source code for up-to-date implementation.

Input:

SATURATION_FUNCTION Name

SATURATION_FUNCTION_TYPE [VAN_GENCHTEN, BROOKS_COREY, THOMEER_COREY, NMT_EXP, PRUESS_1, VAN_GENCHTEN_PARKER]
PERMEABILITY_FUNCTION_TYPE [MUALEM, BURDINE]
RESIDUAL_SATURATION <Value> MODES: RICHARDS, TH
RESIDUAL_SATURATION_LIQUID <Value> MODES: MPHASE
RESIDUAL_SATURATION_GAS <Value>
   MODES: MPHASE (VAN_GENCHTEN_PARKER)
LAMBDA <Value> [—]
ALPHA <Value> [Pa⁻¹]
MAX_CAPILLARY_PRESSURE <Value> [Pa]
BETAC <Value> [—] (Only used in NMT_EXP)
POWER <Value> [—] (Only used in NMT_EXP)

END [/ .]

Explanation:

Example:

SATURATION_FUNCTION sf1
   SATURATION_FUNCTION_TYPE VAN_GENCHTEN
   RESIDUAL_SATURATION 0.1d0
LAMBDA 2.67d0
ALPHA 2.042d-4
MAX_CAPILLARY_PRESSURE 1d8
END

Return to List of Keywords
4.45 Keyword: SOURCE_SINK

**SOURCE_SINK** <name>

- **REGION** <region_name> name of the region the source/sink term is applied to
- **FLOW_CONDITION** <condition_name> name of the flow condition
- **TRANSPORT_CONDITION** <condition_name> name of the transport condition

**END** [/ .]

**Example:**

```plaintext
SOURCE_SINK Well_2-9_1
    FLOW_CONDITION Injection_1
    TRANSPORT_CONDITION Source
    REGION Well_2-9_1
END
```

↪ Return to List of Keywords
4.46  Keyword: STRATIGRAPHY (STRATA)

Couples material IDs and/or properties with a region in the problem domain.

STRATIGRAPHY (STRATA)

  MATERIAL  <string>  name of the material property to be associated with a region

  REGION    <string>  name of region associated with a material property

END  [/ .]

Note: a filename can be provided instead of a material property name from which material IDs are read on a cell by cell basis. In this case there is no need for the REGION keyword.

Examples:

Assign hanford_unit material properties to the region source_zone

STRATA
  MATERIAL hanford_unit
  REGION source_zone
/

Assign material properties through material IDs read from an HDF5 formatted file. In this case there is no need to specify a region as material IDs are assigned to the entire grid on a cell by cell basis.

STRATA
  MATERIAL ..../field_material_ids.h5
/

For a detailed example on creating the .h5 file see the PFLOTRAN wiki: Strata.
Description: the keyword `TIME` controls the simulation time.

Input:

4.47 Keyword: TIME

TIME

FINAL_TIME <Value> Unit (s, m, h, d, mo, y)
INITIAL_TIMESTEP_SIZE <Value> Unit (s, m, h, d, mo, y)
MAXIMUM_TIMESTEP_SIZE <Value> Unit (s, m, h, d, mo, y)
MAXIMUM_TIMESTEP_SIZE <Value> Unit (s, m, h, d, mo, y) AT <Value>
STEADY_STATE

END [/ .]

Explanation:

Example:

```
TIME
  FINAL_TIME 100. h
  INITIAL_TIMESTEP_SIZE 1.d-3 h
  MAXIMUM_TIMESTEP_SIZE 1.d0 h
END

! Ability to change maximum time step size at select times
! during simulation.
TIME
  FINAL_TIME 100. y
  INITIAL_TIMESTEP_SIZE 1. s
  MAXIMUM_TIMESTEP_SIZE 1. y
  MAXIMUM_TIMESTEP_SIZE 1. d at 10. y
  MAXIMUM_TIMESTEP_SIZE 10. y at 11. y
END
```
4.48 Keyword: TIMESTEPPER

Description: the keyword TIMESTEPPER controls time stepping.

Input:

TIMESTEPPER [FLOW, TRAN, TRANSPORT]

NUM_STEPS_AFTER_CUT <int>
Number of time steps after a time step cut that the time step size is held constant [5].

MAX_STEPS <int>
Maximum time step after which the simulation will be terminated [999999].

TS_ACCELERATION <int>
Integer indexing time step acceleration ramp (expert users only) [5].

MAX_TS_CUTS <int>
Maximum number of consecutive time step cuts before the simulation is terminated with plot of the current solution printed to a XXX_cut_to_failure.tec file for debugging [16].

CFL_LIMITER <float>
The maximum CFL number allowed for transport. Enables Courant (CFL) number limiting on the transport time step.

DT_FACTOR <float>
Array of floating point numbers of tfac array (expert users only).

INITIALIZE_TO_STEADY_STATE
Flag indicating that the simulation is to be run as steady state (Warning: not robust).

RUN_AS_STEADY_STATE
Flag requesting that a steady state solution be computed based on boundary and initial conditions at the beginning of the simulation (Warning: not robust).

MAX_PRESSURE_CHANGE <float>
Maximum change in pressure for a time step [5.d4 Pa].

MAX_TEMPERATURE_CHANGE <float>
Maximum change in temperature for a time step [5 C].

MAX_CONCENTRATION_CHANGE <float>
Maximum change in concentration for a time step [1. mol/L].

MAX_SATURATION_CHANGE <float>
Maximum change in saturation for a time step [0.5].

PRESSURE_DAMPENING_FACTOR <float>

END [/ .]
Examples:

```
TIMESTEPPER
  TS_ACCELERATION 8
  MAX_STEPS 10000 ! terminates simulation after 10,000 time steps
  MAX_TS_CUTS 5 ! terminates simulation after 5 consecutive time step cuts
END

TIMESTEPPER
  CFL_LIMITER 1. ! limits time step size to enforce a CFL # of 1.
END
```

↩ Return to List of Keywords
4.49 Keyword: TRANSPORT_CONDITION

Description: Specifies a geochemical solution composition based on various user defined constraints with minerals, gases, pH, charge balance, free ion and total concentrations.

Input:

TRANSPORT_CONDITION Name

    TYPE [dirichlet, dirichlet_zero_gradient, equilibrium, neumann, mole, mole_rate, zero_gradient]
    TIME <Value>
    UNITS s, sec, min, hr, d, day, y, yr
    CONSTRAINT_LIST
        time constraint_name
    END [/ .]
    CONSTRAINT constraint_name

END [/ .]

Explanation:

Example:

TRANSPORT_CONDITION Initial
    TYPE dirichlet_zero_gradient
    CONSTRAINT_LIST
        0.d0 initial
    /
END

TRANSPORT_CONDITION U_source
    TYPE dirichlet
    CONSTRAINT_LIST
        0.d0 U_source
        336000.d0 Initial
    /
END

: the units for time are seconds by default
CONSTRAINT U_source

CONCENTRATIONS
:name concentration constraint constraint species
H+ 3.0 pH
Ca++ 1.20644e-3 T
Cu++ 1.e-8 T
Mg++ 5.09772e-4 T
UO2++ 1.E-6 T
K+ 1.54789e-4 T
Na+ 1.03498e-3 T
HCO3- -3.5 G CO2(g)
Cl- 6.97741e-4 Z
F- 2.09491e-5 T
HPO4-- 1.e-8 M Fluorapatite
NO3- 4.69979e-4 T
SO4-- 6.37961e-4 T
Tracer 1.e0 F
Tracer2 1.e0 F
/
MINERALS
:mineral vol. frac. area
Calcite 0.1 1.
Metatorbernite 0. 1.
/ ! end of minerals
END ! end of constraint

CONSTRAINT initial

CONCENTRATIONS
:name concentration constraint constraint species
H+ 7.3 M Calcite
Ca++ 1.20644e-3 T
Cu++ 1.e-8 T
Mg++ 5.09772e-4 M Dolomite
UO2++ 2.4830E-11 T
K+ 1.54789e-4 T
Na+ 1.03498e-3 Z
HCO3- -3.5 G CO2(g)
Cl- 6.97741e-4 T
F- 2.09491e-5 T
HPO4-- 1.e-8 M Fluorapatite
NO3- 4.69979e-4 T
SO4-- 6.37961e-4 T
Tracer 1.e-7 F
Tracer2 1.e-7 F
/
MINERALS
:mineral vol. frac. area
Calcite 0.1 1.
Metatorbernite 0. 1.
/
END
4.50  Keyword: UNIFORM_VELOCITY  Optional

Description:

Input:

UNIFORM_VELOCITY  vlx  vly  vlz [m/s]

Explanation: Set uniform velocity for transport mode.

Example:

UNIFORM_VELOCITY  3.84259d-6  0.d0  0.d0  ! 1.38333 cm/h
4.51 Keyword: USE_TOUCH_OPTIONS

Description:

Input:

USE_TOUCH_OPTIONS \{use\_touch\_options = .true.\}

Explanation:

Example:

\[\text{\textarrow Return to List of Keywords}\]
4.52 Keyword: VELOCITY_DATASET

Description: Set time-dependent velocity for transport mode.

Input:

VELOCITY_DATASET

UNIT cm/h
CYCLIC
INTERP step [default]
INTERP linear
VELOCITY
  Time velx vely velz
END [/ .]
END [/ .]

Explanation:

Example:

VELOCITY_DATASET
  UNIT cm/h
  CYCLIC ! cycles the data set using last time as offset
  :INTERP step [default] ! interpolation method (step [default] or linear)
  VELOCITY
    !time velx vely velz
    !time units = time unit in velocity units
    0.d0  1.38333d0  0.d0  0.d0
    12.d0 -1.38333d0  0.d0  0.d0
    24.d0  1.38333d0  0.d0  0.d0
  /
END
**4.53 Keyword: WALLCLOCK_STOP**

**WALLCLOCK_STOP** <real> <char>

**Explanation:** Specifies a wall clock time at which the simulation will shut down gracefully generating a restart file, if specified. The option is especially useful when there is an upper limit on wall clock time that can be requested (e.g. on a supercomputer) and it is not certain if the run will be completed within that time.

**Example:** WALLCLOCK_STOP 9.5 h
5 Example Input Files

5.1 Richards Equation

```
:Description: 1D test problem for tracer transport

=========================== flow mode ============================
MODE RICHARDS

=========================== chemistry ============================
CHEMISTRY
  PRIMARY_SPECIES
    Tracer
 /
END

=========================== solver options ========================
TIMESTEPPER
  MAX_STEPS 10
  TS_ACCELERATION 8
END

NEWTON_SOLVER FLOW
  :RTOL 1.d-4
  :ATOL 1.d-4
  :STOL 1.e-60
  :DTOL 1.e4
  :ITOL_UPDATE 1.d0
  :NO_INFINITY_NORM
  ::NO_PRINT_CONVERGENCE
  ::PRINT_DETAILED_CONVERGENCE
END

LINEAR_SOLVER FLOW
  KSP_TYPE GMRES
  PC_TYPE NONE
  ::KSP_TYPE PREONLY
  ::PC_TYPE LU
  ::SOLVER GMRES
END

NEWTON_SOLVER TRANSPORT
  :RTOL 1.d-4
  :ATOL 1.d-4
  :STOL 1.e-60
  :DTOL 1.e4
  :ITOL_UPDATE 1.d-4
  NO_INFINITY_NORM
  :NO_PRINT_CONVERGENCE
  ::PRINT_DETAILED_CONVERGENCE
END
```
LINEAR_SOLVER TRANSPORT
   KSP_TYPE GMRES
   PC_TYPE NONE
   :KSP_TYPE PREONLY
   :PC_TYPE LU
   :SOLVER GMRES
END

:============================ discretization =============================
GRID
   TYPE structured
   ORIGIN 0.d0 0.d0 0.d0
   NXYZ 32 32 32
   BOUNDS
      0.d0 100.d0
      0.d0 100.d0
      0.d0 100.d0
/
END

:============================ fluid properties ==============================
FLUID_PROPERTY
   DIFFUSION_COEFFICIENT 1.d-9
END

:============================ material properties ===========================
MATERIALPROPERTY soil1
   ID 1
   POROSITY 0.25d0
   TORTUOSITY 1.d0
   SATURATION_FUNCTION default
   PERMEABILITY
      PERM_X 1.d-12
      PERM_Y 1.d-12
      PERM_Z 1.d-12
/
END

MATERIALPROPERTY soil2
   ID 2
   POROSITY 0.25d0
   TORTUOSITY 1.d0
   SATURATION_FUNCTION default
   PERMEABILITY
      PERM_X 5.d-13
      PERM_Y 5.d-13
      PERM_Z 5.d-13
/
END

:============================ saturation functions ==========================
SATURATION_FUNCTION default
END

:============================ output options ===============================
OUTPUT
:MASS_BALANCE
TIMES y 0.25d0 0.5d0 0.75d0
FORMAT TECPLIT BLOCK
VELOCITIES
END

:=========================== times ============================================
TIME
FINAL_TIME 1.d0 y
INITIAL_TIMESTEP_SIZE 1.d-3 y
MAXIMUM_TIMESTEP_SIZE 1.d-1 y
END

:=========================== regions ==========================================  
REGION all
COORDINATES
  0.d0 0.d0 0.d0  
  100.d0 100.d0 100.d0
/
END

REGION top_layer
COORDINATES
  0.d0 0.d0 60.d0
  100.d0 100.d0 100.d0
/
END

REGION bottom_layer
COORDINATES
  0.d0 0.d0 0.d0
  100.d0 100.d0 60.d0
/
END

REGION west
FACE WEST
COORDINATES
  0.d0 0.d0 0.d0
  0.d0 100.d0 100.d0
/
END

REGION east
FACE EAST
COORDINATES
  100.d0 0.d0 0.d0
  100.d0 100.d0 100.d0
/
END

REGION north
FACE NORTH
COORDINATES
  0.d0 100.d0 0.d0
  100.d0 100.d0 100.d0
REGION south
  FACE SOUTH
  COORDINATES
    0.d0 0.d0 0.d0
    100.d0 0.d0 100.d0
  /
END

REGION top
  FACE TOP
  COORDINATES
    0.d0 0.d0 100.d0
    100.d0 100.d0 100.d0
  /
END

REGION bottom
  FACE BOTTOM
  COORDINATES
    0.d0 0.d0 0.d0
    100.d0 100.d0 0.d0
  /
END

REGION well
  COORDINATES
    50.d0 50.d0 50.d0
    50.d0 50.d0 50.d0
  /
END

inicio
  TYPE
    PRESSURE hydrostatic
  /
    PRESSURE 1956741.84 ! 200 meter piezometric head (200*997.32*9.81)
END

FACET source
  TYPE
    RATE volumetric_rate
  /
    RATE 10 m^3/hr
END

transportnotations
TRANSPORT CONDITION initial
  TYPE zero_gradient
  CONSTRAINT_LIST
    0.d0 initial
  /
END
TRANSPORT_CONDITION source
   TYPE dirichlet
   CONSTRAINT_LIST
      0.d0 well
   /
END

:=========================== constraints ================================
CONSTRAINT well
   CONCENTRATIONS
      Tracer 1.d0 T
   /
END

CONSTRAINT initial
   CONCENTRATIONS
      Tracer 1.d-40 T
   /
END

:=========================== condition couplers ===========================
: initial condition
INITIAL_CONDITION
   FLOW_CONDITION initial
   TRANSPORT_CONDITION initial
   REGION all
END

: west boundary condition
BOUNDARY_CONDITION west
   FLOW_CONDITION initial
   TRANSPORT_CONDITION initial
   REGION west
END

: east boundary condition
BOUNDARY_CONDITION east
   FLOW_CONDITION initial
   TRANSPORT_CONDITION initial
   REGION east
END

: north boundary condition
BOUNDARY_CONDITION north
   FLOW_CONDITION initial
   TRANSPORT_CONDITION initial
   REGION north
END

: south boundary condition
BOUNDARY_CONDITION south
   FLOW_CONDITION initial
   TRANSPORT_CONDITION initial
   REGION south
END
: top boundary condition
BOUNDARY_CONDITION top
   FLOW_CONDITION initial
   TRANSPORT_CONDITION initial
   REGION top
END

: bottom boundary condition
BOUNDARY_CONDITION bottom
   FLOW_CONDITION initial
   TRANSPORT_CONDITION initial
   REGION bottom
END

: well source/sink
SOURCE_SINK well
   FLOW_CONDITION source
   TRANSPORT_CONDITION source
   REGION well
END

: stratigraphy couplers
STRATA
   REGION top_layer
   MATERIAL soil1
END

STRATA
   REGION bottom_layer
   MATERIAL soil1
END

5.2 Carbon Sequestration Problem with Reaction with Calcite

MODE FLASH2
:MODE MPHASE

:CHECKPOINT 100
:RESTART pflotran.chk800

: discretization
GRID
   TYPE structured
   ORIGIN 0.d0 0.d0 0.d0
   NXYZ 160 160 25
   BOUNDS
      0.d0 7000.d0
      0.d0 7000.d0
      0.d0 250.d0
/

< 124 >
END

:PROC 2 1 1

:=========================== solver options ==============================
TIMESTEPPER
 :MAX_STEPS 50
 TS_ACCELERATION 8
 MAX_PRESSURE_CHANGE 5.D4
 MAX_TEMPERATURE_CHANGE 2.D0
 MAX_CONCENTRATION_CHANGE 0.005
 MAX_SATURATION_CHANGE 0.01
 /

NEWTON_SOLVER FLOW
 ATOL 1D-8
 RTOL 1D-8
 STOL 1D-30
 DTOL 1D15
 ITOL 1D-8
 MAXIT 25
 MAXF 100
END

:=========================== times ========================================
TIME
 FINAL_TIME 300.d0 y
 INITIAL_TIMESTEP_SIZE 1.d-3 y
 MAXIMUM_TIMESTEP_SIZE 0.05 y at 10 y
 MAXIMUM_TIMESTEP_SIZE 0.1 y at 100 y
 MAXIMUM_TIMESTEP_SIZE 0.25 y at 200 y
 MAXIMUM_TIMESTEP_SIZE 0.5 y at 300 y
 MAXIMUM_TIMESTEP_SIZE 1. y at 500 y
 MAXIMUM_TIMESTEP_SIZE 2.5 y at 1000 y
END

:=========================== output options ===============================
OUTPUT
 MASS_BALANCE
 TIMES y 10. 50. 100. 200. 300.
 FORMAT TECPLOT BLOCK
 :FORMAT TECPLOT POINT
 VELOCITIES
END

:=========================== fluid properties ==============================
FLUID_PROPERTY
 DIFFUSION_COEFFICIENT 1.d-9
END

:=========================== material properties ===========================
MATERIAL_PROPERTY rock2
 ID 1
 POROSITY 0.38d0
 TORTUOSITY 1d-1
ROCK_DENSITY 2.65E3
SPECIFIC_HEAT 1E3
THERMAL_CONDUCTIVITY_DRY 0.5
THERMAL_CONDUCTIVITY_WET 0.5
SATURATION_FUNCTION sf2
PERMEABILITY
   PERM_X 2.d-12
   PERM_Y 2.d-12
   PERM_Z 2.d-12
/
END

:========================== saturation functions ===========================
SATURATION_FUNCTION sf2
   PERMEABILITY_FUNCTION_TYPE PRUESS_1
   SATURATION_FUNCTION_TYPE PRUESS_1
   RESIDUAL_SATURATION LIQUID_PHASE 0.25
   RESIDUAL_SATURATION GAS_PHASE 0.1
   LAMBDA 0.75d0
   ALPHA 5d-3
   MAX_CAPILLARY_PRESSURE 1.d6
   BETAC 2.d0
   POWER 1.d0
END

:========================== regions =========================================
REGION all
   COORDINATES
      0.d0 0.d0 0.d0
      7000.d0 7000.d0 250.d0
/
END

REGION west
   FACE WEST
   COORDINATES
      0.d0 0.d0 0.d0
      0.d0 7000.d0 250.d0
/
END

REGION east
   FACE EAST
   COORDINATES
      7000.d0 0.d0 0.d0
      7000.d0 7000.d0 250.d0
/
END

REGION well
   BLOCK 77 84 77 84 5 5
   COORDINATES
      50.d0 0.d0 50.d0
      51.d0 1.d0 51.d0
/
END

< 126 >
VFLOTRAN User Manual §5.2.0 January 20, 2015

:================================ flow conditions ==================================
FLOW_CONDITION initial
  UNITS Pa,C,M,yr
  TYPE
    PRESSURE hydrostatic
    TEMPERATURE zero_gradient
    CONCENTRATION zero_gradient
  /
  IPHASE 1
  PRESSURE 2D7 2D7
  TEMPERATURE 50 C
  CONCENTRATION 1d-6 M
  ENTHALPY 0.d0 0.d0
END

skip
FLOW_CONDITION top
  UNITS Pa,C,M,yr
  TYPE
    PRESSURE dirichlet
    TEMPERATURE zero_gradient
    CONCENTRATION zero_gradient
  /
  IPHASE 1
  PRESSURE 2D7 2D7
  TEMPERATURE 50
  CONCENTRATION 1d-6
  ENTHALPY 0.d0 0.d0
END

noskip

FLOW_CONDITION side
  UNITS Pa,C,M,yr
  TYPE
    PRESSURE hydrostatic
      PRESSURE dirichlet
      TEMPERATURE zero_gradient
      CONCENTRATION zero_gradient
  /
  IPHASE 1
  PRESSURE 2D7 2D7
  TEMPERATURE 50
  CONCENTRATION 1d-6
  ENTHALPY 0.d0 0.d0
END

FLOW_CONDITION source
  UNITS Pa,C,M,yr
  TYPE
    PRESSURE mass_rate
      TEMPERATURE dirichlet
      CONCENTRATION dirichlet
  /
      PRESSURE file c2i.txt  ! kg/s
      TEMPERATURE 50.d0
CONCENTRATION 0.D0
ENTHALPY 0.d0 0.d0
END

:============================= condition couplers ==============================
: initial condition
INITIAL_CONDITION
   FLOW_CONDITION initial
   REGION all
END

: top boundary condition
BOUNDARY_CONDITION top
   FLOW_CONDITION top
   REGION top
:END

: boundary condition
BOUNDARY_CONDITION east
   FLOW_CONDITION side
   REGION east
END

BOUNDARY_CONDITION west
FLOW_CONDITION side
REGION west
END

SOURCE_SINK
   FLOW_CONDITION source
   REGION well
END

:============================= stratigraphy couplers ==============================
STRATA
   REGION all
   MATERIAL rock2
END
6 PFLOTRAN Regression Test Manager

The test manager for PFLOTRAN is a python program that is responsible for reading a configuration file, identifying the tests declared in the file, running PFLOTRAN on the appropriate input files, and then comparing the results to a known “gold standard” output file.

6.1 Running the Test Manager

The test manager can be run in two ways, either as part of the build system using “make” or manually.

There are two options for calling the test manager through make: “make check” and “make test”. The “check” target runs a small set of tests that verify that PFLOTRAN is built and running on a given system. This would be run by user to verify that their installation of PFLOTRAN is working. The “test” target runs a fuller set of regression tests intended to identify when changes to the code cause significant changes to PFLOTRAN’s results.

Calling the test manager through make relies on make variables from PETSc to determine the correct version of python to use, if PFLOTRAN was built with MPI, and optional configurations such as unstructured meshes. The version of python used to call the test manager can be changed from the command line by specifying python:

$ cd ${PFLOTRAN_DEV}/src/pflotran
$ make PYTHON=/opt/local/bin/python3.3 check

To call the test manager manually:

$ cd ${PFLOTRAN_DEV}/regression_tests
$ python regression-tests.py \
    --executable ../src/pflotran/pflotran \
    --config-file shortcourse/copper_leaching/cu_leaching.cfg \
    --tests cu_leaching

Some important command line arguments when running manually are:

- executable : the path to the PFLOTRAN executable
- mpiexec : the name of the executable for launching parallel jobs, (mpiexec, mpirun, aprun, etc).
- config-file : the path to the configuration file containing the tests you want to run
- recursive-search : the path to a directory. The test manager searches the directory and all its sub-directories for configuration files.
• tests : a list of test names that should be run

• suites : a list of test suites that should be run

• update : indicate that the gold standard test file for a given test should be updated to the current output.

• new-tests : indicate that the test is new and current output should be used for gold standard test file.

• check-performance : include the performance metrics ('SOLUTION' blocks) in regression checks.

The full list of command line options and a brief description can be found by running with the “–help” flag:

$ python regression-tests.py --help

6.2 Test output

The test manager produces terse screen output, only printing critical warnings, a progress bar as each test is run, and a summary of the overall test results. Example results from running make test are:

Test log file : pflotran-tests-2013-05-06_13-07-05.testlog

** WARNING ** : mpiexec was not provided on the command line.
All parallel tests will be skipped!

Running pflotran regression tests :
..........S.......SSSS.......F..........SSS

Regression test summary:
   Total run time: 179.259 [s]
   Total tests : 44
   Skipped : 8
   Tests run : 36
   Failed : 1

The progress bar records a period, “.”, for each successful test, an “S” if a test was skipped, an “F” for failed tests, a “W” for a test that generated a warning, and an “E” for a test that generated an internal error.

Each time the test suite is run, a log file is generated in the regression test directory. The log file contains a detailed record of every test run, including: the directory containing the test, the
command line call to PFLOTRAN used to run the test, a diff command to compare the regression files, and a list of failures. The log file can quickly be searched for “skip”, “fail”, “error” to identify the tests that generated the message.

The test directories contain any files generated by PFLOTRAN during the run. Screen output for each test is contained in the file “${TEST_NAME}.stdout”.

### 6.3 Configuration Files

The regression test manager reads tests specified in a series of configuration files in standard “cfg” (or windows “ini” file) format. They consist of a series of sections with key-value pairs:

```
[section-name]
key = value
```

Section names should be all lower case, and spaces must be replaced by a hyphen or underscore. Comments are specified by a “#” character.

A test is declared as a section in the configuration file. It is assumed that there will be a PFLOTRAN input file with the same name as the test section. The key-value pairs in a test section define how the test is run and the output is compared to the gold standard file.

```
[calcite-kinetics]
#look for an input file named ‘calcite-kinetics.in’
np = 2
timeout = 30.0
concentration = 1.0e-10 absolute
```

- np = N, (optional), indicates a parallel test run with N processors. Default is serial. If mpiexec is not provided on the command line, then parallel tests are skipped.
- timeout = N, (optional), indicates that the test should be allowed to run for N seconds before it is killed. Default is 60.0 seconds.
- TYPE = TOLERANCE COMPARISON, indicates that data in the regression file of type TYPE should be compared using a tolerance of TOLERANCE. Known data types are listed below.

The data types and default tolerances are:

- time = 5 percent
- concentration = $1 \times 10^{-12}$ absolute
- generic = $1 \times 10^{-12}$ absolute
• discrete = 0 absolute
• rate = \(1 \times 10^{-12}\) absolute
• volume_fraction = \(1 \times 10^{-12}\) absolute
• pressure = \(1 \times 10^{-12}\) absolute
• saturation = \(1 \times 10^{-12}\) absolute
• residual = \(1 \times 10^{-12}\) absolute

The default tolerances are deliberately set very tight, and are expected to be overridden on a per-test or per configuration file basis. There are three known comparisons: “absolute”, for absolute differences (\(\delta = |c-g|\)), “relative” for relative differences (\(\delta = |c-g|/g\)), and “percent” for specifying a percent difference (\(\delta = 100 \cdot |c-g|/g\)).

In addition there are two optional sections in configuration files. The section “default-test-criteria” specifies the default criteria to be used for all tests in the current file. Criteria specified in a test section override these value. A section name “suites” defines aliases for a group of tests.

```
[suites]
serial = test-1 test-2 test-3
parallel = test-4 test-5 test-6
```

Common test suites are standard and standard_parallel, used by “make test”, and domain specific test suites, geochemistry, flow, transport, mesh, et cetra.

### 6.4 Creating New Tests

We want running tests to become a habit for developers so that “make pflotran” is always followed by “make test.” With that in mind, ideal test cases are small and fast, and operate on a small subsection of the code so it is easier to diagnose where a problem has occurred. While it may (will) be necessary to create some platform specific tests, we want as many tests as possible to be platform independent and widely used. There is a real danger in having test output become stale if it requires special access to a particular piece of hardware, operating system or compiler to run.

The steps for creating new regression tests are:

- Create the PFLOTRAN input file, and get the simulation running correctly.
- Tell PFLOTRAN to generate a regression file by adding a regression block to the input file, e.g.:

```
REGRESSION
   CELLS
   1
```
6.5 Updating Test Results

The output from PFLOTRAN should be fairly stable, and we consider the current output to be “correct”. Changes to regression output should be rare, and primarily done for bug fixes. Updating the test results is simply a matter of replacing the gold standard file with a new file. This can be done with a simple rename in the file system:

```
mv test_1.regression test_1.regression.gold
```

Or using the regression test manager:

```
$ python regression-tests.py --executable ../src/pflotran/pflotran \   --config-file my_test.cfg --tests test_1 --update
```

Updating through the regression test manager ensures that the output is from your current executable rather than a stale file.

Please document why you updated gold standard files in your revision control commit message.
7 Visualization

Visualization of the results produced by PFLOTRAN can be achieved using several different utilities including commercial and open source software. Plotting 2D or 3D output files can be done using the commercial package Tecplot, or the opensource packages VisIt and ParaView. Paraview is similar to VisIt and both are capable of remote visualization on parallel architectures. In order to change the default file format in TecPlot so that it recognizes .tec files, place the following in the tecplot.cfg file:

```bash
$!FileConfig FNameFilter {InputDataFile = "*.tec"}
```

For 1D problems or for plotting PFLOTRAN observation and mass balance output, the opensource software packages gnuplot and matplotlib are recommended. With gnuplot and matplotlib it is possible to plot data from several files in the same plot. To do this with gnuplot it is necessary that the files have the same number of rows, e.g. time history points. The files can be merged during input by using the paste command as a pipe: e.g.

```bash
plot '< paste file1.dat file2.dat' using 1:($n1*$n2)
```

plots the product of variable in file 1 in column n1 times the variable in file 2 in column n2 of the merged file.
8 Benchmark Problems

In this section several benchmark problems are introduced illustrating the capabilities of PFLOTRAN.

8.1 Ion Exchange

Voegelin et al. (2000) present results of ion exchange in a soil column for the system Ca-Mg-Na. Here PFLOTRAN is applied to this problem using the Gaines-Thomas exchange model. Soil column C1 with length 48.1 cm and diameter 0.3 cm was used for the simulations. A flow rate of 5.6 cm/min was used in the experiment. The inlet solution was changed during the course of the experiment at 20 and 65 pore volumes with cation compositions listed in Table 2 of Voegelin et al. (2000). The CEC of the soil used in the experiments was determined to have a value of 0.06±0.002 mol/kg. As PFLOTRAN requires the CEC in units of mol/m³ this was obtained from the formula

\[ \omega = \frac{N_s}{V} = \frac{N_s M_s V_s}{M_s V_s V}, \]
\[ = \rho_s (1 - \varphi) \text{CEC}. \] (8-12a)

Using a porosity of 0.61 and solid grain density of 3.0344 g/cm³, yielded \( \omega = 71.004 \text{ mol/m³} \). The results of the simulation are shown in Figure 1 along with data reported by Voegelin et al. (2000). Self-sharpening fronts can be observed at approximately 10 and 71 at pore volumes, and a self-broadening front from 30-55 pore volumes where agreement with experiment is not as good.

The input file for the simulation is listed in Table 8.1.
Figure 1: Breakthrough curves for Ca\textsuperscript{2+}, Mg\textsuperscript{2+} and Na\textsuperscript{+} compared with experimental results from Voegelin et al. (2000).

```plaintext
#Description: 3D 5-spot copper leaching problem

# == debugging ================================================================
#DEBUG
#MATVIEW_JACOBIAN
#VECVIEW_RESIDUAL
#VECVIEW_SOLUTION
#/

# == mode =====================================================================
#MODE RICHARDS

# m/s
UNIFORM_VELOCITY 5.69333e-4 0.d0 0.d0

#=========================== runtime ==========================================
#CHECKPOINT 100
#WALLCLOCK_STOP 11.75

# == chemistry ================================================================
CHEMISTRY
#OPERATOR_SPLIT
PRIMARY_SPECIES
Na+
#K+
Ca++
```

< 136 >
Mg++
H+
HCO3-
Cl-
Tracer
/
SECONDARY_SPECIES
OH-
CO3--
CO2(aq)
CaOH+
CaCO3(aq)
CaHCO3+
CaCl+
MgCO3(aq)
MgHCO3+
MgCl+
HCl(aq)
#KCl(aq)
NaCl(aq)
NaOH(aq)
/
GAS_SPECIES
CO2(g)
/
MINERALS
Halite
/
# MINERAL_KINETICS
Halite
RATE_CONSTANT 1.e-30
/
/
SORPTION
ION_EXCHANGE_RXN
MINERAL Halite
#CEC 68.676 ! mol/m^3
CEC 71.004
CATIONS
Na+ 7.94328
Ca++ 1.
Mg++ 1.44544
/
/
DATABASE /Users/lichtner/projects/parallel/repository/pflotran/database/hanford.dat
LOG_FORMULATION
ACTIVITY_COEFFICIENTS ! NEWTON_ITERATION
MOLAL
OUTPUT
All
/
/
# == reference variables =====================================================
REFERENCE_POROSITY 0.61d0

# == time stepping ============================================================
TIMESTEPPER
TS_ACCELERATION 8
MAX_STEPS 100000
#MAX_STEPS 1
/

# == discretization ===========================================================
GRID
TYPE structured
NXYZ 250 1 1
BOUNDS
0.d0 0.d0 0.d0
0.481d0 1.d0 1.d0
/
/

# == flow solvers =============================================================
NEWTON_SOLVER FLOW
PRECONDITIONER_MATRIX_TYPE AIJ
RTOL 1.d-8
ATOL 1.d-8
STOL 1.d-30
ITOL_UPDATE 1.d0
#NO_INFINITY_NORM
#NO_PRINT_CONVERGENCE
#PRINT_DETAILED_CONVERGENCE
/

LINEAR_SOLVER FLOW
#KSP_TYPE PREONLY
#PC_TYPE LU
#KSP_TYPE FGMRES !samrai
#PC_TYPE SHELL !samrai
/

# == transport solvers ========================================================
NEWTON_SOLVER TRANSPORT
PRECONDITIONER_MATRIX_TYPE AIJ
RTOL 1.d-8
ATOL 1.d-8
STOL 1.d-30
#NO_INFINITY_NORM
#NO_PRINT_CONVERGENCE
#PRINT_DETAILED_CONVERGENCE
/

LINEAR_SOLVER TRANSPORT
#PC_TYPE LU
#KSP_TYPE PREONLY
#KSP_TYPE FGMRES !samrai
#PC_TYPE SHELL !samrai
/
# == fluid properties =========================================================
FLUID_PROPERTY
DIFFUSION_COEFFICIENT 1.d-9
/*DIFFUSION_COEFFICIENT 9.33333e-7
*/

# == material properties ======================================================
MATERIAL_PROPERTY HD
ID 1
SATURATION_FUNCTION default
POROSITY 0.61
TORTUOSITY 1.0
/*LONGITUDINAL_DISPERSIVITY 0.001
PERMEABILITY
PERM_ISO 5.43d-13
*/

# == saturation / permeability functions ========================================
SATURATION_FUNCTION HD
SATURATION_FUNCTION_TYPE VAN_GENUCHTEN
RESIDUAL_SATURATION 0.115
LAMBDA 0.286
ALPHA 1.9401d-4
/

#=========================== saturation functions ==============================
SATURATION_FUNCTION default
/

# == output ====================================================================
OUTPUT
TIMES s 10307.1 33498.2 41228.6
PERIODIC_OBSERVATION TIMESTEP 1
/*PERIODIC TIMESTEP 1
PERIODIC TIME 0.04 y
SCREEN PERIODIC 10
FORMAT HDF5
FORMAT TECPlot POINT
VELOCITIES
*/

# == times ====================================================================
TIME
FINAL_TIME 41228.6 s
INITIAL_TIMESTEP_SIZE 1. s
MAXIMUM_TIMESTEP_SIZE 20. s
MAXIMUM_TIMESTEP_SIZE 1. s at 10200. s
MAXIMUM_TIMESTEP_SIZE 20. s at 10350 s
MAXIMUM_TIMESTEP_SIZE 1. s at 33300 s
MAXIMUM_TIMESTEP_SIZE 20. s at 33600 s
/

# == regions ==================================================================
REGION all
COORDINATES

< 139 >
0.d0 0.d0 0.d0  
0.481d0 1.d0 1.d0  
/
/

REGION west
FACE WEST
COORDINATES
0. 0. 0.
0. 1. 1.
/
/

REGION east
FACE EAST
COORDINATES
0.481 0. 0.
0.481 1. 1.
/
/

OBSERVATION
REGION east
/
/

# == flow conditions=================================================================

skip
FLOW_CONDITION west
TYPE
FLUX neumann
/
FLUX 0.317098d-7 ! 1 m/y
#FLUX 1.5855d-9 ! 5 cm/y
#FLUX file 200w_recharge_1951-2000_daily.dat
/

noskip

FLOW_CONDITION Initial
TYPE
PRESSURE hydrostatic
/
DATUM 0.d0 0.d0 0.d0
PRESSURE 101325.d0
/

FLOW_CONDITION west
TYPE
PRESSURE hydrostatic
/
DATUM 0.d0 0.d0 0.d0
PRESSURE 101425.d0
END

FLOW_CONDITION east
TYPE
PRESSURE hydrostatic
/ 
DATUM 0.d0 0.d0 0.d0
PRESSURE 101325.d0
END

# == transport conditions =========================================================
TRANSPORT_CONDITION Initial
TYPE dirichlet
CONSTRAINT_LIST
0.d0 Initial
/
/

TRANSPORT_CONDITION east
TYPE dirichlet
CONSTRAINT_LIST
0.d0 Initial
/
/

TRANSPORT_CONDITION west
TYPE dirichlet
CONSTRAINT_LIST
0.d0 Inlet1
10307.1 Inlet2
33498.2 Inlet3
/
/

# == couplers ==============================================================
INITIAL_CONDITION Initial
#FLOW_CONDITION Initial
TRANSPORT_CONDITION Initial
REGION all
/
/

BOUNDARY_CONDITION
#FLOW_CONDITION west
TRANSPORT_CONDITION west
REGION west
END
/

BOUNDARY_CONDITION
#FLOW_CONDITION east
TRANSPORT_CONDITION east
REGION east
END
/

# == stratigraphy ==============================================================
STRATA
MATERIAL HD
REGION all
/
/

# == transport constraints =======================================================
CONSTRAINT Initial
CONCENTRATIONS
Na+   4.65d-3 T
#K+   2.d-4  T
Ca++  5.2d-3 T
Mg++  4.55e-3 T
H+    4.6   pH
HCO3- -3.5   G CO2(g)
Cl-   1.d-3  Z
Tracer 4.65d-3 T
/
MINERALS
Halite 0.5 1.
/
/

CONSTRAINT Inlet1
CONCENTRATIONS
Na+ 1.d-16 T
#K+ 1.d-10  T
Ca++ 5.3d-3 T
Mg++ 1.e-16 T
H+ 4.6  pH
HCO3- -3.5   G CO2(g)
Cl- 3.d-4  Z
Tracer 9.4d-3 T
/
/

CONSTRAINT Inlet2
CONCENTRATIONS
Na+ 4.6d-3 T
#K+ 1.d-10  T
Ca++ 1.d-16 T
Mg++ 2.4e-3 T
H+ 4.6  pH
HCO3- -3.5   G CO2(g)
Cl- 3.d-4  Z
Tracer 9.4d-3 T
/
/

CONSTRAINT Inlet3
CONCENTRATIONS
Na+ 4.65d-3 T
#K+ 1.d-10  T
Ca++ 5.2d-3 T
Mg++ 4.55e-3 T
H+ 4.6  pH
HCO3- -3.5   G CO2(g)
Cl- 3.d-4  Z
Tracer 9.4d-3 T
/
/

< 142 >
8.2 SX-115 Hanford Tank Farm

8.2.1 Problem Description

The saturation profile is computed for both steady-state and transient conditions in a 1D vertical column consisting of a layered porous medium representing the Hanford sediment in the vicinity of the S/SX tank farm. The transient case simulates a leak from the base of the SX-115 tank. This problem description is taken from Lichtner et al. (2004).

8.2.2 Governing Equations

The moisture profile is calculated using parameters related to the Hanford sediment at the S/SX tank farm based on the Richards equation for variably saturated porous media. The Hanford sediment is composed of five layers with the properties listed in Tables 6 and 7. The governing equations consist of Richards equation for variably saturated fluid flow given by

\[
\frac{\partial}{\partial t} \varphi s \rho + \nabla \cdot q \rho = Q, \tag{8-13}
\]

and solute transport of a tracer

\[
\frac{\partial}{\partial t} \varphi C + \nabla \cdot (q C - \varphi s \tau D \nabla C) = Q_C. \tag{8-14}
\]

In these equations \( \varphi \) denotes the spatially variable porosity of the porous medium assumed to constant within each stratigraphic layer, \( s \) gives the saturation state of the porous medium, \( \rho \) represents the fluid density in general a function of pressure and temperature, \( C \) denotes the solute concentration, \( D \) denotes the diffusion/dispersion coefficient, \( \tau \) represents tortuosity, \( Q \) and \( Q_C \) denote source/sink terms, and \( q \) denotes the Darcy velocity defined by

\[
q = - \frac{k_{\text{sat}} k_r}{\mu} \nabla (p - \rho g z), \tag{8-15}
\]

with saturated permeability \( k_{\text{sat}} \), relative permeability \( k_r \), fluid viscosity \( \mu \), pressure \( p \), formula weight of water \( W \), acceleration of gravity \( g \), and height \( z \). Van Genuchten capillary properties are used for relative relative permeability according to the relation

\[
k_r = \sqrt{s_{\text{eff}}} \left\{ 1 - \left[ 1 - (s_{\text{eff}})^{1/m} \right]^m \right\}^2, \tag{8-16}
\]

where \( s_{\text{eff}} \) is related to capillary pressure \( P_c \) by the equation

\[
s_{\text{eff}} = [1 + (\alpha |P_c|)^n]^{-m}, \tag{8-17}
\]

where \( s_{\text{eff}} \) is defined by

\[
s_{\text{eff}} = \frac{s - s_r}{1 - s_r}, \tag{8-18}
\]

and where \( s_r \) denotes the residual saturation. The quantity \( n \) is related to \( m \) by the expression

\[
m = 1 - \frac{1}{n}, \quad n = \frac{1}{1 - m}. \tag{8-19}
\]

The capillary pressure \( P_c \) and fluid pressure \( p \) are related by the (constant) gas pressure \( p_g^0 \)

\[
P_c = p_g^0 - p, \tag{8-20}
\]

where \( p_g^0 = 101,325 \) Pa is set to atmospheric pressure.
8.2.2.1 Semi-Analytical Solution for Steady-State Conditions

For steady-state conditions the saturation profile satisfies the equation

$$\frac{d}{dz} \rho q_z = 0,$$  \hspace{1cm} (8-21)

or assuming an incompressible fluid

$$q_z = q_z^0,$$  \hspace{1cm} (8-22)

where \( q_z^0 \) denotes infiltration at the surface. Thus the pressure is obtained as a function of \( z \) by solving the ODE

$$\frac{dp}{dz} = -\frac{\mu q_z^0}{k_{sat} k_r} - \rho g,$$  \hspace{1cm} (8-23)

using Eqns.(8-16) and (8-17) to express the relative permeability \( k_r \) as a function of pressure. For the special case of zero infiltration it follows that

$$p(z) = p_0 - \rho g (z - z_0),$$  \hspace{1cm} (8-24)

with \( p(z_0) = p_0 \). The saturation profile is obtained from Eqns.(8-17) and (10-4).

8.2.2.2 Watertable

The position of the watertable is defined by vanishing of the capillary pressure

$$P_c(z_{wt}) = 0,$$ \hspace{1cm} (8-25)

where \( z_{wt} \) denotes the height of the watertable. For the case with no infiltration at the surface it follows that

$$z_{wt} = z_0 + \frac{p_0 - p_g}{\rho g},$$ \hspace{1cm} (8-26)

with the boundary condition \( p(z_0) = p_0 \) and \( z_0 \) denotes the datum. If \( p_0 \) is set equal to \( p_g \), then \( z_{wt} = z_0 \), or the height of the watertable is equal to the datum. The same holds true also with constant nonzero infiltration.

8.2.3 Model Parameters

Model parameters used in the simulations are listed in Tables 6 and 7. Although not needed here, thermal properties are also listed. Diffusivity was set to \( 10^{-9} \text{ m}^2 \text{ s}^{-1} \) and tortuosity was set to one.

<table>
<thead>
<tr>
<th>Formation</th>
<th>Abbrev.</th>
<th>Thickness [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backfill</td>
<td>BF</td>
<td>16.0</td>
</tr>
<tr>
<td>Hanford Fine Sand</td>
<td>HF</td>
<td>23.0</td>
</tr>
<tr>
<td>Plio-Pleistocene</td>
<td>PP</td>
<td>6.0</td>
</tr>
<tr>
<td>Upper Ringold Gravel</td>
<td>URG</td>
<td>3.0</td>
</tr>
<tr>
<td>Middle Ringold Gravel</td>
<td>MRG</td>
<td>20.0</td>
</tr>
</tbody>
</table>
Table 7: Parameters for material and thermal properties for intrinsic rock density $\rho_s$, heat capacity $c$, thermal conductivity $\kappa$, porosity $\varphi$, residual water saturation $s_r$, van Genuchten parameters $\alpha$ and $\lambda$, and vertical water saturated permeability $k_{sat}$. Data taken from Khaleel and Freeman (1995), Khaleel et al. (2001), and Pruess et al. (2002).

<table>
<thead>
<tr>
<th>Formation</th>
<th>$\rho_s$</th>
<th>$c$</th>
<th>$\kappa_{dry}$</th>
<th>$\kappa_{wet}$</th>
<th>$\varphi$</th>
<th>$s_r$</th>
<th>$\alpha$</th>
<th>$m$</th>
<th>$k_{sat}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BF</td>
<td>2.8</td>
<td>800</td>
<td>0.5</td>
<td>2</td>
<td>0.2585</td>
<td>0.0774</td>
<td>1.008e-3</td>
<td>0.6585</td>
<td>1.240e-12</td>
</tr>
<tr>
<td>HF</td>
<td>2.8</td>
<td>800</td>
<td>0.5</td>
<td>2</td>
<td>0.3586</td>
<td>0.0837</td>
<td>9.408e-5</td>
<td>0.4694</td>
<td>3.370e-13</td>
</tr>
<tr>
<td>PP</td>
<td>2.8</td>
<td>800</td>
<td>0.5</td>
<td>2</td>
<td>0.4223</td>
<td>0.2595</td>
<td>6.851e-5</td>
<td>0.4559</td>
<td>3.735e-14</td>
</tr>
<tr>
<td>URG</td>
<td>2.8</td>
<td>800</td>
<td>0.5</td>
<td>2</td>
<td>0.2625</td>
<td>0.2130</td>
<td>2.966e-5</td>
<td>0.3859</td>
<td>1.439e-13</td>
</tr>
<tr>
<td>MRG</td>
<td>2.8</td>
<td>800</td>
<td>0.5</td>
<td>2</td>
<td>0.1643</td>
<td>0.0609</td>
<td>6.340e-5</td>
<td>0.3922</td>
<td>2.004e-13</td>
</tr>
</tbody>
</table>

8.2.4 Simulation Results

The calculations are carried out for an isothermal system using Richards equation. First, the steady-state saturation profile is obtained without the tank leak present. Then using the steady-state profile as the initial condition the tank leak is turned on. The results for the steady-state saturation and pressure profiles are shown in Figure 2 for infiltration rates at the surface of 0, 8 and 80 mm/y. The mean infiltration rate at the Hanford site is approximately 8 mm/y. A 1D column 68 m height with the water table located at a height of 6 m from the bottom is used in the simulation. A uniform grid spacing of 0.5 m is used to discretize Richards equation.

Shown in Figure 3 is the saturation at different times following a two week leak releasing 60,000 gallons from the SX-115 tank at a depth of 16 m. In the simulation a release rate of $1.87 \times 10^{-3}$ kg/s is used.
Figure 2: Steady-state saturation and pressure profiles for infiltration rates of 0, 8 and 80 mm/y. The water table is located at 6 m from the bottom of the computational domain.
Figure 3: Simulation of a tank leak with a duration of two weeks showing the saturation profile for different times indicated in the figure.

Figure 4: The solute concentration profile corresponding to Figure 7 for different times indicated in the figure.
8.2.5 PFLOTRAN Input Files

Listing of PFLOTRAN input file including a tracer. Note that the stratigraphic zone specification in REGION is grid independent as is the grid size specification in keyword GRID. Therefore to change the grid spacing only the line: NXYZ 1 1 136, needs to be changed. Also note that a line beginning with # is read as a comment.

PFLOTRAN input file pflotran.in:

```plaintext
#Description: 1D test problem for tracer transport

#=========================== flow mode ========================================
MODE RICHARDS

#=========================== chemistry ========================================
CHEMISTRY
   PRIMARY_SPECIES
     Tracer
/
END

#=========================== runtime ========================================
CHECKPOINT 100000
#RESTART restart.chk 0.d0
#OVERWRITE_RESTART_TRANSPORT
#WALLCLOCK_STOP 11.75

#=========================== solver options ===================================
TIMESTEPPER
   #MAX_STEPS -1
   TS_ACCELERATION 8
   INITIALIZE_TO_STEADY_STATE 1.d0
END

NEWTON_SOLVER FLOW
   #RTOL 1.d-12
   RTOL 1.d-20
   #ATOL 1.d-12
   #STOL 1.e-60
   #DTOL 1.e4
   ITOL_UPDATE 1.d0
   #NO_INFINITY_NORM
   #NO_PRINT_CONVERGENCE
   #PRINT_DETAILED_CONVERGENCE
END

LINEAR_SOLVER FLOW
   #KSP_TYPE GMRES
   #PC_TYPE NONE
   #KSP_TYPE PREONLY
   #PC_TYPE LU
   #SOLVER GMRES
END
```
NEWTON_SOLVER TRANSPORT
  RTOL 1.d-12
  ATOL 1.d-12
  STOL 1.e-60
  DTOL 1.e4
  #ITOL_UPDATE 1.d-4
  #NO_INFINITY_NORM
  #NO_PRINT_CONVERGENCE
  #PRINT_DETAILED_CONVERGENCE
END

LINEAR_SOLVER TRANSPORT
  #KSP_TYPE GMRES
  #PC_TYPE NONE
  #KSP_TYPE PREONLY
  #PC_TYPE LU
  #SOLVER GMRES
END

#=========================== discretization ===================================
GRID
  TYPE structured
  ORIGIN 0.d0 0.d0 0.d0
  NXYZ 1 1 136
  BOUNDS
    0.d0 0.d0 0.d0
    1.d0 1.d0 68.d0
/
END

#=========================== fluid properties ==================================
FLUID_PROPERTY
  DIFFUSION_COEFFICIENT 1.d-9
END

#=========================== material properties ==============================
MATERIAL_PROPERTY Backfill
  ID 1
  POROSITY 0.2585d0
  TORTUOSITY 0.5d0
  SATURATION_FUNCTION BF
  PERMEABILITY
    PERM_X 1.24e-12
    PERM_Y 1.24e-12
    PERM_Z 1.24e-12
/
END

MATERIAL_PROPERTY Hanford-Fine-Sand
  ID 2
  POROSITY 0.3586
  TORTUOSITY 0.5d0
  SATURATION_FUNCTION HF
  PERMEABILITY
    PERM_X 3.37028e-13
    PERM_Y 3.37028e-13

< 149 >
PERM_Z 3.37028e-13
/
END

MATERIALPROPERTY Plio-Pleistocene
   ID 3
   POROSITY 0.4223d0
   TORTUOSITY 0.5d0
   SATURATIONFUNCTION PP
   PERMEABILITY
      PERM_X 3.73463e-14
      PERM_Y 3.73463e-14
      PERM_Z 3.73463e-14
   /
   END

MATERIALPROPERTY Upper-Ringold-Gravel
   ID 4
   POROSITY 0.2625d0
   TORTUOSITY 0.5d0
   SATURATIONFUNCTION URG
   PERMEABILITY
      PERM_X 1.4392e-13
      PERM_Y 1.4392e-13
      PERM_Z 1.4392e-13
   /
   END

MATERIALPROPERTY Middle-Ringold-Gravel
   ID 5
   POROSITY 0.1643
   TORTUOSITY 0.5d0
   SATURATIONFUNCTION MRG
   PERMEABILITY
      PERM_X 2.00395e-13
      PERM_Y 2.00395e-13
      PERM_Z 2.00395e-13
   /
   END

#=========================== saturation functions =============================
SATURATIONFUNCTION BF
   SATURATIONFUNCTIONTYPE VAN_GENUCHTEN
   PERMEABILITYFUNCTIONTYPE MUalem
   RESIDUALSATURATION 0.0774d0
   LAMBDA 0.6585d0
   ALPHA 1.008d-3
   MAX_CAPILLARY_PRESSURE 1.d6
   BETAC 0.d0
   POWER 1.d0
   END

SATURATIONFUNCTION HF
   SATURATIONFUNCTIONTYPE VAN_GENUCHTEN
   PERMEABILITYFUNCTIONTYPE MUalem
   RESIDUALSATURATION 0.08366d0

< 150 >
LAMBDA 0.46944d0
ALPHA 9.40796d-5
MAX_CAPILLARY_PRESSURE 1.d6
BETAC 0.d0
POWER 1.d0
END

SATURATION_FUNCTION_PP
    SATURATION_FUNCTION_TYPE VAN_GENUCHTEN
    PERMEABILITY_FUNCTION_TYPE MUALEM
    RESIDUAL_SATURATION 0.25953d0
    LAMBDA 0.45587d0
    ALPHA 6.85145d-5
    MAX_CAPILLARY_PRESSURE 1.d6
    BETAC 0.d0
    POWER 1.d0
END

SATURATION_FUNCTION_URG
    SATURATION_FUNCTION_TYPE VAN_GENUCHTEN
    PERMEABILITY_FUNCTION_TYPE MUALEM
    RESIDUAL_SATURATION 0.21295d0
    LAMBDA 0.38594d0
    ALPHA 2.96555d-5
    MAX_CAPILLARY_PRESSURE 1.d6
    BETAC 0.d0
    POWER 1.d0
END

SATURATION_FUNCTION_MRG
    SATURATION_FUNCTION_TYPE VAN_GENUCHTEN
    PERMEABILITY_FUNCTION_TYPE MUALEM
    RESIDUAL_SATURATION 0.06086d0
    LAMBDA 0.39217d0
    ALPHA 6.34015e-5
    MAX_CAPILLARY_PRESSURE 1.d6
    BETAC 0.d0
    POWER 1.d0
END

#=============== output options ================================
OUTPUT
    #SCREEN PERIODIC 10
    #MASS_BALANCE
    #TIMES y 0.5 1.0 2.0 5.0 10.0
    TIMES y 100. 250. 500. 750. 1000.
    FORMAT TECPLOT POINT
    VELOCITIES
END

#=============== times ================================
TIME
    FINAL_TIME 1000.d0 y
    INITIAL_TIMESTEP_SIZE 1.d-6 y
    MAXIMUM_TIMESTEP_SIZE 1.d-0 y
END
#=========================== regions =========================================
REGION all
  COORDINATES
    0.d0 0.d0 0.d0
    1.d0 1.d0 136.d0
/
END

REGION MRG
  COORDINATES
    0.d0 0.d0 0.d0
    1.d0 1.d0 20.d0
/
END

REGION URG
  COORDINATES
    0.d0 0.d0 20.d0
    1.d0 1.d0 23.d0
/
END

REGION PP
  COORDINATES
    0.d0 0.d0 23.d0
    1.d0 1.d0 29.d0
/
END

REGION HF
  COORDINATES
    0.d0 0.d0 29.d0
    1.d0 1.d0 52.d0
/
END

REGION BF
  COORDINATES
    0.d0 0.d0 52.d0
    1.d0 1.d0 68.d0
/
END

#================= boundaries ==================
REGION west
  FACE WEST
  COORDINATES
    0.d0 0.d0 0.d0
    0.d0 1.d0 68.d0
/
END

REGION east
  FACE EAST
COORDINATES
  1.d0 0.d0 0.d0
  1.d0 1.d0 68.d0
/
END

REGION north
  FACE NORTH
  COORDINATES
    0.d0 1.d0 0.d0
    1.d0 1.d0 68.d0
/
END

REGION south
  FACE SOUTH
  COORDINATES
    0.d0 0.d0 0.d0
    1.d0 0.d0 68.d0
/
END

REGION top
  FACE TOP
  COORDINATES
    0.d0 0.d0 68.d0
    1.d0 1.d0 68.d0
/
END

REGION bottom
  FACE BOTTOM
  COORDINATES
    0.d0 0.d0 0.d0
    1.d0 1.d0 0.d0
/
END

REGION well
  COORDINATES
    1.d0 1.d0 52.d0
    1.d0 1.d0 52.d0
/
END

#================================ flow conditions ===============================
FLOW_CONDITION initial
  TYPE
    PRESSURE hydrostatic
  /
    DATUM 0.d0 0.d0 6.d0
    PRESSURE 101325.d0
END

FLOW_CONDITION infiltration
  TYPE

< 153 >
FLUX neumann
/
#FLUX 2.53678e-9 ! 0.08 m/yr
#FLUX 2.53678e-10 ! 8 mm/yr
FLUX 0.d0
END

FLOW_CONDITION water_table
  TYPE
   PRESSURE hydrostatic
 /
  DATUM 0.d0 0.d0 6.d0
  PRESSURE 101325.d0
  #PRESSURE 1.4e5 ! 200 meter piezometric head (200*997.32*9.81)
END

FLOW_CONDITION source
  TYPE
   RATE mass_rate
 /
  #RATE 8.5d-4 ! kg/s
  RATE file src.dat ! 8.4e-4 kg/s for 2 weeks (60,000 gallons)
END

#============== transport conditions ========================
TRANSPORT_CONDITION initial
  TYPE zero_gradient
  CONSTRAINT_LIST
   0.d0 initial
 /
END

TRANSPORT_CONDITION boundary
  TYPE zero_gradient
  CONSTRAINT_LIST
   0.d0 initial
 /
END

TRANSPORT_CONDITION infiltration
  TYPE dirichlet
  CONSTRAINT_LIST
   0.d0 infiltration
 /
END

TRANSPORT_CONDITION source
  TYPE dirichlet
  CONSTRAINT_LIST
   0.d0 well
 /
/

#=================== condition couplers ======================
# initial condition
INITIAL_CONDITION
FLOW_CONDITION initial
TRANSPORT_CONDITION initial
REGION all
END

# top boundary condition
BOUNDARY_CONDITION top
  #FLOW_CONDITION initial
  FLOW_CONDITION infiltration
  TRANSPORT_CONDITION initial
  REGION top
END

# bottom boundary condition
BOUNDARY_CONDITION bottom
  FLOW_CONDITION water_table
  TRANSPORT_CONDITION initial
  REGION bottom
END

# well source/sink
skip
SOURCE_SINK well
  FLOW_CONDITION source
  TRANSPORT_CONDITION source
  REGION well
END

noskip

# infiltration source/sink
skip
SOURCE_SINK infil
  FLOW_CONDITION infiltration
  TRANSPORT_CONDITION infiltration
  REGION top
END

noskip

#=========================== stratigraphy couplers ============================
STRATA
  REGION MRG
  MATERIAL Middle-Ringold-Gravel
END

STRATA
  REGION URG
  MATERIAL Upper-Ringold-Gravel
END

STRATA
  REGION PP
  MATERIAL Plio-Pleistocene
END

STRATA
  REGION HF
MATERIAL Hanford-Fine-Sand
END

STRATA
  REGION BF
  MATERIAL Backfill
END

skip
STRATA
  REGION all
  MATERIAL Middle-Ringold-Gravel
END

noskip

#================================== constraints ================================

CONSTRAINT well
  CONCENTRATIONS
    Tracer 1.d0 T
  /
END

CONSTRAINT infiltration
  CONCENTRATIONS
    Tracer 1.d0 T
  /
END

CONSTRAINT initial
  CONCENTRATIONS
    Tracer 1.d-16 T
  /
END
Source/sink file src.dat:

```
#time rate [kg/s]
0.d0 0.187e-4
1.21292e6 0.187e-4
1.21293e6 0.
1.e12 0.
```
9 References


10 Appendix: Governing Equations

10.1 Mode: RICHARDS

RICHARDS Mode applies to single phase, variably saturated, isothermal systems. The governing mass conservation equation is given by

\[
\frac{\partial}{\partial t} (\varphi s \rho) + \nabla \cdot (\rho q) = Q_w, \tag{10-1}
\]

with Darcy flux \( q \) defined as

\[
q = -\frac{k k_r(s)}{\mu} \nabla (P - W_w \rho g z). \tag{10-2}
\]

Here, \( \varphi \) denotes porosity [-], \( s \) saturation \([m^3 m^{-3}]\), \( \rho \) water density \([kmol m^{-3}]\), \( q \) Darcy flux \([m s^{-1}]\), \( k \) intrinsic permeability \([m^2]\), \( k_r \) relative permeability [-], \( \mu \) viscosity \([Pa s]\), \( P \) pressure \([Pa]\), \( W_w \) formula weight of water \([kg kmol^{-1}]\), \( g \) gravity \([m s^{-2}]\), and \( z \) the vertical component of the position vector \([m]\). Supported relative permeability functions \( k_r \) for Richards’ equation include van Genuchten, Books-Corey and Thomeer-Corey, while the saturation functions include Burdine and Mualem. Water density and viscosity are computed as a function of temperature and pressure through an equation of state for water. The source/sink term \( Q_w \) \([kmol m^{-3} s^{-1}]\) has the form

\[
Q_w = \frac{q_M}{W_w} \delta(r - r_{ss}), \tag{10-3}
\]

where \( q_M \) denotes a mass rate in \( kg/m^3 s \), and \( r_{ss} \) denotes the location of the source/sink.

10.2 Capillary Pressure Relations

Capillary pressure is related to saturation by various phenomenological relations, one of which is the van Genuchten (1980) relation

\[
s_e = \left[1 + \left( \frac{p_c}{p_c^0} \right)^n \right]^{-m}, \tag{10-4}
\]

where \( p_c \) represents the capillary pressure \([Pa]\), and the effective saturation \( s_e \) is defined by

\[
s_e = \frac{s - s_r}{s_0 - s_r}, \tag{10-5}
\]

where \( s_r \) denotes the residual saturation, and \( s_0 \) denotes the maximum saturation. The inverse relation is given by

\[
p_c = p_c^0 \left( s_e^{-1/m} - 1 \right)^{1/n}. \tag{10-6}
\]

The quantities \( m, n \) and \( p_c^0 \) are empirical constants determined by fitting to experimental data.

< 160 >
10.2.1 Brooks-Corey Saturation Function

The Brooks-Corey saturation function is a limiting form of the van Genuchten relation for $p_c/p_c^0 \gg 1$, with the form

$$s_e = \left( \frac{p_c}{p_c^0} \right)^{-\lambda}, \quad (10-7)$$

with $\lambda = mn$ and inverse relation

$$p_c = p_c^0 s_e^{-1/\lambda}. \quad (10-8)$$

10.2.2 Relative Permeability

Two forms of the relative permeability function are implemented based on the Mualem and Burdine formulations. The quantity $n$ is related to $m$ by the expression

$$m = 1 - \frac{1}{n}, \quad n = \frac{1}{1 - m}, \quad (10-9)$$

for the Mualem formulation and by

$$m = 1 - \frac{2}{n}, \quad n = \frac{2}{1 - m}, \quad (10-10)$$

for the Burdine formulation.

For the Mualem relative permeability function based on the van Genuchten saturation function is given by the expression

$$k_r = \sqrt{s_e} \left\{ 1 - \left[ 1 - (s_e)^{1/m} \right]^m \right\}^2. \quad (10-11)$$

The Mualem relative permeability function based on the Brooks-Corey saturation function is defined by

$$k_r = \left( s_e \right)^{5/2+2/\lambda},$$

$$= \left( p_c/p_c^0 \right)^{-(5\lambda/2+2)}. \quad (10-12)$$

For the Burdine relative permeability function based on the van Genuchten saturation function is given by the expression

$$k_r = s_e^2 \left\{ 1 - \left[ 1 - (s_e)^{1/m} \right]^m \right\}.$$  \quad (10-14)

The Burdine relative permeability function based on the Brooks-Corey saturation function has the form

$$k_r = \left( s_e \right)^{2+3/\lambda},$$

$$= \left( p_c/p_c^0 \right)^{-(2+3\lambda)}. \quad (10-15)$$
10.2.3 Smoothing

At the end points of the saturation and relative permeability functions it is sometimes necessary to smooth the functions in order for the Newton-Raphson equations to converge. This is accomplished using a third order polynomial interpolation by matching the values of the function to be fit (capillary pressure or relative permeability), and imposing zero slope at the fully saturated end point and matching the derivative at a chosen variably saturated point that is close to fully saturated. The resulting equations for coefficients \( a_i, i = 0 - 3 \), are given by

\[
\begin{align*}
  a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 &= f_1, \\
  a_0 + a_1 x_2 + a_2 x_2^2 + a_3 x_2^3 &= f_1, \\
  a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 &= f_1', \\
  a_1 x_2 + a_2 x_2^2 + a_3 x_2^3 &= f_2',
\end{align*}
\] (10-17a to 10-17d)

for chosen points \( x_1 \) and \( x_2 \). In matrix form these equations become

\[
\begin{bmatrix}
  1 & x_1 & x_1^2 & x_1^3 \\
  1 & x_2 & x_2^2 & x_2^3 \\
  0 & 1 & 2x_1 & 3x_1^2 \\
  0 & 1 & 2x_2 & 3x_2^2
\end{bmatrix}
\begin{bmatrix}
  a_0 \\
  a_1 \\
  a_2 \\
  a_3
\end{bmatrix}
= \begin{bmatrix}
  f_1 \\
  f_2 \\
  f_1' \\
  f_2'
\end{bmatrix}.
\]
(10-18)

The conditions imposed on the smoothing equations for capillary pressure \( f = s_e(p_c) \) are \( x_1 = 2p_0^c, \) \( x_2 = p_0^c/2, \) \( f_1 = (s_e)_1, \) \( f_2 = 1, \) \( f_1' = (s'_e)_1, \) \( f_2' = 0. \) For relative permeability \( f = k_r(s_e), \) \( x_1 = 1, \) \( x_2 = 0.99, \) \( f_1 = 1, \) \( f_2 = (k_r)_2, \) \( f_1' = 0, \) \( f_2' = (k'_r)_2. \)

10.3 Mode: MPHASE

Local equilibrium is assumed between phases for modeling multiphase systems with PFLOTRAN. The multiphase partial differential equations for mass and energy conservation solved by PFLOTRAN have the general form:

\[
\frac{\partial}{\partial t} \left( \varphi \sum_{\alpha} s_\alpha \eta_\alpha x_\alpha^i \right) + \nabla \cdot \sum_{\alpha} F_\alpha^i = Q_i,
\]
(10-19a)

for the \( i \)th component where the flux \( F_\alpha^i \) is given by

\[
F_\alpha^i = q_\alpha \eta_\alpha x_\alpha^i - \varphi s_\alpha D_\alpha \eta_\alpha \nabla x_\alpha^i,
\]
(10-19b)

and

\[
\frac{\partial}{\partial t} \left( \varphi \sum_{\alpha} s_\alpha \eta_\alpha U_\alpha + (1 - \varphi) \rho_r c_r T \right) + \nabla \cdot \sum_{\alpha} \left[ q_\alpha \eta_\alpha H_\alpha - \kappa \nabla T \right] = Q_e,
\]
(10-19c)

for energy. In these equations \( \alpha \) designates a fluid phase (\( \alpha = l, sc \)) at temperature \( T \) and pressure \( P_\alpha \) with the sums over all fluid phases present in the system, and source/sink terms \( Q_i \) and \( Q_e \) described in more detail below. Species are designated by the subscript \( i (i = \text{H}_2\text{O}, \text{CO}_2); \) \( \varphi \)
denotes the porosity of the porous medium; \( s_\alpha \) denotes the phase saturation state; \( x_i^\alpha \) denotes the mole fraction of species \( i \) satisfying
\[
\sum_i x_i^\alpha = 1, \tag{10-20}
\]
the quantities \( \eta_\alpha, H_\alpha, U_\alpha \) refer to the molar density, enthalpy, and internal energy of each fluid phase, respectively; and \( q_\alpha \) denotes the Darcy flow rate for phase \( \alpha \) defined by
\[
q_\alpha = -\frac{k k_\alpha}{\mu_\alpha} \nabla (P_\alpha - \rho_\alpha g z), \tag{10-21}
\]
where \( k \) refers to the intrinsic permeability, \( k_\alpha \) denotes the relative permeability, \( \mu_\alpha \) denotes the fluid viscosity, \( W_\alpha \) denotes the formula weight, \( g \) denotes the acceleration of gravity, and \( z \) designates the vertical of the position vector. The mass density \( \rho_\alpha \) is related to the molar density by the expression
\[
\rho_\alpha = W_\alpha \eta_\alpha, \tag{10-22}
\]
where the formula weight \( W_\alpha \) is a function of composition according to the relation
\[
W_\alpha = \frac{\rho_\alpha}{\eta_\alpha} = \sum_i W_i x_i^\alpha. \tag{10-23}
\]
The quantities \( \rho_r, c_r, \) and \( \kappa \) refer to the mass density, heat capacity, and thermal conductivity of the porous rock.

### 10.3.1 Source/Sink Terms

The source/sink terms, \( Q_i \) and \( Q_e \), describe injection and extraction of mass and heat, respectively, for various well models. Several different well models are available. The simplest is a volume or mass rate injection/production well given by
\[
Q_i = \sum_n \sum_\alpha q_V^\alpha \eta_\alpha x_i^\alpha \delta (r - r_n), \tag{10-24a}
\]
\[
= \sum_n \sum_\alpha \frac{\eta_\alpha}{\rho_\alpha} q_M^\alpha x_i^\alpha \delta (r - r_n), \tag{10-24b}
\]
\[
= \sum_n \sum_\alpha W_\alpha^{-1} q_M^\alpha x_i^\alpha \delta (r - r_n), \tag{10-24c}
\]
where \( q_V^\alpha, q_M^\alpha \) refer to volume and mass rates with units m\(^3\)/s, kg/s, respectively, related by the density
\[
q_M^\alpha = \rho_\alpha q_V^\alpha. \tag{10-25}
\]
The position vector \( r_n \) refers to the location of the \( n \)th source/sink.

A less simplistic approach is to specify the bottom well pressure to regulate the flow rate in the well. In this approach the mass flow rate is determined from the expression
\[
q_M^\alpha = \Gamma \rho_\alpha \frac{k_\alpha}{\mu_\alpha} (p_\alpha - p_{lw}^\alpha), \tag{10-26}
\]
< 163 >
Table 8: Choice of primary variables.

<table>
<thead>
<tr>
<th>State</th>
<th>X_1</th>
<th>X_2</th>
<th>X_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>liquid</td>
<td>p_l</td>
<td>T</td>
<td>x_l^{CO_2}</td>
</tr>
<tr>
<td>gas</td>
<td>p_g</td>
<td>T</td>
<td>x_g^{CO_2}</td>
</tr>
<tr>
<td>two-phase</td>
<td>p_g</td>
<td>T</td>
<td>s_g</td>
</tr>
</tbody>
</table>

with bottom well pressure $p_{bw}$, and where $\Gamma$ denotes the well factor (production index) given by

$$\Gamma = \frac{2\pi k \Delta z}{\ln \left( \frac{r_e}{r_w} \right) + \sigma - 1/2}.$$  \hspace{1cm} (10-27)

In this expression $k$ denotes the permeability of the porous medium, $\Delta z$ refers to the layer thickness, $r_e$ denotes the grid block radius, $r_w$ denotes the well radius, and $\sigma$ refers to the skin thickness factor. For a rectangular grid block of area $A = \Delta x \Delta y$, $r_e$ can be obtained from the relation

$$r_e = \sqrt{\frac{A}{\pi}}.$$ \hspace{1cm} (10-28)

See Peaceman (1977) and Coats and Ramesh (1982) for more details.

### 10.3.2 Variable Switching

In PFLOTRAN a variable switching approach is used to account for phase changes enforcing local equilibrium. According to the Gibbs phase rule there are a total of $N_C+1$ degrees of freedom where $N_C$ denotes the number of independent components. This can be seen by noting that the intensive degrees of freedom are equal to $N_{int} = N_C - N_P + 2$, where $N_P$ denotes the number of phases. The extensive degrees of freedom equals $N_{ext} = N_P - 1$. This gives a total number of degrees of freedom $N_{dof} = N_{int} + N_{ext} = N_C + 1$, independent of the number of phases $N_P$ in the system. Primary variables for liquid, gas and two-phase systems are listed in Table 8. The conditions for phase changes to occur are considered in detail below.

#### 10.3.2.1 Gas: $(p_g, T, x_g^{CO_2}) \rightarrow$ Two-Phase: $(p_g, T, s_g)$

- gas $\rightarrow$ 2-ph: $x_g^{CO_2} \leq 1 - \frac{P_{sat}(T)}{p_g}$, or equivalently: $x_{H_2O}^{g} \geq \frac{P_{sat}(T)}{p_g}$

#### 10.3.2.2 Liquid: $(p_l, T, x_l^{CO_2}) \rightarrow$ Two-phase: $(p_g, T, s_g)$

- liq $\rightarrow$ 2-ph: $x_l^{CO_2} \geq x_{eq}^{CO_2}$

The equilibrium mole fraction $x_{eq}^{CO_2}$ is given by

$$x_{eq}^{CO_2} = \frac{m_{CO_2}}{W_{H_2O} + m_{CO_2} + \sum_{l \neq H_2O,C02} m_l},$$ \hspace{1cm} (10-29)
where the molality at equilibrium is given by

$$m_{\text{CO}_2}^{eq} = \left(1 - \frac{P_{\text{sat}}(T)}{p}\right)\frac{\phi_c p}{K_{\text{CO}_2} \gamma_{\text{CO}_2}},$$  \hspace{1cm} (10-30)

where it is assumed that

$$y_{\text{CO}_2} = 1 - \frac{P_{\text{sat}}(T)}{p}. \hspace{1cm} (10-31)$$

### 10.3.2.3 Two-Phase: \((p_g, T, s_g) \rightarrow \text{Liquid} (p_l, T, x_{\text{CO}_2}^l) \text{ or Gas} (p_g, T, x_{\text{CO}_2}^g)\)

Equilibrium in a two-phase \(\text{H}_2\text{O}–\text{CO}_2\) system is defined as the equality of chemical potentials between the two phases as expressed by the relation

$$f_{\text{CO}_2} = y_{\text{CO}_2} \phi_{\text{CO}_2} p_g = K_{\text{CO}_2} \left(\gamma_{\text{CO}_2} m_{\text{CO}_2}\right),$$  \hspace{1cm} (10-32)

where

$$y_{\text{CO}_2} = x_{\text{CO}_2}^g, \quad x_{\text{CO}_2} = x_{\text{CO}_2}^l,$$

$$x_{\text{H}_2\text{O}}^g = \frac{P_{\text{sat}}(T)}{p_g}, \hspace{1cm} (10-34)$$

$$y_{\text{CO}_2} = 1 - x_{\text{H}_2\text{O}}^g = 1 - \frac{P_{\text{sat}}(T)}{p_g}, \hspace{1cm} (10-35)$$

From these equations a Henry coefficient-like relation can be written as

$$y_{\text{CO}_2} = \tilde{K}_{\text{CO}_2} x_{\text{CO}_2},$$  \hspace{1cm} (10-36)

where

$$\tilde{K}_{\text{CO}_2} = \frac{\gamma_{\text{CO}_2} K_{\text{CO}_2} m_{\text{CO}_2}}{\phi_{\text{CO}_2} p_g x_{\text{CO}_2}}. \hspace{1cm} (10-37)$$

- A phase change to single liquid or gas phase occurs if \(s_g \leq 0\) or \(s_g \geq 1\), respectively.

Conversion relations between mole fraction \((x_i)\), mass fraction \((w_i)\) and molality \((m_i)\) are as follows:

**Molality–mole fraction:**

$$m_i = \frac{n_i}{M_{\text{H}_2\text{O}}} = \frac{n_i}{W_{\text{H}_2\text{O}} n_{\text{H}_2\text{O}}} = \frac{x_i}{W_{\text{H}_2\text{O}} x_{\text{H}_2\text{O}}} = \frac{x_i}{W_{\text{H}_2\text{O}} \left(1 - \sum_{l \neq \text{H}_2\text{O}} x_l\right)} \hspace{1cm} (10-38)$$

**Mole fraction–molality:**

$$x_i = \frac{n_i}{N} = \frac{n_i}{M_{\text{H}_2\text{O}}} \frac{M_{\text{H}_2\text{O}}}{N} = \frac{m_i}{\sum m_i} = \frac{W_{\text{H}_2\text{O}} m_i}{1 + W_{\text{H}_2\text{O}} \sum_{l \neq \text{H}_2\text{O}} m_l} \hspace{1cm} (10-39)$$

**Mole fraction–mass fraction:**

$$x_i = \frac{n_i}{N} = \frac{W_i^{-1} W_i n_i}{\sum W_i^{-1} W_i n_i} = \frac{W_i^{-1} w_i}{\sum W_i^{-1} w_i} \hspace{1cm} (10-40)$$

**Mass fraction–mole fraction:**

$$w_i = \frac{M_i}{M} = \frac{W_i n_i}{\sum W_i n_i} = \frac{W_i x_i}{\sum W_i x_i} \hspace{1cm} (10-41)$$
10.4 Mode: **IMMIS**

The **IMMIS** mode applies to multiple completely immiscible phases. The code PIMS, parallel immiscible multiphase flow simulator, is a simplified version of the MPHASE mode in which the dependency on thermodynamic relations have been removed, since for immiscible systems the solubility is identically zero for each component. In this case the number of components is equal to the number of phases, or degrees of freedom associated with each node for an isothermal system. The immiscible property removes the variable switching strategy used in MPHASE, which may be the most numerically difficult part of PFLOTRAN, and may cause problems for multi-level solvers.

The governing equations solved by PIMS are given by

\[
\frac{\partial}{\partial t} \left( \varphi \rho_\alpha s_\alpha \right) + \nabla \cdot \left( \rho_\alpha q_\alpha \right) = Q_\alpha, \tag{10-42}
\]

where the subscript \( \alpha \) denotes an immiscible phase. In this equation \( \varphi \) is porosity, \( s_\alpha, \rho_\alpha \) refer to the \( \alpha \)th phase saturation and density, respectively, \( q_\alpha \) is the Darcy velocity of the \( \alpha \)th phase given by

\[
q_\alpha = -\frac{k k_\alpha}{\mu_\alpha} \left( \nabla p - \rho_\alpha g \hat{z} \right), \tag{10-43}
\]

with permeability \( k \), relative permeability \( k_\alpha \), fluid viscosity \( \mu_\alpha \), and \( Q_\alpha \) is the source/sink term. The selection of primary variables are pressure \( p \) and \( n-1 \) independent phase saturation variables \( s_\alpha, \alpha = 1, \ldots, n-1 \) with

\[
\sum_{\alpha=1}^{n} s_\alpha = 1. \tag{10-44}
\]

The mass conservation equations are coupled to the energy balance equation given by

\[
\frac{\partial}{\partial t} \left( \varphi \sum_\alpha s_\alpha \rho_\alpha U_\alpha + (1 - \varphi) \rho_r C_r T \right) + \nabla \cdot \left( \sum_\alpha \rho_\alpha q_\alpha H_\alpha - \kappa \nabla T \right) = Q_e, \tag{10-45}
\]

where \( U_\alpha, H_\alpha \) denote the internal energy and enthalpy of the \( \alpha \)th fluid phase, \( \kappa \) denotes the thermal conductivity of the bulk porous medium, \( \rho_r, C_r \) denote the rock density and heat capacity, and \( T \) refers to the temperature. Thus the number of equations is equal to number of phases plus one, which is equal to the number of unknowns: \( (p, T, s_1, \ldots, s_{n-1}) \).

10.5 Mode: **MISCIBLE**

The miscible mode applies to a mixture of water and proplylene glycol (PPG). In terms of molar density for the mixture \( \eta \) and mole fractions \( x_i, i=1 \) (water), \( i=2 \) (PPG), the mass conservation equations have the form

\[
\frac{\partial}{\partial t} \varphi \eta x_i + \nabla \cdot [\eta q x_i - \varphi D_\eta \nabla x_i] = Q_i, \tag{10-46}
\]

with source/sink term \( Q_i \). It should be noted that the mass- and mole-fraction formulations of the conservation equations are not exactly equivalent. This is due to the diffusion term which gives an extra term when transformed from the mole-fraction to mass-fraction gradient.
The molar density $\eta$ is related to the mass density by

$$\eta = W^{-1} \rho,$$  \hspace{1cm} (10-47)

and

$$W_i \eta x_i = \rho y_i.$$  \hspace{1cm} (10-48)

It follows that

$$W_i \eta \nabla x_i = \rho \nabla y_i + \rho y_i \nabla \ln W.$$  \hspace{1cm} (10-49)

The second term on the right-hand side is ignored.

Simple equations of state are provided for density $[g/cm^3]$, viscosity $[Pa \ s]$, and diffusivity $[m^2/s]$. The density is a function of both composition and pressure with the form

$$\rho(y_1, p) = \rho(y_1, p_0) + \frac{\partial \rho}{\partial p} \bigg|_{p=p_0} (p-p_0),$$  \hspace{1cm} (10-50)

$$= \rho(y_1, p_0)(1 + \beta(p-p_0)),$$  \hspace{1cm} (10-51)

with the compressibility $\beta(y_1)$ given by

$$\beta = \frac{1}{\rho} \frac{\partial \rho}{\partial p} \bigg|_{p=p_0},$$  \hspace{1cm} (10-52)

$$= 4.49758 \times 10^{-10} y_1 + 5 \times 10^{-10}(1-y_1),$$  \hspace{1cm} (10-53)

and the mixture density at the reference pressure $p_0$ taken as atmospheric pressure is given by

$$\rho(y_1, p_0) = \left( (0.0806y_1 - 0.203)y_1 + 0.0873 \right) y_1 + 1.0341 \right) 10^3,$$  \hspace{1cm} (10-54)

with mass fraction of water $y_1$. The viscosity and diffusivity have the forms

$$\mu(y_1) = 10^{1.6743(1-y_1) - 0.0758} 10^{-3},$$  \hspace{1cm} (10-55)

and

$$D(y_1) = \left( (-4.021y_1 + 9.1181)y_1 - 5.9703y_1 + 0.4043 \right) y_1 + 0.5687 \right) 10^{-9},$$  \hspace{1cm} (10-56)

The mass fraction is related to mole fraction according to

$$y_1 = \frac{x_1 W_{H_2O}}{W},$$  \hspace{1cm} (10-57)

where the mean formula weight $W$ is given by

$$W = x_1 W_{H_2O} + x_2 W_{PPG},$$  \hspace{1cm} (10-58)

with formula weights for water and propylene glycol equal to $W_{H_2O} = 18.01534$ and $W_{PPG} = 76.09 [kg/kmol]$.

Global mass conservation satisfies the relation

$$\frac{d}{dt} M_i = - \int F_i \cdot dS + \int Q_i dV,$$  \hspace{1cm} (10-59)

with

$$M_i = \int \varphi \eta x_i dV.$$  \hspace{1cm} (10-60)

In terms of mass fractions and mass density

$$M_i^m = W_i M_i = \int \varphi y_i dV.$$  \hspace{1cm} (10-61)
### 10.6 Mode: Air-Water

The **Air-Water** mode involves two phase liquid water-gas flow coupled to the reactive transport mode. Mass conservation equations have the form

\[
\frac{\partial}{\partial t} \varphi \left( s_l \rho_l x_i^l + s_g \rho_g x_i^g \right) + \nabla \cdot \left( q_l \rho_l x_i^l + q_g \rho_g x_i^g - \varphi s_l D_l \rho_l \nabla x_i^l - \varphi s_g D_g \rho_g \nabla x_i^g \right) = Q_i, \tag{10-62}
\]

for liquid and gas saturation \( s_{l,g} \), density \( \rho_{l,g} \), diffusivity \( D_{l,g} \), Darcy velocity \( q_{l,g} \) and mole fraction \( x_{l,g}^i \). The energy conservation equation can be written in the form

\[
\sum_{\alpha=l,g} \left\{ \frac{\partial}{\partial t} (\varphi s_{\alpha} \rho_{\alpha} U_{\alpha}) + \nabla \cdot (q_{\alpha} \rho_{\alpha} H_{\alpha}) \right\} + \frac{\partial}{\partial t} \left( (1 - \varphi) \rho_c C_p T \right) - \nabla \cdot (\kappa \nabla T) = Q, \tag{10-63}
\]

as the sum of contributions from liquid and gas fluid phases and rock, with internal energy \( U_{\alpha} \) and enthalpy \( H_{\alpha} \) of fluid phase \( \alpha \), rock heat capacity \( C_p \) and thermal conductivity \( \kappa \). Note that

\[
U_{\alpha} = H_{\alpha} - \frac{P_{\alpha}}{\rho_{\alpha}}. \tag{10-64}
\]

Thermal conductivity \( \kappa \) is determined from the equation (Somerton et al., 1974)

\[
\kappa = \kappa_{\text{dry}} + \sqrt{s_l (\kappa_{\text{sat}} - \kappa_{\text{dry}})}, \tag{10-65}
\]

where \( \kappa_{\text{dry}} \) and \( \kappa_{\text{sat}} \) are dry and fully saturated rock thermal conductivities.

### 10.7 Mode: TH (Thermal-Hydrologic)

The current implementation of the **TH** mode applies to mass and energy conservation equations which are solved fully coupled. The fluid density only a function of \( T \) and \( P \). Future generalizations of the **TH** mode will include multicomponent variable density fluids. The **TH** equations may be coupled to the reactive transport mode (see Section 10.9).

**TH** mode applies to single phase, variably saturated, nonisothermal systems with incorporation of density variations coupled to fluid flow. The governing equations for mass and energy are given by

\[
\frac{\partial}{\partial t} (\varphi sp) + \nabla \cdot (\rho q) = Q_w, \tag{10-66}
\]

and

\[
\frac{\partial}{\partial t} (\varphi sp U + (1 - \varphi) \rho_p c_p T) + \nabla \cdot (\rho q H - \kappa \nabla T) = Q_e. \tag{10-67}
\]

The Darcy flow velocity \( q \) is given by

\[
q = -\frac{kk_r}{\mu} \nabla (P - W \rho g z). \tag{10-68}
\]

Here, \( \varphi \) denotes porosity, \( s \) saturation, \( \rho \) mixture density of the brine, \( q \) Darcy flux, \( k \) intrinsic permeability, \( k_r \) relative permeability, \( \mu \) viscosity, \( P \) pressure, \( g \) gravity, and \( z \) the vertical component of the position vector. Supported relative permeability functions \( k_r \) for Richards’ equation
include van Genuchten, Books-Corey and Thomeer-Corey, while the saturation functions include Burdine and Mualem. Water density and viscosity are computed as a function of temperature and pressure through an equation of state for water. The quantities $\rho$, $c_p$, and $\kappa$ denote the density, heat capacity, and thermal conductivity of the porous medium-fluid system. The internal energy and enthalpy of the fluid, $U$ and $H$, are obtained from an equation of state for pure water. These two quantities are related by the thermodynamic expression

$$U = H - \frac{P}{\rho}. \quad (10-69)$$

Thermal conductivity is determined from the equation (Somerton et al., 1974)

$$\kappa = \kappa_{\text{dry}} + \sqrt{s_l} (\kappa_{\text{sat}} - \kappa_{\text{dry}}), \quad (10-70)$$

where $\kappa_{\text{dry}}$ and $\kappa_{\text{sat}}$ are dry and fully saturated rock thermal conductivities.

### 10.7.1 Ice Model

In PFLOTRAN, the formulation used to model ice and water vapor involves solving a modified Richards equation coupled with an energy balance equation. This formulation is different from Painter (2011), where a multiphase approach was used and mass balance for air was also solved for. In this formulation, the movement of air is not tracked, and hence the mass balance for air is not considered. The balance equations for mass and energy involving three phases (liquid, gas, ice) for the water component are given by

$$\frac{\partial}{\partial t} \left[ \phi \left( s_l \eta_l X_l^l + s_g \eta_g X_g^g + s_i \eta_i X_i^i \right) \right] + \nabla \cdot \left[ X_w^l \mathbf{v}_l \eta_l + X_w^g \eta_g \mathbf{v}_g \right] - \nabla \cdot \left[ \phi s_g \tau_g \eta_g D_g \nabla X_g^g \right] = Q_w, \quad (10-71a)$$

$$\frac{\partial}{\partial t} \left[ \phi \left( s_l \eta_l U_l + s_g \eta_g U_g + s_i \eta_i U_i \right) + (1 - \phi) \rho_r c_r T \right] + \nabla \cdot \left[ \mathbf{v}_l \rho_l H_l + \mathbf{v}_g \rho_g H_g \right] - \nabla \cdot \left[ \kappa \nabla T \right] = Q_e, \quad (10-71b)$$

where the subscripts $l, i, g$ denote the liquid, ice and gas phases, respectively; $\phi$ is the porosity; $s_\alpha (\alpha = i, l, g)$ is the saturation of the $\alpha$-th phase; $\eta_\alpha (\alpha = i, l, g)$ is the molar density of the $\alpha$-th phase; $\rho_g, \rho_l$ are the mass densities of the gas and liquid phases; $Q_w$ is the mass source of $\text{H}_2\text{O}$; $X_w^\alpha (\alpha = i, l, g)$ is the mole fraction of $\text{H}_2\text{O}$ in the $\alpha$-th phase; $\tau_g$ is the tortuosity of the gas phase; $D_g$ is the diffusion coefficient in the gas phase; $T$ is the temperature (assuming all the phases and the rock are in thermal equilibrium); $c_r$ is the specific heat of the rock; $\rho_r$ is the density of the rock; $U_\alpha (\alpha = i, l, g)$ is the molar internal energy of the $\alpha$-th phase; $H_\alpha (\alpha = l, g)$ is the molar enthalpy of the $\alpha$-th phase; $Q_e$ is the heat source; $\nabla (\ )$ is the gradient operator; $\nabla \cdot (\ )$ is the divergence operator.

The Darcy velocity for the gas and liquid phases are given as follows:

$$\mathbf{v}_g = - \frac{k_{rg}}{\mu_g} \nabla \left[ p_g - \rho_g g z \right], \quad (10-72a)$$

$$\mathbf{v}_l = - \frac{k_{rl}}{\mu_l} \nabla \left[ p_l - \rho_l g z \right], \quad (10-72b)$$
where $k$ is the absolute permeability; $k_{ra}(\alpha = l, g)$ is the relative permeability of the $\alpha$-th phase; 
$\mu_{\alpha}(\alpha = l, g)$ is the viscosity of the $\alpha$-th phase; $p_{\alpha}(\alpha = l, g)$ is the partial pressure of the $\alpha$-th phase; $g$ is acceleration due to gravity, and $z$ is the vertical distance from a reference datum.

The constraint on the saturations of the various phases of water is given by

$$s_l + s_g + s_i = 1.$$  \hfill (10-73)

Furthermore, neglecting the amount of air in liquid and ice phases, it follows that

$$X^l_a = 0, X^i_a = 0 \Rightarrow X^l_w = 1, X^i_w = 1,$$  \hfill (10-74)

and so Eqns. (10-71), (10-72), based on the assumption that $p_g$ is hydrostatic (i.e., $p_g = (p_g)_0 + \rho_g g z$), reduce to

$$\frac{\partial}{\partial t}[\phi (s_g \eta_g X^g_w + s_l \eta_l + s_i \eta_i)] + \nabla \cdot [\mathbf{v}_l \eta_l] - \nabla \cdot [\phi s_g \tau_g \eta_g D_g \nabla X^g_w] = Q_w,$$  \hfill (10-75a)

$$\frac{\partial}{\partial t} [\phi (s_l \eta_l U_l + s_g \eta_g U_g + s_i \eta_i U_i) + (1 - \phi) \rho_c c_r T] + \nabla \cdot [\mathbf{v}_l \eta_l H_l] - \nabla \cdot [\kappa \nabla T] = Q_e,$$  \hfill (10-75b)

$$\mathbf{v}_l = -\frac{k_{rl}}{\mu_l} \nabla [p_l - \rho_l g z].$$  \hfill (10-75c)

In the above formulation, temperature and liquid pressure are chosen to be primary variables. It is ensured that complete dry-out does not occur, and that liquid is present at all times. With this approach, it is not necessary to change the primary variables based on the phases present.

In addition to the previously described mass and energy balance equations, additional constitutive relations are required to model non-isothermal, multiphase flow of water. Assuming thermal equilibrium among the ice, liquid and vapor phases, the mole fraction of water in vapor phase is given by the relation,

$$X^g_w = \frac{p_v}{p_g},$$  \hfill (10-76)

where $p_v$ is the vapor pressure, and $p_g$ is the gas pressure (It is assumed that $p_g = 1$ atm throughout the domain). Vapor pressure is calculated using Kelvin’s relation which includes vapor pressure lowering due to capillary effects as follows

$$p_v = P_{sat}(T) \exp \left[ \frac{P_{cgl}}{\eta_l R(T + 273.15)} \right],$$  \hfill (10-77)

where $P_{sat}$ is the saturated vapor pressure, $P_{cgl}$ is the liquid-gas capillary pressure, and $R$ is the gas constant. Empirical relations for saturated vapor pressure are used for both above and below freezing conditions.

To calculate the partition of ice, liquid and vapor phases, at a known temperature and liquid pressure, the following two relations are used (see Painter, 2011):

$$\frac{s_l}{s_l + s_g} = S_* \left( P_{cgl} \right),$$  \hfill (10-78a)

$$\frac{s_l}{s_l + s_i} = S_* \left( \frac{\sigma_{gl}}{\sigma_{il}} \right) P_{csl},$$  \hfill (10-78b)
$S_*$ is the retention curve for unfrozen liquid-gas phases, $P_{cgl}$ is the gas-liquid capillary pressure, $P_{cil}$ is the ice-liquid capillary pressure, $\sigma_{il}$ and $\sigma_{gl}$ are the ice-liquid and gas-liquid interfacial tensions. Also, $P_{cil} = -\rho_i h_{iw} \vartheta$, where $h_{iw}^0$ is the heat of fusion of ice at 273.15 K, $\rho_i$ is the mass density of ice, $\vartheta = \frac{T - T_0}{T_0}$ with $T_0 = 273.15$ K.

For $S_*$ the van Genuchten model is used:

$$S_* = \begin{cases} 
[1 + (\alpha P_c)^\gamma]^{-\lambda}, & P_c > 0 \\
1, & P_c \leq 0
\end{cases} \quad (10-79)$$

with the Mualem model implemented for the relative permeability of liquid water,

$$k_{rl} = (s_l)^{\frac{1}{2}} \left[ 1 - \left( 1 - (s_l)^{\frac{1}{2}} \right)^\lambda \right]^2, \quad (10-80)$$

where $\lambda, \alpha$ are parameters, with $\gamma = \frac{1}{1-\lambda}$.

The thermal conductivity for the frozen soil is chosen to be

$$\kappa = Ke_f \kappa_{wet,f} + Ke_u \kappa_{wet,u} + (1 - Ke_u - Ke_f) \kappa_{dry}, \quad (10-81)$$

where $\kappa_{wet,f}$, $\kappa_{wet,u}$ are the liquid- and ice-saturated thermal conductivities, $\kappa_{dry}$ is the dry thermal conductivity, $Ke_f$, $Ke_u$ are the Kersten numbers in frozen and unfrozen conditions and are assumed to be related to the ice and liquid saturations by power law relations as follows

$$Ke_f = (s_i)^{\alpha_f}, \quad (10-82a)$$
$$Ke_u = (s_l)^{\alpha_u}, \quad (10-82b)$$

with $\alpha_f, \alpha_u$ being the power law coefficients. Care is also taken to ensure that the derivatives of the Kersten numbers do not blow up when $s_i, s_l$ go to zero when $\alpha_f, \alpha_u$ are less than one.

The gas diffusion coefficient $D_g$ is assumed to depend on temperature and pressure as follows:

$$D_g = D_g^0 \left( \frac{P_{ref}}{P} \right) \left( \frac{T}{T_{ref}} \right)^{1.8}, \quad (10-83)$$

where $D_g^0$ is the reference diffusion coefficient at some reference temperature, $T_{ref}$, and pressure $P_{ref}$.

### 10.8 Thermal Conduction Multiple Continuum Model

A thermal conduction model employing a multiple continuum model has been added to modes MPHASE and TH. The formulation is based on Pruess and Narasimhan (1985) using a integrated finite volume approach to develop equations for fracture and matrix temperatures $T_\alpha$ and $T_\beta$, respectively. The DCDM (dual continuum discrete matrix) model following the classification given in Lichtner (2000) is implemented. In what follows the matrix porosity is assumed to be zero. Using the control volume configuration shown in Figure 5, with fracture nodes designated by the
The subscript \( n \) and matrix nodes by \( m \), the integrated finite volume form of the heat transport equation for the \( n \)th fracture control volume is given by

\[
\varphi \left( \sum_{\alpha} (s_{\alpha}\rho_{\alpha} U_{\alpha})_{n}^{k+1} - \sum_{\alpha} (s_{\alpha}\rho_{\alpha} U_{\alpha})_{n}^{k} \right) + (1 - \varphi) \left( (\rho_{r}C_{r}T_{\alpha})_{n}^{k+1} - (\rho_{r}C_{r}T_{\alpha})_{n}^{k} \right) \frac{v_{n}}{\Delta t} \\
+ \sum_{n'} \left[ \sum_{\alpha} (q_{\alpha}\rho_{\alpha} H_{\alpha})_{nn'} + \frac{k_{\alpha}}{d_{n} + d_{n'}} (T_{\alpha n} - T_{\alpha n'}) \right] A_{\alpha nn'}^{n} \\
+ \sum_{\alpha} \frac{k_{M}}{d_{n} + d_{M}} (T_{\alpha n} - T_{\beta M}) A_{\alpha nM}^{M} = 0,
\]

(10-84a)

where \( v_{n} \) denotes the fracture volume, and

\[
\left( (\rho_{r}C_{r}T_{\beta})_{m}^{k+1} - (\rho_{r}C_{r}T_{\beta})_{m}^{k} \right) \frac{v_{m}}{\Delta t} + \sum_{m'} \frac{k_{M}}{d_{m} + d_{m'}} (T_{\beta m} - T_{\beta m'}) A_{\beta mm'}^{M} + \delta_{mM} \frac{k_{M}}{d_{n} + d_{M}} (T_{\alpha n} - T_{\beta M}) A_{\alpha nM}^{M} = 0,
\]

(10-84b)

for the \( m \)th matrix node with volume \( v_{m} \). The matrix node designated by \( M \) refers to the outer most node in contact with the fracture. More than one type of matrix geometry is included in the above equations as indicated by the sum over \( \beta \) in Eqn.(10-84a).

For better convergence uniform logarithmic spacing is used for the matrix nodes

\[
\Delta \xi_{m} = \rho \Delta \xi_{m-1}.
\]

(10-85a)

specifying \( \Delta \xi_{M} \) and \( l_{M} \) for the outer most matrix node and matrix block size, respectively. The factor \( \rho \) is determined from the constraint

\[
l_{M} = 2 \sum_{m=1}^{M} \Delta \xi_{m},
\]

(10-85b)

which requires that \( \rho \) satisfy the equation

\[
\frac{l_{M}}{2 \Delta \xi_{1}} = \frac{\rho_{M}^{M} - 1}{\rho - 1},
\]

(10-85c)

with the inner and outer grid spacing related by

\[
\Delta \xi_{M} = \rho^{M-1} \Delta \xi_{1}.
\]

(10-85d)

Thermal conductivity at the interface between two control volumes is calculated using the harmonic average

\[
k_{ll'} = \frac{k_{l}k_{l'}(d_{l} + d_{l'})}{d_{l}k_{l'} + d_{l'}k_{l}}.
\]

(10-86)

The fracture volume \( v_{n} \) is related to the REV volume \( V_{n} \) by

\[
\epsilon = \frac{v_{n}}{V_{n}}.
\]

(10-87)
According to the geometry in Figure 5 assuming a 3D orthogonal set of fractures,

\[ V_n = (d + 2\delta)^3, \]  
(10-88)

and

\[ v_n = (d + 2\delta)^3 - d^3, \]  
(10-89)

giving

\[ \epsilon = 1 - \frac{d^3}{(d + 2\delta)^3} = 1 - \left( \frac{1}{1 + \frac{2\delta}{d}} \right)^3, \]  
(10-90a)

\[ \approx \frac{6\delta}{d}. \]  
(10-90b)

The fracture aperture \(2\delta\) is found to be in terms of \(\epsilon\) and \(d\)

\[ 2\delta = d \left( \frac{1}{(1 - \epsilon)^{1/3}} - 1 \right). \]  
(10-91)

A list of different sub-continua geometries and parameters implemented in PFLOTRAN is given in Table 9. Different independent and dependent parameters for the nested cube geometry are listed in Table 10. The interfacial area \(A_{nM}^{\alpha}\) between fracture control volumes is equal to \(\Delta y \Delta z, \Delta z \Delta x, \Delta x \Delta y\) for \(x, y,\) and \(z\) directions, respectively.

In the case of nested cubes there are four possible parameters \((\epsilon, 2\delta, l_m, l_f)\), where \(l_m\) denotes the matrix block size and \(l_f\) refers to the fracture spacing, two of which are independent.

The fracture-matrix interfacial area \(A_{nM}^{\beta}\) per unit volume is equal to

\[ A_{nM}^{\beta} = \frac{N_{\beta}}{V} A_{\beta}^0, \]  
(10-92)
Table 9: DCDM geometric parameters.

<table>
<thead>
<tr>
<th>Geometry</th>
<th>Area $A_\beta$</th>
<th>Volume $V_\beta^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slab</td>
<td>$A_\beta$</td>
<td>$V_\beta^0$</td>
</tr>
<tr>
<td>Nested Cubes</td>
<td>$6d^2$</td>
<td>$d^3$</td>
</tr>
<tr>
<td>Nested Spheres</td>
<td>$4\pi R^2$</td>
<td>$\frac{4}{3}\pi R^3$</td>
</tr>
</tbody>
</table>

Table 10: Independent and dependent nested cube parameters.

<table>
<thead>
<tr>
<th>Independent</th>
<th>Dependent</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon$</td>
<td>$l_f$</td>
</tr>
<tr>
<td>$2\delta = l_f - l_m$</td>
<td>$l_m = l_f(1 - \epsilon)^{1/3}$</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>$l_m$</td>
</tr>
<tr>
<td>$2\delta = l_f - l_m$</td>
<td>$l_f = l_m(1 - \epsilon)^{-1/3}$</td>
</tr>
<tr>
<td>$2\delta$</td>
<td>$l_f$</td>
</tr>
<tr>
<td>$\epsilon = 1 - (l_m/l_f)^3$</td>
<td>$l_m = l_f - 2\delta$</td>
</tr>
<tr>
<td>$2\delta$</td>
<td>$l_m$</td>
</tr>
<tr>
<td>$\epsilon = 1 - (l_m/l_f)^3$</td>
<td>$l_f = l_m + 2\delta$</td>
</tr>
<tr>
<td>$2\delta$</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$l_m = 2\delta\left(\frac{1}{(1 - \epsilon)^{1/3}} - 1\right)^{-1}$</td>
<td>$l_m = l - 2\delta$</td>
</tr>
</tbody>
</table>

where the number density $N_\beta/V$ of secondary continua of type $\beta$ is equal to

$$\frac{N_\beta}{V} = \sqrt[3]{\frac{V_\beta}{V_\beta^0}} = \frac{\epsilon_\beta}{V_\beta^0}.\quad (10-93)$$

and $A_\beta^0$ and $V_\beta^0$ refer to the area and volume of each geometric type as listed in Table 9. The primary-secondary coupling term can then be written in the form

$$\sum_{\beta} \kappa_{nM}^{\alpha\beta} (T_n^\alpha - T_M^\beta) A_n^\alpha = V_n \sum_{\beta} \frac{\epsilon_\beta \kappa_{nM}^{\alpha\beta} (T_n^\alpha - T_M^\beta) A_n^\alpha}{V_\beta^0}.\quad (10-94)$$

In terms of partial differential equations the heat conservation equations may be written as

$$\frac{\partial}{\partial t} \left[ \varphi \sum_{\alpha} s_{\alpha} \rho_{\alpha} U_{\alpha} + (1 - \varphi) \rho_f C_f T_f \right] + \nabla \cdot \left( \sum_{\alpha} q_{\alpha} \rho_{\alpha} H_{\alpha} - \kappa_f \nabla T_f \right) = -A_f m \kappa_f \frac{\partial T_f}{\partial n},\quad (10-95a)$$

and

$$\frac{\partial}{\partial t} \rho_f C_f T_m + \frac{\partial}{\partial \xi} \left( - \kappa_m \frac{\partial T_m}{\partial \xi} \right) = 0,\quad (10-95b)$$

for fracture and matrix temperatures $T_f$ and $T_m$, respectively, where $\xi$ represents the matrix coordinate assumed to be an effective 1D domain. The boundary condition

$$T_m(\xi_d, t | r) = T_f(r, t),\quad (10-95c)$$

between fracture and matrix continua is imposed, where $\xi_d$ denotes the outer boundary of the matrix.
10.9 Mode: Reactive Transport (Keyword CHEMISTRY)

The governing mass conservation equations for the geochemical transport mode for a multiphase system written in terms of a set of independent aqueous primary or basis species with the form

\[ \frac{\partial}{\partial t} \left( \varphi \sum_{\alpha} s_{\alpha} \Psi_{j}^{\alpha} \right) + \nabla \cdot \sum_{\alpha} \Omega_{j}^{\alpha} = Q_{j} - \sum_{m} v_{jm} I_{m} - \frac{\partial S_{j}}{\partial t}, \]  

(10-96)

and

\[ \frac{\partial \varphi_{m}}{\partial t} = \nabla m I_{m}, \]  

(10-97)

for minerals with molar volume \( V_{m} \), mineral reaction rate \( I_{m} \) and mineral volume fraction \( \varphi_{m} \) referenced to an REV. Sums over \( \alpha \) in Eqn.(10-96) are over all fluid phases in the system. The quantity \( \Psi_{j}^{\alpha} \) denotes the total concentration of the \( j \)th primary species \( A_{pri}^{j} \) in the \( \alpha \)th fluid phase defined by

\[ \Psi_{j}^{\alpha} = \delta_{\alpha l} l_{j} C_{l}^{j} + \sum_{i=1}^{N_{sec}} \nu_{ji}^{\alpha} C_{i}^{\alpha}. \]  

(10-98)

In this equation the subscript \( l \) represents the aqueous electrolyte phase from which the primary species are chosen. The secondary species concentrations \( C_{i}^{\alpha} \) are obtained from mass action equations corresponding to equilibrium conditions of the reactions

\[ \sum_{j} \nu_{ji}^{\alpha} A_{l}^{j} \rightleftharpoons A_{\alpha i}^{j}, \]  

(10-99)

yielding

\[ C_{i}^{\alpha} = \frac{K_{i}^{\alpha}}{\gamma_{i}^{\alpha}} \prod_{j} \left( \frac{\gamma_{l}^{\alpha} C_{l}^{j}}{\nu_{ji}^{\alpha}} \right), \]  

(10-100)

with equilibrium constant \( K_{i}^{\alpha} \), and activity coefficients \( \gamma_{i}^{\alpha} \). The total flux \( \Omega_{j}^{\alpha} \) for species-independent diffusion is given by

\[ \Omega_{j}^{\alpha} = (q_{\alpha} - \varphi_{s} s_{\alpha} D_{\alpha} \nabla) \Psi_{j}^{\alpha}. \]  

(10-101)

The diffusion/dispersion coefficient \( D_{\alpha} \) may be different for different phases, e.g. an aqueous electrolyte solution or gas phase, but is assumed to be species independent. Dispersivity currently must be described through a diagonal dispersion tensor. The Darcy velocity \( q_{\alpha} \) for phase \( \alpha \) is given by

\[ q_{\alpha} = -\frac{k k_{\alpha}}{\mu_{\alpha}} \nabla \left( \rho_{\alpha} - \rho_{\alpha} g z \right), \]  

(10-102)

with bulk permeability of the porous medium \( k \) and relative permeability \( k_{\alpha} \), fluid viscosity \( \mu_{\alpha} \), pressure \( \rho_{\alpha} \), density \( \rho_{\alpha} \), and acceleration of gravity \( g \). The diffusivity/dispersivity tensor \( D_{\alpha} \) is the sum of contributions from molecular diffusion and dispersion which for an isotropic medium has the form

\[ D_{\alpha} = \tau D_{m} I + a_{T} v I + \left( a_{L} - a_{T} \right) \frac{v_{v}}{v}, \]  

(10-103)

with longitudinal and transverse dispersivity coefficients \( a_{L}, a_{T} \), respectively, \( \tau \) refers to tortuosity, and \( D_{m} \) to the molecular diffusion coefficient. Currently, only longitudinal dispersion is implemented in PFLOTRAN.
The porosity may be calculated from the mineral volume fractions according to the relation

\[ \varphi = 1 - \sum_m \varphi_m. \]  

(10-104)

The temperature dependence of the diffusion coefficient is defined through the relation

\[ D_m(T) = D_m^0 \exp \left[ \frac{A_D}{R} \left( \frac{1}{T_0} - \frac{1}{T} \right) \right], \]  

(10-105)

with diffusion activation energy \( A_D \) in kJ/mol. The quantity \( D_m^0 \) denotes the diffusion coefficient at the reference temperature \( T_0 \) taken as 25°C and the quantity \( R \) denotes the gas constant \((8.317 \times 10^{-3} \text{ kJ/mol/K})\). The temperature \( T \) is in Kelvin.

The quantity \( Q_j \) denotes a source/sink term

\[ Q_j = \sum_n \frac{qM}{\rho} \Psi_j \delta(r - r_n), \]  

(10-106)

where \( qM \) denotes a mass rate in units of kg/s, \( \rho \) denotes the fluid density in kg/m\(^3\), and \( r_n \) refers to the location of the \( n \)th source/sink. The quantity \( S_j \) represents the sorbed concentration of the \( j \)th primary species considered in more detail in the next section.

Molality \( m_i \) and molarity \( C_i \) are related by the density of water \( \rho_w \) according to

\[ C_i = \rho_w m_i. \]  

(10-107)

The activity of water is calculated from the approximate relation

\[ a_{H_2O} = 1 - 0.017 \sum_i m_i. \]  

(10-108)

### 10.9.1 Mineral Precipitation and Dissolution

The reaction rate \( I_m \) is based on transition state theory with the form

\[ I_m = -a_m \left( \sum_l k_{ml}(T)P_{ml} \right) \left( 1 - K_m Q_m \right), \]  

(10-109)

where the sum over \( l \) represents contributions from parallel reaction mechanisms such as pH dependence etc., and where \( K_m \) denotes the equilibrium constant, \( a_m \) refers to the specific mineral surface area, and the ion activity product \( Q_m \) is defined as

\[ Q_m = \prod_j (\gamma_j m_j)^{\nu_{jm}}, \]  

(10-110)

with molality \( m_j \). The rate constant \( k_{ml} \) is a function of temperature given by the Arrhenius relation

\[ k_{ml}(T) = k_{ml}^0 \exp \left[ \frac{E_{ml}}{R} \left( \frac{1}{T_0} - \frac{1}{T} \right) \right], \]  

(10-111)
where $k^0_{ml}$ refers to the rate constant at the reference temperature $T_0$ taken as $25^\circ C$, with $T$ in units of Kelvin, $E_{ml}$ denotes the activation energy (kJ/mol), and the quantity $P_{ml}$ denotes the prefactor for the $l$th parallel reaction with the form

$$P_{ml} = \prod \frac{(\gamma_i m_i)^{\alpha_i m}}{1 + K_m l (\gamma_i m_i)^{\beta_i m}}, \quad (10-112)$$

where the product index $i$ generally runs over both primary and secondary species, the quantities $\alpha_{il}^m$ and $\beta_{il}^m$ refer to prefactor coefficients, and $K_{ml}$ is an attenuation factor. The quantity $R$ denotes the gas constant ($8.317 \times 10^{-3}$ kJ/mol/K).

**10.9.1.1 Rate Limiter** A rate-limited form of the mineral kinetic rate law can be devised according to the expression

$$\hat{I}_m = -a_m \left( \sum l P_{ml} k_{ml} \right) \left[ \frac{1 - (K_m Q_m)^{1/\sigma_m}}{1 + k_{ml} l (K_m Q_m)^{1/\sigma_m}} \right], \quad (10-113)$$

with rate-limiter $r_{lim}$. In the limit $K_m Q_m \to \infty$, the rate becomes

$$\lim_{K_m Q_m \to \infty} \hat{I}_m = k_{ml} l a_m \sum l P_{ml}. \quad (10-114)$$

Defining the affinity factor

$$\Omega_m = 1 - (K_m Q_m)^{1/\sigma_m}, \quad (10-115)$$

or

$$K_m Q_m = \left( 1 - \Omega_m \right)^{\sigma_m}, \quad (10-116)$$

the rate may be expressed alternatively as

$$\hat{I}_m = -a_m \sum l P_{ml} k_{ml} \frac{\Omega_m}{1 + k_{ml} l (1 - \Omega_m)}. \quad (10-117)$$

**10.9.1.2 Changes in Material Properties** Porosity, permeability, tortuosity and mineral surface area may be updated optionally due to mineral precipitation and dissolution reactions according to the relations

$$\varphi = 1 - \sum m \varphi_m, \quad (10-118)$$

$$k = k_0 f(\varphi, \varphi_0, \varphi_c, a), \quad (10-119)$$

with

$$f = \left( \frac{\varphi - \varphi_c}{\varphi_0 - \varphi_c} \right)^a, \quad (10-120)$$

$$= f_{min} \text{ if } \varphi \leq \varphi_c, \quad (10-121)$$
\[
\tau = \tau_0 \left( \frac{\varphi}{\varphi_0} \right)^b, \quad (10-122)
\]

and
\[
a_m = a_0^0 \left( \frac{\varphi_m}{\varphi_0^m} \right)^n \left( \frac{1 - \varphi}{1 - \varphi_0} \right)^{n'}, \quad (10-123)
\]

where the super/subscript 0 denotes initial values, with a typical value for \( n \) of 2/3 reflecting the surface to volume ratio. Note that this relation only applies to primary minerals \((\varphi_m^0 \neq 0)\). The quantity \( \varphi_c \) refers to a critical porosity below which the permeability is assumed to be constant with scale factor \( f_{\text{min}} \).

In PFLOTRAN the solid is represented as an aggregate of minerals described quantitatively by specifying its porosity \( \varphi \) and the volume fraction \( \varphi_m \) of each primary mineral. It is not necessary that Eqn.(10-118) relating porosity and mineral volume fractions holds. Typically, however, the solid composition is specified by giving the mass fraction \( y_m \) of each of the primary minerals making up the solid phase. The volume fraction is related to mole \( x_m \) and mass \( y_m \) fractions by the expressions

\[
\varphi_m = (1 - \varphi) \frac{x_m V_m}{\sum_{m'} x_m' V_m'}, \quad (10-124a)
\]
\[
= (1 - \varphi) \frac{y_m \rho_m^{-1}}{\sum_{m'} y_m' \rho_{m'}^{-1}}, \quad (10-124b)
\]

with inverse relation
\[
x_m = \frac{\varphi_m}{V_m \eta_s (1 - \varphi)}, \quad (10-125)
\]

and similarly for the mass fraction, where
\[
\rho_m = W_m V_m^{-1}, \quad (10-126)
\]

and the solid molar density \( \eta_s \) is given by
\[
\eta_s = \frac{1}{\sum_m x_m V_m}. \quad (10-127)
\]

In these relations \( W_m \) refers to the formula weight and \( V_m \) the molar volume of the \( m \)th mineral. The solid molar density is related to the mass density \( \rho_s \) by
\[
\rho_s = W_s \eta_s, \quad (10-128)
\]

with the mean molecular weight \( W_s \) of the solid phase equal to
\[
W_s = \sum_m x_m W_m = \frac{1}{\sum_m W_m^{-1} y_m}. \quad (10-129)
\]

Mass and mole fractions are related by the expression
\[
W_m x_m = W_s y_m. \quad (10-130)
\]
10.9.1.3 Affinity Threshold An affinity threshold \( f \) for precipitation may be introduced which only allows precipitation to occur if \( K_m Q_m > f > 1 \).

10.9.1.4 Surface Armoring Surface armoring occurs when one mineral precipitates on top of another mineral, blocking that mineral from reacting. Thus suppose mineral \( \mathcal{M}_m \) is being replaced by the secondary mineral \( \mathcal{M}_{m'} \). Blocking may be described phenomenologically by the surface area relation

\[
a_m(t) = a_m^0 \left( \frac{\varphi_m}{\varphi_m^0} \right)^n \left( \frac{1 - \varphi}{1 - \varphi_0} \right)^{n'} \left( \frac{\varphi_{m'} - \varphi_m^c}{\varphi_{m'}^c - \varphi_m^c} \right)^{n''},
\]

for \( \varphi_{m'} < \varphi_{m'}^c \), and

\[
a_m = 0,
\]

if \( \varphi_{m'}(t) \geq \varphi_{m'}^c \), where \( \varphi_{m'}^c \) represents the critical volume fraction necessary for complete blocking of the reaction of mineral \( \mathcal{M}_m \).

10.9.2 Sorption

Sorption reactions incorporated into PFLOTRAN consist of ion exchange and surface complexation reactions for both equilibrium and multirate formulations.

10.9.2.1 Ion Exchange Ion exchange reactions may be represented either in terms of bulk- or mineral-specific rock properties. Changes in bulk sorption properties can be expected as a result of mineral reactions. However, only the mineral-based formulation enables these effects to be captured in the model. The bulk rock sorption site concentration \( \omega_\alpha \), in units of moles of sites per bulk sediment volume (mol/dm\(^3\)), is related to the bulk cation exchange capacity \( Q_\alpha \) (mol/kg) by the expression

\[
\omega_\alpha = \frac{N_{\text{site}}}{V} = \frac{N_{\text{site}} M_s V_s}{M_s V_s V} = Q_\alpha \rho_s (1 - \phi).
\]

The cation exchange capacity associated with the \( m \)th mineral is defined on a molar basis as

\[
\omega_{m,\text{CEC}} = \frac{N_m}{V} = \frac{N_m M_m V_m}{M_m V_m V} = Q_{m,\text{CEC}} \rho_m \phi_m.
\]

In PFLOTRAN ion exchange reactions are expressed in the form

\[
z_i A_j + z_j X_{\alpha}^\alpha A_i \rightleftharpoons z_j A_i + z_i X_{\alpha}^\alpha A_j,
\]

with valencies \( z_j, z_i \) of cations \( A_j \) and \( A_i \), respectively. The reference cation is denoted by \( A_j \) and \( A_i, i \neq j \) represents all other cations. The corresponding mass action equation is given by

\[
K_{ji}^\alpha = \frac{(k_j^\alpha)^{z_i}}{(k_i^\alpha)^{z_j}} = \left( \frac{X_{\alpha}^\alpha}{a_j} \right)^{z_i} \left( \frac{a_i}{X_{\alpha}^\alpha} \right)^{z_j}.
\]

Using the Gaines-Thomas convention, the equivalent fractions \( X_k^\alpha \) are defined by

\[
X_k^\alpha = \sum l z_l S_k^\alpha = \frac{z_k S_k^\alpha}{\omega_\alpha S_k^\alpha},
\]

\(< 179 >\)
with
\[ \sum_{k} X_{k}^{\alpha} = 1. \] (10-138)

The site concentration \( \omega_{\alpha} \) is defined by
\[ \omega_{\alpha} = \sum_{k} z_{k} S_{k}^{\alpha}, \] (10-139)

where \( \omega_{\alpha} \) is related to the cation exchange capacity \( Q_{\alpha} \) (CEC) by the expression
\[ \omega_{\alpha} = (1 - \varphi) \rho_s Q_{\alpha}, \] (10-140)

with solid density \( \rho_s \) and porosity \( \varphi \).

An alternative form of reactions 10-135 often found in the literature is
\[ \frac{1}{z_{j}} A_{j} + \frac{1}{z_{i}} X_{z_{i}}^{\alpha} A_{i} \rightleftharpoons \frac{1}{z_{i}} A_{i} + \frac{1}{z_{j}} X_{z_{j}}^{\alpha} A_{j}, \] (10-141)

obtained by dividing reaction 10-135 through by the product \( z_{i} z_{j} \). In addition the reaction may be written in reverse order. The mass action equations corresponding to reactions 10-141 have the form
\[ K'_{ji}^{\alpha} = \left( \frac{k'_{j}^{\alpha}}{k'_{i}^{\alpha}} \right)^{1/z_{j}} = \left( \frac{X_{j}^{\alpha}}{X_{i}^{\alpha}} \right)^{1/z_{i}} \] (10-142)

The selectivity coefficients corresponding to the two forms are related by the expression
\[ K_{ji}^{\alpha} = \left( K'_{ji}^{\alpha} \right)^{z_{j} z_{j}}, \] (10-143)

and similarly for \( k_{i}^{\alpha}, k_{j}^{\alpha} \). When comparing with other formulations it is important that the user determine which form of the ion exchange reactions are being used and make the appropriate transformations.

For equivalent exchange \( (z_{j} = z_{i} = z) \), an explicit expression exists for the sorbed concentrations given by
\[ S_{j}^{\alpha} = \omega_{\alpha} \frac{k_{j}^{\alpha} x_{j} m_{j}}{z \sum_{l} k_{l}^{\alpha} x_{l} m_{l}}, \] (10-144)

where \( m_{k} \) denotes the \( k \)th cation molality. This expression follows directly from the mass action equations and conservation of exchange sites.

In the more general case \( (z_{i} \neq z_{j}) \) it is necessary to solve the nonlinear equation
\[ X_{j}^{\alpha} + \sum_{i \neq j} X_{i}^{\alpha} = 1, \] (10-145)

for the reference cation mole fraction \( X_{j} \). From the mass action equation Eqn.(10-136) it follows that
\[ X_{i}^{\alpha} = k_{i}^{\alpha} a_{i} \left( \frac{X_{j}^{\alpha}}{k_{j}^{\alpha} a_{j}} \right)^{z_{i}/z_{j}}. \] (10-146)
Defining the function
\[ f(X_j^\alpha) = X_j^\alpha + \sum_{i \neq j} X_i^\alpha X_j^\alpha - 1, \]  
its derivative is given by
\[ \frac{df}{dX_j^\alpha} = 1 - \frac{1}{z_j X_j^\alpha} \sum_{i \neq j} z_i k_i^{\alpha} \frac{a_i}{k_j^{\alpha} a_j} \cdot \]  
The reference mole fraction is then obtained by Newton-Raphson iteration
\[ (X_j^\alpha)^{k+1} = (X_j^\alpha)^k - \frac{f[(X_j^\alpha)^k]}{df[(X_j^\alpha)^k]} \cdot \]  
The sorbed concentration for the \( j \)th cation appearing in the accumulation term is given by
\[ S_j^\alpha = \frac{\omega_\alpha}{z_j} X_j^\alpha, \]  
with the derivatives for \( j \neq l \)
\[ \frac{\partial S_j^\alpha}{\partial m_l} = -\frac{\omega_\alpha}{m_l} \sum_l z_l X_l^\alpha, \]  
and for \( j = l \)
\[ \frac{\partial S_j^\alpha}{\partial m_j} = \frac{\omega_\alpha X_j^\alpha}{z_j m_j} \left( 1 - \frac{z_j X_j^\alpha}{\sum_l z_l X_l^\alpha} \right), \]  
\[ = \frac{S_j^\alpha}{m_j} \left( 1 - \frac{z_j^2 S_j^\alpha}{\sum_l z_l^2 S_l^\alpha} \right). \]  

10.9.2.2 Surface Complexation  
Surface complexation reactions are assumed to have the form
\[ \nu_\alpha > \chi_\alpha + \sum_j \nu_{ji} A_j \rightleftharpoons > S_{i\alpha}, \]  
for the \( i \)th surface complex \( > S_{i\alpha} \) on site \( \alpha \) and empty site \( > \chi_\alpha \). As follows from the corresponding mass action equation the equilibrium sorption concentration \( S_{i\alpha}^{eq} \) is given by
\[ S_{i\alpha}^{eq} = \frac{\omega_\alpha K_i Q_i}{1 + \sum_l K_l Q_l}, \]  
\[ < 181 > \]
and the empty site concentration by

$$S_{\alpha}^{\text{eq}} = \frac{\omega_{\alpha}}{1 + \sum_{l} K_{l} Q_{l}}.$$  

(10-155)

where the ion activity product $Q_i$ is defined by

$$Q_i = \prod_{j} (\gamma_j C_j)^{\nu_{ji}}.$$  

(10-156)

The site concentration $\omega_{\alpha}$ satisfies the relation

$$\omega_{\alpha} = S_{\alpha} + \sum_{i} S_{i\alpha},$$  

(10-157)

and is constant. The equilibrium sorbed concentration $S_{\text{eq}}^{\alpha}$ is defined as

$$S_{\text{eq}}^{\alpha} = \sum_{i} \nu_{ji} S_{i\alpha} = \frac{\omega_{\alpha}}{1 + \sum_{l} K_{l} Q_{l}} \sum_{i} \nu_{ji} K_{i} Q_{i}.$$  

(10-158)

### 10.9.2.3 Multirate Sorption

In the multirate model the rates of sorption reactions are described through a kinetic relation given by

$$\frac{\partial S_{i\alpha}}{\partial t} = k_{\alpha} (S_{i\alpha}^{\text{eq}} - S_{i\alpha}),$$  

(10-159)

for surface complexes, and

$$\frac{\partial S_{\alpha}}{\partial t} = - \sum_{i} k_{\alpha} (S_{i\alpha}^{\text{eq}} - S_{i\alpha}),$$  

(10-160)

$$= k_{\alpha} (S_{\alpha}^{\text{eq}} - S_{\alpha}),$$  

(10-161)

for empty sites, where $S_{\text{eq}}^{\alpha}$ denotes the equilibrium sorbed concentration. For simplicity, in what follows it is assumed that $\nu_{\alpha} = 1$. With each site $\alpha$ is associated a rate constant $k_{\alpha}$ and site concentration $\omega_{\alpha}$. These quantities are defined through a given distribution of sites $\varphi(\alpha)$, such that

$$\int_{0}^{\infty} \varphi(k_{\alpha}) dk_{\alpha} = 1.$$  

(10-162)

The fraction of sites $f_{\alpha}$ belonging to site $\alpha$ is determined from the relation

$$f_{\alpha} = \int_{k_{\alpha} - \Delta k_{\alpha}/2}^{k_{\alpha} + \Delta k_{\alpha}/2} \varphi(k_{\alpha}) dk_{\alpha} \approx \varphi(k_{\alpha}) \Delta k_{\alpha},$$  

(10-163)

with the property that

$$\sum_{\alpha} f_{\alpha} = 1.$$  

(10-164)

Given that the total site concentration is $\omega$, then the site concentration $\omega_{\alpha}$ associated with site $\alpha$ is equal to

$$\omega_{\alpha} = f_{\alpha} \omega.$$  

(10-165)
An alternative form of these equations is obtained by introducing the total sorbed concentration for the \( j \)th primary species for each site defined as

\[
S_{j\alpha} = \sum_i \nu_{ji} S_{i\alpha}.
\]  

(10-166)

Then the transport equations become

\[
\frac{\partial}{\partial t} \left( \varphi \Psi_j + \sum_{\alpha} S_{j\alpha} \right) + \nabla \cdot \Omega_j = -\sum_m \nu_{jm} I_m.
\]  

(10-167)

The total sorbed concentrations are obtained from the equations

\[
\frac{\partial S_{j\alpha}}{\partial t} = k_\alpha (S_{j\alpha}^{eq} - S_{j\alpha}).
\]  

(10-168)

10.9.3 Sorption Isotherm <Under Revision>

The distribution coefficient \( \tilde{K}^D \) [m³ kg⁻¹] is customarily defined as the ratio of sorbed to aqueous concentrations with the sorbed concentration referenced to the mass of solid as given by

\[
\tilde{K}^D_j = \frac{M_j^s}{M_j^a V_l},
\]  

(10-169a)

\[
= \frac{N_j^s/M_s}{N_j^a/V_l},
\]  

(10-169b)

\[
= \frac{\tilde{S}_j}{C_j} = \frac{1}{\rho_w m_j},
\]  

(10-169c)

where \( M_j^s = W_j N_j^s / M_s \), \( M_j^a = W_j N_j^a / M_s \), refers to the mass and number of moles of sorbed and aqueous solute related by the formula weight \( W_j \) of the \( j \)th species, \( M_s \) refers to the mass of the solid, \( V_l \) denotes the aqueous volume, \( \tilde{S}_j = N_j^s / M_s \) [mol kg⁻¹] represents the sorbed concentration referenced to the mass of solid, \( C_j = N_j^a / V_l \) denotes molarity, and \( m_j = C_j / \rho_w \) represents molality, where \( \rho_w \) is the density of pure water.

The distribution coefficient \( \tilde{K}^D \) may be related to its dimensionless counterpart \( K^D_j [\text{—}] \) defined by

\[
K^D_j = \frac{N_j^s}{N_j^a V} = \frac{1}{\varphi s_l C_j},
\]  

(10-170)

by writing

\[
K^D_j = \frac{N_j^s M_s V_s V_p V_l}{M_s V_s V_p V_l N_j^a},
\]  

(10-171)

\[
= \rho_s \frac{1 - \varphi}{\varphi s_l} \tilde{K}^D_j = \frac{\rho_b}{\varphi s_l} \tilde{K}^D_j,
\]  

(10-172)

with grain density \( \rho_s = M_s/V_s \), bulk density \( \rho_b = (1 - \varphi) \rho_s \), porosity \( \varphi = V_p / V \), and saturation \( s_l = V_l / V_p \).
An alternative definition of the distribution coefficient denoted by $\hat{K}_D^j$ [kg m$^{-3}$] is obtained by using molality to define the solute concentration and referencing the sorbed concentration to the bulk volume $V$

$$\hat{K}_D^j = \frac{N_j^s/V}{N_j^{aq}/M_w} = \frac{S_j}{m_j}. \quad (10-173)$$

A sorption isotherm $S_j$ may be specified for any primary species $A_j$ resulting in the transport equation

$$\frac{\partial}{\partial t} \varphi_{s_l} C_j + \nabla \cdot F_j = - \frac{\partial S_j}{\partial t}, \quad (10-174)$$

for a partially saturated medium. Substituting $S_j = \varphi_{s_l} K_D^j C_j$ from Eqn.(10-170) and introducing the retardation $R_j$ gives

$$\frac{\partial}{\partial t} R_j \varphi_{s_l} C_j + \nabla \cdot F_j = 0, \quad (10-175)$$

with the retardation given by the alternative forms

$$R_j = 1 + K_D^j, \quad \text{(dimensionless)}, \quad (10-176a)$$

$$= 1 + \frac{\rho_b}{\varphi_{s_l}} \hat{K}_D^j, \quad \text{(conventional)}, \quad (10-176b)$$

$$= 1 + \frac{1}{\varphi_{s_l} \rho_w} \hat{K}_D^j, \quad \text{(molality-based)}. \quad (10-176c)$$

Three distinct models are available for the sorption isotherm $S_j$ in PFLOTRAN:

- linear $K_D$ model:
  $$S_j = \varphi_{s_l} K_D^j C_j = \hat{K}_D^j m_j, \quad (10-177)$$
  with distribution coefficient $\hat{K}_D^j$,

- Langmuir isotherm:
  $$S_j = \frac{K_L^j b_L^j C_j / \rho_w}{1 + K_L^j C_j / \rho_w} = \frac{K_L^j b_L^j m_j}{1 + K_L^j m_j}, \quad (10-178)$$
  with Langmuir coefficients $K_L^j$ and $b_L^j$,

- Freundlich isotherm:
  $$S_j = K_F^j \left( \frac{C_j}{\rho_w} \right)^{(1/n_F^j)} = K_F^j (m_j)^{(1/n_F^j)}, \quad (10-179)$$
  with coefficients $K_F^j$ and $n_F^j$.

## 10.9.4 Colloid-Facilitated Transport

Colloid-facilitated transport is implemented into PFLOTRAN based on surface complexation reactions. Competition between mobile and immobile colloids and stationary mineral surfaces is taken into account. Colloid filtration processes are not currently implemented into PFLOTRAN. A
colloid is treated as a solid particle suspended in solution or attached to a mineral surface. Colloids may be generated through nucleation of minerals in solution, although this effect is not included currently in the code.

Three separate reactions may take place involving competition between mobile and immobile colloids and mineral surfaces

\[ > X^m_k + \sum_j \nu_{jk} A_j \rightleftharpoons > S^{m}_k, \]  
(10-180)

\[ > X^{im}_k + \sum_j \nu_{jk} A_j \rightleftharpoons > S^{im}_k, \]  
(10-181)

\[ > X^s_k + \sum_j \nu_{jk} A_j \rightleftharpoons > S^s_k, \]  
(10-182)

with corresponding reaction rates \( I^m_k \), \( I^{im}_k \), and \( I^s_k \), where the superscripts \( s \), \( m \), and \( im \) denote mineral surfaces, and mobile and immobile colloids, respectively. In addition, reaction with minerals \( \mathcal{M}_s \) may occur according to the reaction

\[ \sum_j \nu_{js} A_j \rightleftharpoons \mathcal{M}_s. \]  
(10-183)

The transport equations for primary species, mobile and immobile colloids, read

\[ \frac{\partial}{\partial t} \varphi_{si} \Psi^l_j + \nabla \cdot \Omega^l_j = - \sum_k \nu_{jk} (I^m_k + I^{im}_k + \sum_s I^s_k) - \sum_s \nu_{js} I_s, \]  
(10-184)

\[ \frac{\partial}{\partial t} \varphi_{si} S^{m}_k + \nabla \cdot q_c S^{m}_k = I^m_k, \]  
(10-185)

\[ \frac{\partial}{\partial t} S^{im}_k = I^{im}_k, \]  
(10-186)

\[ \frac{\partial}{\partial t} S^s_k = I^s_k, \]  
(10-187)

where \( q_c \) denotes the colloid Darcy velocity which may be greater than the fluid velocity \( q \). For conditions of local equilibrium the sorption reaction rates may be eliminated and replaced by algebraic sorption isotherms to yield

\[ \frac{\partial}{\partial t} \left[ \varphi_{si} \Psi^l_j + \sum_k \nu_{jk} (\varphi_{si} S^{m}_k + S^{im}_k + \sum_s S^s_k) \right] + \nabla \cdot \left( \Omega^l_j + q_c \sum_k \nu_{jk} S^{m}_k \right) = - \sum_s \nu_{js} I_s. \]  
(10-188)

In the kinetic case either form of the primary species transport equations given by Eqn.(10-184) or (10-188) can be used provided it is coupled with the appropriate kinetic equations Eqns.(10-185)–(10-187). The mobile case leads to additional equations that must be solved simultaneously with the primary species equations. A typical expression for \( I^m_k \) might be

\[ I^m_k = k_k (S^{eq}_k - S^{eq}_{km}), \]  
(10-189)

with rate constant \( k_k \) and where \( S^{eq}_{km} \) is a known function of the solute concentrations. In this case, Eqn.(10-185) must be added to the primary species transport equations. Further reduction of the transport equations for the case where a flux term is present in the kinetic equation is not possible in general for complex flux terms.
10.10 Tracer Mean Age

PFLOTRAN implements the Eulerian formulation of solute age for a nonreactive tracer following Goode (1996). PFLOTRAN solves the advection-diffusion/dispersion equation for the mean age given by

\[
\frac{\partial}{\partial t} \varphi sAC + \nabla \cdot \left( qAC - \varphi sD \nabla (AC) \right) = \varphi sC,
\]

(10-190)

where \( A \) denotes the mean age of the tracer with concentration \( C \). Other quantities appearing in the age equation are identical to the tracer transport equation for a partially saturated porous medium with saturation state \( s \). The age and tracer transport equations are solved simultaneously for the age-concentration \( \alpha = AC \) and tracer concentration \( C \). The age-concentration \( \alpha \) satisfies the usual advection-diffusion-dispersion equation with a source term on the right-hand side.

The mean tracer is calculated in PFLOTRAN by adding the species Tracer_Age together with Tracer to the list of primary species

```plaintext
PRIMARY_SPECIES
  Tracer
  Tracer_Age
/
```

including sorption through a constant \( K_d \) model if desired

```plaintext
SORPTION
  ISOTHERM_REACTIONS
    Tracer
      TYPE LINEAR
      DISTRIBUTION_COEFFICIENT 500. ! kg water/m^3 bulk
    /
    Tracer_Age
      TYPE LINEAR
      DISTRIBUTION_COEFFICIENT 500. ! kg water/m^3 bulk
    /
/
```

and specifying these species in the initial and boundary CONSTRAINT condition as e.g.: 

```plaintext
CONSTRAINT initial
  CONCENTRATIONS
    Tracer 1.e-8 F
    Tracer_Age 1.e-16 F
/
```

Output is given in terms of \( \alpha \) and \( C \) from which the mean age \( A \) can be obtained as \( A = \alpha/C \).
10.11 Thermodynamic Database

PFLOTRAN reads thermodynamic data from a database that may be customized by the user. Reactions included in the database consist of aqueous complexation, mineral precipitation and dissolution, gaseous reactions, and surface complexation. Ion exchange reactions and their selectivity coefficients are entered directly from the input file. A standard database supplied with the code is referred to as `hanford.dat` and is found in the `.database` directory in the PFLOTRAN mercurial repository. This database is an ascii text file that can be edited by any editor and is equivalent to the EQ3/6 database:

data0.com.V8.R6
THERMODYNAMIC DATABASE
generated by GEMBOCHS.V2-Jewel.src.R5 03-dec-1996 14:19:25

The database provides equilibrium constants in the form of log $K$ values at a specified set of temperatures listed in the top line of the database. A least squares fit is used to interpolate the log $K$ values between the database temperatures using a Maier-Kelly expansion of the form

$$
\log K = c_{-1} \ln T + c_0 + c_1 T + \frac{c_2}{T} + \frac{c_3}{T^2},
$$

(10-191)

with fit coefficients $c_i$. The thermodynamic database stores all chemical reaction properties (equilibrium constant $\log K_r$, reaction stoichiometry $\nu_{ir}$, species valence $z_i$, Debye parameter $a_i$, mineral molar volume $V_m$, and formula weight $w_i$) used in PFLOTRAN. The database is divided into 5 blocks as listed in Table 11, consisting of database primary species, aqueous complex reactions, gaseous reactions, mineral reactions, and surface complexation reactions. Each block is terminated by a line beginning with ‘null’. The quantity $N_{temp}$ refers to the number of temperatures at which log $K$ values are stored in the database. In the `hanford.dat` database $N_{temp} = 8$ with equilibrium constants stored at the temperatures: 0, 25, 60, 100, 150, 200, 250, and 300°C. The pressure is assumed to lie along the saturation curve of pure water for temperatures above 25°C and is equal to 1 bar at lower temperatures. Reactions in the database are assumed to be written in the canonical form

$$
\mathcal{A}_r \equiv \sum_{i=1}^{n_{spec}} \nu_{ir} \mathcal{A}_i,
$$

(10-192)

for species $\mathcal{A}_r$, where $n_{spec}$ refers to the number of aqueous or gaseous species $\mathcal{A}_i$ on the right-hand side of the reaction. Redox reactions in the standard database `hanford.dat` are usually written in terms of $O_2(g)$. Complexation reactions involving redox sensitive species are written in such a manner as to preserve the redox state.

10.12 Eh, pe

Output for Eh and pe is calculated from the half-cell reaction

$$
2 \text{H}_2\text{O} - 4 \text{H}^+ - 4 e^- \rightleftharpoons \text{O}_2,
$$

(10-193)
Table 11: Format of thermodynamic database.

<table>
<thead>
<tr>
<th>Species Type</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Species</td>
<td>name, $a_0$, $z$, $w$</td>
</tr>
<tr>
<td>Secondary Species</td>
<td>name, nspec, ($\nu(n)$, name($n$), $n=1$, nspec), $\log K(1:N_{\text{temp}})$, $a_0$, $z$, $w$</td>
</tr>
<tr>
<td>Gaseous Species</td>
<td>name, $\overline{V}$, nspec, ($\nu(n)$, name($n$), $n=1$, nspec), $\log K(1:N_{\text{temp}})$, $w$</td>
</tr>
<tr>
<td>Minerals</td>
<td>name, $\overline{V}$, nspec, ($\nu(n)$, name($n$), $n=1$, nspec), $\log K(1:N_{\text{temp}})$, $w$</td>
</tr>
<tr>
<td>Surface Complexes</td>
<td>$&gt;name$, nspec, $\nu$, $&gt;site$, ($\nu(n)$, name($n$), $n=1$, nspec-1), $\log K(1:N_{\text{temp}})$, $z$, $w$</td>
</tr>
</tbody>
</table>

with the corresponding equilibrium constant fit to the Maier-Kelly expansion Eqn.(10-191). The fit coefficients are listed in Table 12.

Table 12: Fit coefficients for $\log K$ of reaction 10-193.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_{-1}$</td>
<td>6.745529048</td>
</tr>
<tr>
<td>$c_0$</td>
<td>-48.295936593</td>
</tr>
<tr>
<td>$c_1$</td>
<td>0.000557816</td>
</tr>
<tr>
<td>$c_2$</td>
<td>27780.749538022</td>
</tr>
<tr>
<td>$c_3$</td>
<td>4027.337694858</td>
</tr>
</tbody>
</table>

10.13 Geomechanics

In PFLOTRAN, linear elasticity model is assumed as the constitutive model for deformation of the rock. Biot’s model is used to incorporate the effect of flow on the geomechanics. In addition, the effect of temperature on geomechanics is considered via coefficient of thermal expansion. The following governing equations are used:

\[
\nabla \cdot [\sigma] + \rho\mathbf{b} = 0 \quad \text{in } \Omega, \tag{10-194a}
\]

\[
\sigma = \lambda \text{tr} (\varepsilon) + 2\mu \varepsilon - \beta (p - p_0) I - \alpha (T - T_0) I, \tag{10-194b}
\]

\[
\varepsilon = \frac{1}{2} \left( \nabla \mathbf{u}(x) + [\nabla \mathbf{u}(x)]^T \right), \tag{10-194c}
\]

\[
\mathbf{u}(x) = \mathbf{u}^p(x) \quad \text{on } \Gamma_D, \tag{10-194d}
\]

\[
\sigma \mathbf{n}(x) = \mathbf{t}^p(x) \quad \text{on } \Gamma_N, \tag{10-194e}
\]

where $\mathbf{u}$ is the unknown displacement field, $\sigma$ is the Cauchy stress tensor, $\lambda$, $\mu$ are Lamé parameters (Young’s modulus and Poisson’s ratio can be related to these two parameters), $\mathbf{b}$ is the specific body force (which is gravity in most cases), $\mathbf{n}$ is the outward normal to the boundary $\Gamma_N$. Also, $\mathbf{u}^p$ is the prescribed values of $\mathbf{u}$ on the Dirichlet part of the boundary $\Gamma_D$, and $\mathbf{t}^p$ is the prescribed traction on $\Gamma_N$. Additionally, $\beta$ is the Biot’s coefficient, $\alpha$ is the coefficient of thermal expansion, $p$, $T$ are the fluid pressure and temperature, obtained by solving subsurface flow problem. Also, $p_0$ and $T_0$ are set to initial pressure and temperature in the domain, $\varepsilon$ is the strain tensor and tr is the trace
of a second order tensor, $\Omega$ is the domain, and $I$ is the identity tensor. Note that stress is assumed 
*positive under tension*. The effect of deformation on the pore structure is accounted for via

$$\phi = \frac{\phi_0}{1 + (1 - \phi_0)\text{tr}(\epsilon)}.$$ (10-195)

Note that the above equations are solved using finite element method (Galerkin finite element) with the displacements solved for at the vertices. Since, the flow equations are solved via finite volume method with unknowns such as pressure and temperature solved for at the cell centers, in order to transfer data from the subsurface to geomechanics grid without interpolation, the geomechanics grid is constructed such that the vertices of the geomechanics grid coincide with the cell centers of the subsurface mesh. That is, the dual mesh of the subsurface mesh is used for the geomechanics solve.

Also, geomechanics grid is read as an unstructured grid. Even if one needs to work with a structured grid, they have to set it up in the unstructured grid format.

## 11 Appendix: Method of Solution

The flow and heat equations (Modes: RICHARDS, MPHASE, FLASH2, TH, . . .) are solved using a fully implicit backward Euler approach based on Newton-Krylov iteration. Both fully implicit backward Euler and operator splitting solution methods are supported for reactive transport.

### 11.1 Integrated Finite Volume Discretization

The governing partial differential equations for mass conservation can be written in the general form

$$\frac{\partial}{\partial t} A_j + \nabla \cdot F_j = Q_j,$$ (11-1)

with accumulation $A_j$, flux $F_j$ and source/sink $Q_j$. Integrating over a REV corresponding to the $n$th grid cell with volume $V_n$ yields

$$\frac{d}{dt} \int_{V_n} A_j \, dV + \int_{V_n} \nabla \cdot F_j \, dV = \int_{V_n} Q_j \, dV.$$ (11-2)

The accumulation term has the finite volume form

$$\frac{d}{dt} \int_{V_n} A_j \, dV = \frac{A_j^{t+\Delta t} - A_j^t}{\Delta t} V_n,$$ (11-3)

with time step $\Delta t$. The flux term can be expanded as a surface integral using Gauss’ theorem

$$\int_{V_n} \nabla \cdot F_j = \int_{\partial V_n} F_j \cdot dS,$$ (11-4)

$$= \sum_{n'} F_{j,n'n} A_{nn'},$$ (11-5)

< 189 >
where the latter finite volume form is based on the two-point flux approximation, where the sum over \( n' \) involves nearest neighbor grid cells connected to the \( n \)th node with interfacial area \( A_{nn'} \).

The discretized flux has the form for fluid phase \( \alpha \)

\[
F_{j,nn'}^{\alpha} = (q_{\alpha}X_{\alpha})_{nn'} - (\varphi s_{\alpha}r_{\alpha}D_{\alpha})_{nn'}\frac{X_{\alpha}^{n'} - X_{\alpha}^{n}}{d_{nn'} + d_{n}},
\]

(11-6)

with perpendicular distances to interface \( nn' \) from nodes \( n \) and \( n' \) denoted by \( d_{nn'} \) and \( d_{n} \), respectively. Upstream weighting is used for the advective term

\[
(q_{\alpha}X_{\alpha})_{nn'} = \begin{cases} 
q_{\alpha}^{n'}X_{n'}, & q_{nn'} > 0 \\
q_{\alpha}^{n}X_{n}, & q_{nn'} < 0 
\end{cases}
\]

(11-7)

Depending on the type of source/sink term, the finite volume discretization has the form

\[
\int_{V_{n}} Q_{j} \, dV = Q_{jn} V_{n},
\]

(11-8)

for reaction rates that are distributed continuously over a control volume, or for a well with point source \( Q_{j} = \hat{Q}_{j} \delta (r - r_{0}) \):

\[
\int_{V_{n}} Q_{j} \, dV = \hat{Q}_{jn}.
\]

(11-9)

### 11.2 Fully Implicit

In a fully implicit formulation the nonlinear equations for the residual function \( R \) given by

\[
R(x) = 0,
\]

(11-10)

are solved using an iterative solver based on the Newton-Raphson equations

\[
J^{(i)} \delta x^{(i+1)} = -R^{(i)},
\]

(11-11)

at the \( i \)th iteration. Iteration stops when

\[
\left| R^{(i+1)} \right| < \epsilon,
\]

(11-12)

or if

\[
\left| \delta x^{(i+1)} \right| < \delta.
\]

(11-13)

However, the latter criteria does not necessarily guarantee that the residual equations are satisfied. The solution is updated from the relation

\[
x^{(i+1)} = x^{(i)} + \delta x^{(i+1)}.
\]

(11-14)

For the logarithm of the concentration with \( x = \ln y \), the solution is updated according to the equation

\[
y^{(i+1)} = y^{(i)} e^{\delta \ln y^{(i+1)}}.
\]

(11-15)
11.2.1 Multirate Sorption

The residual function incorporating the multirate sorption model can be further simplified by solving analytically the finite difference form of kinetic sorption equations. This is possible when these equations are linear in the sorbed concentration $S_{j\alpha}$ and because they do not contain a flux term. Thus discretizing Eqn.(10-168) in time using the fully implicit backward Euler method gives

$$\frac{S_{j\alpha}^{t+\Delta t} - S_{j\alpha}^t}{\Delta t} = k_\alpha (f_\alpha S_{j\alpha}^{eq} - S_{j\alpha}^{t+\Delta t}). \quad (11-16)$$

Solving for $S_{j\alpha}^{t+\Delta t}$ yields

$$S_{j\alpha}^{t+\Delta t} = \frac{S_{j\alpha}^t + k_\alpha \Delta t f_\alpha S_{j\alpha}^{eq}}{1 + k_\alpha \Delta t}. \quad (11-17)$$

From this expression the reaction rate can be calculated as

$$\frac{S_{j\alpha}^{t+\Delta t} - S_{j\alpha}^t}{\Delta t} = \frac{k_\alpha}{1 + k_\alpha \Delta t} (f_\alpha S_{j\alpha}^{eq} - S_{j\alpha}^t). \quad (11-18)$$

The right-hand side of this equation is a known function of the solute concentration and thus by substituting into Eqn.(10-167) eliminates the appearance of the unknown sorbed concentration. Once the transport equations are solved over a time step, the sorbed concentrations can be computed from Eqn.(11-17).

11.3 Operator Splitting

Operator splitting involves splitting the reactive transport equations into a nonreactive part and a part incorporating reactions. This is accomplished by writing Eqns.(10-96) as the two coupled equations

$$\frac{\partial}{\partial t} \left( \varphi \sum_\alpha s_\alpha \Psi_{j\alpha}^t \right) + \nabla \cdot \left( \sum_\alpha (q_\alpha - \varphi s_\alpha D_\alpha \nabla) \Psi_{j\alpha}^t \right) = Q_j, \quad (11-19)$$

and

$$\frac{d}{dt} \left( \sum_\alpha s_\alpha \Psi_{j\alpha}^t \right) = - \sum_m \nu_{jm} I_m - \frac{\partial S_j^t}{\partial t}, \quad (11-20)$$

The first set of equations are linear in $\Psi_j$ (for species-independent diffusion coefficients) and solved over over a time step $\Delta t$ resulting in $\Psi_j^*$. The result for $\Psi_j^*$ is inverted to give the concentrations $C_j^*$ by solving the equations

$$\Psi_j^* = C_j^* + \sum_i \nu_{ji} C_i^*, \quad (11-21)$$

where the secondary species concentrations $C_i^*$ are nonlinear functions of the primary species concentrations $C_j^*$. With this result the second set of equations are solved implicitly for $C_j$ at $t + \Delta t$ using $\Psi_j^*$ for the starting value at time $t$. 

< 191 >
11.3.1 Constant $K_d$

As a simple example of operator splitting consider a single component system with retardation described by a constant $K_d$. According to this model the sorbed concentration $S$ is related to the aqueous concentration by the linear equation

$$S = K_d C.$$  
(11-22)

The governing equation is given by

$$\frac{\partial}{\partial t} \varphi C + \nabla \cdot (q C - \varphi D \nabla C) = -\frac{\partial S}{\partial t}.$$  
(11-23)

If $C(x, t; q, D)$ is the solution to the case with no retardation (i.e. $K_d = 0$), then $C(x, t; q/R, D/R)$ is the solution with retardation ($K_d > 0$), with

$$R = 1 + \frac{1}{\varphi} K_d.$$  
(11-24)

Thus propagation of a front is retarded by the retardation factor $R$.

In operator splitting form this equation becomes

$$\frac{\partial}{\partial t} \varphi C + \nabla \cdot (q C - \varphi D \nabla C) = 0,$$  
(11-25)

and

$$\frac{d}{dt} \varphi C = -\frac{dS}{dt}.$$  
(11-26)

The solution to the latter equation is given by

$$\varphi C^{t+\Delta t} - \varphi C^* = -(S^{t+\Delta t} - S^t),$$  
(11-27)

where $C^*$ is the solution to the nonreactive transport equation. Using Eqn.(11-22), this result can be written as

$$C^{t+\Delta t} = \frac{1}{R} C^* + \left(1 - \frac{1}{R}\right) C^t.$$  
(11-28)

Thus for $R = 1$, $C^{t+\Delta t} = C^*$ and the solution advances unretarded. As $R \rightarrow \infty$, $C^{t+\Delta t} \rightarrow C^t$ and the front is fully retarded.
12 Appendix: Reaction Sandbox

12.1 Background

Researchers often have a suite of reactions tailored to a unique problem scenario, but these reaction networks only exist in their respective research codes. The “reaction sandbox” provides these researchers with a venue for implementing user-defined reactions within PFLOTRAN. Reaction networks developed within the reaction sandbox can leverage existing biogeochemical capability within PFLOTRAN (e.g. equilibrium aqueous complexation, mineral precipitation–dissolution, etc.) or function independently. Please note that although the reaction sandbox facilitates the integration of user-defined reactions, the process still requires a basic understanding of PFLOTRAN and its approach to solving reaction through the Newton-Raphson method. For instance, One must understand the purpose and function of the rt_auxvar and global_auxvar objects.

12.2 Implementation

The core framework of reaction sandbox leverages Fortran 2003 object–oriented extendable derived types and methods and consists of two modules:

\[
\text{Reaction\_Sandbox\_module (reaction\_sandbox.F90)}
\]
\[
\text{Reaction\_Sandbox\_Base\_class (reaction\_sandbox\_base.F90)}.
\]

To implement a new reaction within the reaction sandbox, one creates a new class by extending the Reaction\_Sandbox\_Base\_class and adds the new class to the Reaction\_Sandbox\_module. The following steps illustrate this process through the creation of the class Reaction\_Sandbox\_Example\_class that implements a first order decay reaction.

1. Copy reaction\_sandbox\_template.F90 to a new filename (e.g. reaction\_sandbox\_example.F90).
2. Replace all references to Template/template with the new reaction name.
   \[
   \text{Template} \rightarrow \text{Example} \\
   \text{template} \rightarrow \text{example}
   \]
3. Add necessary variables to the module and/or the extended derived type.
   \[
   \text{character (len=MAXWORDLENGTH) :: species\_name} \\
   \text{PetscInt :: species\_id} \\
   \text{PetscReal :: rate\_constant}
   \]
4. Add the necessary functionality within the following subroutines:
a ExampleCreate: Allocate the reaction object, initializing all variables to zero and nullifying arrays. **Be sure to nullify ExampleCreate%next which comes from the base class.** E.g.,

```fortran
allocate(ExampleCreate)
ExampleCreate%species_name = ''
ExampleCreate%species_id = 0
ExampleCreate%rate_constant = 0.d0
nullify(ExampleCreate%next)
```

b ExampleRead: Read parameters in from the input file block EXAMPLE. E.g.,

```fortran
...  
  case('SPECIES_NAME')
      call InputReadWord(input,option,this%species_name, &
        PETSC_TRUE)
      call InputErrorMsg(input,option,'species_name', &
        'CHEMISTRY,REACTION_SANDBOX,EXAMPLE')
  ...
...  
  this%species_id = &
      GetPrimarySpeciesIDFromName(this%species_name, &
        reaction,option)
...  
```

c ExampleSetup: Construct the reaction network (e.g. array allocation, establishing linkages, etc.). E.g.,

```fortran
...  
  Residual(this%species_id) = &
      Residual(this%species_id) - &
      this%rate_constant*porosity* &
      global_auxvar%sat(iphase)*volume*1.d3* &
      rt_auxvar%total(this%species_id,iphase)
...  
  Jacobian(this%species_id,this%species_id) = &
      Jacobian(this%species_id,this%species_id) + &
      this%rate_constant*porosity* &
      global_auxvar%sat(iphase)*volume*1.d3* &
      rt_auxvar%aqueous%dtotal(this%species_id, &
        this%species_id,iphase)
...  
```
e ExampleDestroy: Deallocate any dynamic memory within the class (without deallocating the object itself).

5. Ensure that the methods within the extended derived type point to the proper procedures in the module

```fortran
procedure, public :: ReadInput => ExampleRead
procedure, public :: Setup => ExampleSetup
procedure, public :: Evaluate => ExampleReact
procedure, public :: Destroy => ExampleDestroy
```

6. Within reaction_sandbox.F90:

   a Add Reaction_Sandbox_Example_class+ to the list of modules to be “used” at the top of the file.

   b Add a case statement in RSandboxRead2 for the keyword defining the new reaction and create the reaction within. I.e.

```fortran
  case('EXAMPLE')
    new_sandbox => ExampleCreate()
```

< 195 >