LA-UR-15-20403

Title: PFLOTRAN User Manual: A Massively Parallel Reactive Flow and Transport Model for Describing Surface and Subsurface Processes

Author(s):
Lichtner, Peter
Karra, Satish
Hammond, Glenn
Lu, Chuan
Bisht, Gautam
Kumar, Jitendra
Mills, Richard
Andre, Ben

Intended for: Report

Issued: 2015-01-23

PFLOTRAN User Manual

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

Peter C. Lichtner[α], Glenn E. Hammond[β], Chuan Lu[σ],
Satish Karra[δ], Gautam Bisht[γ1], Benjamin Andre[γ2],
Richard Mills[ϵ1], Jitu Kumar[ϵ2]

[α]	OFM Research	peter.lichtner@gmail.com
[β]	SNL	gehammo@sandia.gov
[σ]	INL	chuan.lu@inl.gov
[δ]	LANL	satkarra@lanl.gov
[γ1]	LBNL	gbisht@lbl.gov
[γ2]	LBNL	bandre@lbl.gov
[ϵ1]	Intel	rtm@eecs.utk.edu
[ϵ2]	ORNL	jitu1503@gmail.com

January 20, 2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

PFLOTRAN solves a system of generally nonlinear partial differential equations describing multiphase, 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:

```
setenv PETSC_DIR 'path to petsc-dev'
```

and

```
setenv PETSC_ARCH MacOSX-10.8.2-gcc4.8.
```

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

```
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.

	Option	Package	
MPI	Parallel	OpenMPI	–download-openmpi=1
		MPICH	–download-mpich=1
Grid	Structured	—	
	Unstructured	METIS and ParMETIS*	–download-metis=1
			–download-parmetis=1
Output	TecPlot, VTK	—	
	HDF5	Parallel HDF5	–download-hdf5=1
Solvers	Iterative (GMRES)/Direct	—	
	Parallel Direct	MUMPS	–download-mumps=1
	Multigrid	ML	–download-ml=1
	Preconditioners	Hypre	–download-hypre=1

* 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.

Option	Description
scco2=1	MPHASE mode for supercritical CO_2
scorpio=1	SCORPIO* parallel IO

* See §3.5 for installing SCORPIO.

```
cd PLOTRAN_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

MODE	Description	Required Databases†
RICHARDS	Solves Richards equation	—
MPHASE	Supercritcal CO_2	`co2data0.dat`
TH	Thermal-Hydrology mode	—
CHEMISTRY	Reactive Transport	`hanford.dat`‡

† 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, ParMETIS/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 ($\leq$ 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:

```
./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-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 \
```

```
--known-sizeof-size_t=8 \
--known-bits-per-byte=8 \
--known-sizeof-MPI_Comm=4 \
--known-sizeof-MPI_Fint=4 \
--known-mpi-long-double=0 \
--with-batch=1 \
--with-shared-libraries=0 \
--with-dynamic=0 \
--with-cc=cc \
--with-cxx=CC \
--with-fc=ftn \
--COPTFLAGS="-tp barcelona-64 -fastsse -Mipa=fast" \
--CXXOPTFLAGS="-tp barcelona-64 -fastsse -Mipa=fast" \
--FOPTFLAGS="-tp barcelona-64 -fastsse" \
--with-debugging=0 \
--with-blas-lib=sci \
--with-lapack-lib=sci \
--with-x=0 \
--with-mpi-dir=$MPICH_DIR \
--download-hypre=1 \
--download-parmetis=1 \
--with-hdf5=1 \
--with-hdf5-dir=$HDF5_DIR \
--known-mpi-shared=0
```

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:

Compile Option	Description
`scorpio=1`:	–enable parallel IO
`coll=1`:	–enable colloid-facilitated transport option
`mfd=1`:	–enable MFD full permeability tensor capability (not operational)

Thus for compiling with the supercritical CO_2 option use: `make ssco2=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 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:

```
:========================== 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`

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:

   `export SCORPIO_DIR=<your-iolib-install-dir>` *(bash shell)*

   or

   `setenv SCORPIO_DIR <your-iolib-install-dir>` *(tcsh/csh shell)*

   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.

   `cd DIRNAME/src`

   `make MACHINE=pflotran_machine`

   `make install` *(compile with* `mpicc`*)*

   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:

   ```
   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>`:

   `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:

Option	Description
`-pflotranin <string>`	specify input file [Default: `pflotran.in`]
`-input_prefix <string>`	specify input (and output) file prefix [default: `pflotran`]
`-output_prefix <string>`	specify output file prefix [default: `pflotran`]
`-screen_output off`	turn off screen output
`-realization_id <integer>`	run specified realization ID
`-multisimulation`	run multiple input files in one run
`-stochastic`	Monte Carlo multiple realization run
PETSc	
`-help`	print list of command line options
`-log_summary`	print out run performance
`-snes_converged_reason`	print the reason for convergence/divergence after each solve
`-on_error_abort`	aborts run on hitting NaNs
`-v`	<int> verbose
`-options_left`	for debugging: lists any PETSc objects that have not been freed
`-snes_monitor`	print to screen function norm with each iteration
`-snes_linesearch_monitor`	print to screen line search information
`-snes_view`	print line search method
`-snes_ls`	options: [cubic, quadratic, basic, basicnonorms]
`-snes_type newtonls`	
`-snes_type newtontr`	
`-snes_tr_delta0`	<delta0>
`-snes_monitor_lg_residualnorm`	draws line graph of the residual norm convergence using X11
`-ksp_monitor_lg_residualnorm`	

3.7 Multiple Realization Simulation Mode

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

```
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:

```
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:

```
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 multiple 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.

Keyword	Description
BOUNDARY_CONDITION	
BRINE	
CHARCTERISTIC_CURVES	
CHECKPOINT	
CHEMISTRY	
COMPUTE_STATISTICS	
CO2_DATABASE	
CONSTRAINT	transport (optional)
DATASET	
DBASE_DATASET	
DEBUG	
EOS	
FLOW_CONDITION	
FLUID_PROPERTY	
GEOMECHANICS	
GEOMECHANICS_GRID	
GEOMECHANICS_OUTPUT	
GEOMECHANICS_MATERIAL_PROPERTY	
GEOMECHANICS_REGION	
GEOMECHANICS_CONDITION	
GEOMECHANICS_BOUNDARY_CONDITION	
GEOMECHANICS_STRATA	
GRID	(required)

INITIAL_CONDITION	
INTEGRAL_FLUX	
LINEAR_SOLVER	
MATERIAL_PROPERTY	
MODE	
MULTIPLE_CONTINUUM	
NEWTON_SOLVER	
NONUNIFORM_VELOCITY	
NUMERICAL_JACOBIAN_FLOW	
NUMERICAL_JACOBIAN_RXN	
NUMERICAL_JACOBIAN_MULTI_COUPLE	
OBSERVATION	
ORIG, ORIGIN	
OUTPUT	
OVERWRITE_RESTART_TRANSPORT	
PROC	(optional)
REGION	
RESTART	
SATURATION_FUNCTION	
SOURCE_SINK	
STRATIGRAPHY (STRATA)	
TIME	
TIMESTEPPER	
TRANSPORT_CONDITION	
UNIFORM_VELOCITY	
USE_TOUCH_OPTIONS	
VELOCITY_DATASET	(optional)
WALLCLOCK_STOP	

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

Quantity	Units
Pressure:	Pascal [Pa] (absolute)
Temperature:	Celcius [C]
Distance:	meter [m]
Volume:	meter^3 [m^3]
Time:	second [s]
Velocity:	meter/second [m/s]
Concentration:	molarity [M] or molality [m] (see MOLAL keyword)
Enthalpy:	kiloJoule/mole [kJ/mol]
Mass:	kilogram [kg]
Rate:	kilogram/second [kg/s] or cubic meter/second [m^3/s]
Surface Site Density:	mole/meter^3 [mol/m^3]

4.2 Example Input File

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


# == debugging ======================================================================
#DEBUG
# 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
/
#DXYZ
#1.
#1.
#1.
#/
/

# == 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
/
```

```
# == 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
/
/
```

```
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-4 M Calcite
H+ 8.d0 pH
CO2(aq) 1.d-2 G CO2(g)
Tracer 1.d-8 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:

Keyword	**Description**
BOUNDARY_CONDITION	Defines the beginning of a boundary condition entry and the name of the boundary condition.
FLOW_CONDITION	Defines the name of the flow condition to be linked to this boundary condition.
TRANSPORT_CONDITION	Defines the name of the transport condition to be linked to this boundary condition.
REGION	Defines the name of the region to which the conditions are linked.
END	Terminates the boundary condition entry.

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.d0 MOLAL
```

↪ Return to List of Keywords

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 10^9 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
  /
/
```

↪ Return to List of Keywords

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:

Keyword	**Description**
CHECKPOINT	toggles on checkpointing
`checkpoint_frequency`	frequency at which checkpoint files are printed `<integer>`

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/m^2/s], *If* `<float>` <0, `<float>` $= 10^{\texttt{<float>}}$

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>` [$\mathrm{mol/m^2/s}$] [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>` [$\mathrm{mol/m^2/s}$], *If* `<float>` < 0, `<float>` = $10^{\texttt{<float>}}$
  ACTIVATION_ENERGY `<float>` [kJ/mol/K]
  PREFACTOR_SPECIES `<Char>`
    ALPHA `<float>` [—]
    BETA `<float>` [—]
    ATTENUATION_COEF `<float>`
  END [/ .]
END [/ .]

END [/ .]
END [/ .]

SORPTION
  SURFACE_COMPLEXATION_RXN
    EQUILIBRIUM
    MULTIRATE_KINETIC
    KINETIC
    SITE_FRACTION `<float>`[—] (Continuation line ‘\’)

RATE, RATES `<float>` [1/s] (Continuation line '\')

MULTIRATE_SCALE_FACTOR `<float>` [—]

MINERAL `Mineral Name`

SITE `Name Site Density` [mol/m^3]

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: $k_b = K_{\text{eq}} k_f$, where K_{eq} is the corresponding equilibrium constant of the reaction.

END [/ .]

END [/ .]

COLLOID `Name`

SITE `Name Site Density` [mol/m^3]

COMPLEXES

`Surface_Complex Name`

END [/ .]

END [/ .]

END [/ .]

ION_EXCHANGE_RXN

MINERAL `Mineral Name`

CEC `<float>` [mol/m^3]

CATIONS

`Name Selectivity_Coefficient`

END [/ .]

END [/ .]

END [/ .]

ISOTHERM_REACTIONS

Species_Name

TYPE LINEAR, LANGMUIR, FREUNDLICH
DISTRIBUTION_COEF, KD `<float>` [kg water/m^3 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

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
    ...
/
```

Keyword	Description
Primary_Species	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.
Secondary_Species	List of aqueous species in equilibrium with primary species.
Gas_Species	List of gas species.
Sorption	Surface complexation, ion exchange and specified isotherm.

```
  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
    /
```

```
  /
  OUTPUT
    ALL
    Gibbsite
  /
  ...
END
```

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:

PREFACTOR (see Eqn.(10-112)):

```
MINERAL_KINETICS
  Quartz
    RATE_CONSTANT -17.99d0 mol/cm^2-sec
    ACTIVATION_ENERGY 87.7d0
  /
  Albite
    PREFACTOR
      RATE_CONSTANT -16.56d0 mol/cm^2-sec
      ACTIVATION_ENERGY 69.8d0
    /
    PREFACTOR
      RATE_CONSTANT -14.16d0 mol/cm^2-sec
      ACTIVATION_ENERGY 65.0d0
      PREFACTOR_SPECIES H+
        ALPHA           0.457d0
      ! BETA
      ! ATTENUATION_COEF
      /
    /
    PREFACTOR
```

```
      RATE_CONSTANT -19.6d0 mol/cm^2-sec
      ACTIVATION_ENERGY 71.0d0
      PREFACTOR_SPECIES H+
        ALPHA           -0.572d0
      ! BETA
      ! ATTENUATION_COEF
      /
    /
  /
END
```

↪ Return to List of Keywords

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 CO_2 database file providing a lookup table for CO_2 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 $T : 0 - 375$°C and $P : 0.01 - 1250$ bars. This keyword is required for MPHASE, FLASH2 and IMMIS modes.

Input:

CO2_DATABASE `{Path/Database_Name}` The default database file is `./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:

F, FREE	–Free ion/species concentration
T, TOTAL	–Total aqueous concentration
TOTAL_SORB	–Total aqueous and sorbed concentration
P, pH	–pH
L, LOG	–Log base 10 of free ion concentration
M, MINERAL, MNRL	–Mineral equilibrium constraint
G, GAS	–Gaseous species constraint [bars]
SC, CONSTRAINT_SUPERCRIT_CO2	–Supercritical CO_2 EOS
Z, CHG	–Charge balance

Example:

```
CONSTRAINT initial
  CONCENTRATIONS
    H+        7.3              P
```

```
    O2(aq)   1.78132e-4       T
    Al+++    1.e-9            M K-Feldspar
    Ca++     1.20644e-3       M Calcite
    Cu++     1.e-6            T
    Fe++     1.e-9            M Ferrihydrite
    Mg++     5.09772e-4       T
    UO2++    2.34845e-7       T
    K+       1.54789e-4       T
    Na+      2.03498e-3       T
    HCO3-    2.57305e-3       T
    Cl-      6.97741e-4       T
    F-       2.09491e-5       T
    HPO4--   1.e-6            T
    NO3-     4.69979e-3       T
    SO4--    6.37961e-4       T
    SiO2(aq) 5.36989e-4       T
    Tracer   2.34845e-7       F
  /
  MINERALS
    Quartz          0.35  1. cm^2/cm^3
    Calcite         0.    1. cm^2/cm^3
    Metatorbernite 0.    1. cm^2/cm^3
  /
END
```

```
:====== secondary constraint ===================
SECONDARY_CONSTRAINT sec
CONCENTRATIONS
Ca++          1.e-3    M Calcite
H+            7.      pH
SO4--         1.e-3    M Gypsum
HCO3-        -3.5      G CO2(g)
Cl-           1.e-3    Z
SiO2(aq)      1.e-4    M Quartz
Tracer        1.d-8    F
/
MINERALS
Quartz      0.019175   1.d2
Calcite     0.730825   1.d2
Gypsum      0.0        1.d2
/
/
```

↪ Return to List of Keywords

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
```

↪ Return to List of Keywords

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.

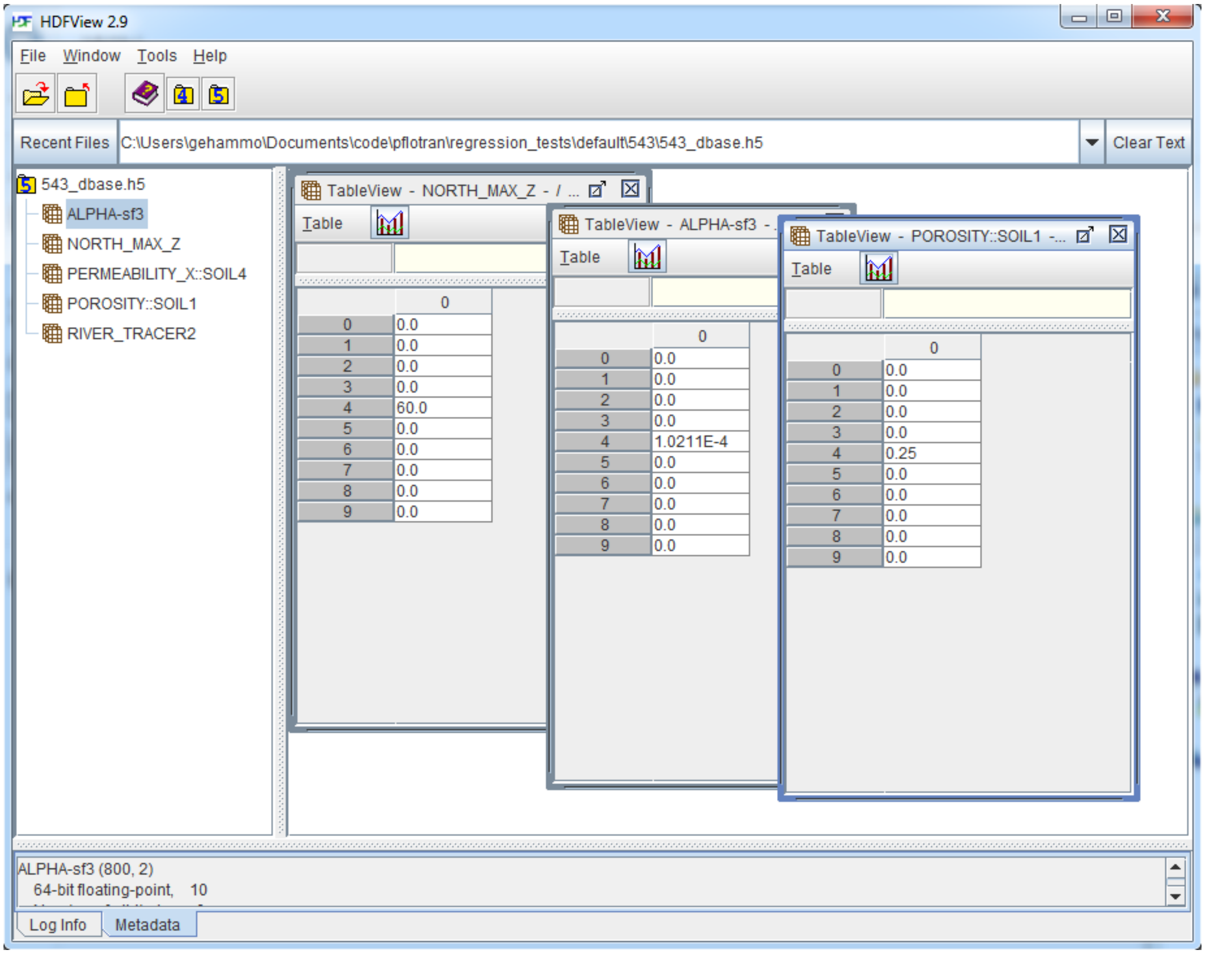
DBASE_FILENAME <string> The path/filename of the external database to be employed.

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:

```
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
```

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`.



C:\Users\gehammo\Documents\code\pflotran\regression_tests\default\543\543_dbase.h5

TableView - NORTH_MAX_Z

	0
0	0.0
1	0.0
2	0.0
3	0.0
4	60.0
5	0.0
6	0.0
7	0.0
8	0.0
9	0.0

TableView - ALPHA-sf3

	0
0	0.0
1	0.0
2	0.0
3	0.0
4	1.0211E-4
5	0.0
6	0.0
7	0.0
8	0.0
9	0.0

TableView - POROSITY::SOIL1

	0
0	0.0
1	0.0
2	0.0
3	0.0
4	0.25
5	0.0
6	0.0
7	0.0
8	0.0
9	0.0

```
ALPHA-sf3 (800, 2)
  64-bit floating-point,    10
```

↪ Return to List of Keywords

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>

ENTHALPY CONSTANT <float>

VISCOSITY <string> <optional parameters>

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 [m^3/s], and a volumetric injection/extraction rate [m^3/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_{dx} d_{dy} d_{dz}

FILE file_name

FLUX

TEMP, TEMPERATURE

CONC, CONCENTRATION

H, ENTHALPY

END [/ .]

TEMPERATURE, TEMP <float>

ENTHALPY, H <float>

PRESSURE, PRES, RESS <float>

RATE <float>

FLUX, VELOCITY, VEL <float>

CONC, CONCENTRATION <float>

SAT, SATURATION <float>

CONDUCTANCE <float>

END [/ .]

Explanation:

IPHASE	Phases present	CO_2 concentration
1	H_2O	$X_{CO_2}^{H_2O}$
2	SC CO_2	$X_{CO_2}^{SCCO_2}$
3	H_2O – SC CO_2	$\left(X_{CO_2}^{H_2O}\right)_{eq}$

Keyword	Description
FLOW/TRANSPORT_CONDITION	Initiates a condition entry and defines its name
CYCLIC	Instructs PFLOTRAN to cycle the transient data set should the simulation time exceed the last time in the data set
INTERPOLATION	Defines the method for interpolating between data set times
SYNC_TIMESTEP_WITH_UPDATE	Synchronizes time step with waypoints
DATUM	Location is space where prescribed scalar (e.g. pressure, temperature concentration, etc.) is defined
TYPE	Specifies condition type
PRESSURE	Specifies pressure condition type
TEMPERATURE	Specifies temperature condition type
CONCENTRATION	Specifies the type of concentration condition
SATURATION	Specifies saturation condition type
ENTHALPY	Specifies enthalpy condition type
END	Terminates type entry
GRADIENT	Gradient of the scalar field in 3D space
PRESSURE	Pressure gradient in x-, y-, and z-directions
TEMPERATURE	Temperature gradient in x-, y-, and z-directions
CONCENTRATION	Concentration gradient in x-, y-, and z-directions
ENTHALPY	Enthalpy gradient in x-, y-, and z-directions
END	Terminates gradient entry
PRESSURE	Absolute fluid pressure at the datum
FLUX	Darcy velocity of fluid defining flux across specified boundary
TEMPERATURE	Temperature in °C at datum
CONCENTRATION	Solute concentration at datum
SATURATION	Solute saturation at datum
ENTHALPY	Enthalpy at datum
CONSTRAINT_LIST	Specifies list of concentration constraints for solute transport
END	Terminates condition entry

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
```

↪ Return to List of Keywords

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>` [m^2/s] [Default: 0×10^{-9} m^2/s]
- **DIFFUSION_ACTIVATION_ENERGY** `<float>` `[kJ/mol]` [Default: 0 kJ/mol]

END [/ .]

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

$$D_m(T) \;=\; D_m^\circ \exp\left[\frac{A_D}{R}\left(\frac{1}{T_0} - \frac{1}{T}\right)\right], \tag{4-1}$$

where D_m° 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
/
```

↪ 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/s^2]

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:

```
MATERIAL_PROPERTY 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
```

↪ Return to List of Keywords

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:

Keyword	**Description**
GEOMECHANICS_CONDITION	Initiates a geomechanics condition entry and defines its name
TYPE	Specifies condition type
DISPLACEMENT_X	Specifies $x-$ displacement condition type
DISPLACEMENT_Y	Specifies $y-$ displacement condition type
DISPLACEMENT_Z	Specifies $z-$ displacement condition type
FORCE_X	Specifies in $x-$ force condition type
FORCE_Y	Specifies in $y-$ force condition type
FORCE_Z	Specifies in $z-$ force condition type
END	Terminates type entry
DISPLACEMENT_X	Displacement in the $x-$ direction
DISPLACEMENT_Y	Displacement in the $y-$ direction
DISPLACEMENT_Z	Displacement in the $z-$ direction
FORCE_X	Force in the $x-$ direction
FORCE_X	Force in the $y-$ direction
FORCE_X	Force in the $z-$ direction
END	Terminates condition entry

Example:

```
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:

Keyword	Description
GEOMECHANICS_BOUNDARY_CONDITION	Defines the beginning of a geomechanics boundary condition entry and the name of the geomechanics boundary condition.
GEOMECHANICS_CONDITION	Defines the name of the geomechanics condition to be linked to this geomechanics boundary condition.
REGION	Defines the name of the region to which the conditions are linked.
END	Terminates the geomechanics boundary condition entry.

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 Required

Description: this keyword defines the descritization 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/s^2]

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
```

```
! ...
! ...
! id_up_M id_dn_M x_M y_M z_M area_M
! -----------------------------------------------------------------
```

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
```

↪ Return to List of Keywords

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_AGG_NL** `<float>`
  - **BOOMERAMG_AGG_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_COARSEN_TYPE `<Value>`

BOOMERAMG_INTERPOLATION_TYPE, BOOMERAMG_INTERP_TYPE `<Value>`

BOOMERAMG_NODAL_COARSEN `<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
/
```

↪ Return to List of Keywords

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>` [m^2] 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
```

```
  TORTUOSITY 1.
  PERMEABILITY
    PERM_X 1.d-13
    PERM_Y 1.d-13
    PERM_Z 1.d-13
  /
  SECONDARY_CONTINUUM
    TYPE  SLAB
    LENGTH 50
    AREA 2500
    NUM_CELLS 10
    EPSILON 0.02
  /
END
```

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
```

↪ Return to List of Keywords

4.30 Keyword: MODE

Description: determines the flow mode: Richards (variably saturated porous media); MPH, MPHASE, FLASH2 ($CO_2 + H_2O$); TH (Thermal-Hydrologic); IMMIS, THS (Immisible).

MODE <option>

Option	Description
GENERAL	Two-phase air-water, nonisothermal, variably saturated groundwater flow
RICHARDS	Single-phase, isothermal, variably saturated groundwater flow using Richards equation
MPHASE (MPH)	Two-phase supercritical CO_2-brine-energy based on variable switching for phase changes
FLASH2	Two-phase supercritical CO_2-brine-energy based on the flash method for phase changes with a persistent set of unknowns
TH	Thermal-Hydrologic coupled groundwater flow
IMMIS (IMS, THS)	Immissible CO_2-water-energy
MIS	Missible H_2O-glycol fluid

Example:

```
MODE TH
```

↪ 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>` Checks the infinite norm of secondary continuum residual for convergence when using transport. (default set to 1.d-10)

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_DIVERGED_FUNCTION_DOMAIN     = -1, /* the new x location passed the function is
                                               not in the domain of F */
    SNES_DIVERGED_FUNCTION_COUNT      = -2,
    SNES_DIVERGED_LINEAR_SOLVE        = -3, /* the linear solve failed */
    SNES_DIVERGED_FNORM_NAN           = -4,
    SNES_DIVERGED_MAX_IT              = -5,
    SNES_DIVERGED_LINE_SEARCH         = -6, /* the line search failed */
    SNES_DIVERGED_INNER               = -7, /* inner solve failed */
    SNES_DIVERGED_LOCAL_MIN           = -8, /* || J^T b || is small, implies converged to
                                               local minimum of F() */
    SNES_CONVERGED_ITERATING          =  0} SNESConvergedReason;
```

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 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.
```

```
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
```

↪ Return to List of Keywords

4.34 Keyword: NUMERICAL_JACOBIAN_FLOW

NUMERICAL_JACOBIAN_FLOW Uses numerically evaluated Jacobian for flow.

↪ Return to List of Keywords

4.35 Keyword: NUMERICAL_JACOBIAN_RXN

NUMERICAL_JACOBIAN_RXN Uses numerically evaluated Jacobian for reactions.

↪ Return to List of Keywords

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
```

↪ Return to List of Keywords

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_2. Output times are controlled by `PERIODIC_OBSERVATION TIMESTEP and TIME`, and printout times.

END [/ .]

Explanation:

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

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 ith component in phase α 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 ith component

$$N_i = \sum_\alpha N_i^\alpha, \tag{4-3}$$

is given by

$$\frac{dN_i}{dt} = \sum_\alpha \frac{dN_i^\alpha}{dt} \tag{4-4}$$

$$= -\sum_\alpha \int_{\partial V} \boldsymbol{F}_i^\alpha \cdot \boldsymbol{dS} + Q_i, \tag{4-5}$$

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

$$\mathcal{F}_i^\alpha = -\int_{\partial V} \boldsymbol{F}_i^\alpha \cdot \boldsymbol{dS}, \tag{4-6}$$

defined as positive for flow into and negative for flow out of the domain V. Integrating the flux over time gives the cumulative flux $\overline{\mathcal{F}}_i^\alpha$

$$\overline{\mathcal{F}}_i^\alpha = \int_0^t \mathcal{F}_i^\alpha \, dt', \tag{4-7}$$

and time integrated source/sink $\overline{Q}_i$

$$\overline{Q}_i = \int_0^t Q_i \, dt'. \tag{4-8}$$

Global conservation of mass implies the equality

$$N_i(t) = N_i(0) + \sum_\alpha \overline{\mathcal{F}}_i^\alpha + \overline{Q}_i. \tag{4-9}$$

The header for the `MPHASE` mass balance file reads (for time in years, l=aqueous liquid phase, sc=supercritical CO_2):

$$\begin{aligned}
t,\ \Delta t,\ [y]\\
N_{\mathrm{H_2O}}^l,\ N_{\mathrm{CO_2}}^l,\ \widetilde{N}_{\mathrm{CO_2}}^l,\ N_{\mathrm{H_2O}}^{\mathrm{sc}},\ N_{\mathrm{CO_2}}^{\mathrm{sc}},\ \widetilde{N}_{\mathrm{CO_2}}^{\mathrm{sc}},\ \text{[kmol]}\\
\overline{\mathcal{F}}_{\mathrm{H_2O}}^l,\ \overline{\mathcal{F}}_{\mathrm{CO_2}}^l,\ \overline{\mathcal{F}}_{\mathrm{H_2O}}^{\mathrm{sc}},\ \overline{\mathcal{F}}_{\mathrm{CO_2}}^{\mathrm{sc}},\ \text{[kmol]}\\
\mathcal{F}_{\mathrm{H_2O}}^l,\ \mathcal{F}_{\mathrm{CO_2}}^l,\ \mathcal{F}_{\mathrm{H_2O}}^{\mathrm{sc}},\ \mathcal{F}_{\mathrm{CO_2}}^{\mathrm{sc}},\ \text{[kmol/y]}\\
\overline{Q}_{\mathrm{H_2O}},\ \overline{Q}_{\mathrm{CO_2}},\ \text{[kmol]}\\
Q_{\mathrm{H_2O}},\ Q_{\mathrm{CO_2}},\ \text{[kmol/y]}
\end{aligned} \tag{4-10}$$

The quantity $\widetilde{N}_i^\alpha$ denotes the trapped component in phase α defined by

$$\widetilde{N}_i^\alpha = \int_{V,\, s_\alpha < s_\alpha^0} \varphi s_\alpha \eta_\alpha x_i^\alpha \, dV, \tag{4-11}$$

where s_α^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
```

↪ Return to List of Keywords

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
```

↪ Return to List of Keywords

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
```

↪ Return to List of Keywords

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_GENUCHTEN**, **BROOKS_COREY**, **THOMEER_COREY**, **NMT_EXP**, **PRUESS_1**, **VAN_GENUCHTEN_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_GENUCHTEN_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_GENUCHTEN
  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:

```
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.

↪ Return to List of Keywords

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> Unit (s, m, h, d, mo, y)`

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
```

↪ Return to List of Keywords

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 pressure for a time step [1. mol/L].

MAX_SATURATION_CHANGE `<float>`

Maximum change in pressure 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
```

↪ Return to List of Keywords

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
```

↪ Return to List of Keywords

4.51 Keyword: USE_TOUCH_OPTIONS

Description:

Input:

USE_TOUCH_OPTIONS {use_touch_options = .true.}

Explanation:

Example:

↪ Return to List of Keywords

4.52 Keyword: VELOCITY_DATASET

Description: Set time-dependent velocity for transport mode.

Input:

VELOCITY_DATASET

UNITS cm/h

CYCLIC

INTERPOLATION **step** [default]

INTERPOLATION **linear**

VELOCITY

`Time velx vely velz`

END **[/ .]**

END **[/ .]**

Explanation:

Example:

```
VELOCITY_DATASET
  UNITS cm/h
  CYCLIC ! cycles the data set using last time as offset
  :INTERPOLATION STEP ! 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
```

↪ Return to List of Keywords

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`

↪ Return to List of Keywords

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 ==============================
MATERIAL_PROPERTY 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

MATERIAL_PROPERTY 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 TECPLOT 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
```

```
  /
END

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

:=========================== flow conditions ====================================
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 conditions ===============================
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
  /
```

```
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
```

```
:========================== 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 build 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 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 in 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. Know data types are listed below.

The data types and default tolerances are:

- time = 5 percent
- concentration = 1×10^{-12} absolute
- generic = 1×10^{-12} absolute

- discrete = 0 absolute
- rate = 1×10^{-12} absolute
- volume_fraction = 1×10^{-12} absolute
- pressure = 1×10^{-12} absolute
- saturation = 1×10^{-12} absolute
- residual = 1×10^{-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
```

```
      3978
    /
    CELLS_PER_PROCESS 4
  END
```

- Add the test to the configuration file
- Refine the tolerances so that they will be tight enough to identify problems, but loose enough that they do not create a lot of false positives and discourage users and developers from running the tests.
- Add the test to the appropriate test suite.
- Add the configuration file, input file and "gold" file to revision control.

Guidelines for setting tolerances go here, once we figure out what to recommend.

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:

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

For 1D problems or for plotting PFLOTRAN observation and mass balance output, the open-source 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.

```
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 coarse 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/m^3 this was obtained from the formula

$$\omega = \frac{N_s}{V} = \frac{N_s}{M_s}\frac{M_s}{V_s}\frac{V_s}{V}, \tag{8-12a}$$

$$= \rho_s(1-\varphi)\text{CEC}. \tag{8-12b}$$

Using a porosity of 0.61 and solid grain density of 3.0344 g/cm^3, yielded $\omega = 71.004$ mol/m^3. 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.

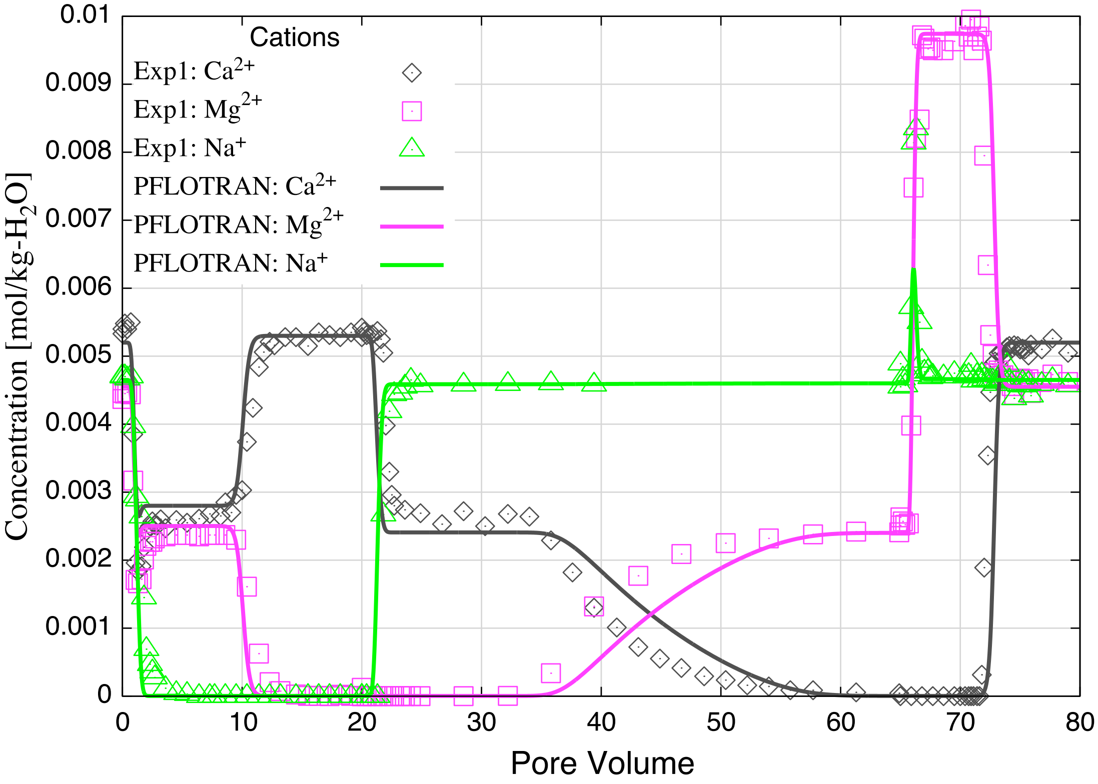


Figure 1: Breakthrough curves for Ca^{2+}, Mg^{2+} and Na^{+} compared with experimental results from Voegelin et al. (2000).

```
#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++
```

```
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
```

```
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
/
/
```

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

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

and solute transport of a tracer

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

In these equations φ 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, ρ represents the fluid density in general a function of pressure and temperature, C denotes the solute concentration, D denotes the diffusion/dispersion coefficient, τ represents tortuosity, Q and Q_C denote source/sink terms, and $\boldsymbol{q}$ denotes the Darcy velocity defined by

$$\boldsymbol{q} \;=\; -\frac{k_{\rm sat}k_r}{\mu}\boldsymbol{\nabla}(p-\rho g z), \tag{8-15}$$

with saturated permeability $k_{\rm sat}$, relative permeability k_r, fluid viscosity μ, 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

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

where $s_{\rm eff}$ is related to capillary pressure P_c by the equation

$$s_{\rm eff} \;=\; \left[1+(\alpha|P_c|)^n\right]^{-m}, \tag{8-17}$$

where $s_{\rm eff}$ is defined by

$$s_{\rm eff} \;=\; \frac{s-s_r}{1-s_r}, \tag{8-18}$$

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

$$m \;=\; 1-\frac{1}{n}, \qquad n \;=\; \frac{1}{1-m}. \tag{8-19}$$

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

$$\frac{d}{dz}\rho q_z = 0, \tag{8-21}$$

or assuming an incompressible fluid

$$q_z = q_z^0, \tag{8-22}$$

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

$$\frac{dp}{dz} = -\frac{\mu q_z^0}{k_{\rm sat} k_r} - \rho g, \tag{8-23}$$

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

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

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

$$P_c(z_{\rm wt}) = 0, \tag{8-25}$$

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

$$z_{\rm wt} = z_0 + \frac{p_0 - p_g}{\rho g}, \tag{8-26}$$

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 $z_{\rm wt} = z_0$, 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} m^2 s^{-1} and tortuosity was set to one.

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

Formation	Abbrev.	Thickness [m]
Backfill	BF	16.0
Hanford Fine Sand	HF	23.0
Plio-Pleistocene	PP	6.0
Upper Ringold Gravel	URG	3.0
Middle Ringold Gravel	MRG	20.0

Table 7: Parameters for material and thermal properties for intrinsic rock density ρ_s, heat capacity c, thermal conductivity κ, porosity φ, residual water saturation s_r, van Genuchten parameters α and λ, and vertical water saturated permeability k_{sat}. Data taken from Khaleel and Freeman (1995), Khaleel et al. (2001), and Pruess et al. (2002).

Formation	ρ_s	c	κ_{dry}	κ_{wet}	φ	s_r	α	m	k_{sat}
	g cm^{-3}	J kg^{-1} K^{-1}	W m^{-1}		—	—	Pa^{-1}	—	m^2
BF	2.8	800	0.5	2	0.2585	0.0774	1.008e-3	0.6585	1.240e-12
HF	2.8	800	0.5	2	0.3586	0.0837	9.408e-5	0.4694	3.370e-13
PP	2.8	800	0.5	2	0.4223	0.2595	6.851e-5	0.4559	3.735e-14
URG	2.8	800	0.5	2	0.2625	0.2130	2.966e-5	0.3859	1.439e-13
MRG	2.8	800	0.5	2	0.1643	0.0609	6.340e-5	0.3922	2.004e-13

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 heigh 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 1.87×10^{-3} kg/s is used.

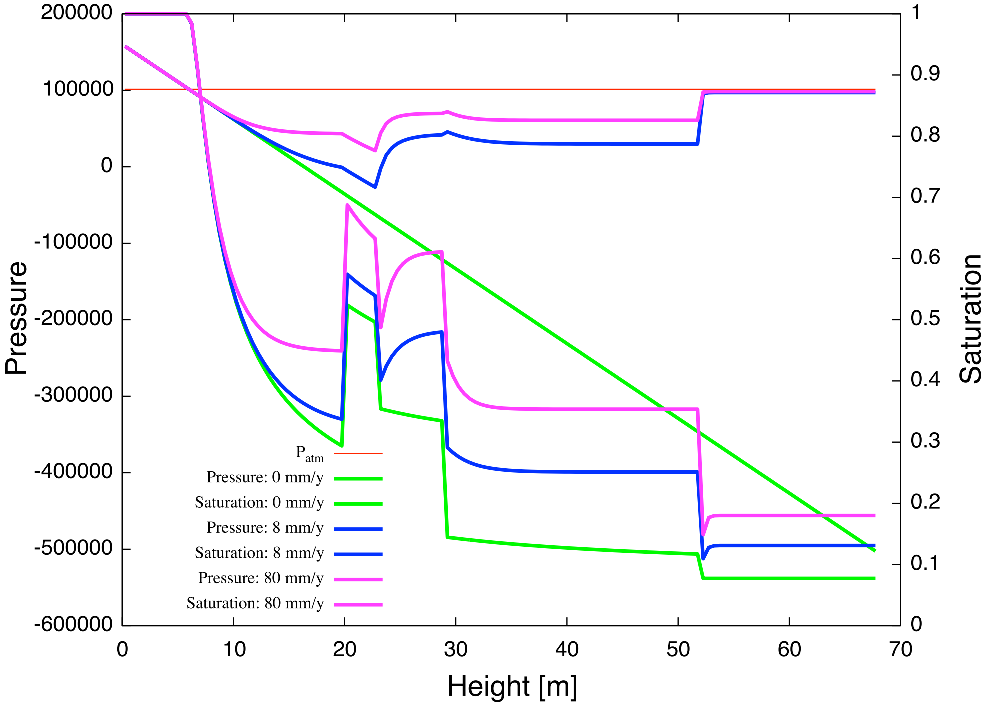


Figure 2: Steady-state saturation and pressure profiles for infiltration rates of 0, 8 and 80 mm/y. The water table is located at 6 m from the bottom of the computational domain.

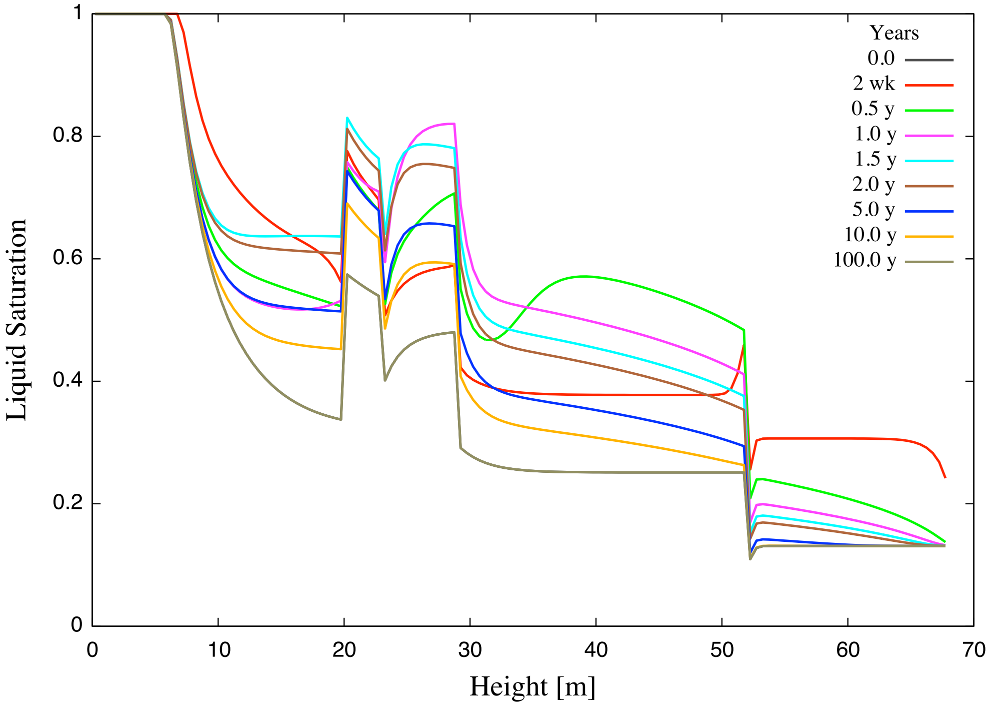


Figure 3: Simulation of a tank leak with a duration of two weeks showing the saturation profile for different times indicated in the figure.

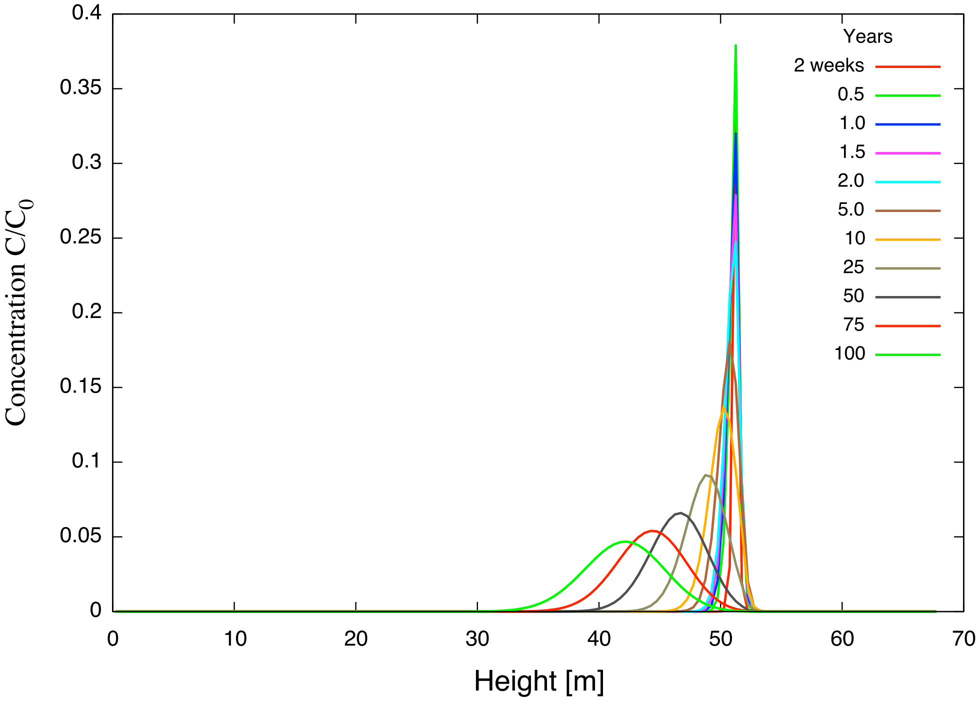


Figure 4: The solute concentration profile corresponding to Figure 7 for different times indicated in the figure.

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`:

```
#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
```

```
    PERM_Z 3.37028e-13
  /
END

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

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

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

#=========================== saturation functions ==============================
SATURATION_FUNCTION BF
  SATURATION_FUNCTION_TYPE VAN_GENUCHTEN
  PERMEABILITY_FUNCTION_TYPE MUALEM
  RESIDUAL_SATURATION 0.0774d0
  LAMBDA 0.6585d0
  ALPHA 1.008d-3
  MAX_CAPILLARY_PRESSURE 1.d6
  BETAC 0.d0
  POWER 1.d0
END

SATURATION_FUNCTION HF
  SATURATION_FUNCTION_TYPE VAN_GENUCHTEN
  PERMEABILITY_FUNCTION_TYPE MUALEM
  RESIDUAL_SATURATION 0.08366d0
```

```
  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
```

```
    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

Balay S., V. Eijkhout V, W.D. Gropp, L.C. McInnes and B.F. Smith (1997) Modern Software Tools in Scientific Computing, Eds. Arge E, Bruaset AM and Langtangen HP (Birkhaüser Press), pp. 163–202.

Coats, K.H. and A.B. Ramesh (1982) Effects of Grid Type and Difference Scheme on Pattern Steamflood Simulation Results, paper SPE-11079, presented at the 57th Annual Fall Technical Conference and Exhibition of the Society of Petroleum Engineers, New Orleans, LA, September 1982.

Fenghour, A., W.A. Wakeham, and V. Vesovic (1998) The viscosity of carbon dioxide, J. Phys. Chem. Ref. Data, 27(1), 31–44.

Goode, D.J. (1996) Direct simulation of groundwater age, Water Resources Research, 32, 289–296.

Hammond, G.E., P.C. Lichtner, C. Lu, and R.T. Mills (2011) PFLOTRAN: Reactive Flow & Transport Code for Use on Laptops to Leadership-Class Supercomputers, Editors: Zhang, F., G. T. Yeh, and J. C. Parker, *Ground Water Reactive Transport Models*, Bentham Science Publishers. ISBN 978-1-60805-029-1.

Khaleel, R., E.J. Freeman (1995) Variability and scaling of hydraulic properties for 200 area soils, Hanford Site. Report WHC-EP-0883. Westinghouse Hanford Company, Richland, WA.

Khaleel, R., T.E. Jones, A.J. Knepp, F.M. Mann, D.A. Myers, P.M. Rogers, R.J. Serne, and M.I. Wood (2000) Modeling data package for S-SX Field Investigation Report (FIR). Report RPP-6296, Rev. 0. CH2M Hill Hanford Group, Richland, WA.

Lichtner, P.C., Yabusaki, S.B., Pruess K., and Steefel, C.I. (2004) Role of Competitive Cation Exchange on Chromatographic Displacement of Cesium in the Vadose Zone Beneath the Hanford S/SX Tank Farm, *VJZ*, **3**, 203–219.

Lichtner P.C. (2000) Critique of Dual Continuum Formulations of Multicomponent Reactive Transport in Fractured Porous Media, Ed. Boris Faybishenko, *Dynamics of Fluids in Fractured Rock*, Geophysical Monograph **122**, 281–298.

Painter, S.L. (2011) Three-phase numerical model of water migration in partially frozen geological media: model formulation, validation, and applications, Computational Geosciences **15**, 69-85.

Peaceman, D.W. (1977) Interpretation of Well-Block Pressures in Numerical Reservoir Simulation with Nonsquare Grid Blocks and Anisotropic Permeability, paper SPE-10528, presented at the Sixth SPE Symposium on Reservoir Simulation of the Society of Petroleum Engineers, New Orleans, LA, January 1982.

Pruess, K., S. Yabusaki, C. Steefel, and P. Lichtner (2002) Fluid flow, heat transfer, and solute transport at nuclear waste storage tanks in the Hanford vadose zone. Available at www.vadosezonejournal.org. Vadose Zone J. 1:68–88.

Pruess, K., and Narasimhan (1985) A practical method for modeling fluid and heat flow in fractured porous media, SPE 10509, 14–26.

Somerton, W.H., A.H. El-Shaarani, and S.M. Mobarak (1974) High temperature behavior of rocks associated with geothermal-type reservoirs. Paper SPE-4897. Proceedings of the 44th Annual California Regional Meeting of the Society of Petroleum Engineers. Richardson, TX: Society of Petroleum Engineers.

Andreas Voegelin, Vijay M. Vulava, Florian Kuhnen, Ruben Kretzschmar (2000) Multicomponent transport of major cations predicted from binary adsorption experiments, Journal of Contaminant Hydrology, 46, 319–338.

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

$$\frac{\partial}{\partial t}\left(\varphi s \rho\right) + \boldsymbol{\nabla}\cdot\left(\rho \boldsymbol{q}\right) = Q_w, \tag{10-1}$$

with Darcy flux $\boldsymbol{q}$ defined as

$$\boldsymbol{q} = -\frac{k k_r(s)}{\mu}\boldsymbol{\nabla}\left(P - W_w \rho g z\right). \tag{10-2}$$

Here, φ denotes porosity [-], s saturation [m^{3}m^{-3}], ρ water density [kmol m^{-3}], $\boldsymbol{q}$ Darcy flux [m s^{-1}], k intrinsic permeability [m^2], k_r relative permeability [-], μ 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

$$Q_w \;=\; \frac{q_M}{W_w}\delta(\boldsymbol{r} - \boldsymbol{r}_{ss}), \tag{10-3}$$

where q_M denotes a mass rate in kg/m^3/s, and $\boldsymbol{r}_{ss}$ 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 impirical constants determined by fitting to experimental data.

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

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

with $\lambda = mn$ and inverse relation

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

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

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

for the Mualem formulation and by

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

for the Burdine formulation.

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

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

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

$$\begin{aligned} k_r &= \left(s_e\right)^{5/2+2/\lambda}, && (10\text{-}12)\\ &= \left(p_c/p_c^0\right)^{-(5\lambda/2+2)}. && (10\text{-}13) \end{aligned}$$

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

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

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

$$\begin{aligned} k_r &= \left(s_e\right)^{2+3/\lambda}, && (10\text{-}15)\\ &= \left(\frac{p_c}{p_c^0}\right)^{-(2+3\lambda)}. && (10\text{-}16) \end{aligned}$$

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

$$\begin{aligned} a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 &= f_1, & \text{(10-17a)} \\ a_0 + a_1 x_2 + a_2 x_2^2 + a_3 x_2^3 &= f_1, & \text{(10-17b)} \\ a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 &= f_1', & \text{(10-17c)} \\ a_1 x_2 + a_2 x_2^2 + a_3 x_2^3 &= f_2', & \text{(10-17d)} \end{aligned}$$

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

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

The conditions imposed on the smoothing equations for capillary pressure $f = s_e(p_c)$ are $x_1 = 2p_c^0$, $x_2 = p_c^0/2$, $f_1 = (s_e)_1$, $f_2 = 1$, $f_1' = (s_e')_1$, $f_2' = 0$. 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:

$$\frac{\partial}{\partial t}\Big(\varphi \sum_\alpha s_\alpha \eta_\alpha x_i^\alpha\Big) + \boldsymbol{\nabla}\cdot\sum_\alpha \boldsymbol{F}_i^\alpha = Q_i, \qquad \text{(10-19a)}$$

for the ith component where the flux $\boldsymbol{F}_i^\alpha$ is given by

$$\boldsymbol{F}_i^\alpha = \boldsymbol{q}_\alpha \eta_\alpha x_i^\alpha - \varphi s_\alpha D_\alpha \eta_\alpha \boldsymbol{\nabla} x_i^\alpha, \qquad \text{(10-19b)}$$

and

$$\frac{\partial}{\partial t}\Big(\varphi \sum_\alpha s_\alpha \eta_\alpha U_\alpha + (1-\varphi)\rho_r c_r T\Big) + \boldsymbol{\nabla}\cdot\sum_\alpha \Big[\boldsymbol{q}_\alpha \eta_\alpha H_\alpha - \kappa \boldsymbol{\nabla} T\Big] = Q_e, \qquad \text{(10-19c)}$$

for energy. In these equations α designates a fluid phase ($\alpha = l$, sc) at temperature T and pressure P_α 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 = H_2O, CO_2); φ

denotes the porosity of the porous medium; s_α denotes the phase saturation state; x_i^α denotes the mole fraction of species i satisfying

$$\sum_i x_i^\alpha = 1, \tag{10-20}$$

the quantities η_α, H_α, U_α refer to the molar density, enthalpy, and internal energy of each fluid phase, respectively; and $\boldsymbol{q}_\alpha$ denotes the Darcy flow rate for phase α defined by

$$\boldsymbol{q}_\alpha = -\frac{kk_\alpha}{\mu_\alpha}\boldsymbol{\nabla}\big(P_\alpha - \rho_\alpha g \boldsymbol{z}\big), \tag{10-21}$$

where k refers to the intrinsic permeability, k_α denotes the relative permeability, μ_α denotes the fluid viscosity, W_α denotes the formula weight, g denotes the acceleration of gravity, and z designates the vertical of the position vector. The mass density ρ_α is related to the molar density by the expression

$$\rho_\alpha = W_\alpha \eta_\alpha, \tag{10-22}$$

where the formula weight W_α 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 ρ_r, c_r, and κ 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

$$\begin{aligned} Q_i &= \sum_n \sum_\alpha q_\alpha^V \eta_\alpha x_i^\alpha \delta(\boldsymbol{r}-\boldsymbol{r}_n), && \text{(10-24a)}\\ &= \sum_n \sum_\alpha \frac{\eta_\alpha}{\rho_\alpha} q_\alpha^M x_i^\alpha \delta(\boldsymbol{r}-\boldsymbol{r}_n), && \text{(10-24b)}\\ &= \sum_n \sum_\alpha W_\alpha^{-1} q_\alpha^M x_i^\alpha \delta(\boldsymbol{r}-\boldsymbol{r}_n), && \text{(10-24c)} \end{aligned}$$

where q_α^V, q_α^M refer to volume and mass rates with units m^3/s, kg/s, respectively, related by the density

$$q_\alpha^M = \rho_\alpha q_\alpha^V. \tag{10-25}$$

The position vector $\boldsymbol{r}_n$ refers to the location of the nth 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

$$q_\alpha^M = \Gamma \rho_\alpha \frac{k_\alpha}{\mu_\alpha}\big(p_\alpha - p_\alpha^{\rm bw}\big), \tag{10-26}$$

Table 8: Choice of primary variables.

State	X_1	X_2	X_3
liquid	p_l	T	$x^l_{\mathrm{CO_2}}$
gas	p_g	T	$x^g_{\mathrm{CO_2}}$
two-phase	p_g	T	s_g

with bottom well pressure p_α^{bw}, and where Γ denotes the well factor (production index) given by

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

In this expression k denotes the permeability of the porous medium, Δz refers to the layer thickness, r_e denotes the grid block radius, r_w denotes the well radius, and σ 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

$$r_e = \sqrt{A/\pi}. \tag{10-28}$$

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 $N_{\mathrm{int}} = N_C - N_P + 2$, where N_P denotes the number of phases. The extensive degrees of freedom equals $N_{\mathrm{ext}} = N_P - 1$. This gives a total number of degrees of freedom $N_{\mathrm{dof}} = N_{\mathrm{int}} + N_{\mathrm{ext}} = N_C + 1$, 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: $(p_g,\, T,\, x^g_{\mathrm{CO_2}}) \rightarrow$ Two-Phase: $(p_g,\, T,\, s_g)$

- gas → 2-ph: $x^g_{\mathrm{CO_2}} \leq 1 - \dfrac{P_{\mathrm{sat}}(T)}{p_g}$, or equivalently: $x^g_{\mathrm{H_2O}} \geq \dfrac{P_{\mathrm{sat}}(T)}{p_g}$

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

- liq → 2-ph: $x^l_{\mathrm{CO_2}} \geq x^{eq}_{\mathrm{CO_2}}$

The equilibrium mole fraction $x^{eq}_{\mathrm{CO_2}}$ is given by

$$x^{eq}_{\mathrm{CO_2}} = \frac{m_{\mathrm{CO_2}}}{W^{-1}_{\mathrm{H_2O}} + m_{\mathrm{CO_2}} + \sum_{l \neq \mathrm{H_2O},\,\mathrm{CO_2}} m_l}, \tag{10-29}$$

where the molality at equilibrium is given by

$$m_{\mathrm{CO_2}}^{eq} = \left(1 - \frac{P_{\mathrm{sat}}(T)}{p}\right)\frac{\phi_c p}{K_{\mathrm{CO_2}}\gamma_{\mathrm{CO_2}}}, \tag{10-30}$$

where it is assumed that

$$y_{\mathrm{CO_2}} = 1 - \frac{P_{\mathrm{sat}}(T)}{p}. \tag{10-31}$$

10.3.2.3 Two-Phase: $(p_g,\, T,\, s_g) \rightarrow$ Liquid $(p_l,\, T,\, x_{\mathrm{CO_2}}^l)$ or Gas $(p_g,\, T,\, x_{\mathrm{CO_2}}^g)$

Equilibrium in a two-phase H_2O–CO_2 system is defined as the equality of chemical potentials between the two phases as expressed by the relation

$$f_{\mathrm{CO_2}} = y_{\mathrm{CO_2}}\phi_{\mathrm{CO_2}}p_g = K_{\mathrm{CO_2}}\big(\gamma_{\mathrm{CO_2}}m_{\mathrm{CO_2}}\big), \tag{10-32}$$

where

$$y_{\mathrm{CO_2}} = x_{\mathrm{CO_2}}^g, \quad x_{\mathrm{CO_2}} = x_{\mathrm{CO_2}}^l, \tag{10-33}$$

$$x_{\mathrm{H_2O}}^g = \frac{P_{\mathrm{sat}}(T)}{p_g}, \tag{10-34}$$

$$y_{\mathrm{CO_2}} = 1 - x_{\mathrm{H_2O}}^g = 1 - \frac{P_{\mathrm{sat}}(T)}{p_g}, \tag{10-35}$$

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

$$y_{\mathrm{CO_2}} = \widetilde{K}_{\mathrm{CO_2}}x_{\mathrm{CO_2}}, \tag{10-36}$$

where

$$\widetilde{K}_{\mathrm{CO_2}} = \frac{\gamma_{\mathrm{CO_2}}K_{\mathrm{CO_2}}}{\phi_{\mathrm{CO_2}}p_g}\frac{m_{\mathrm{CO_2}}}{x_{\mathrm{CO_2}}}. \tag{10-37}$$

- 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:

$$m_i = \frac{n_i}{M_{\mathrm{H_2O}}} = \frac{n_i}{W_{\mathrm{H_2O}}n_{\mathrm{H_2O}}} = \frac{x_i}{W_{\mathrm{H_2O}}x_{\mathrm{H_2O}}} = \frac{x_i}{W_{\mathrm{H_2O}}\big(1 - \sum_{l\neq \mathrm{H_2O}} x_l\big)} \tag{10-38}$$

Mole fraction–molality:

$$x_i = \frac{n_i}{N} = \frac{n_i}{M_{\mathrm{H_2O}}}\frac{M_{\mathrm{H_2O}}}{N} = \frac{m_i}{\sum m_l} = \frac{W_{\mathrm{H_2O}}m_i}{1 + W_{\mathrm{H_2O}}\sum_{l\neq \mathrm{H_2O}} m_l} \tag{10-39}$$

Mole fraction–mass fraction:

$$x_i = \frac{n_i}{N} = \frac{W_i^{-1}W_i n_i}{\sum W_l^{-1}W_l n_l} = \frac{W_i^{-1}w_i}{\sum W_l^{-1}w_l} \tag{10-40}$$

Mass fraction–mole fraction:

$$w_i = \frac{M_i}{M} = \frac{W_i n_i}{\sum W_l n_l} = \frac{W_i x_i}{\sum W_l x_l} \tag{10-41}$$

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

$$\frac{\partial}{\partial t}\big(\varphi\rho_\alpha s_\alpha\big) + \boldsymbol{\nabla}\cdot\big(\rho_\alpha \boldsymbol{q}_\alpha\big) \;=\; Q_\alpha, \tag{10-42}$$

where the subscript α denotes an immiscible phase. In this equation φ is porosity, s_α, ρ_α refer to the αth phase saturation and density, respectively, $\boldsymbol{q}_\alpha$ is the Darcy velocity of the αth phase given by

$$\boldsymbol{q}_\alpha \;=\; -\frac{kk_\alpha}{\mu_\alpha}\big(\boldsymbol{\nabla}p - \rho_\alpha g\hat{\boldsymbol{z}}\big), \tag{10-43}$$

with permeability k, relative permeability k_α, fluid viscosity μ_α, and Q_α is the source/sink term. The selection of primary variables are pressure p and $n-1$ independent phase saturation variables $s_\alpha, \alpha = 1, ..., n-1$ with

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

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

$$\frac{\partial}{\partial t}\Big(\varphi\sum_\alpha s_\alpha\rho_\alpha U_\alpha + (1-\varphi)\rho_r C_r T\Big) + \boldsymbol{\nabla}\cdot\Big(\sum_\alpha \rho_\alpha \boldsymbol{q}_\alpha H_\alpha - \kappa\boldsymbol{\nabla}T\Big) \;=\; Q_e, \tag{10-45}$$

where U_α, H_α denote the internal energy and enthalpy of the αth fluid phase, κ denotes the thermal conductivity of the bulk porous medium, ρ_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 proplyene glycol (PPG). In terms of molar density for the mixture η and mole fractions x_i, i=1 (water), i=2 (PPG), the mass conservation equations have the form

$$\frac{\partial}{\partial t}\varphi\eta x_i + \boldsymbol{\nabla}\cdot\left[\boldsymbol{q}\eta x_i - \varphi D\eta\boldsymbol{\nabla}x_i\right] \;=\; Q_i, \tag{10-46}$$

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 η is related to the mass density by

$$\eta \;=\; W^{-1}\rho, \tag{10-47}$$

and

$$W_i \eta x_i \;=\; \rho y_i. \tag{10-48}$$

It follows that

$$W_i \eta \boldsymbol{\nabla} x_i \;=\; \rho \boldsymbol{\nabla} y_i + \rho y_i \boldsymbol{\nabla} \ln W. \tag{10-49}$$

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 are pressrue with the form

$$\begin{aligned}\rho(y_1,\, p) \;&=\; \rho(y_1,\, p_0) + \frac{\partial \rho}{\partial p}\bigg|_{p=p_0} (p - p_0), \qquad &(10\text{-}50)\\ &=\; \rho(y_1,\, p_0)\big(1 + \beta(p - p_0)\big), &(10\text{-}51)\end{aligned}$$

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

$$\begin{aligned}\beta \;&=\; \frac{1}{\rho}\frac{\partial \rho}{\partial p}\bigg|_{p=p_0}, \qquad &(10\text{-}52)\\ &=\; 4.49758 \times 10^{-10} y_1 + 5 \times 10^{-10}(1 - y_1), &(10\text{-}53)\end{aligned}$$

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

$$\rho(y_1,\, p_0) \;=\; \Big(((0.0806 y_1 - 0.203) y_1 + 0.0873) y_1 + 1.0341\Big) 10^3, \tag{10-54}$$

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

$$\mu(y_1) \;=\; 10^{\big(1.6743(1-y_1) - 0.0758\big)} 10^{-3}, \tag{10-55}$$

and

$$D(y_1) \;=\; \Big((((-4.021 y_1 + 9.1181) y_1 - 5.9703) y_1 + 0.4043) y_1 + 0.5687\Big) 10^{-9}, \tag{10-56}$$

The mass fraction is related to mole fraction according to

$$y_1 = \frac{x_1 W_{\mathrm{H_2O}}}{W}, \tag{10-57}$$

where the mean formula weight W is given by

$$W = x_1 W_{\mathrm{H_2O}} + x_2 W_{\mathrm{PPG}}, \tag{10-58}$$

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

Global mass conservation satisfies the relation

$$\frac{d}{dt} M_i \;=\; -\int \boldsymbol{F}_i \cdot \boldsymbol{dS} + \int Q_i dV, \tag{10-59}$$

with

$$M_i \;=\; \int \varphi \eta x_i dV. \tag{10-60}$$

In terms of mass fractions and mass density

$$M_i^m \;=\; W_i M_i \;=\; \int \varphi \rho y_i dV. \tag{10-61}$$

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

$$\frac{\partial}{\partial t}\varphi\Big(s_l\rho_l x_i^l + s_g\rho_g x_i^g\Big) + \boldsymbol{\nabla}\cdot\Big(\boldsymbol{q}_l\rho_l x_i^l + \boldsymbol{q}_g\rho_g x_i^g - \varphi s_l D_l\rho_l\boldsymbol{\nabla} x_i^l - \varphi s_g D_g\rho_g\boldsymbol{\nabla} x_i^g\Big) \;=\; Q_i, \tag{10-62}$$

for liquid and gas saturation $s_{l,\,g}$, density $\rho_{l,\,g}$, diffusivity $D_{l,\,g}$, Darcy velocity $\boldsymbol{q}_{l,\,g}$ and mole fraction $x_i^{l,\,g}$. The energy conservation equation can be written in the form

$$\sum_{\alpha=l,\,g}\left\{\frac{\partial}{\partial t}\big(\varphi s_\alpha\rho_\alpha U_\alpha\big) + \boldsymbol{\nabla}\cdot\big(\boldsymbol{q}_\alpha\rho_\alpha H_\alpha\big)\right\} + \frac{\partial}{\partial t}\Big((1-\varphi)\rho_r C_p T\Big) - \boldsymbol{\nabla}\cdot(\kappa\boldsymbol{\nabla}T)\Big) \;=\; Q, \tag{10-63}$$

as the sum of contributions from liquid and gas fluid phases and rock, with internal energy U_α and enthalpy H_α of fluid phase α, rock heat capacity C_p and thermal conductivity κ. Note that

$$U_\alpha \;=\; H_\alpha - \frac{P_\alpha}{\rho_\alpha}. \tag{10-64}$$

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

$$\kappa \;=\; \kappa_{\rm dry} + \sqrt{s_l}(\kappa_{\rm sat} - \kappa_{\rm dry}), \tag{10-65}$$

where $\kappa_{\rm dry}$ and $\kappa_{\rm sat}$ 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

$$\frac{\partial}{\partial t}\left(\varphi s\rho\right) + \boldsymbol{\nabla}\cdot\left(\rho\boldsymbol{q}\right) = Q_w, \tag{10-66}$$

and

$$\frac{\partial}{\partial t}\big(\varphi s\rho U + (1-\varphi)\rho_p c_p T\big) + \boldsymbol{\nabla}\cdot\big(\rho\boldsymbol{q}H - \kappa\boldsymbol{\nabla}T\big) = Q_e, \tag{10-67}$$

The Darcy flow velocity $\boldsymbol{q}$ is given by

$$\boldsymbol{q} = -\frac{kk_r}{\mu}\boldsymbol{\nabla}\left(P - W\rho g z\right). \tag{10-68}$$

Here, φ denotes porosity, s saturation, ρ mixture density of the brine, $\boldsymbol{q}$ Darcy flux, k intrinsic permeability, k_r relative permeability, μ 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 ρ_p, c_p, and κ denotes 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

$$U \;=\; H - \frac{P}{\rho}. \tag{10-69}$$

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

$$\kappa \;=\; \kappa_{\rm dry} + \sqrt{s_l}(\kappa_{\rm sat} - \kappa_{\rm dry}), \tag{10-70}$$

where $\kappa_{\rm dry}$ and $\kappa_{\rm sat}$ 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

$$\begin{aligned}\frac{\partial}{\partial t}\left[\phi\left(s_l\eta_l X_w^l + s_g\eta_g X_w^g + s_i\eta_i X_w^i\right)\right] + \boldsymbol{\nabla}\cdot\left[X_w^l \boldsymbol{v}_l\eta_l + X_w^g\eta_g\boldsymbol{v}_g\right]&\\ - \boldsymbol{\nabla}\cdot\left[\phi s_g\tau_g\eta_g D_g\boldsymbol{\nabla}X_w^g\right] = Q_w,& \qquad \text{(10-71a)}\\ \frac{\partial}{\partial t}\left[\phi\left(s_l\eta_l U_l + s_g\eta_g U_g + s_i\eta_i U_i\right) + (1-\phi)\rho_r c_r T\right] + \boldsymbol{\nabla}\cdot\left[\boldsymbol{v}_l\eta_l H_l + \boldsymbol{v}_g\eta_g H_g\right]&\\ - \boldsymbol{\nabla}\cdot\left[\kappa\boldsymbol{\nabla}T\right] = Q_e,& \qquad \text{(10-71b)}\end{aligned}$$

where the subscripts l, i, g denote the liquid, ice and gas phases, respectively; ϕ is the porosity; $s_\alpha(\alpha = i, l, g)$ is the saturation of the α-th phase; $\eta_\alpha(\alpha = i, l, g)$ is the molar density of the α-th phase; ρ_g, ρ_l are the mass densities of the gas and liquid phases; Q_w is the mass source of H_2O; $X_w^\alpha(\alpha = i, l, g)$ is the mole fraction of H_2O in the α-th phase; τ_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; ρ_r is the density of the rock; $U_\alpha(\alpha = i, l, g)$ is the molar internal energy of the α-th phase; $H_\alpha(\alpha = l, g)$ is the molar enthalpy of the α-the phase; Q_e is the heat source; $\boldsymbol{\nabla}\,()$ is the gradient operator; $\boldsymbol{\nabla}\cdot()$ is the divergence operator.

The Darcy velocity for the gas and liquid phases are given as follows:

$$\boldsymbol{v}_g = -\frac{k_{rg}k}{\mu_g}\boldsymbol{\nabla}\left[p_g - \rho_g g z\right], \tag{10-72a}$$

$$\boldsymbol{v}_l = -\frac{k_{rl}k}{\mu_l}\boldsymbol{\nabla}\left[p_l - \rho_l g z\right], \tag{10-72b}$$

where k is the absolute permeability; $k_{r\alpha}(\alpha = l, g)$ is the relative permeability of the α-th phase; $\mu_\alpha(\alpha = l, g)$ is the viscosity of the α-th phase; $p_\alpha(\alpha = l, g)$ is the partial pressure of the α-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

$$s_l + s_g + s_i = 1. \tag{10-73}$$

Furthermore, neglecting the amount of air in liquid and ice phases, it follows that

$$X_a^l = 0, X_a^i = 0 \Rightarrow X_w^l = 1, X_w^i = 1, \tag{10-74}$$

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

$$\frac{\partial}{\partial t}\left[\phi\left(s_g\eta_g X_w^g + s_l\eta_l + s_i\eta_i\right)\right] + \boldsymbol{\nabla}\cdot\left[\boldsymbol{v}_l\eta_l\right] - \boldsymbol{\nabla}\cdot\left[\phi s_g\tau_g\eta_g D_g\boldsymbol{\nabla}X_w^g\right] = Q_w, \tag{10-75a}$$

$$\frac{\partial}{\partial t}\left[\phi\left(s_l\eta_l U_l + s_g\eta_g U_g + s_i\eta_i U_i\right) + (1-\phi)\rho_r c_r T\right] + \boldsymbol{\nabla}\cdot\left[\boldsymbol{v}_l\eta_l H_l\right] - \boldsymbol{\nabla}\cdot\left[\kappa\boldsymbol{\nabla}T\right] = Q_e, \tag{10-75b}$$

$$\boldsymbol{v}_l = -\frac{k_{rl}k}{\mu_l}\boldsymbol{\nabla}\left[p_l - \rho_l g z\right]. \tag{10-75c}$$

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,

$$X_w^g = \frac{p_v}{p_g}, \tag{10-76}$$

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

$$p_v = P_{\text{sat}}(T)\text{exp}\left[\frac{P_{cgl}}{\eta_l R(T + 273.15)}\right], \tag{10-77}$$

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):

$$\frac{s_l}{s_l + s_g} = S_*\left(P_{cgl}\right), \tag{10-78a}$$

$$\frac{s_l}{s_l + s_i} = S_*\left[\frac{\sigma_{gl}}{\sigma_{il}}P_{cil}\right], \tag{10-78b}$$

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, σ_{il} and σ_{gl} are the ice-liquid and gas-liquid interfacial tensions. Also, $P_{cil} = -\rho_i h_{iw} \vartheta$, where h^0_{iw} is the heat of fusion of ice at 273.15 K, ρ_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:

$$S_* = \begin{cases} \left[1+(\alpha P_c)^\gamma\right]^{-\lambda}, & P_c > 0 \\ 1, & P_c \leq 0 \end{cases} \tag{10-79}$$

with the Mualem model implemented for the relative permeability of liquid water,

$$k_{rl} = (s_l)^{\frac{1}{2}} \left[1 - \left(1 - (s_l)^{\frac{1}{\lambda}}\right)^{\lambda}\right]^2, \tag{10-80}$$

where λ, α are parameters, with $\gamma = \frac{1}{1-\lambda}$.

The thermal conductivity for the frozen soil is chosen to be

$$\kappa = Ke_f \kappa_{\text{wet},f} + Ke_u \kappa_{\text{wet},u} + (1 - Ke_u - Ke_f)\kappa_{\text{dry}}, \tag{10-81}$$

where $\kappa_{\text{wet},f}$, $\kappa_{\text{wet},u}$ are the liquid- and ice-saturated thermal conductivities, κ_{dry} is the dry thermal conducitivity, 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

$$Ke_f = (s_i)^{\alpha_f}, \tag{10-82a}$$

$$Ke_u = (s_l)^{\alpha_u}, \tag{10-82b}$$

with α_f, α_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 α_f, α_u are less than one.

The gas diffusion coefficient D_g is assumed to dependend on temperature and pressure as follows:

$$D_g = D^0_g \left(\frac{P_{\text{ref}}}{P}\right) \left(\frac{T}{T_{\text{ref}}}\right)^{1.8}, \tag{10-83}$$

where D^0_g 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_α and T_β, 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 subscript n and matrix nodes by m, the integrated finite volume form of the heat transport equation for the nth fracture control volume is given by

$$\begin{aligned}
&\Big[\varphi\Big(\sum_\alpha \big(s_\alpha\rho_\alpha U_\alpha\big)_n^{k+1} - \sum_\alpha \big(s_\alpha\rho_\alpha U_\alpha\big)_n^{k}\Big) + (1-\varphi)\Big(\big(\rho_r C_r T_\alpha\big)_n^{k+1} - \big(\rho_r C_r T_\alpha\big)_n^{k}\Big)\Big]\frac{v_n}{\Delta t} \\
&\quad + \sum_{n'}\Big[\sum_\alpha \big(q_\alpha\rho_\alpha H_\alpha\big)_{nn'} + \frac{\kappa_{nn'}^\alpha}{d_n+d_{n'}}\big(T_{\alpha n}-T_{\alpha n'}\big)\Big]A_{nn'}^\alpha \\
&\quad + \sum_\beta \frac{\kappa_{nM}^{\alpha\beta}}{d_n+d_M}\big(T_{\alpha n}-T_{\beta M}\big)A_{nM}^\beta \;=\; 0,
\end{aligned} \tag{10-84a}$$

where v_n denotes the fracture volume, and

$$\begin{aligned}
\Big(\big(\rho_r C_r T_\beta\big)_m^{k+1} - \big(\rho_r C_r T_\beta\big)_m^{k}\Big)\frac{v_m}{\Delta t} + \sum_{m'}\frac{\kappa_{mm'}^\beta}{d_m+d_{m'}}\big(T_{\beta m}-T_{\beta m'}\big)A_{mm'}^\beta& \\
+\,\delta_{mM}\frac{\kappa_{nM}^{\alpha\beta}}{d_n+d_M}\big(T_{\alpha n}-T_{\beta M}\big)A_{nM}^\beta \;=\; 0,&
\end{aligned} \tag{10-84b}$$

for the mth 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 β in Eqn.(10-84a).

For better convergence uniform logarithmic spacing is used for the matrix nodes

$$\Delta\xi_m \;=\; \rho\,\Delta\xi_{m-1}, \tag{10-85a}$$

specifying $\Delta\xi_M$ and l_M for the outer most matrix node and matrix block size, respectively. The factor ρ is determined from the constraint

$$l_M \;=\; 2\sum_{m=1}^{M}\Delta\xi_m, \tag{10-85b}$$

which requires that ρ satisfy the equation

$$\frac{l_M}{2\Delta\xi_1} \;=\; \frac{\rho^M-1}{\rho-1}, \tag{10-85c}$$

with the inner and outer grid spacing related by

$$\Delta\xi_M \;=\; \rho^{M-1}\Delta\xi_1. \tag{10-85d}$$

Thermal conductivity at the interface between two control volumes is calculated using the harmonic average

$$\kappa_{ll'} \;=\; \frac{\kappa_l\kappa_{l'}(d_l+d_{l'})}{d_l\kappa_{l'}+d_{l'}\kappa_l}. \tag{10-86}$$

The fracture volume v_n is related to the REV volume V_n by

$$\epsilon \;=\; \frac{v_n}{V_n}. \tag{10-87}$$

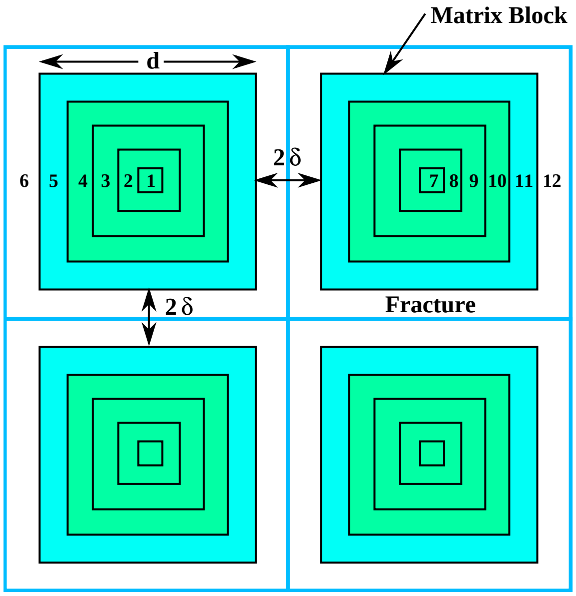


Figure 5: Control volumes in DCDM multiple continuum model with fracture aperture 2δ and matrix block size d.

According to the geometry in Figure 5 assuming a 3D orthogonal set of fractures,

$$V_n = (d+2\delta)^3, \tag{10-88}$$

and

$$v_n = (d+2\delta)^3 - d^3, \tag{10-89}$$

giving

$$\epsilon = 1 - \frac{d^3}{(d+2\delta)^3} = 1 - \left(\frac{1}{1+\frac{2\delta}{d}}\right)^3, \tag{10-90a}$$

$$\simeq \frac{6\delta}{d}. \tag{10-90b}$$

The fracture aperture 2δ is found to be in terms of ϵ and d

$$2\delta = d\left(\frac{1}{(1-\epsilon)^{1/3}} - 1\right). \tag{10-91}$$

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 $A_{nn'}^{\alpha}$ 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 (ϵ, 2δ, 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} per unit volume is equal to

$$A_{nM}^{\beta} = \frac{\mathcal{N}_\beta}{V} A_\beta^0, \tag{10-92}$$

Table 9: DCDM geometric parameters.

Geometry	Area A_β^0	Volume V_β^0
Slab	A	Al
Nested Cubes	$6d^2$	d^3
Nested Spheres	$4\pi R^2$	$\frac{4}{3}\pi R^3$

Table 10: Independent and dependent nested cube parameters.

Independent		Dependent	
ϵ	l_f	$2\delta = l_f - l_m$	$l_m = l_f(1-\epsilon)^{1/3}$
ϵ	l_m	$2\delta = l_f - l_m$	$l_f = l_m(1-\epsilon)^{-1/3}$
2δ	l_f	$\epsilon = 1-(l_m/l_f)^3$	$l_m = l_f - 2\delta$
2δ	l_m	$\epsilon = 1-(l_m/{}_f)^3$	$l_f = l_m + 2\delta$
2δ	ϵ	$l_m = 2\delta\Big(\frac{1}{(1-\epsilon)^{1/3}} - 1\Big)^{-1}$	$l_m = l - 2\delta$

where the number density $\mathcal{N}_\beta/V$ of secondary continua of type β is equal to

$$\frac{\mathcal{N}_\beta}{V} \;=\; \frac{1}{V}\frac{V_\beta}{V_\beta^0} \;=\; \frac{\epsilon_\beta}{V_\beta^0}, \tag{10-93}$$

and A_β^0 and V_β^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

$$\sum_\beta \frac{\kappa_{nM}^{\alpha\beta}}{d_n + d_M}\big(T_n^\alpha - T_M^\beta\big) A_{nM}^\beta \;=\; V_n \sum_\beta \frac{\epsilon_\beta \kappa_{nM}^{\alpha\beta}}{d_n + d_M}\big(T_n^\alpha - T_M^\beta\big)\frac{A_\beta^0}{V_\beta^0}. \tag{10-94}$$

In terms of partial differential equations the heat conservation equations may be written as

$$\begin{aligned}\frac{\partial}{\partial t}\epsilon\Big[\varphi\sum_\alpha s_\alpha\rho_\alpha U_\alpha + (1-\varphi)\rho_r C_r T_f\Big] + \boldsymbol{\nabla}\cdot\Big(\sum_\alpha \boldsymbol{q}_\alpha\rho_\alpha H_\alpha - \kappa_f \boldsymbol{\nabla} T_f\Big)&\\ = -A_{fm}\kappa_{fm}\frac{\partial T_m}{\partial n},&\end{aligned} \tag{10-95a}$$

and

$$\frac{\partial}{\partial t}\rho_r C_r T_m + \frac{\partial}{\partial \xi}\Big(-\kappa_m\frac{\partial T_m}{\partial \xi}\Big) \;=\; 0, \tag{10-95b}$$

for fracture and matrix temperatures T_f and T_m, respectively, where ξ represents the matrix coordinate assumed to be an effective 1D domain. The boundary condition

$$T_m(\xi_d,\, t\,|\,\boldsymbol{r}) \;=\; T_f(\boldsymbol{r},\, t), \tag{10-95c}$$

between fracture and matrix continua is imposed, where ξ_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

$$\frac{\partial}{\partial t}\big(\varphi\sum_\alpha s_\alpha \Psi_j^\alpha\big) + \nabla\cdot\sum_\alpha \boldsymbol{\Omega}_j^\alpha \;=\; Q_j - \sum_m \nu_{jm} I_m - \frac{\partial S_j}{\partial t}, \tag{10-96}$$

and

$$\frac{\partial \varphi_m}{\partial t} \;=\; \overline{V}_m I_m, \tag{10-97}$$

for minerals with molar volume $\overline{V}_m$, mineral reaction rate I_m and mineral volume fraction φ_m referenced to an REV. Sums over α in Eqn.(10-96) are over all fluid phases in the system. The quantity Ψ_j^α denotes the total concentration of the jth primary species $\mathcal{A}_j^{\rm pri}$ in the αth fluid phase defined by

$$\Psi_j^\alpha = \delta_{l\alpha} C_j^l + \sum_{i=1}^{N_{\rm sec}} \nu_{ji}^\alpha C_i^\alpha. \tag{10-98}$$

In this equation the subscript l represents the aqueous electrolyte phase from which the primary species are chosen. The secondary species concentrations C_i^α are obtained from mass action equations corresponding to equilibrium conditions of the reactions

$$\sum_j \nu_{ji}^\alpha \mathcal{A}_j^l \;\rightleftharpoons\; \mathcal{A}_i^\alpha, \tag{10-99}$$

yielding

$$C_i^\alpha \;=\; \frac{K_i^\alpha}{\gamma_i^\alpha}\prod_j \Big(\gamma_j^l C_j^l\Big)^{\nu_{ji}^\alpha}, \tag{10-100}$$

with equilibrium constant K_i^α, and activity coefficients γ_k^α. The total flux $\boldsymbol{\Omega}_j^\alpha$ for species-independent diffusion is given by

$$\boldsymbol{\Omega}_j^\alpha \;=\; \big(\boldsymbol{q}_\alpha - \varphi s_\alpha \boldsymbol{D}_\alpha \boldsymbol{\nabla}\big)\Psi_j^\alpha. \tag{10-101}$$

The diffusion/dispersion coefficient $\boldsymbol{D}_\alpha$ 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 $\boldsymbol{q}_\alpha$ for phase α is given by

$$\boldsymbol{q}_a \;=\; -\frac{kk_\alpha}{\mu_\alpha}\boldsymbol{\nabla}\big(p_\alpha - \rho_\alpha g z\big), \tag{10-102}$$

with bulk permeability of the porous medium k and relative permeability k_α, fluid viscosity μ_α, pressure p_α, density ρ_α, and acceleration of gravity g. The diffusivity/dispersivity tensor $\boldsymbol{D}_\alpha$ is the sum of contributions from molecular diffusion and dispersion which for an isotropic medium has the form

$$\boldsymbol{D}_\alpha \;=\; \tau D_m \boldsymbol{I} + a_T v \boldsymbol{I} + \big(a_L - a_T\big)\frac{\boldsymbol{v}\boldsymbol{v}}{v}, \tag{10-103}$$

with longitudinal and transverse dispersivity coefficients a_L, a_T, respectively, τ 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

$$\varphi \;=\; 1-\sum_m \varphi_m. \tag{10-104}$$

The temperature dependence of the diffusion coefficient is defined through the relation

$$D_m(T) \;=\; D_m^\circ \exp\left[\frac{A_D}{R}\left(\frac{1}{T_0}-\frac{1}{T}\right)\right], \tag{10-105}$$

with diffusion activation energy A_D in kJ/mol. The quantity D_m° denotes the diffusion coefficient at the reference temperature T_0 taken as 25°C and the quantity R denotes the gas constant (8.317×10^{-3} kJ/mol/K). The temperature T is in Kelvin.

The quantity Q_j denotes a source/sink term

$$Q_j \;=\; \sum_n \frac{q_M}{\rho}\Psi_j \delta(\boldsymbol{r}-\boldsymbol{r}_n), \tag{10-106}$$

where q_M denotes a mass rate in units of kg/s, ρ denotes the fluid density in kg/m^3, and $\boldsymbol{r}_n$ refers to the location of the nth source/sink. The quantity S_j represents the sorbed concentration of the jth primary species considered in more detail in the next section.

Molality m_i and molarity C_i are related by the density of water ρ_w according to

$$C_i \;=\; \rho_w m_i. \tag{10-107}$$

The activity of water is calculated from the approximate relation

$$a_{\mathrm{H_2O}} \;=\; 1-0.017\sum_i m_i. \tag{10-108}$$

10.9.1 Mineral Precipitation and Dissolution

The reaction rate I_m is based on transition state theory with the form

$$I_m \;=\; -a_m\left(\sum_l k_{ml}(T)\mathcal{P}_{ml}\right)\Big(1-K_m Q_m\Big), \tag{10-109}$$

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

$$Q_m \;=\; \prod_j \big(\gamma_j m_j\big)^{\nu_{jm}}, \tag{10-110}$$

with molality m_j. The rate constant k_{ml} is a function of temperature given by the Arrhenius relation

$$k_{ml}(T) \;=\; k_{ml}^0 \exp\left[\frac{E_{ml}}{R}\Big(\frac{1}{T_0}-\frac{1}{T}\Big)\right], \tag{10-111}$$

where k_{ml}^0 refers to the rate constant at the reference temperature T_0 taken as 25°C, with T in units of Kelvin, E_{ml} denotes the activation energy (kJ/mol), and the quantity $\mathcal{P}_{ml}$ denotes the prefactor for the lth parallel reaction with the form

$$\mathcal{P}_{ml} \;=\; \prod_i \frac{\big(\gamma_i m_i\big)^{\alpha_{il}^m}}{1+K_{ml}\big(\gamma_i m_i\big)^{\beta_{il}^m}}, \tag{10-112}$$

where the product index i generally runs over both primary and secondary species, the quantities α_{il}^m and β_{il}^m refer to prefactor coefficients, and K_{ml} is an attenuation factor. The quantity R denotes the gas constant (8.317×10^{-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

$$\widehat{I}_m \;=\; -a_m\left(\sum_l \mathcal{P}_{ml} k_{ml}\right)\left[\frac{1-\left(K_m Q_m\right)^{1/\sigma_m}}{1+\dfrac{k_{ml}}{k_{ml}^{\text{lim}}}\left(K_m Q_m\right)^{1/\sigma_m}}\right], \tag{10-113}$$

with rate-limiter r_{lim}. In the limit $K_m Q_m \to \infty$, the rate becomes

$$\lim_{K_m Q_m\to\infty} \widehat{I}_m \;=\; k_{ml}^{\text{lim}} a_m \sum_l \mathcal{P}_{ml}. \tag{10-114}$$

Defining the affinity factor

$$\Omega_m \;=\; 1-\left(K_m Q_m\right)^{1/\sigma_m}, \tag{10-115}$$

or

$$K_m Q_m \;=\; \Big(1-\Omega_m\Big)^{\sigma_m}, \tag{10-116}$$

the rate may be expressed alternatively as

$$\widehat{I}_m \;=\; -a_m \sum_l \mathcal{P}_{ml} k_{ml} \frac{\Omega_m}{1+\dfrac{k_{ml}}{k_{ml}^{\text{lim}}}\big(1-\Omega_m\big)}. \tag{10-117}$$

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

$$\varphi \;=\; 1-\sum_m \varphi_m, \tag{10-118}$$

$$k \;=\; k_0 f(\varphi,\, \varphi_0,\, \varphi_c,\, a), \tag{10-119}$$

with

$$f \;=\; \left(\frac{\varphi-\varphi_c}{\varphi_0-\varphi_c}\right)^a, \tag{10-120}$$

$$\;=\; f_{\text{min}} \quad \text{if} \quad \varphi \le \varphi_c, \tag{10-121}$$

$$\tau = \tau_0 \left(\frac{\varphi}{\varphi_0}\right)^b, \tag{10-122}$$

and

$$a_m = a_m^0 \left(\frac{\varphi_m}{\varphi_m^0}\right)^n \left(\frac{1-\varphi}{1-\varphi_0}\right)^{n'}, \tag{10-123}$$

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 φ_c refers to a critical porosity below which the permeability is assumed to be constant with scale factor $f_{\rm min}$.

In PFLOTRAN the solid is represented as an aggregate of minerals described quantitatively by specifying its porosity φ and the volume fraction φ_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

$$\varphi_m = (1-\varphi)\frac{x_m \overline{V}_m}{\sum_{m'} x_{m'} \overline{V}_{m'}}, \tag{10-124a}$$

$$= (1-\varphi)\frac{y_m \rho_m^{-1}}{\sum_{m'} y_{m'} \rho_{m'}^{-1}}, \tag{10-124b}$$

with inverse relation

$$x_m = \frac{\varphi_m}{\overline{V}_m \eta_s (1-\varphi)}, \tag{10-125}$$

and similarly for the mass fraction, where

$$\rho_m = W_m \overline{V}_m^{-1}, \tag{10-126}$$

and the solid molar density η_s is given by

$$\eta_s = \frac{1}{\sum_m x_m \overline{V}_m}. \tag{10-127}$$

In these relations W_m refers to the formula weight and $\overline{V}_m$ the molar volume of the mth mineral. The solid molar density is related to the mass density ρ_s by

$$\rho_s = W_s \eta_s, \tag{10-128}$$

with the mean molecular weight W_s of the solid phase equal to

$$W_s = \sum_m x_m W_m = \frac{1}{\sum_m W_m^{-1} y_m}. \tag{10-129}$$

Mass and mole fractions are related by the expression

$$W_m x_m = W_s y_m. \tag{10-130}$$

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

$$a_m(t) \;=\; a_m^0 \left(\frac{\varphi_m}{\varphi_m^0}\right)^n \left(\frac{1-\varphi}{1-\varphi_0}\right)^{n'} \left(\frac{\varphi_{m'}^c - \varphi_{m'}}{\varphi_{m'}^c}\right)^{n''}, \tag{10-131}$$

for $\varphi_{m'} < \varphi_{m'}^c$, and

$$a_m \;=\; 0, \tag{10-132}$$

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 ω_α, in units of moles of sites per bulk sediment volume (mol/dm^3), is related to the bulk cation exchange capacity Q_α (mol/kg) by the expression

$$\omega_\alpha \;=\; \frac{N_{\rm site}}{V} \;=\; \frac{N_{\rm site}}{M_s}\frac{M_s}{V_s}\frac{V_s}{V} \;=\; Q_\alpha \rho_s (1-\phi). \tag{10-133}$$

The cation exchange capacity associated with the mth mineral is defined on a molar basis as

$$\omega_m^{\rm CEC} \;=\; \frac{N_m}{V} \;=\; \frac{N_m}{M_m}\frac{M_m}{V_m}\frac{V_m}{V} \;=\; Q_m^{\rm CEC} \rho_m \phi_m. \tag{10-134}$$

In PFLOTRAN ion exchange reactions are expressed in the form

$$z_i \mathcal{A}_j + z_j X_{z_i}^\alpha \mathcal{A}_i \;\rightleftharpoons\; z_j \mathcal{A}_i + z_i X_{z_j}^\alpha \mathcal{A}_j, \tag{10-135}$$

with valencies z_j, z_i of cations $\mathcal{A}_j$ and $\mathcal{A}_i$, respectively. The reference cation is denoted by $\mathcal{A}_j$ and $\mathcal{A}_i,\, i \neq j$ represents all other cations. The corresponding mass action equation is given by

$$K_{ji}^\alpha \;=\; \frac{(k_j^\alpha)^{z_i}}{(k_i^\alpha)^{z_j}} \;=\; \left(\frac{X_j^\alpha}{a_j}\right)^{z_i} \left(\frac{a_i}{X_i^\alpha}\right)^{z_j}. \tag{10-136}$$

Using the Gaines-Thomas convention, the equivalent fractions X_k^α are defined by

$$X_k^\alpha = \frac{z_k S_k^\alpha}{\displaystyle\sum_l z_l S_l^\alpha} = \frac{z_k}{\omega_\alpha} S_k^\alpha, \tag{10-137}$$

with

$$\sum_k X_k^\alpha = 1. \tag{10-138}$$

The site concentration ω_α is defined by

$$\omega_\alpha = \sum_k z_k S_k^\alpha, \tag{10-139}$$

where ω_α is related to the cation exchange capacity Q_α (CEC) by the expression

$$\omega_\alpha = (1-\varphi)\rho_s Q_\alpha, \tag{10-140}$$

with solid density ρ_s and porosity φ.

An alternative form of reactions 10-135 often found in the literature is

$$\frac{1}{z_j}\mathcal{A}_j + \frac{1}{z_i}X_{z_i}^\alpha \mathcal{A}_i \rightleftharpoons \frac{1}{z_i}\mathcal{A}_i + \frac{1}{z_j}X_{z_j}^\alpha \mathcal{A}_j, \tag{10-141}$$

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

$$K'^\alpha_{ji} = \frac{(k'^\alpha_j)^{1/z_j}}{(k'^\alpha_i)^{1/z_i}} = \left(\frac{X_j^\alpha}{a_j}\right)^{1/z_j} \left(\frac{a_i}{X_i^\alpha}\right)^{1/z_i}. \tag{10-142}$$

The selectivity coefficients corresponding to the two forms are related by the expression

$$K_{ji}^\alpha = \left(K'^\alpha_{ji}\right)^{z_i z_j}, \tag{10-143}$$

and similarly for k_i^α, k_j^α. 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

$$S_j^\alpha = \frac{\omega_\alpha}{z}\frac{k_j^\alpha \gamma_j m_j}{\displaystyle\sum_l k_l^\alpha \gamma_l m_l}, \tag{10-144}$$

where m_k denotes the kth 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

$$X_j^\alpha + \sum_{i\neq j} X_i^\alpha = 1, \tag{10-145}$$

for the reference cation mole fraction X_j. From the mass action equation Eqn.(10-136) it follows that

$$X_i^\alpha = k_i^\alpha a_i \left(\frac{X_j^\alpha}{k_j^\alpha a_j}\right)^{z_i/z_j}. \tag{10-146}$$

Defining the function

$$f(X_j^\alpha) \;=\; X_j^\alpha + \sum_{i\neq j} X_i^\alpha(X_j^\alpha) - 1, \tag{10-147}$$

its derivative is given by

$$\frac{df}{dX_j^\alpha} \;=\; 1 - \frac{1}{z_j X_j^\alpha}\sum_{i\neq j} z_i k_i^\alpha a_i \left(\frac{X_j^\alpha}{k_j^\alpha a_j}\right)^{z_i/z_j}. \tag{10-148}$$

The reference mole fraction is then obtained by Newton-Raphson iteration

$$(X_j^\alpha)^{k+1} \;=\; (X_j^\alpha)^k - \frac{f[(X_j^\alpha)^k]}{\dfrac{df[(X_j^\alpha)^k]}{dX_j^\alpha}}. \tag{10-149}$$

The sorbed concentration for the jth cation appearing in the accumulation term is given by

$$S_j^\alpha \;=\; \frac{\omega_\alpha}{z_j} X_j^\alpha, \tag{10-150}$$

with the derivatives for $j \neq l$

$$\frac{\partial S_j^\alpha}{\partial m_l} \;=\; -\frac{\omega_\alpha}{m_l}\frac{X_j^\alpha X_l^\alpha}{\displaystyle\sum_l z_l X_l^\alpha}, \tag{10-151a}$$

$$\;=\; -\frac{1}{m_l}\frac{z_j z_l S_j^\alpha S_l^\alpha}{\displaystyle\sum_l z_l^2 S_l^\alpha}, \tag{10-151b}$$

and for $j = l$

$$\frac{\partial S_j^\alpha}{\partial m_j} \;=\; \frac{\omega_\alpha X_j^\alpha}{z_j m_j}\left(1 - \frac{z_j X_j^\alpha}{\displaystyle\sum_l z_l X_l^\alpha}\right), \tag{10-152a}$$

$$\;=\; \frac{S_j^\alpha}{m_j}\left(1 - \frac{z_j^2 S_j^\alpha}{\displaystyle\sum_l z_l^2 S_l^\alpha}\right). \tag{10-152b}$$

10.9.2.2 Surface Complexation Surface complexation reactions are assumed to have the form

$$\nu_\alpha > \!\chi_\alpha + \sum_j \nu_{ji}\mathcal{A}_j \;\rightleftharpoons\; >\!\mathcal{S}_{i\alpha}, \tag{10-153}$$

for the ith surface complex $>\!\mathcal{S}_{i\alpha}$ on site α and empty site $>\!\chi_\alpha$. As follows from the corresponding mass action equation the equilibrium sorption concentration $S_{i\alpha}^{\rm eq}$ is given by

$$S_{i\alpha}^{\rm eq} \;=\; \frac{\omega_\alpha K_i Q_i}{1 + \sum_l K_l Q_l}, \tag{10-154}$$

and the empty site concentration by

$$S_\alpha^{\rm eq} \;=\; \frac{\omega_\alpha}{1+\sum_l K_l Q_l}, \tag{10-155}$$

where the ion activity product Q_i is defined by

$$Q_i \;=\; \prod_j \big(\gamma_j C_j\big)^{\nu_{ji}}. \tag{10-156}$$

The site concentration ω_α satisfies the relation

$$\omega_\alpha \;=\; S_\alpha + \sum_i S_{i\alpha}, \tag{10-157}$$

and is constant. The equilibrium sorbed concentration $S_{j\alpha}^{\rm eq}$ is defined as

$$S_{j\alpha}^{\rm eq} \;=\; \sum_i \nu_{ji} S_{i\alpha}^{\rm eq} \;=\; \frac{\omega_\alpha}{1+\sum_l K_l Q_l} \sum_i \nu_{ji} K_i Q_i. \tag{10-158}$$

10.9.2.3 Multirate Sorption In the multirate model the rates of sorption reactions are described through a kinetic relation given by

$$\frac{\partial S_{i\alpha}}{\partial t} \;=\; k_\alpha\big(S_{i\alpha}^{\rm eq} - S_{i\alpha}\big), \tag{10-159}$$

for surface complexes, and

$$\frac{\partial S_\alpha}{\partial t} \;=\; -\sum_i k_\alpha\big(S_{i\alpha}^{\rm eq} - S_{i\alpha}\big), \tag{10-160}$$

$$\;=\; k_\alpha\big(S_\alpha^{\rm eq} - S_\alpha\big), \tag{10-161}$$

for empty sites, where $S_\alpha^{\rm eq}$ denotes the equilibrium sorbed concentration. For simplicity, in what follows it is assumed that $\nu_\alpha = 1$. With each site α is associated a rate constant k_α and site concentration ω_α. These quantities are defined through a given distribution of sites $\wp(\alpha)$, such that

$$\int_0^\infty \wp(k_\alpha) dk_\alpha \;=\; 1. \tag{10-162}$$

The fraction of sites f_α belonging to site α is determined from the relation

$$f_\alpha \;=\; \int_{k_\alpha-\Delta k_\alpha/2}^{k_\alpha+\Delta k_\alpha/2} \wp(k_\alpha) dk_\alpha \simeq \wp(k_\alpha)\Delta k_\alpha, \tag{10-163}$$

with the property that

$$\sum_\alpha f_\alpha = 1. \tag{10-164}$$

Given that the total site concentration is ω, then the site concentration ω_α associated with site α is equal to

$$\omega_\alpha \;=\; f_\alpha \omega. \tag{10-165}$$

An alternative form of these equations is obtained by introducing the total sorbed concentration for the jth primary species for each site defined as

$$S_{j\alpha} \;=\; \sum_i \nu_{ji} S_{i\alpha}. \tag{10-166}$$

Then the transport equations become

$$\frac{\partial}{\partial t}\left(\varphi\Psi_j + \sum_\alpha S_{j\alpha}\right) + \boldsymbol{\nabla}\cdot\boldsymbol{\Omega}_j \;=\; -\sum_m \nu_{jm} I_m. \tag{10-167}$$

The total sorbed concentrations are obtained from the equations

$$\frac{\partial S_{j\alpha}}{\partial t} \;=\; k_\alpha\big(S_{j\alpha}^{\rm eq} - S_{j\alpha}\big). \tag{10-168}$$

10.9.3 Sorption Isotherm <Under Revision>

The distribution coefficient $\widetilde{K}_j^D$ [m^3 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

$$\widetilde{K}_j^D \;=\; \frac{M_j^s/M_s}{M_j^{\rm aq}/V_l}, \tag{10-169a}$$

$$\;=\; \frac{N_j^s/M_s}{N_j^{\rm aq}/V_l}, \tag{10-169b}$$

$$\;=\; \frac{\widetilde{S}_j}{C_j} \;=\; \frac{1}{\rho_w}\frac{\widetilde{S}_j}{m_j}, \tag{10-169c}$$

where $M_j^s = W_j N_j^s$, $M_j^{\rm aq} = W_j N_j^{\rm aq}$, refers to the mass and number of moles of sorbed and aqueous solute related by the formula weight W_j of the jth species, M_s refers to the mass of the solid, V_l denotes the aqueous volume, $\widetilde{S}_j = N_j^s/M_s$ [mol kg^{-1}] represents the sorbed concentration referenced to the mass of solid, $C_j = N_j^{\rm aq}/V_l$ denotes molarity, and $m_j = C_j/\rho_w$ represents molality, where ρ_w is the density of pure water.

The distribution coefficient $\widetilde{K}_j^D$ may be related to its dimensionless counterpart K_j^D [—] defined by

$$K_j^D \;=\; \frac{N_i^s}{N_i^{\rm aq}} \;=\; \frac{N_i^s/V}{N_i^{\rm aq}/V} \;=\; \frac{1}{\varphi s_l}\frac{S_j}{C_j}, \tag{10-170}$$

by writing

$$K_j^D \;=\; \frac{N_i^s}{M_s}\frac{M_s}{V_s}\frac{V_s}{V_p}\frac{V_p}{V_l}\frac{V_l}{N_i^{\rm aq}}, \tag{10-171}$$

$$\;=\; \rho_s\frac{1-\varphi}{\varphi s_l}\widetilde{K}_j^D \;=\; \frac{\rho_b}{\varphi s_l}\widetilde{K}_j^D, \tag{10-172}$$

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 $\widehat{K}_j^D$ [kg m^{-3}] is obtained by using molality to define the solute concentration and referencing the sorbed concentration to the bulk volume V

$$\widehat{K}_j^D \;=\; \frac{N_j^s/V}{N_j^{\rm aq}/M_w} \;=\; \frac{S_j}{m_j}. \tag{10-173}$$

A sorption isotherm S_j may be specified for any primary species $\mathcal{A}_j$ resulting in the transport equation

$$\frac{\partial}{\partial t}\varphi s_l C_j + \boldsymbol{\nabla}\cdot\boldsymbol{F}_j \;=\; -\frac{\partial S_j}{\partial t}, \tag{10-174}$$

for a partially saturated medium. Substituting $S_j = \varphi s_l K_j^D C_j$ from Eqn.(10-170) and introducing the retardation $\mathcal{R}_j$ gives

$$\frac{\partial}{\partial t}R_j\varphi s_l C_j + \boldsymbol{\nabla}\cdot\boldsymbol{F}_j \;=\; 0, \tag{10-175}$$

with the retardation given by the alternative forms

$$\begin{aligned}
R_j \;&=\; 1 + K_j^D, \qquad \text{(dimensionless)}, && \text{(10-176a)}\\
&=\; 1 + \frac{\rho_b}{\varphi s_l}\widetilde{K}_j^D, \qquad \text{(conventional)}, && \text{(10-176b)}\\
&=\; 1 + \frac{1}{\varphi s_l \rho_w}\widehat{K}_j^D, \qquad \text{(molality-based)}. && \text{(10-176c)}
\end{aligned}$$

Three distinct models are available for the sorption isotherm S_j in PFLOTRAN:

- linear K_D model:

  $$S_j \;=\; \varphi s_l K_j^D C_j \;=\; \widehat{K}_j^D m_j, \tag{10-177}$$

  with distribution coefficient $\widehat{K}_j^D$,

- Langmuir isotherm:

  $$S_j \;=\; \frac{K_j^L b_j^L C_j/\rho_w}{1+K_j^L C_j/\rho_w} \;=\; \frac{K_j^L b_j^L m_j}{1+K_j^L m_j}, \tag{10-178}$$

  with Langmuir coefficients K_j^L and b_j^L, and

- Freundlich isotherm:

  $$S_j \;=\; K_j^F\left(\frac{C_j}{\rho_w}\right)^{(1/n_j^F)} \;=\; K_j^F\big(m_j\big)^{(1/n_j^F)}, \tag{10-179}$$

  with coefficients K_j^F and n_j^F.

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

$$>X_k^{\rm m} + \sum_j \nu_{jk}\mathcal{A}_j \ \rightleftharpoons\ >S_k^{\rm m}, \tag{10-180}$$

$$>X_k^{\rm im} + \sum_j \nu_{jk}\mathcal{A}_j \ \rightleftharpoons\ >S_k^{\rm im}, \tag{10-181}$$

$$>X_k^{s} + \sum_j \nu_{jk}\mathcal{A}_j \ \rightleftharpoons\ >S_k^{s}, \tag{10-182}$$

with corresponding reaction rates $I_k^{\rm m}$, $I_k^{\rm im}$, and I_k^s, 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

$$\sum_j \nu_{js}\mathcal{A}_j \ \rightleftharpoons\ \mathcal{M}_s. \tag{10-183}$$

The transport equations for primary species, mobile and immobile colloids, read

$$\frac{\partial}{\partial t}\varphi s_l \Psi_j^l + \boldsymbol{\nabla}\cdot\boldsymbol{\Omega}_j^l \ = \ -\sum_k \nu_{jk}\big(I_k^{\rm m} + I_k^{\rm im} + \sum_s I_k^s\big) - \sum_s \nu_{js} I_s, \tag{10-184}$$

$$\frac{\partial}{\partial t}\varphi s_l S_k^{\rm m} + \boldsymbol{\nabla}\cdot\boldsymbol{q}_c S_k^{\rm m} \ = \ I_k^{\rm m}, \tag{10-185}$$

$$\frac{\partial}{\partial t} S_k^{\rm im} \ = \ I_k^{\rm im}, \tag{10-186}$$

$$\frac{\partial}{\partial t} S_k^{s} \ = \ I_k^{s}, \tag{10-187}$$

where $\boldsymbol{q}_c$ denotes the colloid Darcy velocity which may be greater than the fluid velocity $\boldsymbol{q}$. For conditions of local equilibrium the sorption reaction rates may be eliminated and replaced by algebraic sorption isotherms to yield

$$\frac{\partial}{\partial t}\Big[\varphi s_l \Psi_j^l + \sum_k \nu_{jk}\big(\varphi s_l S_k^{\rm m} + S_k^{\rm im} + \sum_s S_k^s\big)\Big] + \boldsymbol{\nabla}\cdot\Big(\boldsymbol{\Omega}_j^l + \boldsymbol{q}_c \sum_k \nu_{jk} S_k^{\rm m}\Big) \ = \ -\sum_s \nu_{js} I_s. \tag{10-188}$$

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_k^m might be

$$I_k^m \ = \ k_k\big(S_k^m - S_{km}^{\rm eq}\big), \tag{10-189}$$

with rate constant k_k and where $S_{km}^{\rm eq}$ 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

$$\frac{\partial}{\partial t}\varphi s A C + \boldsymbol{\nabla}\cdot\Big(\boldsymbol{q} A C - \varphi s D \boldsymbol{\nabla}(AC)\Big) \;=\; \varphi s C, \tag{10-190}$$

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 α 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

```
  PRIMARY_SPECIES
    Tracer
    Tracer_Age
  /
```

including sorption through a constant K_d model if desired

```
  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.:

```
CONSTRAINT initial
  CONCENTRATIONS
    Tracer     1.e-8        F
    Tracer_Age 1.e-16       F
  /
/
```

Output is given in terms of α 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
CII: GEMBOCHS.V2-EQ8-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

$$\log K \;=\; c_{-1}\ln T + c_0 + c_1 T + \frac{c_2}{T} + \frac{c_3}{T^2}, \tag{10-191}$$

with fit coefficients c_i. The thermodynamic database stores all chemical reaction properties (equilibrium constant $\log K_r$, reaction stoichiometry ν_{ir}, species valence z_i, Debye parameter a_i, mineral molar volume $\overline{V}_m$, 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 $N_{\rm temp}$ refers to the number of temperatures at which log K values are stored in the database. In the `hanford.dat` database $N_{\rm temp} = 8$ 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

$$\mathcal{A}_r \;\rightleftharpoons\; \sum_{i=1}^{\rm nspec} \nu_{ir}\mathcal{A}_i, \tag{10-192}$$

for species $\mathcal{A}_r$, where `nspec` 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

$$2\,\mathrm{H_2O} - 4\,\mathrm{H^+} - 4\,\mathrm{e^-} \;\rightleftharpoons\; \mathrm{O_2}, \tag{10-193}$$

Table 11: Format of thermodynamic database.

Primary Species:	name, a_0, z, w
Secondary Species:	name, nspec, (ν(n), name(n), n=1, nspec), log$K(1{:}N_{\text{temp}})$, a_0, z, w
Gaseous Species:	name, $\overline{V}$, nspec, (ν(n), name(n), n=1, nspec), log$K(1{:}N_{\text{temp}})$, w
Minerals:	name, $\overline{V}$, nspec, (ν(n), name(n), n=1, nspec), log$K(1{:}N_{\text{temp}})$, w
Surface Complexes:	>name, nspec, ν, >site, (ν(n), name(n), n=1, nspec-1),
	log$K(1{:}N_{\text{temp}})$, z, w

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.

c_{-1}	6.745529048
c_0	−48.295936593
c_1	−0.000557816
c_2	27780.749538022
c_3	4027.337694858

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:

$$\nabla \cdot [\boldsymbol{\sigma}] + \rho \boldsymbol{b} = 0 \quad \text{in } \Omega, \tag{10-194a}$$

$$\boldsymbol{\sigma} = \lambda \text{tr}\,(\boldsymbol{\varepsilon}) + 2\mu\boldsymbol{\varepsilon} - \beta(p - p_0)\boldsymbol{I} - \alpha(T - T_0)\boldsymbol{I}, \tag{10-194b}$$

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u}(\boldsymbol{x}) + [\nabla \boldsymbol{u}(\boldsymbol{x})]^T\right), \tag{10-194c}$$

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}^p(\boldsymbol{x}) \quad \text{on } \Gamma^D, \tag{10-194d}$$

$$\boldsymbol{\sigma}\boldsymbol{n}(\boldsymbol{x}) = t^p(\boldsymbol{x}) \quad \text{on } \Gamma^N, \tag{10-194e}$$

where $\boldsymbol{u}$ is the unknown displacement field, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, λ, μ are Lamé parameters (Young's modulus and Poisson's ratio can be related to these two parameters), $\boldsymbol{b}$ is the specific body force (which is gravity in most cases), $\boldsymbol{n}$ is the outward normal to the boundary Γ^N. Also, $\boldsymbol{u}^p$ is the prescribed values of $\boldsymbol{u}$ on the Dirichlet part of the boundary Γ^D, and $\boldsymbol{t}^p$ is the prescribed traction on Γ^N. Additionally, β is the Biot's coefficient, α 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, $\boldsymbol{\varepsilon}$ is the strain tensor and tr is the trace of a second order tensor, Ω is the domain, and $\boldsymbol{I}$ is the identity tensor. Note that stress is assumed *positive under tension*. The effect of deformation on the pore structure is accounted for via

$$\phi = \frac{\phi_0}{1+(1-\phi_0)\mathrm{tr}(\boldsymbol{\varepsilon})}. \tag{10-195}$$

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

$$\frac{\partial}{\partial t}A_j + \boldsymbol{\nabla}\cdot\boldsymbol{F}_j \;=\; Q_j, \tag{11-1}$$

with accumulation A_j, flux $\boldsymbol{F}_j$ and source/sink Q_j. Integrating over a REV corresponding to the nth grid cell with volume V_n yields

$$\frac{d}{dt}\int_{V_n} A_j\,dV + \int_{V_n}\boldsymbol{\nabla}\cdot\boldsymbol{F}_j \;=\; \int_{V_n} Q_j\,dV. \tag{11-2}$$

The accumulation term has the finite volume form

$$\frac{d}{dt}\int_{V_n} A_j\,dV \;=\; \frac{A_{jn}^{t+\Delta t}-A_{jn}^{t}}{\Delta t}V_n, \tag{11-3}$$

with time step Δt. The flux term can be expanded as a surface integral using Gauss' theorem

$$\int_{V_n}\boldsymbol{\nabla}\cdot\boldsymbol{F}_j \;=\; \int_{\partial V_n}\boldsymbol{F}_j\cdot d\boldsymbol{S}, \tag{11-4}$$

$$\;=\; \sum_{n'} F_{j,nn'}A_{nn'}, \tag{11-5}$$

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 nth node with interfacial area $A_{nn'}$. The discretized flux has the form for fluid phase α

$$F^\alpha_{j,nn'} \;=\; \big(q_\alpha X_\alpha\big)_{nn'} - \big(\varphi s_\alpha \tau_\alpha D_\alpha\big)_{nn'} \frac{X^\alpha_{n'} - X^\alpha_n}{d_{n'} + d_n}, \tag{11-6}$$

with perpendicular distances to interface nn' from nodes n and n' denoted by $d_{n'}$ and d_n, respectively. Upstream weighting is used for the advective term

$$(q_\alpha X_\alpha)_{nn'} \;=\; \begin{cases} q^\alpha_{nn'} X_{n'}, & q_{nn'} > 0 \\ q^\alpha_{nn'} X_n, & q_{nn'} < 0 \end{cases}. \tag{11-7}$$

Depending on the type of source/sink term, the finite volume discretization has the form

$$\int_{V_n} Q_j \, dV \;=\; Q_{jn} V_n, \tag{11-8}$$

for reaction rates that are distributed continuously over a control volume, or for a well with point source $Q_j \;=\; \widehat{Q}_j \delta(\boldsymbol{r} - \boldsymbol{r}_0)$:

$$\int_{V_n} Q_j \, dV \;=\; \widehat{Q}_{jn}. \tag{11-9}$$

11.2 Fully Implicit

In a fully implicit formulation the nonlinear equations for the residual function $\boldsymbol{R}$ given by

$$\boldsymbol{R}(\boldsymbol{x}) \;=\; \boldsymbol{0}, \tag{11-10}$$

are solved using an iterative solver based on the Newton-Raphson equations

$$\boldsymbol{J}^{(i)} \delta \boldsymbol{x}^{(i+1)} \;=\; -\boldsymbol{R}^{(i)}, \tag{11-11}$$

at the ith iteration. Iteration stops when

$$\left|\boldsymbol{R}^{(i+1)}\right| < \epsilon, \tag{11-12}$$

or if

$$\left|\delta \boldsymbol{x}^{(i+1)}\right| < \delta. \tag{11-13}$$

However, the latter criteria does not necessarily guarantee that the residual equations are satisfied. The solution is updated from the relation

$$\boldsymbol{x}^{(i+1)} \;=\; \boldsymbol{x}^{(i)} + \delta \boldsymbol{x}^{(i+1)}. \tag{11-14}$$

For the logarithm of the concentration with $\boldsymbol{x} \;=\; \ln \boldsymbol{y}$, the solution is updated according to the equation

$$\boldsymbol{y}^{(i+1)} \;=\; \boldsymbol{y}^{(i)} \mathrm{e}^{\delta \ln \boldsymbol{y}^{(i+1)}}. \tag{11-15}$$

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

$$\frac{S_{j\alpha}^{t+\Delta t} - S_{j\alpha}^{t}}{\Delta t} = k_\alpha \big(f_\alpha S_{j\alpha}^{\rm eq} - S_{j\alpha}^{t+\Delta t}\big). \tag{11-16}$$

Solving for $S_{j\alpha}^{t+\Delta t}$ yields

$$S_{j\alpha}^{t+\Delta t} = \frac{S_{j\alpha}^{t} + k_\alpha \Delta t f_\alpha S_j^{\rm eq}}{1+k_\alpha\Delta t}. \tag{11-17}$$

From this expression the reaction rate can be calculated as

$$\frac{S_{j\alpha}^{t+\Delta t} - S_{j\alpha}^{t}}{\Delta t} = \frac{k_\alpha}{1+k_\alpha\Delta t}\big(f_\alpha S_{j\alpha}^{\rm eq} - S_{j\alpha}^{t}\big). \tag{11-18}$$

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

$$\frac{\partial}{\partial t}\big(\varphi\sum_\alpha s_\alpha \Psi_j^\alpha\big) + \nabla\cdot\sum_\alpha\big(\boldsymbol{q}_\alpha - \varphi s_\alpha \boldsymbol{D}_\alpha \boldsymbol{\nabla}\big)\Psi_j^\alpha = Q_j, \tag{11-19}$$

and

$$\frac{d}{dt}\big(\varphi\sum_\alpha s_\alpha \Psi_j^\alpha\big) = -\sum_m \nu_{jm} I_m - \frac{\partial S_j}{\partial t}, \tag{11-20}$$

The first set of equations are linear in Ψ_j (for species-independent diffusion coeffients) and solved over over a time step Δt resulting in Ψ_j^*. The result for Ψ_j^* is inverted to give the concentrations C_j^* by solving the equations

$$\Psi_j^* = C_j^* + \sum_i \nu_{ji} C_i^*, \tag{11-21}$$

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 Ψ_j^* for the starting value at time t.

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

$$S \;=\; K_d C. \tag{11-22}$$

The governing equation is given by

$$\frac{\partial}{\partial t}\varphi C + \boldsymbol{\nabla}\cdot\big(\boldsymbol{q}C - \varphi D\boldsymbol{\nabla}C\big) \;=\; -\frac{\partial S}{\partial t}. \tag{11-23}$$

If $C(x,\,t;\,\boldsymbol{q},\,D)$ is the solution to the case with no retardation (i.e. $K_d = 0$), then $C(x,\,t;\,\boldsymbol{q}/R,\,D/R)$ is the solution with retardation $(K_d > 0)$, with

$$R = 1 + \frac{1}{\varphi}K_d. \tag{11-24}$$

Thus propagation of a front is retarded by the retardation factor R.

In operator splitting form this equation becomes

$$\frac{\partial}{\partial t}\varphi C + \boldsymbol{\nabla}\cdot\big(\boldsymbol{q}C - \varphi D\boldsymbol{\nabla}C\big) \;=\; 0, \tag{11-25}$$

and

$$\frac{d}{dt}\varphi C \;=\; -\frac{dS}{dt}. \tag{11-26}$$

The solution to the latter equation is given by

$$\varphi C^{t+\Delta t} - \varphi C^* \;=\; -\big(S^{t+\Delta t} - S^t\big), \tag{11-27}$$

where C^* is the solution to the nonreactive transport equation. Using Eqn.(11-22), this result can be written as

$$C^{t+\Delta t} \;=\; \frac{1}{R}C^* + \left(1 - \frac{1}{R}\right)C^t. \tag{11-28}$$

Thus for $R = 1$, $C^{t+\Delta t} = C^*$ and the solution advances unretarded. As $R \to \infty$, $C^{t+\Delta t} \to C^t$ 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 in 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:

Reaction_Sandbox_module (reaction_sandbox.F90)

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.

   `Template` → `Example`

   `template` → `example`
3. Add necessary variables to the module and/or the extended derived type.

```
character(len=MAXWORDLENGTH) :: species_name
PetscInt :: species_id
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.,

```
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.,

```
...
case('SPECIES_NAME')
  call InputReadWord(input,option,this%species_name, &
                     PETSC_TRUE)
  call InputErrorMsg(input,option,'species_name', &
                 'CHEMISTRY,REACTION_SANDBOX,EXAMPLE')
...
```

c ExampleSetup: Construct the reaction network (e.g. array allocation, establishing linkages, etc.). E.g.,

```
...
this%species_id = &
  GetPrimarySpeciesIDFromName(this%species_name, &
                              reaction,option)
...
```

d ExampleReact: Calculate contribution of reaction to the residual (units = moles/sec) and Jacobian (units = kg water/sec). E.g.,

```
...
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

```
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.

```
case('EXAMPLE')
  new_sandbox => ExampleCreate()
```