



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 ^{γ} , Benjamin Andre ^{ζ} ,
Richard Mills ^{ϵ} , Jitu Kumar ^{ρ}**

^{α} OFM Research peter.lichtner@gmail.com

^{β} SNL gehammo@sandia.gov

^{σ} IHEG luchuancn@163.com

^{δ} LANL satkarra@lanl.gov

^{γ} LBNL gbisht@lbl.gov

^{ζ} NCAR andre@ucar.edu

^{ϵ} Intel rtm@eecs.utk.edu

^{ρ} ORNL jitu1503@gmail.com

August 6, 2015



TABLE OF CONTENTS

1	Introduction	11
2	Quick Start	12
2.1	Installing PFLOTRAN	12
2.2	Running on Big Iron Parallel Architectures	15
3	Installation	16
3.1	Compilers	16
3.2	Building PETSc	16
3.2.1	Mac OS X	17
3.2.1.1	Lion, Mavericks (\leq Mac OS X 10.9.x)	17
3.2.1.2	Yosemite (Mac OS X 10.10.x)	17
3.2.2	Windows	18
3.2.3	ORNL’s Jaguar XT4/5	18
3.3	Building PFLOTRAN	19
3.4	Updating PFLOTRAN	19
3.5	Parallel I/O using Sarat Sreepathi’s SCORPIO library with PFLOTRAN	20
3.6	Running PFLOTRAN	23
3.7	Multiple Realization Simulation Mode	24
3.8	Multiple Simulation Mode	24
4	Creating the Input File: PFLOTRAN Keywords	25
4.1	Conventions and Notation	25
4.2	Input Deck Specification	25
4.2.1	Keyword: SUBSURFACE_FLOW	28
4.2.2	Keyword: SUBSURFACE_TRANSPORT	28
4.3	Subsurface Keyword Links	30
4.4	Example Input File	32

4.5	Keyword: BOUNDARY_CONDITION	37
4.6	Keyword: BRINE	39
4.7	Keyword: CHARACTERISTIC_CURVES	40
4.8	Keyword: CHECKPOINT	43
4.9	Keyword: CHEMISTRY	44
4.9.1	Keyword: MINERAL_KINETICS	47
4.9.2	Keyword: SORPTION	49
4.9.3	Keyword: RADIOACTIVE_DECAY_REACTION	52
4.9.4	Keyword: GENERAL_REACTION	54
4.9.5	Keyword: MICROBIAL_REACTION	55
4.9.6	Keyword: REACTION_SANDBOX	58
4.9.7	Keyword: OUTPUT	60
4.10	Keyword: COMPUTE_STATISTICS	63
4.11	Keyword: CO2_DATABASE	64
4.12	Keyword: CONSTRAINT	65
4.13	Keyword: DATASET	67
4.14	Keyword: DBASE_FILENAME	69
4.15	Keyword: DEBUG	71
4.16	Keyword: EOS	72
4.17	Keyword: FLOW_CONDITION	73
4.18	Keyword: FLUID_PROPERTY	77
4.19	Keyword: GEOMECHANICS	78
4.20	Keyword: GEOMECHANICS_BOUNDARY_CONDITION	79
4.21	Keyword: GEOMECHANICS_CONDITION	81
4.22	Keyword: GEOMECHANICS_GRID	83
4.23	Keyword: GEOMECHANICS_MATERIAL_PROPERTY	84
4.24	Keyword: GEOMECHANICS_OUTPUT	85
4.25	Keyword: GEOMECHANICS_REGION	86

4.26	Keyword: GEOMECHANICS_STRATA	87
4.27	Keyword: GRID Required	88
4.27.1	Implicit Unstructured Grids	89
4.27.2	Explicit Unstructured Grids	90
4.28	Keyword: IMMOBILE_DECAY_REACTION	95
4.29	Keyword: INITIAL_CONDITION	96
4.30	Keyword: INITIALIZE_FLOW_FROM_FILE	97
4.31	Keyword: INITIALIZE_TRANSPORT_FROM_FILE	97
4.32	Keyword: INTEGRAL_FLUX	98
4.33	Keyword: LINEAR_SOLVER	99
4.34	Keyword: MATERIAL_PROPERTY	102
4.35	Keyword: MODE	106
4.36	Keyword: MULTIPLE_CONTINUUM	107
4.37	Keyword: NEWTON_SOLVER	108
4.38	Keyword: NONUNIFORM_VELOCITY	110
4.39	Keyword: NUMERICAL_JACOBIAN_FLOW	112
4.40	Keyword: NUMERICAL_JACOBIAN_RXN	113
4.41	Keyword: NUMERICAL_JACOBIAN_MULTI_COUPLE	114
4.42	Keyword: OBSERVATION	115
4.43	Keyword: ORIGIN (ORIG)	117
4.44	Keyword: OUTPUT	118
4.45	Keyword: OVERWRITE_RESTART_FLOW_PARAMS	123
4.46	Keyword: OVERWRITE_RESTART_TRANSPORT	123
4.47	Keyword: OVERWRITE_RESTART_TRANSPORT	123
4.48	Keyword: PROC	124
4.49	Keyword: REGION	125
4.50	Keyword: RESTART	127
4.51	Keyword: SATURATION_FUNCTION	128

4.52	Keyword: SOURCE_SINK	130
4.53	Keyword: STRATIGRAPHY (STRATA)	131
4.54	Keyword: TIME	133
4.55	Keyword: TIMESTEP	134
4.56	Keyword: TRANSPORT_CONDITION	136
4.57	Keyword: UNIFORM_VELOCITY	139
	Optional	
4.58	Keyword: USE_TOUCH_OPTIONS	140
4.59	Keyword: VELOCITY_DATASET	141
4.60	Keyword: WALLCLOCK_STOP	142
5	PFLOTTRAN Regression Test Manager	143
5.1	Running the Test Manager	143
5.2	Test output	144
5.3	Configuration Files	145
5.4	Creating New Tests	146
5.5	Updating Test Results	147
6	Visualization	148
6.1	Tecplot	148
6.2	VisIt	148
6.3	Gnuplot, Matplotlib	149
7	Benchmark Problems	150
7.1	Ion Exchange	150
7.2	GENERAL_REACTION Example	157
7.2.1	Problem Description	157
7.2.2	Simulation Results	157
7.2.3	PFLOTTRAN Input File	158
7.3	RICHARDS Mode with Tracer: SX-115 Hanford Tank Farm	161
7.3.1	Problem Description	161

7.3.2	Governing Equations	161
7.3.2.1	Semi-Analytical Solution for Steady-State Conditions	162
7.3.2.2	Watertable	162
7.3.3	Model Parameters	162
7.3.4	Simulation Results	163
7.3.5	PFLOTTRAN Input File	166
7.4	MPHASE	174
7.4.1	CO ₂ Sequestration: 1D Example Problem and Comparison with TOUGHREACT	174
7.4.2	CO ₂ Sequestration in the Presence of a Leaky Well	178
8	References	185
	Appendix A: Governing Equations	187
A-1	Mode: RICHARDS	187
A-2	Capillary Pressure Relations	187
A-2.1	Brooks-Corey Saturation Function	188
A-2.2	Relative Permeability	188
A-2.3	Smoothing	189
A-3	Mode: MPHASE	189
A-3.1	Source/Sink Terms	190
A-3.2	Variable Switching	191
A-3.2.1	Gas: $(p_g, T, x_{\text{CO}_2}^g) \rightarrow$ Two-Phase: (p_g, T, s_g)	191
A-3.2.2	Liquid: $(p_l, T, x_{\text{CO}_2}^l) \rightarrow$ Two-phase: (p_g, T, s_g)	192
A-3.2.3	Two-Phase: $(p_g, T, s_g) \rightarrow$ Liquid $(p_l, T, x_{\text{CO}_2}^l)$ or Gas $(p_g, T, x_{\text{CO}_2}^g)$	192
A-3.3	Sequentially Coupling MPHASE and CHEMISTRY	193
A-4	Mode: IMMIS	193
A-5	Mode: MISCIBLE	194
A-6	Mode: Air-Water	196

A-7	Mode: TH (Thermal-Hydrologic)	196
A-7.1	Ice Model	197
A-8	Thermal Conduction Multiple Continuum Model	199
A-9	Mode: Reactive Transport (Keyword CHEMISTRY)	203
A-9.1	Mineral Precipitation and Dissolution	205
A-9.1.1	Rate Limiter	206
A-9.1.2	Changes in Material Properties	206
A-9.1.3	Affinity Threshold	207
A-9.1.4	Surface Armoring	207
A-9.2	Sorption	208
A-9.2.1	Ion Exchange	208
A-9.2.2	Surface Complexation	210
A-9.2.3	Multirate Sorption	211
A-9.3	Sorption Isotherm <Under Revision>	212
A-9.4	Colloid-Facilitated Transport	213
A-9.5	Microbial Reactions	214
A-10	Tracer Mean Age	215
A-11	Thermodynamic Database	216
A-12	Eh, pe	218
A-13	Geomechanics	219
Appendix B: Method of Solution		220
B-1	Integrated Finite Volume Discretization	220
B-2	Fully Implicit Newton-Raphson Iteration with Linear and Logarithm Update	221
B-2.1	Example	221
B-2.2	Multirate Sorption	222
B-3	Operator Splitting	223
B-3.1	Constant K_d	223

Appendix C: Reaction Sandbox	225
C-1 Background	225
C-2 Implementation	225

LIST OF FIGURES

4-1	Side set numbering of supported cell types within unstructured grid	89
4-2	(a) Example of a $5 \times 4 \times 3$ unstructured grid problem with default decomposition across two processors. Cells labelled in natural ID. (b) ParMETIS decomposition of the domain; (c) Representation of a global PETSc vector	91
4-3	(a-b) Local and ghost cells on proc-0 and proc-1 with cell IDs in PETSc order. (c-d) Local PETSc Vector with cell IDs in local ghosted order.	92
7-4	Breakthrough curves for Ca^{2+} , Mg^{2+} and Na^{+} compared with experimental results from Voegelin et al. (2000).	151
7-5	Comparison of concentrations for species A, B, C plotted as a function of distance for an elapsed time of 5 years for PFLOTRAN and a prototype code based on PETSc's TS class.	158
7-6	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.	164
7-7	Simulation of a tank leak with a duration of two weeks showing the saturation profile for different times indicated in the figure for an infiltration rate of 8 mm/y.	165
7-8	The solute concentration profile corresponding to Figure 7-8 for different times indicated in the figure for an infiltration rate of 8 mm/y.	165
7-9	Comparison with TOUGHREACT (dashed curves) and PFLOTRAN (solid curves) after an elapsed time of 0.4 y corresponding to the end of injection. Reasonable agreement is obtained between the two codes.	176
7-10	Liquid (blue curve) and supercritical CO_2 (red curve) pressures predicted by PFLOTRAN after an elapsed time of 0.4 y corresponding to the end of injection. Also shown is the CO_2 saturation (green curve).	177
7-11	Left: pressure; right: supercritical CO_2 saturation; for an elapsed time of 250 days.	179
7-12	Leakage rate relative to injection rate.	179
A-1	Node configuration for secondary (β) continua (nodes $1, \dots, l, \dots, M$), associated with the n th primary (α) continuum node.	201
A-2	Control volumes in DCDM multiple continuum model with fracture aperture 2δ and matrix block size d	201

LIST OF TABLES

2-1	Options for configuring petsc-dev.	13
2-2	Compile options for PFLOTTRAN.	14
2-3	Available modes in PFLOTTRAN	14
4-4	Units	25
4-5	SUBSURFACE Keywords listed in alphabetical order.	30
4-6	Side set node ordering	90
7-7	Stratigraphic sequence used in the calculations, after Ward et al. (1996).	162
7-8	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).	163
7-9	Problem definition and parameters used in the 1D CO ₂ sequestration example. . . .	174
7-10	Initial concentration of primary and secondary species. Mineral reactions are not considered.	175
7-11	Model parameters.	178
7-12	Input file for 3D CO ₂ sequestration example problem with a leaky well.	180
A-1	Choice of primary variables.	191
A-2	DCDM geometric parameters.	202
A-3	Independent and dependent nested cube parameters.	203
A-4	Definition of symbols used in modeling microbial reactions.	215
A-5	Format of thermodynamic database.	217
A-6	Fit coefficients for log K of reaction A-212.	219

1 Introduction

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

PFLOTTRAN 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, for example, gcc 4.7.x or Intel 13.x.x or later. As a requirement of running problems with a large number of degrees of freedom, PFLOTTRAN allows reading input data that would be too large to fit into memory if allocated to a single processor core. The current limitation to the problem size PFLOTTRAN 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 several billion degrees of freedom for the maximum problem size that can be currently run with PFLOTTRAN. Hopefully this limitation will be remedied in the near future.

Currently PFLOTTRAN can handle a number of subsurface processes involving flow and transport in porous media including Richards equation, two-phase flow involving supercritical CO_2 , and multicomponent reactive transport including aqueous complexing, sorption and mineral precipitation and dissolution. Reactive transport equations are solved using a fully implicit Newton-Raphson algorithm. Operator splitting is currently not implemented. In addition to single continuum processes, 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. Finally, an elastic geomechanical model is implemented.

A novel feature of the code is its ability to run multiple input files and multiple realizations simultaneously, for example with different permeability and porosity fields, 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 information can be found on the PFLOTTRAN [wiki home page](#) which should be consulted for the most up-to-date information. Questions regarding installing PFLOTTRAN on workstations, small clusters, and super computers, and bug reports, may be directed to: `pflotran-dev` at `googlegroups dot com`. For questions regarding running PFLOTTRAN 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 PFLOTTRAN in four easy steps. On the MacOSX platform Xcode must be installed with the commandline tools.

Required Software Packages

- Compilers (compatible with Fortran 2003: gcc 4.7.x, Intel 14.x.x)
For MacOSX 10.10.x the latest Fortran compiler binaries (gcc/gfortran 4.9 at the time of this writing) may be downloaded from the web site [HPC for MacOSX](#).
- Mercurial (hg) (version control system)
- CMAKE (version 3.0.x, needed for installing METIS and ParMETIS)
- Developer version of PETSc: [petsc-dev](#)
 - MPI (for running in parallel)
 - BLAS/LAPACK libraries
 - HDF5 (parallel output)
 - METIS/ParMETIS (unstructured grids)
- [PFLOTTRAN](#)

2.1 Installing PFLOTTRAN

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 10.10.x and gcc-4.9 enter into the `.tcshrc` file:

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

and

```
setenv PETSC_ARCH MacOSX-10.10.x-gcc4.9.
```

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 PFLOTTRAN: structured grids are the simplest; if unstructured grids are needed then install METIS and ParMETIS. See Table [2-1](#) 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 2-1: Options for configuring `petsc-dev`.

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

* 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](https://bitbucket.org/pflotran/pflotran-dev) 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 2-2](#).

Table 2-2: 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 PLOTTRAN_DIR/src/pflotran
make [options] ./pflotran
```

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

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

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

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

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

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

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

The final challenge is setting up an input file. Various modes available are listed in Table 2-3 and example input decks are stored in the `./pflotran/example_problems` and `./pflotran/shortcourse` directories.

Table 2-3: Available modes in PFLOTRAN

MODE	Description	Required Databases [†]
RICHARDS	Solves Richards equation	—
MPHASE	Supercritical 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 §A-11 for a description of the database format).

An example input file is listed in §4.4 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: e.g.

```
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](https://bitbucket.org/pflotran/pflotran-dev) 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 so 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 (Mac OS X 10.10.x) Special considerations are required to install PETSc and PFLOTRAN on Yosemite (MacOSX 10.10.x). 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 PFLOTTRAN on Windows see instructions on the PFLOTTRAN 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
<code>scorpio=1:</code>	<code>--enable parallel IO</code>
<code>coll=1:</code>	<code>--enable colloid-facilitated transport option</code>
<code>mfd=1:</code>	<code>--enable MFD full permeability tensor capability (not operational)</code>

Thus for compiling with the supercritical CO₂ 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 PFLOTTRAN:

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

```
cd PFLOTTRAN_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 distributed 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 (gmkmkumar@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 file prefix and path [default: pflotran]
-output_prefix <string>	specify output file prefix and path to prefix directory which must already exist [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, basic-nonorms]
-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	

Examples:

```
mpirun -np 1 ./pflotran -output_prefix ./path_to/prefix
```

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 ($\text{num_realizations}/\text{num_groups}$), 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`.

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 4-4.

Table 4-4: 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 <code>MOLAL</code> 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 Input Deck Specification

PFLOTRAN input files are divided into blocks based on the process models employed. Process models are defined within the `SIMULATION` block which for convenience should be located at

the top of the input file (but this is not required). Keywords for subsurface processes are listed in Table 4-5. At present manual pages and keywords for surface processes is still under development. Keywords within the SIMULATION block include:

SIMULATION

Opens the SIMULATION block.

SIMULATION_TYPE <string>

Defines the domain to be modeled and the types of process models employed. Options include:

SUBSURFACE

SURFACE_SUBSURFACE

HYDROGEOPHYSICS

PROCESS_MODELS

Opens a block for defining all process models employed. At this point, the ordering and linkage of these process models is preset. For instance, SUBSURFACE_FLOW processes always precede SUBSURFACE_TRANSPORT processes in timestep execution. Options include:

SUBSURFACE_FLOW <string>

GENERAL, RICHARDS, MPHASE

END [/]

SUBSURFACE_TRANSPORT <string>

GLOBAL_IMPLICIT

END [/]

SURFACE_FLOW <string>

...

END [/]

END [/]

SUBSURFACE

Opens a block of input parameters for subsurface process models (this block is essentially, the old PFLOTTRAN input file with the exception that several of the cards (i.e. flow MODE) have been moved to the SIMULATION block.

END_SUBSURFACE

Closes the subsurface block.

Example:

```
SIMULATION
  SIMULATION_TYPE SUBSURFACE
  PROCESS_MODELS
    SUBSURFACE_FLOW flow
    RICHARDS
  /
  SUBSURFACE_TRANSPORT transport
  GLOBAL_IMPLICIT
/
END

SUBSURFACE

.
.
.

END_SUBSURFACE
```

4.2.1 Keyword: SUBSURFACE_FLOW

SUBSURFACE_FLOW Block

Required Cards

MODE <string>: Specifies the flow mode to be employed.

Water Modes:

RICHARDS: single-phase variably-saturated flow

GENERAL: two-phase air-water-energy

TH: variably saturated flow and energy (and optional ice phase)

CO₂-Related Modes:

MPHASE: two-phase supercritical CO₂-water-energy

IMMISCIBLE: immiscible two-phase CO₂-water-energy

MISCIBLE: miscible H₂O-glycol

FLASH2: two-phase supercritical CO₂-water-energy

Optional Cards

OPTIONS: MODE-dependent block for defining options for each flow process model.

Example:

```

SIMULATION
  SIMULATION_TYPE SUBSURFACE
  PROCESS_MODELS
    SUBSURFACE_FLOW flow
    MODE GENERAL
    OPTIONS
      ISOTHERMAL
      TWO_PHASE_ENERGY_DOF TEMPERATURE
      GAS_COMPONENT_FORMULA_WEIGHT 2.01588D0
      MAXIMUM_PRESSURE_CHANGE 1.0D6
  /
/
END

```

4.2.2 Keyword: SUBSURFACE_TRANSPORT

SUBSURFACE_TRANSPORT Block

Optional Cards

GLOBAL_IMPLICIT Specifies fully implicit coupling of transport and reaction.

OPERATOR_SPLITTING (not currently implemented)

Example:

```
SIMULATION
  SIMULATION_TYPE SUBSURFACE
  PROCESS_MODELS
    SUBSURFACE_TRANSPORT transport
    GLOBAL_IMPLICIT
  /
/
END
```

4.3 Subsurface Keyword Links

Table 4-5: **SUBSURFACE** Keywords listed in alphabetical order.

Keyword	Description
BOUNDARY_CONDITION	
BRINE	
CHARCTERISTIC_CURVES	
CHECKPOINT	
CHEMISTRY	
MINERAL_KINETICS	
SORPTION	
RADIOACTIVE_DECAY_REACTION	
GENERAL_REACTION	
MICROBIAL_REACTION	
REACTION_SANDBOX	
OUTPUT	
COMPUTE_STATISTICS	
CO2_DATABASE	
CONSTRAINT	Transport (required)
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)

IMMOBILE_DECAY_REACTION
INITIAL_CONDITION
INITIALIZE_FLOW_FROM_FILE
INITIALIZE_TRANSPORT_FROM_FILE
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

4.4 Example Input File

#Description: 3D infiltration problem using Richards model with calcite dissolution
 #(# or ! denotes following text a comment)

SIMULATION

```
SIMULATION_TYPE SUBSURFACE
PROCESS_MODELS
SUBSURFACE_FLOW flow
MODE RICHARDS
/
SUBSURFACE_TRANSPORT transport
GLOBAL_IMPLICIT
/
/
END
```

SUBSURFACE ! SUBSURFACE block ends with END_SUBSURFACE

== debugging =====

DEBUG

```
# MATVIEW_JACOBIAN
# VECVIEW_RESIDUAL
# VECVIEW_SOLUTION
# /
```

== 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/m2/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
--------	-------	---

/

/

END_SUBSURFACE

4.5 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.

[↔ Return to List of Keywords](#)

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](#)

4.6 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^- .

[↪ Return to List of Keywords](#)

Example:

```
BRINE 4.d0 MOLAL
```

[↪ Return to List of Keywords](#)

4.7 Keyword: CHARACTERISTIC_CURVES

Description: Specifies relative permeability and saturation functions and parameters to be associated with a material property. **Note: this keyword is currently only supported for GENERAL and RICHARDS modes; keyword SATURATION_FUNCTION should be used in all other flow modes.**

[↔ Return to List of Keywords](#)

Input:

CHARACTERISTIC_CURVES <string>

SATURATION_FUNCTION <string> [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 ($m = 1 - 1/n$) [—].
- 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](#)

4.8 Keyword: CHECKPOINT

[↪ Return to List of Keywords](#)

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 PFLOTTRAN 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](#)

4.9 Keyword: CHEMISTRY

[↪ Return to List of Keywords](#)

Description: Specifies geochemistry for multicomponent transport. The **CHEMISTRY** keyword invokes the reactive transport mode coupling a reaction network consisting of aqueous species, minerals and gases to advection, diffusion and dispersion. This keyword provides input for primary species, secondary species implemented through local equilibrium mass action relations, minerals, gases, colloids. Homogeneous aqueous complexing reactions and heterogeneous mineral precipitation and dissolution, sorption including ion exchange and surface complexation, radioactive decay, microbial reactions are provided. Homogeneous complexing reactions may be described through local equilibrium by specifying secondary species or through a kinetic formulation using the **GENERAL_REACTION** keyword. Mineral precipitation and dissolution reactions are described through a kinetic rate law based on transition state theory. Surface complexation reactions may involve equilibrium, kinetic (reversible or irreversible) or a multirate formulation. Colloid-facilitated transport taking into account sorption on colloids is included.

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 [/]

IMMOBILE_SPECIES

Species Name
END [/]

IMMOBILE_DECAY_REACTION

SPECIES_NAME <string>
RATE_CONSTANT <float>
END [/]

COLLOIDS

Colloid Name Mobile_Fraction [—]
END [/]

Example: Specification of primary species, secondary species, gases and minerals.

```
#===== chemistry =====
CHEMISTRY

PRIMARY_SPECIES
Na+
K+
Ca++
Mg++
Al+++
H+
CO2 (aq)
SiO2 (aq)
Cl-
/

SECONDARY_SPECIES
OH-
CO3--
HCO3-

CaCO3 (aq)
CaHCO3+
CaCl2 (aq)
CaCl+
# CaOH+

MgCO3 (aq)
MgHCO3+
MgCl+
MgOH+

Al (OH) 4-
Al (OH) 3 (aq)
Al (OH) 2+
```

```
AlOH++

H3SiO4-
H2SiO4--

HCl(aq)
KCl(aq)
# KOH(aq)

NaCl(aq)
NaCO3-
# NaHCO3(aq)
NaHSiO3(aq)
NaOH(aq)
/

GAS_SPECIES
CO2(g)
O2(g)
/

MINERALS
Quartz
K-Feldspar
Kaolinite
Gibbsite
Muscovite
Oligoclase
Illite
Albite
Anorthite
Brucite
Magnesite
Calcite
/
```

Example: Deactivate redox equilibrium reactions for the listed redox pairs.

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

[↔ Return to List of Keywords](#)

4.9.1 Keyword: MINERAL_KINETICS

The keyword MINERAL_KINETICS specifies parameters for kinetic mineral precipitation-dissolution reactions based on transition state theory.

[↪ Return to List of Keywords](#)

MINERAL_KINETICS

Mineral Name <Char>

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

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

AFFINITY_THRESHOLD <float> [—]

AFFINITY_POWER <float> [—] [see Eqn.(A-114)]

TEMKIN_CONSTANT <float> [—] [See Eqn.(A-114)]

SURFACE_AREA_POROSITY_POWER <float> [—] [see Eqn.(A-128)]

SURFACE_AREA_VOL_FRAC_POWER <float> [—] [see Eqn.(A-128)]

RATE_LIMITER <float> [mol/m²/s] [see Eqn.(A-118)]

IRREVERSIBLE

ARMOR_MINERAL <Char> [See Eqn.(A-136)]

ARMOR_PWR <float> [—] [See Eqn.(A-136)]

ARMOR_CRIT_VOL_FRAC <float> [—] [See Eqn.(A-136)]

PREFACTOR

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

ACTIVATION_ENERGY <float> [kJ/mol]

PREFACTOR_SPECIES <Char>

ALPHA <float> [—]

BETA <float> [—]

ATTENUATION_COEF <float>

END [/]

END [/]

END [/]

END [/]

Example: Mineral kinetics.

```

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

```

Examples: Using PREFACTOR for mineral TST kinetics [see Eqn.(A-117)].

```

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](#)

4.9.2 Keyword: SORPTION

The keyword SORPTION specifies parameters for ion-exchange and surface-complexation reactions.

[↪ Return to List of Keywords](#)

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/m³]

COMPLEXES

Complex Name

END [/]

COMPLEX_KINETICS

COMPLEX name

FORWARD_RATE_CONSTANT <float> [1/s]

BACKWARD_RATE_CONSTANT <float> [1/s] If value < -999
calculate backward rate constant from expression: $k_b = K_{eq}k_f$,
where K_{eq} is the corresponding equilibrium constant of the reaction.

END [/]

END [/]

COLLOID Name

SITE Name Site Density [mol/m³]

COMPLEXES

Surface_Complex Name

END [/]

END [/]

END [/]

ION_EXCHANGE_RXN**MINERAL** Mineral Name**CEC** <float> [mol/m³]**CATIONS** Exchange reactions must be written in terms of a single reference cation which must appear first in the list of cations [see (A-140)].

Cation Name Selectivity_Coefficient

END [/]**END** [/]**END** [/]**ISOTHERM_REACTIONS****Species_Name****TYPE** LINEAR, LANGMUIR, FREUNDLICH**DISTRIBUTION_COEF, KD** <float> [kg water/m³ bulk] [see Eqn.(A-182)]**LANGMUIR_B** <float> [—] [see Eqn.(A-183)]**FREUNDLICH_N** <float> [—] [see Eqn.(A-184)]**END** [/]**END** [/]**JUMPSTART_KINETIC_SORPTION****NO_CHECKPOINT_KINETIC_SORPTION****NO_RESTART_KINETIC_SORPTION****END** [/]**Example:** Linear sorption distribution coefficient and mean age of a tracer.

```

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
    /
  /
/

```

Example: Ion exchange.

```

SORPTION
  ION_EXCHANGE_RXN
    CEC 0.11 ! eq/m3bulk
  CATIONS
    Na+ 1.
    Ca++ 0.158489
    K+ 0.199526
  /
/
```

4.9.3 Keyword: RADIOACTIVE_DECAY_REACTION

Specifies parameters for radioactive decay reaction. This reaction differs from the GENERAL_REACTION in that only one reactant species may be specified with a unit stoichiometry (i.e. the rate is always first order) and the reactant species is decayed in both the aqueous and sorbed phases.

[↔ Return to List of Keywords](#)

RADIOACTIVE_DECAY_REACTION

REACTION <string> Reaction equation. Only one reactant species may be listed on the left side of the equation (i.e. or on the right side with a negative stoichiometry). The reactant's stoichiometry is fixed at 1.0. The forward rate is applied to that one species as a first order rate constant [1/sec]. Multiple species are supported as daughter products on the right hand side and stoichiometries can be specified.

RATE_CONSTANT or HALF_LIFE (but not both)

RATE_CONSTANT <float> rate constant for 1st-order decay reaction [1/sec]. The rate constant may be calculated from $\ln(2) / \text{half-life}$.

HALF_LIFE <float> half life of species [sec]. Note that rate constant or half life units other than sec or 1/sec may be specified.

END [/]

Examples:

```
RADIOACTIVE_DECAY_REACTION
  REACTION Tracer <-> Tracer2
  RATE_CONSTANT 1.7584d-7 ! half life at 0.125 y
/
-----
CHEMISTRY
  PRIMARY_SPECIES
    A(aq)
    B(aq)
    C(aq)
  /
  ...
RADIOACTIVE_DECAY_REACTION
  REACTION A(aq) <-> B(aq)
  ! Calculating forward rate from half-life
  ! rate = -ln(0.5) / half-life [1/sec]
  RATE_CONSTANT 1.75836d-9 ! 1/s half life = 12.5 yrs
/
RADIOACTIVE_DECAY_REACTION
  REACTION B(aq) <-> C(aq)
  RATE_CONSTANT 8.7918d-10 ! 1/s half life = 25. yrs
/
RADIOACTIVE_DECAY_REACTION
  ! Note that C(aq) decays with no daughter products
  REACTION C(aq) <->
  HALF_LIFE 5. y
```

<div data-bbox="280 123 358 218" style="text-align: center;">/ ... /</div>
--

[↩ Return to List of Keywords](#)

4.9.4 Keyword: GENERAL_REACTION

Specifies rate constants and stoichiometry for a general kinetic homogeneous aqueous forward/backward reaction. It does not apply to reactions containing a solid phase, however.

[↪ Return to List of Keywords](#)

GENERAL_REACTION

REACTION <string> Reaction equation. The forward rate is applied to reaction quotient of species on left side of reaction. The reverse or backward rate is applied to the right side. Reactions can also be irreversible if one of the rate constants is zero.

FORWARD_RATE <float> rate constant for n^{th} -order forward reaction
[kg⁽ⁿ⁻¹⁾/mol⁽ⁿ⁻¹⁾-sec]

BACKWARD_RATE <float> rate constant for n^{th} -order reverse reaction
[kg⁽ⁿ⁻¹⁾/mol⁽ⁿ⁻¹⁾-sec]

END [/]

Examples:

```
GENERAL_REACTION
  REACTION Tracer <-> Tracer2
  FORWARD_RATE 1.7584d-7 ! half life at 0.125 y
  BACKWARD_RATE 0.d0
/
CHEMISTRY
  PRIMARY_SPECIES
    A(aq)
    B(aq)
    C(aq)
  /
  ...
  GENERAL_REACTION
    REACTION A(aq) <-> B(aq)
    ! Calculating forward rate from half-life
    ! forward rate = -ln(0.5) / half-life [1/sec]
    FORWARD_RATE 1.75836d-9 ! 1/s half life = 12.5 yrs
    BACKWARD_RATE 0.d0
  /
  GENERAL_REACTION
    REACTION B(aq) <-> C(aq)
    FORWARD_RATE 8.7918d-10 ! 1/s half life = 25. yrs
    BACKWARD_RATE 0.d0
  /
  GENERAL_REACTION
    ! Note that C(aq) decays with no daughter products
    REACTION C(aq) <->
    FORWARD_RATE 4.3959d-9 ! 1/s half life = 5. yrs
    BACKWARD_RATE 0.d0
  /
  ...
/
```

[↪ Return to List of Keywords](#)

4.9.5 Keyword: MICROBIAL_REACTION

Specifies parameters for microbially-mediated reactions.

[↪ Return to List of Keywords](#)

MICROBIAL_REACTION

REACTION <string> Reaction equation. The rate constant is multiplied by the Monod expressions for electron donor and acceptor for select species on the left side of the equation. The reaction may be inhibited by any species in the system.

RATE_CONSTANT <float> Rate constant for the reaction.

MONOD <string> <float> Specifies the Monod equation for the electron donor and acceptor (named by the string) and the half saturation constant.

INHIBITION Specifies inhibition based on species concentration and an inhibition constant(s). Three types of inhibition are currently supported: MONOD, INVERSE_MONOD, and THRESHOLD.

Monod Inhibition:

INHIBITION

SPECIES_NAME <string>

TYPE MONOD Specifies the type of inhibition to be Monod. The reaction proceeds as long as the species concentration is well below the half saturation constant: $\text{inhibition} = C_{th} / (C_{th} + \text{concentration})$

INHIBITION_CONSTANT <float> Half saturation constant.

Inverse Monod Inhibition:

INHIBITION

SPECIES_NAME <string>

TYPE INVERSE_MONOD Specifies the type of inhibition to be inverse Monod. The reaction proceeds as long as the species concentration is well above the half saturation constant: $\text{inhibition} = \text{concentration} / (C_{th} + \text{concentration})$

INHIBITION_CONSTANT <float> Half saturation constant.

Threshold Inhibition:

INHIBITION

SPECIES_NAME <string>

TYPE THRESHOLD <float> Specifies the type of inhibition to be threshold and the scaling factor to be applied. Inhibition is calculated based on the following equation: $\text{inhibition} = 0.5 + \arctan((\text{concentration} - C_{th}) \times f) / \pi$. Inhibition is above and below C_{th} when the sign of C_{th} is negative or positive, respectively.

INHIBITION_CONSTANT <float> Threshold concentration

END [/]

BIOMASS

SPECIES_NAME <string> Specifies biomass species.

YIELD <float> Biomass yield.

END [/]

END [/]

Examples:

```
CHEMISTRY
  PRIMARY_SPECIES
    A(aq)
    B(aq)
    C(aq)
  /
  MICROBIAL_REACTION
    REACTION A(aq) + 2 B(aq) <-> 1.5 C(aq)
    RATE_CONSTANT 1.d-12
    MONOD A(aq) 1.d-5      ! A is the donor
    MONOD B(aq) 1.d-4      ! B is the acceptor
    INHIBITION
      SPECIES_NAME C(aq)
      TYPE MONOD
      INHIBITION_CONSTANT 6.d-4 ! C is the product and
                                ! inhibits when too high
  /
/
...
```

```
CHEMISTRY
  PRIMARY_SPECIES
    A(aq)
    B(aq)
    C(aq)
  /
  IMMOBILE_SPECIES
    Microbe_D
  /
  MICROBIAL_REACTION
    REACTION A(aq) + 2 B(aq) <-> 1.5 C(aq)
    RATE_CONSTANT 1.d-12
    MONOD A(aq) 1.d-5      ! A is the donor
    MONOD B(aq) 1.d-4      ! B is the acceptor
    INHIBITION C(aq) 6.d-4 ! C is the product and inhibits when too high
    BIOMASS Microbe_D 1.d-1
  /
  ...
END

CONSTRAINT biomass
  CONCENTRATIONS
    A(aq) 1.d-7 T
    B(aq) 1.d-7 T
    C(aq) 1.d-7 T
```



```
    /  
    IMMOBILE  
      Microbe_D 1.d0  
    /  
  END
```

[↩ Return to List of Keywords](#)

4.9.6 Keyword: REACTION_SANDBOX

Specifies parameters for user-defined reactions.

[↩ Return to List of Keywords](#)

REACTION_SANDBOX Opens the reaction sandbox block. Terminate with END, ., /.

CLM-CN Block for specifying CLM-CN reaction parameters

Examples:

```
REACTION_SANDBOX
  CLM-CN
    POOLS
      SOM1 12.d0
      Lit1
    /
  REACTION
    UPSTREAM_POOL Lit1
    DOWNSTREAM_POOL SOM1
    TURNOVER_TIME 20. h
    RESPIRATION_FRACTION 0.39d0
    N_INHIBITION 1.d-10
  /
/
```

[↩ Return to List of Keywords](#)

OPERATOR_SPLITTING Toggles operator-splitting mode (Default implicit, **not currently implemented**)

GEO_THERMAL_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.(A-123)]

UPDATE_TORTUOSITY [see Eqn.(A-127)]

UPDATE_PERMEABILITY (Must activate update_porosity, see Eqn.(A-124).)

UPDATE_MINERAL_SURFACE_AREA [see Eqn.(A-128)]

(Must set SURFACE_AREA_VOL_FRAC_POWER)

UPDATE_MNRL_SURF_AREA_WITH_POR [see Eqn.(A-128)]

(Must set SURFACE_AREA_POROSITY_POWER)

MINIMUM_POROSITY <float>

UPDATE_ARMOR_MINERAL_SURFACE [See Eqn.(A-136)]

MAX_DLNC (Default 5)

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

MAX_RESIDUAL_TOLERANCE Speciation tolerance (Default 1.d-12)

4.9.7 Keyword: OUTPUT[↪ Return to List of Keywords](#)**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]

O2Output $\log f_{\text{O}_2}$ [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 [/]****END [/]**

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

```

OUTPUT
...
ALL
FREE_ION
TOTAL
...
/

```

[↪ Return to List of Keywords](#)

Keyword	Description
Primary_Species	List of primary species that fully describe the chemical composition of the fluid terminated by END, ., or /. 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 terminated by END, ., or /.
Gas_Species	List of gas species terminated by END, ., or /.
Minerals	List of minerals, both passive and active, terminated by END, ., or /. Active minerals must be associated with a kinetic rate law under MINERAL_KINETICS and listed under CONSTRAINTS. For passive mineral only the mineral saturation index is calculated for initial and boundary conditions.
Sorption	Surface complexation, ion exchange and specified isotherm.
Output	To print secondary aqueous complex concentrations, either add the individual names of the secondary species of interest or add the keyword “SECONDARY_SPECIES” for all secondary species to the CHEMISTRY OUTPUT card:

Example:

```

CHEMISTRY
...
OUTPUT
...

```

```
CO2(aq)    ! where co2(aq) is a secondary species
SECONDARY_SPECIES
...
/
...
END
```

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. Saturation indices are not printed when only ALL or MINERALS are specified. Outputting tens to hundreds of minerals, most of which are simply considered as potential candidates for active minerals, but not modeled through a kinetic precipitation-dissolution reaction, would overwhelm the output.

...

[↔ Return to List of Keywords](#)

4.10 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.

[↪ Return to List of Keywords](#)

Input:

COMPUTE_STATISTICS {compute_statistics = .true.}

Explanation:

Example:

```
COMPUTE_STATISTICS
```

[↪ Return to List of Keywords](#)

4.11 Keyword: CO2_DATABASE

[↪ Return to List of Keywords](#)

Description: The keyword CO2_DATABASE is for specifying the path to CO₂ database file providing a lookup table for CO₂ fluid properties density, viscosity, fugacity and fugacity coefficient derived from the Span-Wagner (1996) EOS. The temperature-pressure range in the default lookup table is $T : 0 - 375^{\circ}\text{C}$ 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](#)

4.12 Keyword: CONSTRAINT

[↪ Return to List of Keywords](#)

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](#)

4.13 Keyword: DATASET

[↪ Return to List of Keywords](#)

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 PFLOTTRAN 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, PFLOTTRAN 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](#)

4.14 Keyword: DBASE_FILENAME

[↪ Return to List of Keywords](#)

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.

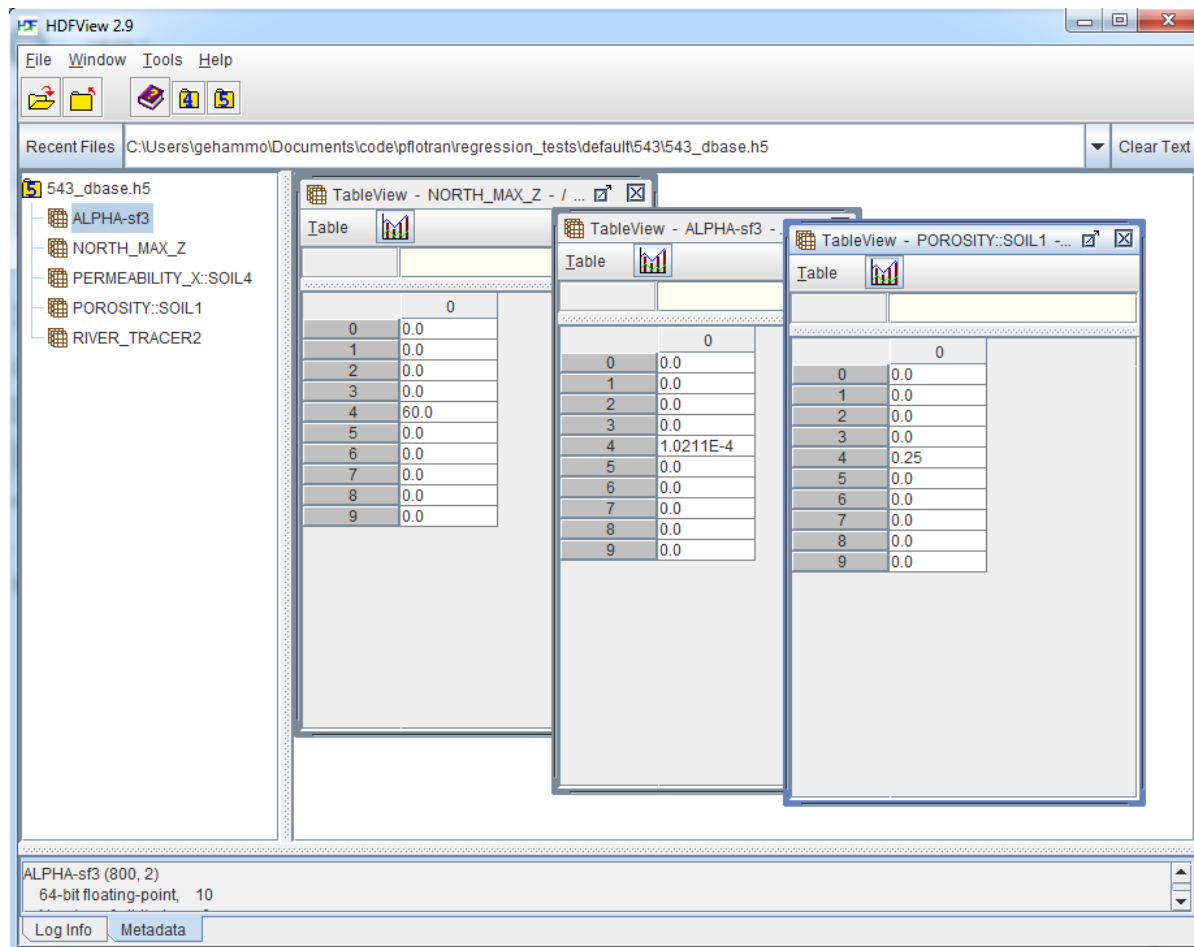
DBASE_VALUE <string1> [: <string2>] The names of the parameters to be read from the database file. This card combination may be entered anywhere a double precision value is read from the input file. The second string <string2> must be preceded by a double colon without spaces and is optional. See example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

tually the 5th realization). An example script for creating the database file may be found in `PFLOTRAN_DIR/src/python/dbase_creator.py`.



↪ [Return to List of Keywords](#)

4.15 Keyword: DEBUG

[↪ Return to List of Keywords](#)

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](#)

4.16 Keyword: EOS

[↪ Return to List of Keywords](#)

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](#)

4.17 Keyword: FLOW_CONDITION

[↪ Return to List of Keywords](#)

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 user would specify a condition with a pressure associated with a point in space (i.e. datum) in space and a gradient, both vector quantities. Note that in the case of a hydrostatic boundary condition, the vertical gradient specified in the input deck must be zero in order to enable the hydrostatic pressure calculation. Otherwise, the specified vertical gradient overrides the hydrostatic pressure. Transient pressures, temperatures, concentrations, datums, gradients, etc. are specified using the **FILE** filename combination for the name of the data set.

Input:

FLOW_CONDITION flow_condition_name

UNITS <char> (not currently supported)

<char> is one of the following entries:

s, min, h (hr), d, day, w, week, mo, month, y (yr) (time)

mm, cm, m, dm, km (length)

kg/s, kg/yr (rate)

Pa, KPa (pressure)

m/s, m/yr (velocity)

C, K (temperature)

M, mol/L (concentration)

KJ/mol (enthalpy)

CYCLIC

INTERPOLATION

step

linear

SYNC_TIMESTEP_WITH_UPDATE

TYPE

PRESSURE [dirichlet, hydrostatic, zero_gradient, conductance, seepage]

RATE [mass_rate, volumetric_rate, scaled_volumetric_rate]: specifies an injection/extraction rate in mass [kg/s], volume [m³/s], and a volumetric injection/extraction rate [m³/s] that is scaled across a well screen, weighted as a function of the interfacial area and permeability of neighboring cells (in x , y).

FLUX [dirichlet, neumann, mass_rate, hydrostatic, conductance, zero_gradient, production_well, seepage, volumetric, volumetric_rate, equilibrium]

```

    TEMPERATURE [dirichlet, hydrostatic, zero_gradient]
    CONCENTRATION [dirichlet, hydrostatic, zero_gradient]
    SATURATION [dirichlet]
    ENTHALPY (H) [dirichlet, hydrostatic, zero_gradient]
END [/]
TIME (not currently supported)
IPHASE <int>
DATUM
    x    y    z
    FILE file_name
GRADIENT, GRAD
    PRESSURE
         $d_{dx} \ d_{dy} \ d_{dz}$ 
        FILE file_name
    FLUX
    TEMPERATURE
    CONCENTRATION
    H, ENTHALPY
END [/]
TEMPERATURE, TEMP <float>
ENTHALPY <float>
PRESSURE <float>
RATE <float>
FLUX, VELOCITY, VEL <float>
CONCENTRATION <float>
SATURATION <float>
CONDUCTANCE <float>
END [/]

```

Explanation:

IPHASE	Phases present	CO ₂ concentration
1	H ₂ O	$X_{\text{CO}_2}^{\text{H}_2\text{O}}$
2	SC CO ₂	$X_{\text{CO}_2}^{\text{SCCO}_2}$
3	H ₂ O – SC CO ₂	$\left(X_{\text{CO}_2}^{\text{H}_2\text{O}}\right)_{\text{eq}}$

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](#)

4.18 Keyword: FLUID_PROPERTY

[↪ Return to List of Keywords](#)

Description: Assign diffusion coefficients and temperature dependence.

Input:

FLUID_PROPERTY

PHASE <name> (LIQUID_PHASE, GAS_PHASE) [Default: LIQUID_PHASE]

DIFFUSION_COEFFICIENT <float> [m²/s] [Default: 0 × 10⁻⁹ m²/s]

DIFFUSION_ACTIVATION_ENERGY <float> [kJ/mol] [Default: 0 kJ/mol]

END [/]

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

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

where D_m° is the reference diffusion coefficient at temperature $T_0 = 25^\circ\text{C}$ 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/mol
/
```

[↪ Return to List of Keywords](#)

4.19 Keyword: GEOMECHANICS

[↩ Return to List of Keywords](#)

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 [/]

4.20 Keyword: GEOMECHANICS_BOUNDARY_CONDITION

[↪ Return to List of Keywords](#)

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](#)

4.21 Keyword: GEOMECHANICS_CONDITION

[↪ Return to List of Keywords](#)

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_Y	Force in the y – direction
FORCE_Z	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](#)

4.22 Keyword: GEOMECHANICS_GRID

[↪ Return to List of Keywords](#)

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

GEOMECHANICS_GRID

Required Input Parameters:

TYPE <type> :

Grid type (unstructured)

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

Optional Input Parameters:

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

END [/]

4.23 Keyword: GEOMECHANICS_MATERIAL_PROPERTY

[↪ Return to List of Keywords](#)

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/m3]
  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](#)

4.24 Keyword: GEOMECHANICS_OUTPUT

[↩ Return to List of Keywords](#)

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.

4.25 Keyword: GEOMECHANICS_REGION

[↩ Return to List of Keywords](#)

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](#)

4.26 Keyword: GEOMECHANICS_STRATA

[↪ Return to List of Keywords](#)

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
```

4.27 Keyword: GRID**Required**[↪ Return to List of Keywords](#)

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)

Symmetry type (cartesian [default], cylindrical, spherical)

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

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

BOUNDS

<x_min, y_min, z_min>

<x_max, y_max, z_max>

END [/]

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

<dx>

<dy>

<dz>

END [/]

Optional Input Parameters:

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

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

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

FILE Read from file

MAX_CELLS_SHARING_A_VERTEX

STENCIL_WIDTH

STENCIL_TYPE

BOX

STAR

END [/]

The following two types of unstructured grid formats, implicit and explicit, are currently supported by PFLOTRAN.

4.27.1 Implicit Unstructured Grids

In this format, the grid is defined by (i) nodal set for each control volume; and (ii) coordinates of cell nodes (or cell vertices). Control volumes in an implicit unstructured grid can be comprised of the following cell types: (i) tetrahedron, (ii) pyramid, (iii) wedge, (iv) hexahedron, and (v) a mix of above mentioned four cell types. The local number of cell nodes and faces is shown in Figure 4-1; while Table 4-6 summarizes the side set node order for the four supported cell types.

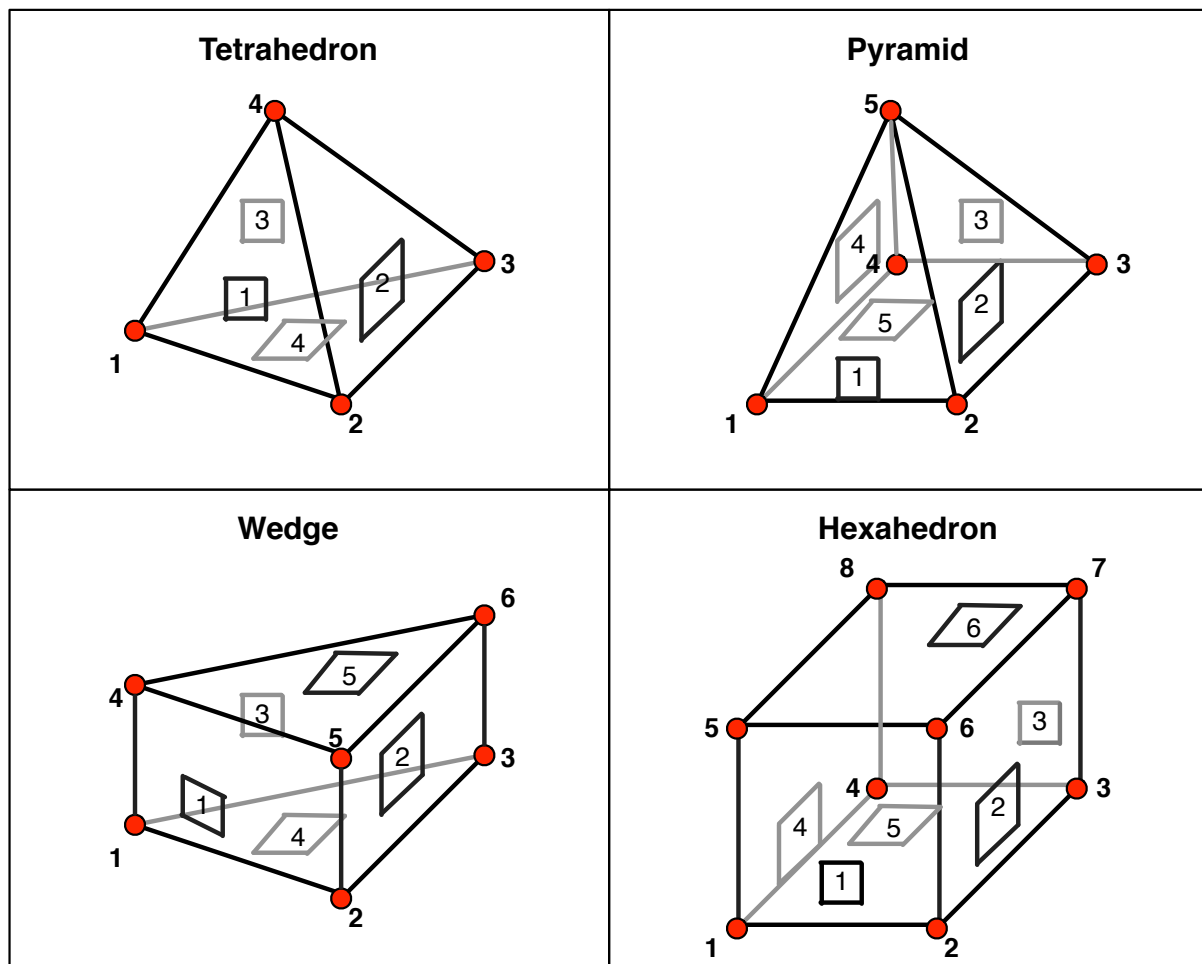


Figure 4-1: Side set numbering of supported cell types within unstructured grid

Table 4-6: Side set node ordering

Element type	Side #	Node order
Tetrahedron	1	1–2–4
	2	2–3–4
	3	1–4–3
	4	1–3–2
Pyramid	1	1–2–5
	2	2–3–5
	3	3–4–5
	4	4–1–5
	5	1–4–3–2
Wedge	1	1–2–5–4
	2	2–3–6–5
	3	3–1–4–6
	4	1–3–2
	5	4–5–6
Hexahedron	1	1–2–6–5
	2	2–3–7–6
	3	3–4–8–7
	4	4–1–5–8
	5	1–4–3–2
	6	5–6–7–8

4.27.2 Explicit Unstructured Grids

The explicit unstructured grid format comprises of information about (i) cells (id, x, y, z, and volume), and (ii) cell connections (upstream and downstream cell ids, centroid and area of side joining cells).

```

CELLS NUM_CELLS
CELL_ID X Y Z VOLUME
...
CONNECTIONS NUM_CONNECTIONS
CELL_ID_UP CELL_ID_DOWN FACE_X FACE_Y FACE_Z AREA
...
```

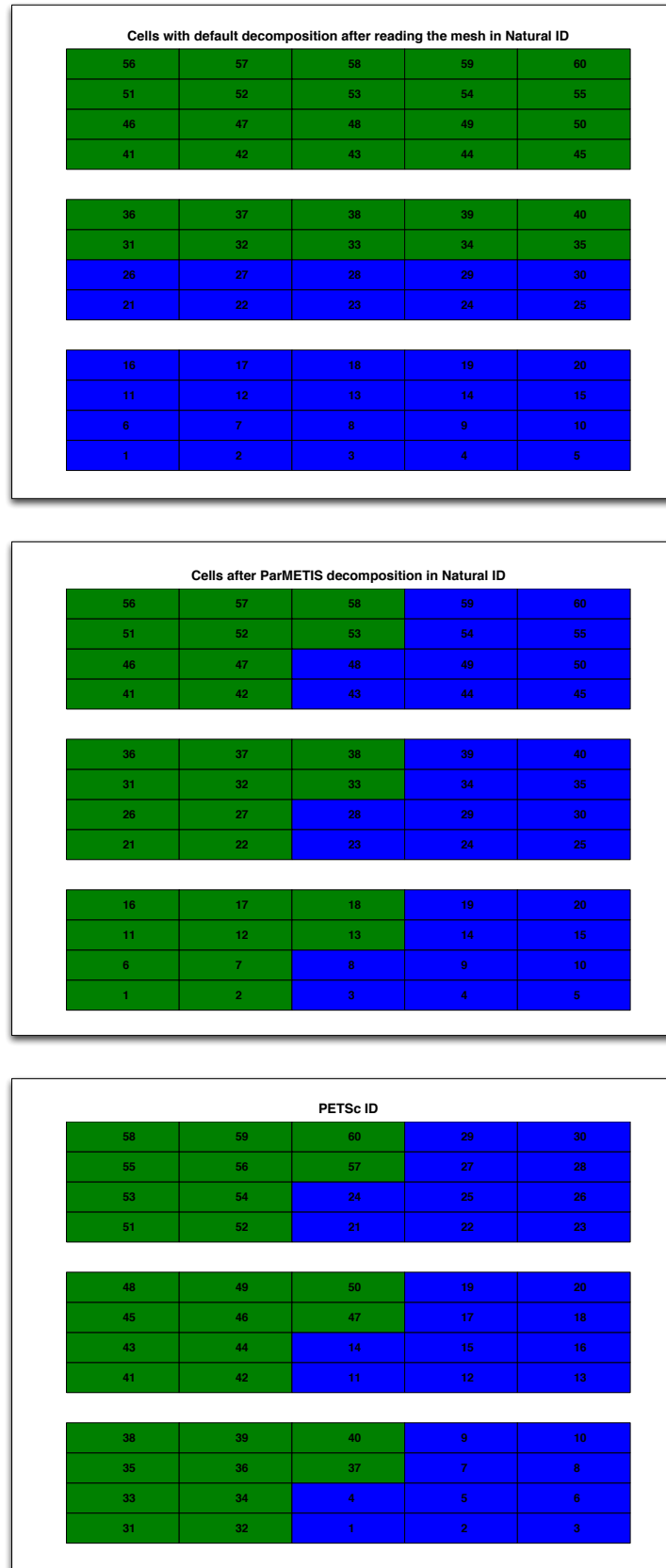


Figure 4-2: (a) Example of a $5 \times 4 \times 3$ unstructured grid problem with default decomposition across two processors. Cells labelled in natural ID. (b) ParMETIS decomposition of the domain; (c) Representation of a global PETSc vector

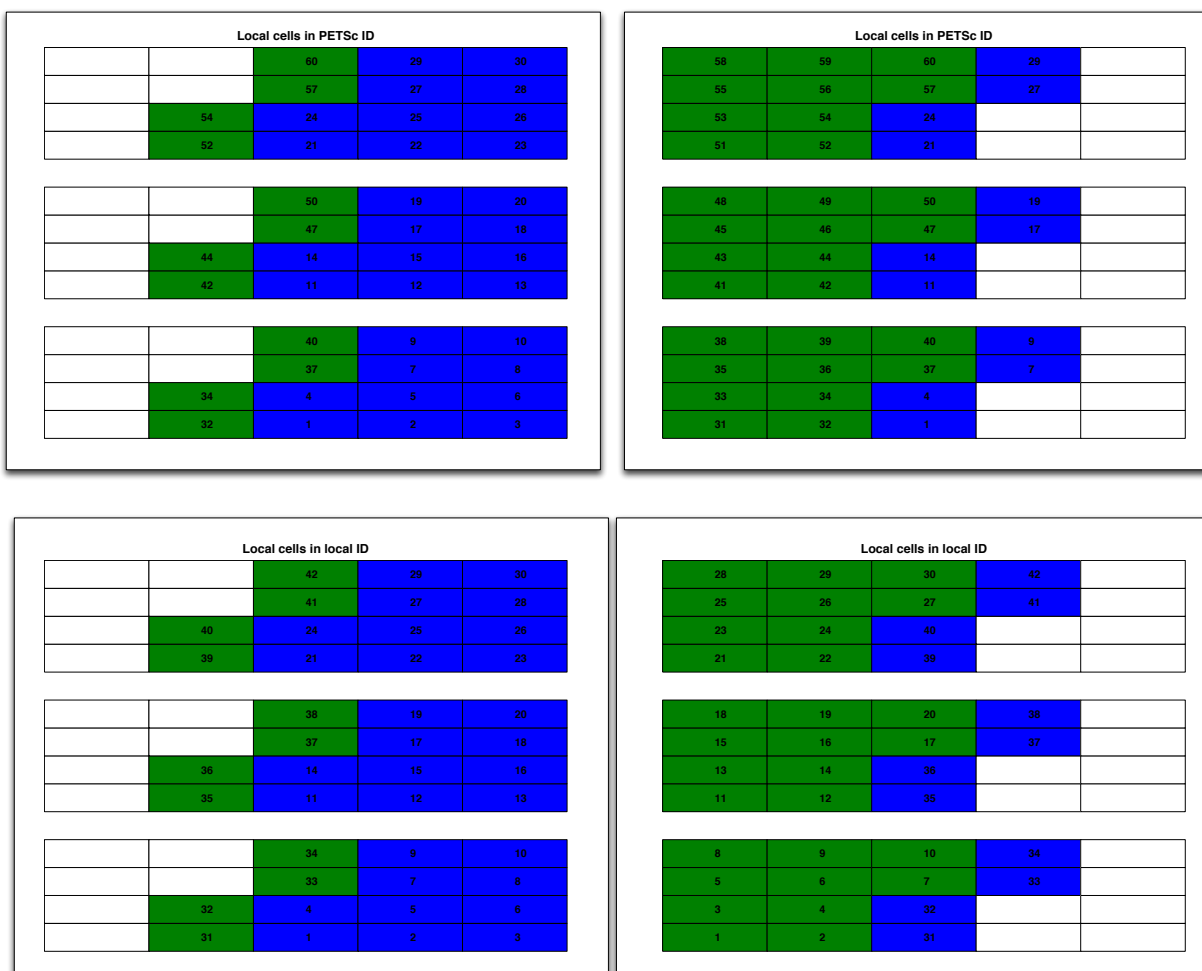


Figure 4-3: (a-b) Local and ghost cells on proc-0 and proc-1 with cell IDs in PETSc order. (c-d) Local PETSc Vector with cell IDs in local ghosted order.

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](#)

4.28 Keyword: IMMOBILE_DECAY_REACTION

[↪ Return to List of Keywords](#)

Description: Specifies parameters for first-order decay of an immobile species.

Input:

IMMOBILE_DECAY_REACTION [Name]

SPECIES_NAME <string> Name of immobile species to undergo first-order decay.

RATE_CONSTANT <float> First-order rate constant [1/sec].

HALF_LIFE <float> Species half life [sec].

END [/]

Explanation: *Note that units for rate constant or half life other than 1/sec or sec, respectively, may be specified. Only specify one of rate constant or half life, but not both.*

Example:

```
CHEMISTRY
  PRIMARY_SPECIES
    ...
  /
  IMMOBILE_SPECIES
    D(im)
  /
  ...
  IMMOBILE_DECAY_REACTION
    SPECIES_NAME D(im)
    RATE_CONSTANT 1.d-9
  /
  ...
/
```

[↪ Return to List of Keywords](#)

4.29 Keyword: INITIAL_CONDITION

[↪ Return to List of Keywords](#)

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](#)

4.30 Keyword: INITIALIZE_FLOW_FROM_FILE

[↪ Return to List of Keywords](#)

INITIALIZE_FLOW_FROM_FILE <string> Filename

Description: Set initial conditions for flow from specified HDF5 file.

Example: `./regression_tests/default/543/543_flow.in`

```
INITIALIZE_FLOW_FROM_FILE 543_initial_pressure.h5
```

4.31 Keyword: INITIALIZE_TRANSPORT_FROM_FILE

INITIALIZE_TRANSPORT_FROM_FILE <string> Filename

Description: Set initial conditions for transport from specified HDF5 file.

Example:

```
INITIALIZE_TRANSPORT_FROM_FILE filename.h5
```

[↪ Return to List of Keywords](#)

4.32 Keyword: INTEGRAL_FLUX

[↪ Return to List of Keywords](#)

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](#)

4.33 Keyword: LINEAR_SOLVER

[↪ Return to List of Keywords](#)

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 [pilot, 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](#)

4.34 Keyword: MATERIAL_PROPERTY

[↪ Return to List of Keywords](#)

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/m³]

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_EXPANSIVITY <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} [m²]

PERM_Y <float> Diagonal permeability k_{yy} [m²]

PERM_Z <float> Diagonal permeability k_{zz} [m²]

PERM_ISO <float> Isotropic permeability values [m²]

PERM_XY <float> Off-diagonal permeability k_{xy} for use with MFD
(mimetic_unstructured grid) [m²] (not currently supported)

PERM_XZ <float> [m²] Off-diagonal permeability k_{xz}

PERM_YZ <float> [m²] Off-diagonal permeability k_{yz}

END [/]

PERMEABILITY_POWER <float> (see Eqn.(A-124))

PERMEABILITY_CRIT_POR <float> (see Eqn.(A-125))

PERMEABILITY_MIN_SCALE_FAC <float> (see Eqn.(A-126))

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.(A-90))

LENGTH <float> [m] length of SLAB type

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

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

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

RADIUS <float> [m] radius for NESTED_SPHERES type

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

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

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

END [/]

END [/]

Explanation:**Example:**

```

MATERIAL_PROPERTY  Hanford
  ID 1
  SATURATION_FUNCTION sf1
  POROSITY 0.332
  TORTUOSITY 1.
  PERMEABILITY
    PERM_X 1.d-12
    PERM_Y 1.d-12
    PERM_Z 1.d-12
  /
END
=====
MATERIAL_PROPERTY soil1
  ID 1
  SATURATION_FUNCTION sf1
  POROSITY 0.1

```

```

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
```

PFLOTTRAN 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](#)

4.35 **Keyword: MODE**

[↪ Return to List of Keywords](#)

Description: determines the flow mode: Richards (variably saturated porous media); MPH, MPHASE, FLASH2 (CO₂ + H₂O); 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 (see Sections 7.4 and A-3)
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)	Immisible CO ₂ -water-energy
MIS	Missible H ₂ O-glycol fluid

Examples:

```
...
PROCESS_MODELS
  SUBSURFACE_FLOW flow
    MODE RICHARDS
  /
/
...

...
PROCESS_MODELS
  SUBSURFACE_FLOW flow
    MODE TH
  /
/
...
```

[↪ Return to List of Keywords](#)

4.36 Keyword: MULTIPLE_CONTINUUM

[↩ Return to List of Keywords](#)

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](#)

4.37 Keyword: NEWTON_SOLVER

[↪ Return to List of Keywords](#)

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](#)

4.38 Keyword: NONUNIFORM_VELOCITY

[↪ Return to List of Keywords](#)

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](#)

4.39 Keyword: NUMERICAL_JACOBIAN_FLOW

[↩ Return to List of Keywords](#)

NUMERICAL_JACOBIAN_FLOW Uses numerically evaluated Jacobian for flow.

[↩ Return to List of Keywords](#)

4.40 Keyword: NUMERICAL_JACOBIAN_RXN[↩ Return to List of Keywords](#)

NUMERICAL_JACOBIAN_RXN Uses numerically evaluated Jacobian for reactions.

[↩ Return to List of Keywords](#)

4.41 Keyword: NUMERICAL_JACOBIAN_MULTI_COUPLE

[↩ Return to List of Keywords](#)

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](#)

4.42 Keyword: OBSERVATION

[↪ Return to List of Keywords](#)

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](#)

4.43 Keyword: ORIGIN (ORIG)

[↩ Return to List of Keywords](#)

ORIGIN (ORIG) X_DIRECTION Y_DIRECTION Z_DIRECTION

[↩ Return to List of Keywords](#)

4.44 Keyword: OUTPUT

[↪ Return to List of Keywords](#)

Description: Specifies output options for biogeochemical transport.

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

MINERAL_POROSITY -Output cell porosity derived from mineral volume fractions

EFFECTIVE_POROSITY -Output cell effective porosity (GENERAL mode only)

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). Note that this is compatible only with TecPlot BLOCK format.

MASS_BALANCE [DETAILED] -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 CO₂. Output times are controlled by PERIODIC_OBSERVATION TIMESTEP and TIME, and printout times. Option DETAILED provides total moles of minerals.

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 i th component in phase α is given as the integral over the entire computational domain (see Eqn.(A-19a))

$$N_i^\alpha = \int_V \varphi s_\alpha \eta_\alpha x_i^\alpha dV. \quad (4-2)$$

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

$$N_i = \sum_\alpha N_i^\alpha, \quad (4-3)$$

is given by

$$\frac{dN_i}{dt} = \sum_{\alpha} \frac{dN_i^{\alpha}}{dt} \quad (4-4)$$

$$= - \sum_{\alpha} \int_{\partial V} \mathbf{F}_i^{\alpha} \cdot d\mathbf{S} + Q_i, \quad (4-5)$$

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

$$\mathcal{F}_i^{\alpha} = - \int_{\partial V} \mathbf{F}_i^{\alpha} \cdot d\mathbf{S}, \quad (4-6)$$

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

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

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

$$\overline{Q}_i = \int_0^t Q_i dt'. \quad (4-8)$$

Global conservation of mass implies the equality

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

The header for the MPHASE mass balance file reads (for time in years, l =aqueous liquid phase, sc =supercritical CO₂):

$$\begin{array}{l} t, \Delta t, [y] \\ N_{\text{H}_2\text{O}}^l, N_{\text{CO}_2}^l, \tilde{N}_{\text{CO}_2}^l, N_{\text{H}_2\text{O}}^{\text{sc}}, N_{\text{CO}_2}^{\text{sc}}, \tilde{N}_{\text{CO}_2}^{\text{sc}}, [\text{kmol}] \\ \overline{\mathcal{F}}_{\text{H}_2\text{O}}^l, \overline{\mathcal{F}}_{\text{CO}_2}^l, \overline{\mathcal{F}}_{\text{H}_2\text{O}}^{\text{sc}}, \overline{\mathcal{F}}_{\text{CO}_2}^{\text{sc}}, [\text{kmol}] \\ \mathcal{F}_{\text{H}_2\text{O}}^l, \mathcal{F}_{\text{CO}_2}^l, \mathcal{F}_{\text{H}_2\text{O}}^{\text{sc}}, \mathcal{F}_{\text{CO}_2}^{\text{sc}}, [\text{kmol/y}] \\ \overline{Q}_{\text{H}_2\text{O}}, \overline{Q}_{\text{CO}_2}, [\text{kmol}] \\ Q_{\text{H}_2\text{O}}, Q_{\text{CO}_2}, [\text{kmol/y}] \end{array} \quad (4-10)$$

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

$$\tilde{N}_i^{\alpha} = \int_{V, s_{\alpha} < s_{\alpha}^0} \varphi s_{\alpha} \eta_{\alpha} x_i^{\alpha} dV, \quad (4-11)$$

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](#)

4.45 Keyword: **OVERWRITE_RESTART_FLOW_PARAMS**

[↩ Return to List of Keywords](#)

OVERWRITE_RESTART_FLOW_PARAMS {overwrite_restart_flow_params = .true.}

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

4.46 Keyword: **OVERWRITE_RESTART_TRANSPORT**

[↩ Return to List of Keywords](#)

OVERWRITE_RESTART_TRANSPORT

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

4.47 Keyword: **OVERWRITE_RESTART_TRANSPORT**

[↩ Return to List of Keywords](#)

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](#)

4.48 Keyword: PROC

[↪ Return to List of Keywords](#)

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](#)

4.49 Keyword: REGION

[↪ Return to List of Keywords](#)

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 through 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](#)

4.50 Keyword: RESTART

[↪ Return to List of Keywords](#)

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](#)

4.51 Keyword: SATURATION_FUNCTION

[↪ Return to List of Keywords](#)



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.

Note: for GENERAL and RICHARDS modes use keyword CHARACTERISTIC_CURVES; SATURATION_FUNCTION should be used in all other flow modes.

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](#)

4.52 Keyword: SOURCE_SINK

[↩ Return to List of Keywords](#)

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](#)

4.53 Keyword: STRATIGRAPHY (STRATA)

[↪ Return to List of Keywords](#)

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

Optional Cards:

START_TIME <float> <string> Start time when STRATA card is relevant. The string specifies units. This keyword allows for changes in a material property at a given point in time (e.g. due to human intrusion).

END_TIME <float> <string> End time when STRATA card is relevant. The string specifies units.

INACTIVE Inactivates all cells in the associated REGION.

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](#).

Evolving strata:

```
MATERIAL_PROPERTY soil1
  ID 1
  ...
  PERMEABILITY
    PERM_ISO 1.d-12
  /
  ...
END
```

```
MATERIAL_PROPERTY soil2
  ID 2
  ...
  PERMEABILITY
    PERM_ISO 5.d-13
  /
  ...
END
```

```
STRATA
  REGION all
  MATERIAL soil1
  START_TIME 0.d0 y
  END_TIME 25.d0 y
END
```

```
STRATA
  REGION all
  MATERIAL soil2
  START_TIME 25.d0 y
  END_TIME 50.d0 y
END
```

[↩ Return to List of Keywords](#)

4.54 Keyword: TIME

[↪ Return to List of Keywords](#)

Description: the keyword **TIME** controls the simulation time.

Input:

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](#)

4.55 Keyword: TIMESTEPPER

Description: the keyword **TIMESTEPPER** controls time stepping.

[↪ Return to List of Keywords](#)

Input:

TIMESTEPPER [**FLOW**, **TRANSPORT**] Must include one of **FLOW** or **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 <float> (steady state convergence relative tolerance). Flag indicating that the simulation is to be initialized 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](#)

4.56 Keyword: TRANSPORT_CONDITION

[↪ Return to List of Keywords](#)

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]

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

```

```

TRANSPORT_CONDITION
TIME_UNITS h
CONSTRAINT_LIST
0. name_of_constraint
1. name_of_constraint
2. name_of_constraint

```

```
    /  
END  
  
TRANSPORT_CONDITION  
  CONSTRAINT_LIST  
    0. h name_of_constraint  
    1. h name_of_constraint  
    2. h name_of_constraint  
  /  
END
```

[↩ Return to List of Keywords](#)

4.57 Keyword: UNIFORM_VELOCITY

Optional[↩ Return to List of Keywords](#)**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](#)

4.58 **Keyword: USE_TOUCH_OPTIONS**

[↩ Return to List of Keywords](#)

Description:

Input:

USE_TOUCH_OPTIONS {use_touch_options = .true.}

Explanation:

Example:

[↩ Return to List of Keywords](#)

4.59 Keyword: VELOCITY_DATASET

[↪ Return to List of Keywords](#)

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](#)

4.60 Keyword: WALLCLOCK_STOP

[↩ Return to List of Keywords](#)

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 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.

5.1 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
```

5.2 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”.

5.3 Configuration Files

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

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

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

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

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

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

- $\text{rate} = 1 \times 10^{-12}$ absolute
- $\text{volume_fraction} = 1 \times 10^{-12}$ absolute
- $\text{pressure} = 1 \times 10^{-12}$ absolute
- $\text{saturation} = 1 \times 10^{-12}$ absolute
- $\text{residual} = 1 \times 10^{-12}$ absolute

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

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

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

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

5.4 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 PFLOTTRAN input file, and get the simulation running correctly.
- Tell PFLOTTRAN 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.

5.5 Updating Test Results

The output from PFLOTTRAN 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.

6 Visualization

Visualization of the results produced by PFLOTTRAN 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.

Several potentially useful hints on using these packages are provided below.

6.1 Tecplot

- 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"}
```

6.2 VisIt

- Inactive cells can be omitted by going to `Controls -> Subset` and unchecking `Material_ID[0]`.
- In 3D to scale the size of one of the coordinate axes go to `Controls -> View` and check box `Scale 3D Axes` and set desired scaling factor in box to right.
- To make 2D plots use `Operators -> Slicing -> Slice`.
- For 3D plots `Operators -> Slicing -> ThreeSlice` is useful.
- Instructions to set up geomechanical and flow data simultaneously in **VisIt**:
 1. Open VisIt.
 2. Select the 1×2 layout button on Window 1 to open a new window (Window 2).
 3. Now make Window 1 active by checking the first button on Window 1.
 4. Click on Open and select `*-geomech-*.xmf` database.
 5. Click on Add \rightarrow Pseudocolor \rightarrow `rel_disp_z` and click Draw (Set Auto apply to avoid clicking Draw repeatedly).
 6. Select the domain and rotate it to a preferred angle.
 7. Select Window 2 and make it active.
 8. Click Open and select `pflotran.h5` or `pflotran-*.h5` (This file contains all the subsurface flow data).
 9. Click on e.g. Add \rightarrow Pseudocolor \rightarrow Gas Saturation (or other desired variable).
 10. Click on Lock view and Lock time on both windows (This will sync views and times on both windows).

11. After a window pops up, select Yes.
12. With Window 2 active select: Operators → Slicing → ThreeSlice.
13. Double click on three slice and change x and y to appropriate values.
14. Select Window 1.
15. Select Controls → Expressions.
16. Click New and add a name (e.g., disp_vector). Select Vector Mesh Variable.
17. Under Definition, add <rel_disp_x _lb_m_rb_>, <rel_disp_y _lb_m_rb_>, <rel_disp_z _lb_m_rb_>, Apply and Dismiss.
18. Click Add → Vector → disp_vector.
19. Double click Vector under pflotran-geomech-*.xmf database:disp_vector.
20. Select Form and set Scale to e.g. 0.5. Then select Rendering and change magnitude color from Default to difference (select a color scheme of your choice).
21. Apply and Dismiss.
22. Double click Pseudocolor → rel_disp_z. Select e.g. hot_desaturated for color table. Set opacity to 50%. Apply and Dismiss.
23. Next, for mesh movement click: Operators → Transforms → Displace. After a window pops up, dismiss it.
24. Double click Displace, change Displacement multiplier to e.g. 10000. Click on Displacement Variable → Vectors → disp_vectors. Dismiss the window saying no data etc. Apply and Dismiss.
25. Finally, click the play button to watch movie. Rotate the domain to a convenient angle before doing so.

6.3 Gnuplot, Matplotlib

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

```
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.

- When using **gnuplot** it is possible to number the output file columns with `PRINT_COLUMN_IDS` added to the `OUTPUT` keyword. This is only useful, however, with `FORMAT TECPLOT POINT` output option.

7 Benchmark Problems

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

7.1 Ion Exchange

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

$$\omega = \frac{N_s}{V} = \frac{N_s}{M_s} \frac{M_s}{V_s} \frac{V_s}{V}, \quad (7-12a)$$

$$= \rho_s(1 - \varphi)\text{CEC}. \quad (7-12b)$$

Using a porosity of 0.61 and solid grain density of 3.0344 g/cm³, yielded $\omega = 71.004$ mol/m³. The results of the simulation are shown in Figure 7-4 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 7.1.

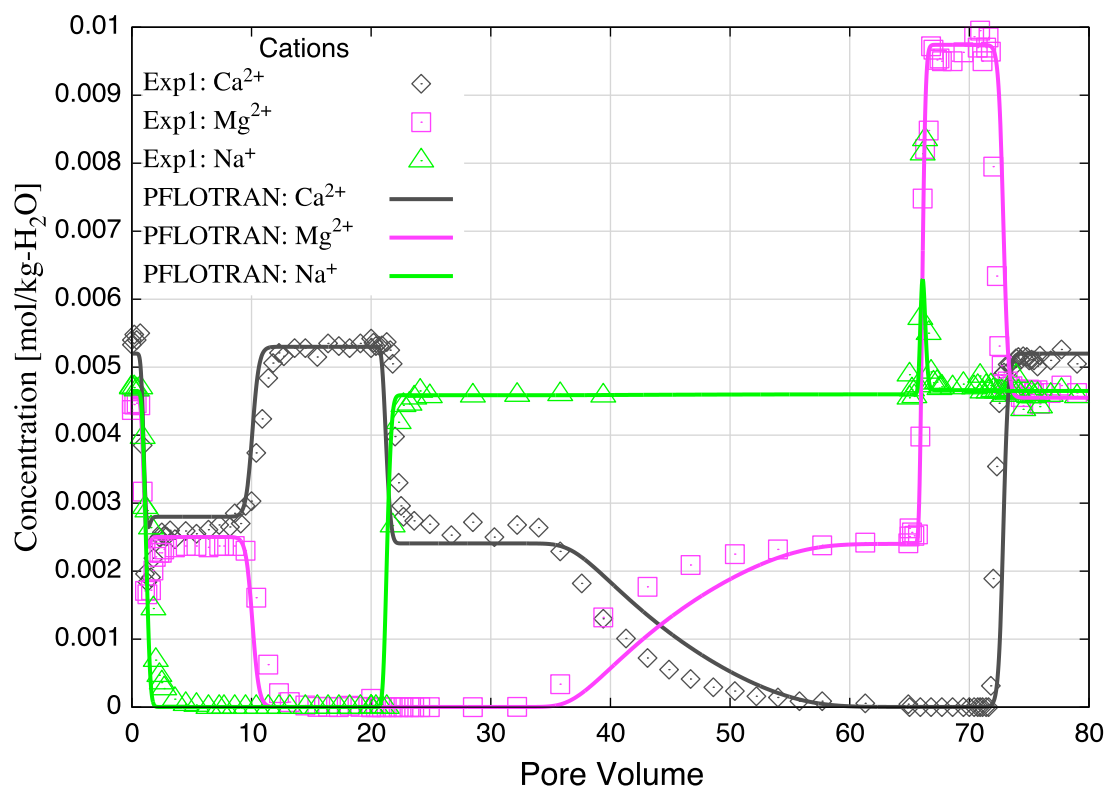


Figure 7-4: Breakthrough curves for Ca^{2+} , Mg^{2+} and Na^{+} compared with experimental results from Voegelin et al. (2000).

#Description: 1D ion exchange problem

```
SIMULATION
  SIMULATION_TYPE SUBSURFACE
  PROCESS_MODELS
    SUBSURFACE_TRANSPORT transport
    GLOBAL_IMPLICIT
  /
END
```

SUBSURFACE

```
# 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 71.004 ! mol/m^3
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  
/  
/
```

```
END_SUBSURFACE
```

7.2 GENERAL_REACTION Example

7.2.1 Problem Description

A single irreversible reaction is considered of the form



for flow in a fully saturated 1D column of length 100 m with a Darcy velocity of 1 m/y, diffusion coefficient of 10^{-9} m²/s and porosity equal to 0.25. The conservation equation for advection, diffusion and reaction is given by

$$\frac{\partial}{\partial t} \varphi C_l + \nabla \cdot \mathbf{F}_l = -\varphi \nu_l \mathcal{R}, \quad (l = A, B, C), \quad (7-14)$$

with stoichiometric coefficients $\nu_A = 1$, $\nu_B = 2$, and $\nu_C = -1$. The flux \mathbf{F}_l consists of contributions from advection and diffusion

$$\mathbf{F}_l = \mathbf{q} C_l - \varphi D \nabla C_l. \quad (7-15)$$

The forward reaction rate is based on a elementary aqueous reaction

$$\mathcal{R} = k_f C_A^{\nu_A} C_B^{\nu_B}. \quad (7-16)$$

Dividing through by porosity (assuming $\varphi = \text{constant}$), the transport equation becomes

$$\frac{\partial C_l}{\partial t} + \nabla \cdot \mathbf{v} C_l - D \nabla \cdot \nabla C_l + \nu_l k_f C_A^{\nu_A} C_B^{\nu_B} = 0, \quad (7-17)$$

with average pore velocity

$$\mathbf{v} = \frac{\mathbf{q}}{\varphi}. \quad (7-18)$$

Initial and boundary conditions imposed on the solution are given by

$$C(x, t = 0) = C_\infty, \quad (7-19a)$$

$$C(x = 0, t) = C_0, \quad (7-19b)$$

$$\left. \frac{\partial C}{\partial x} \right|_{x=l_x} = 0. \quad (7-19c)$$

7.2.2 Simulation Results

Results are shown in Figure 7-5 for the concentrations of species A, B, C at 5 years obtained from PFLOTRAN and a prototype code written in C++ based on the PETSc TS time stepping class. The code uses a backward Euler (TSBEULER) time integrator with nodes placed at the grid cell corners. The slight discrepancy between the results of the two codes may be due to the use of a finite volume cell-centered grid in PFLOTRAN, versus the corner-node grid used in the prototype code.

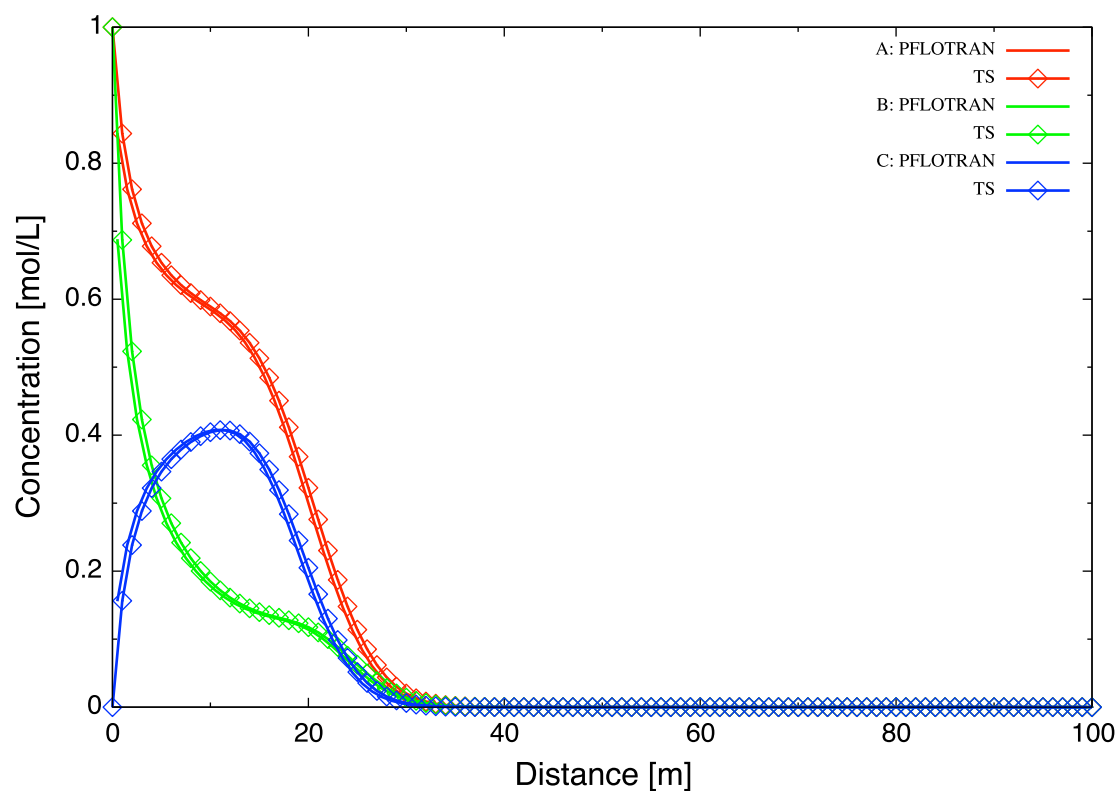


Figure 7-5: Comparison of concentrations for species A, B, C plotted as a function of distance for an elapsed time of 5 years for PFLOTRAN and a prototype code based on PETSc's TS class.

7.2.3 PFLOTRAN Input File

```
#Description: 1D general reaction with the aqueous reaction A + 2 B -> C.

SIMULATION
  SIMULATION_TYPE SUBSURFACE
  PROCESS_MODELS
    SUBSURFACE_TRANSPORT transport
    GLOBAL_IMPLICIT
  /
/
END

SUBSURFACE

#===== useful tranport parameters =====
UNIFORM_VELOCITY 1.d0 0.d0 0.d0 m/yr

REFERENCE_DENSITY 1000.d0

#===== chemistry =====
CHEMISTRY
  PRIMARY_SPECIES
    A(aq)
    B(aq)
    C(aq)
  /
  GENERAL_REACTION
    REACTION A(aq) + 2 * B(aq) <-> C(aq)
    FORWARD_RATE 5.d-8
    BACKWARD_RATE 0.d0
  /
```

```

    DATABASE /Users/lichtner/pflotran/pflotran-dev/database/hanford.dat
    OUTPUT
      all
      TOTAL
    /
  END

#===== solver options =====
LINEAR_SOLVER TRANSPORT
  SOLVER DIRECT
END

#===== discretization =====
GRID
  TYPE structured
  NXYZ 100 1 1
  BOUNDS
    0.d0 0.d0 0.d0
    100.d0 100.d0 1.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
  ROCK_DENSITY 1650.d0
END

#===== output options =====
OUTPUT
  TIMES y 5.
  FORMAT TECPLOT POINT
END

#===== times =====
TIME
  FINAL_TIME 5.d0 y
  INITIAL_TIMESTEP_SIZE 1.d0 h
  MAXIMUM_TIMESTEP_SIZE 1.d-2 y
END

#===== regions =====
REGION all
  COORDINATES
    0.d0 0.d0 0.d0
    100.d0 1.d0 1.d0
  /
END

REGION west
  FACE west
  COORDINATES
    0.d0 0.d0 0.d0
    0.d0 1.d0 1.d0
  /
END

REGION east
  FACE east
  COORDINATES
    100.d0 0.d0 0.d0
    100.d0 1.d0 1.d0
  /
END

```

```

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

TRANSPORT_CONDITION inlet
  TYPE dirichlet
  CONSTRAINT_LIST
    0.d0 inlet
  /
END

TRANSPORT_CONDITION outlet
  TYPE zero_gradient
  CONSTRAINT_LIST
    0.d0 inlet
  /
END

#===== constraints =====
CONSTRAINT initial
  CONCENTRATIONS
    A(aq) 1.d-16 T
    B(aq) 1.d-16 T
    C(aq) 1.d-16 T
  /
END

CONSTRAINT inlet
  CONCENTRATIONS
    A(aq) 1.d0 T
    B(aq) 1.d0 T
    C(aq) 1.d-16 T
  /
END

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

BOUNDARY_CONDITION outlet
  TRANSPORT_CONDITION outlet
  REGION east
END

BOUNDARY_CONDITION inlet
  TRANSPORT_CONDITION inlet
  REGION west
END

#===== stratigraphy couplers =====
STRATA
  REGION all
  MATERIAL soil1
END

END_SUBSURFACE

```


7.3 RICHARDS Mode with Tracer: SX-115 Hanford Tank Farm

7.3.1 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).

7.3.2 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 7-7 and 7-8. The governing equations consist of Richards equation for variably saturated fluid flow given by

$$\frac{\partial}{\partial t} \varphi s \rho + \nabla \cdot \mathbf{q} \rho = Q, \quad (7-20)$$

and solute transport of a tracer

$$\frac{\partial}{\partial t} \varphi C + \nabla \cdot (\mathbf{q} C - \varphi s \tau D \nabla C) = Q_C. \quad (7-21)$$

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 \mathbf{q} denotes the Darcy velocity defined by

$$\mathbf{q} = -\frac{k_{\text{sat}} k_r}{\mu} \nabla (p - \rho g z), \quad (7-22)$$

with saturated permeability k_{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

$$k_r = \sqrt{s_{\text{eff}}} \left\{ 1 - \left[1 - (s_l^{\text{eff}})^{1/m} \right]^m \right\}^2, \quad (7-23)$$

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

$$s_{\text{eff}} = [1 + (\alpha |P_c|)^n]^{-m}, \quad (7-24)$$

where s_{eff} is defined by

$$s_{\text{eff}} = \frac{s - s_r}{1 - s_r}, \quad (7-25)$$

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

$$m = 1 - \frac{1}{n}, \quad n = \frac{1}{1 - m}. \quad (7-26)$$

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

$$P_c = p_g^0 - p, \quad (7-27)$$

where $p_g^0 = 101,325$ Pa is set to atmospheric pressure.

7.3.2.1 Semi-Analytical Solution for Steady-State Conditions For steady-state conditions the saturation profile satisfies the equation

$$\frac{d}{dz}\rho q_z = 0, \quad (7-28)$$

or assuming an incompressible fluid

$$q_z = q_z^0, \quad (7-29)$$

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

$$\frac{dp}{dz} = -\frac{\mu q_z^0}{k_{\text{sat}} k_r} - \rho g, \quad (7-30)$$

using Eqns.(7-23) and (7-24) to express the relative permeability k_r as a function of pressure. For the special case of zero infiltration it follows that

$$p(z) = p_0 - \rho g(z - z_0), \quad (7-31)$$

with $p(z_0) = p_0$. The saturation profile is obtained from Eqns.(7-24) and (A-4).

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

$$P_c(z_{\text{wt}}) = 0, \quad (7-32)$$

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

$$z_{\text{wt}} = z_0 + \frac{p_0 - p_g}{\rho g}, \quad (7-33)$$

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 $z_{\text{wt}} = z_0$, or the height of the watertable is equal to the datum. The same holds true also with constant nonzero infiltration.

7.3.3 Model Parameters

Model parameters used in the simulations are listed in Tables 7-7 and 7-8. Although not needed here, thermal properties are also listed. Diffusivity was set to $10^{-9} \text{ m}^2 \text{ s}^{-1}$ and tortuosity was set to one.

Table 7-7: 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 7-8: 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}	$\text{J kg}^{-1} \text{K}^{-1}$	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

7.3.4 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. This can be easily accomplished using CHECKPOINTING and RESTART keywords. The results for the steady-state saturation and pressure profiles are shown in Figure 7-6 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 high 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 7-7 is the saturation at different times following a two week leak releasing 60,000 gallons from the SX-115 tank at a depth of 16 m. In the simulation a release rate of $1.87 \times 10^{-3} \text{ kg/s}$ is used.

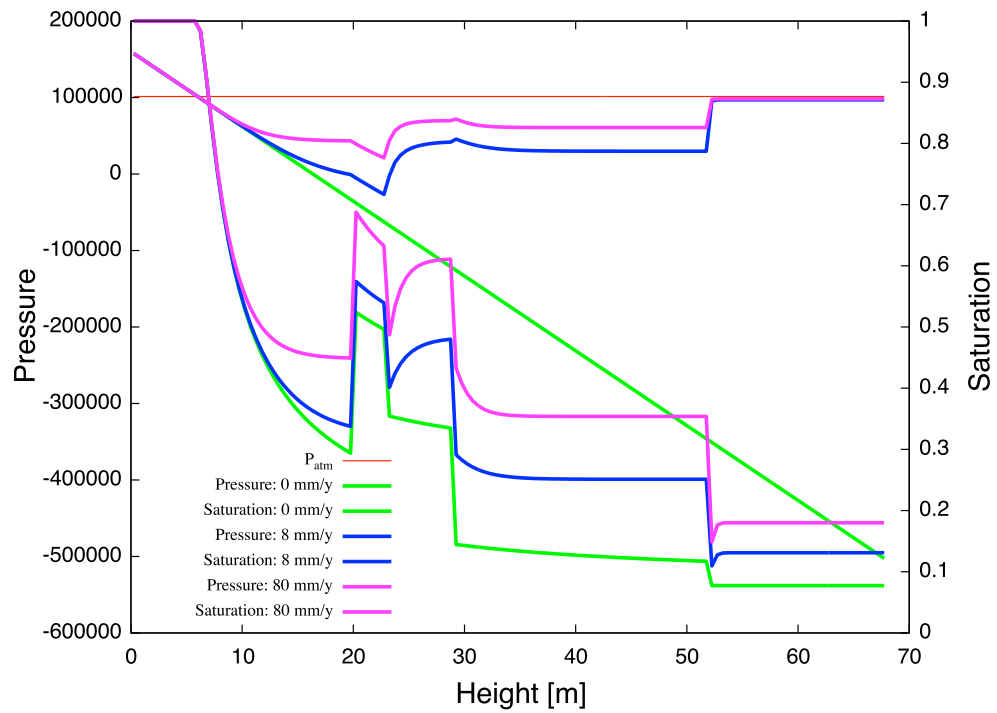


Figure 7-6: Steady-state saturation and pressure profiles for infiltration rates of 0, 8 and 80 mm/y. The water table is located at 6 m from the bottom of the computational domain.

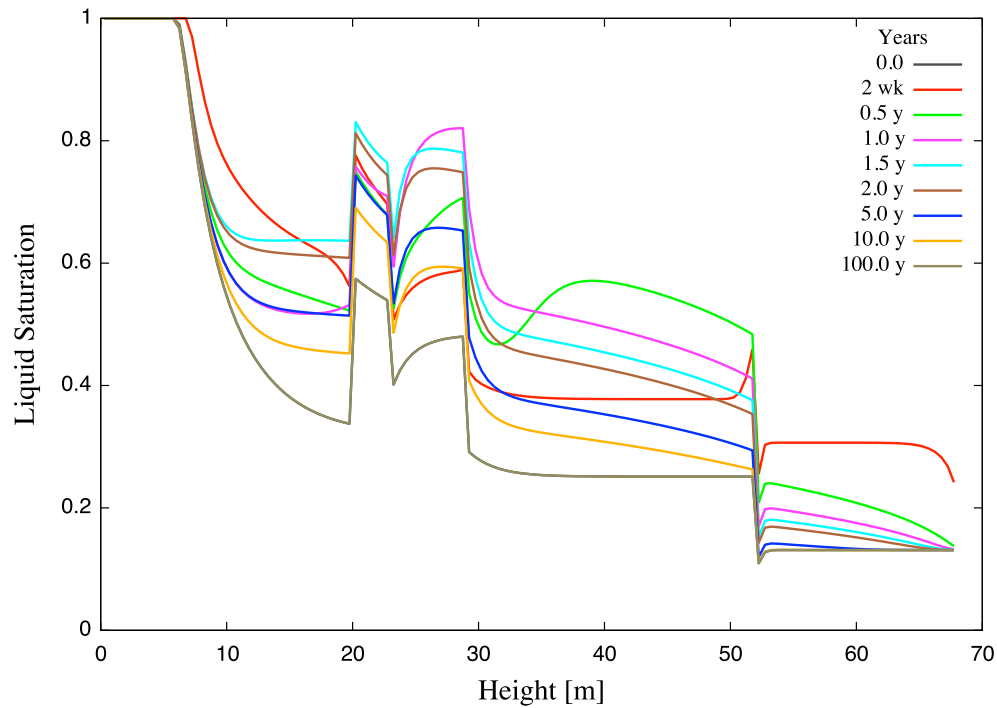


Figure 7-7: Simulation of a tank leak with a duration of two weeks showing the saturation profile for different times indicated in the figure for an infiltration rate of 8 mm/y.

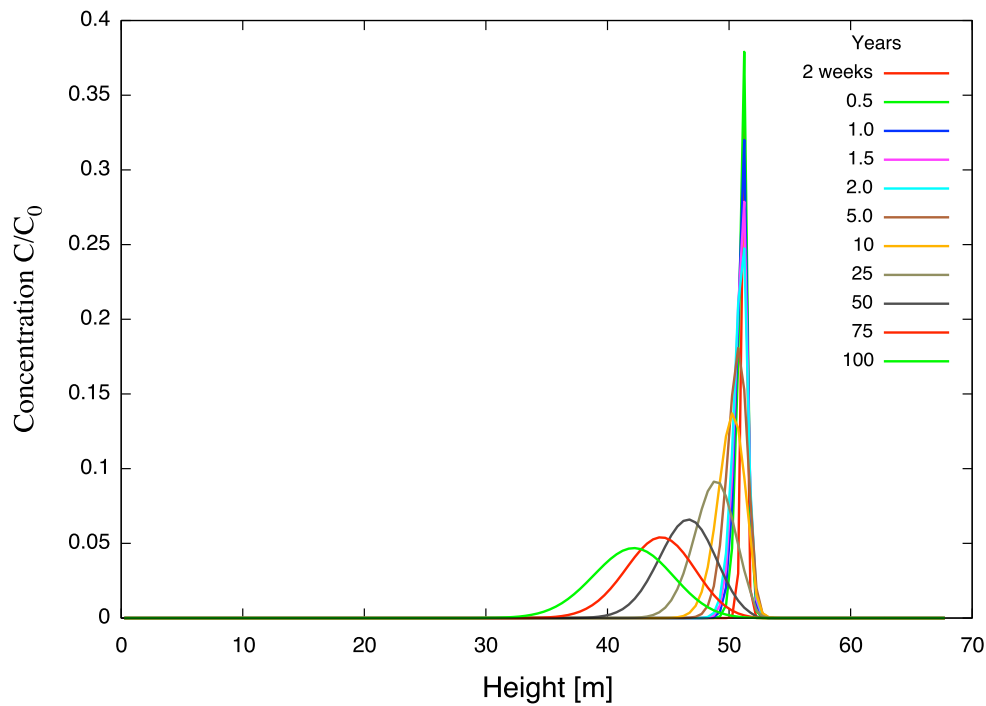


Figure 7-8: The solute concentration profile corresponding to Figure 7-7 for different times indicated in the figure for an infiltration rate of 8 mm/y.

7.3.5 PFLOTRAN Input File

Listing for the PFLOTRAN input file coupling Richards mode to a tracer is given below. 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 lines beginning with # are read as a comment as is input following '!'.

Note that the input file looks for the RESTART file for the transient run in the subdirectory: ./ss/sx115-restart.chk.

PFLOTRAN input file sx115.in:

```
#Description: 1D test problem for tracer transport for Hanford SX-115 waste tank.

SIMULATION
  SIMULATION_TYPE SUBSURFACE
  PROCESS_MODELS
    SUBSURFACE_FLOW flow
    MODE RICHARDS
  /
  SUBSURFACE_TRANSPORT transport
  GLOBAL_IMPLICIT
/
END

SUBSURFACE

#===== chemistry =====
CHEMISTRY
  PRIMARY_SPECIES
    Tracer
  /
  OUTPUT
    all
    FREE_ION
  /
END

#===== runtime =====
#CHECKPOINT 100000
RESTART ./ss/sx115-restart.chk 0.d0
#OVERWRITE_RESTART_TRANSPORT
#WALLCLOCK_STOP 11.75

#===== solver options =====
TIMESTEPPER FLOW
  #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 =====

CHARACTERISTIC_CURVES BF
  SATURATION_FUNCTION VAN_GENUCHTEN
    M 0.6585d0
    ALPHA 1.008d-3
    LIQUID_RESIDUAL_SATURATION 0.0774
  /
  PERMEABILITY_FUNCTION MUALEM_VG_LIQ
    M 0.6585d0
    LIQUID_RESIDUAL_SATURATION 0.0774
  /
END

CHARACTERISTIC_CURVES HF
  SATURATION_FUNCTION VAN_GENUCHTEN
    M 0.46944d0
    ALPHA 9.40796d-5
    LIQUID_RESIDUAL_SATURATION 0.08366d0
  /
  PERMEABILITY_FUNCTION MUALEM_VG_LIQ
    M 0.46944d0
    LIQUID_RESIDUAL_SATURATION 0.08366d0
  /
END

CHARACTERISTIC_CURVES PP
  SATURATION_FUNCTION VAN_GENUCHTEN
    M 0.45587d0
    ALPHA 6.85145d-5
    LIQUID_RESIDUAL_SATURATION 0.25953d0
  /
  PERMEABILITY_FUNCTION MUALEM_VG_LIQ
    M 0.45587d0
    LIQUID_RESIDUAL_SATURATION 0.25953d0
  /
END

```



```

CHARACTERISTIC_CURVES URG
  SATURATION_FUNCTION VAN_GENUCHTEN
    M 0.38594d0
    ALPHA 2.96555d-5
    LIQUID_RESIDUAL_SATURATION 0.21295d0
  /
  PERMEABILITY_FUNCTION MUALEM_VG_LIQ
    M 0.38594d0
    LIQUID_RESIDUAL_SATURATION 0.21295d0
  /
END

CHARACTERISTIC_CURVES MRG
  SATURATION_FUNCTION VAN_GENUCHTEN
    M 0.39217d0
    ALPHA 6.34015e-5
    LIQUID_RESIDUAL_SATURATION 0.06086d0
  /
  PERMEABILITY_FUNCTION MUALEM_VG_LIQ
    M 0.39217d0
    LIQUID_RESIDUAL_SATURATION 0.06086d0
  /
END

#===== output options =====
OUTPUT
  #SCREEN PERIODIC 10
  #MASS_BALANCE
  TIMES y 0.0383562 0.5 1.0 1.5 2.0 5.0 10.0 25. 50. 75. 100.
  FORMAT TECPLOT POINT
# VELOCITIES
  PRINT_COLUMN_IDS
  PERIODIC_OBSERVATION TIMESTEP 1
END

#===== times =====
TIME
  FINAL_TIME 100.d0 y
  INITIAL_TIMESTEP_SIZE 1.d-6 y
  MAXIMUM_TIMESTEP_SIZE 1.d-2 y
  MAXIMUM_TIMESTEP_SIZE 1.d0 y at 10 y
  MAXIMUM_TIMESTEP_SIZE 10.d0 y at 100 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-8 ! 0.08 m/yr
  # 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 LIST
  TIME_UNITS s
  DATA_UNITS kg/s
  0. 0.187e-4
  1.21293e6 0.
  /
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

END_SUBSURFACE
```

7.4 MPHASE

7.4.1 CO₂ Sequestration: 1D Example Problem and Comparison with TOUGHREACT

In this example problem involves sequentially coupling of MPHASE and CHEMISTRY. The chemical system consists of four primary species and 5 secondary species. Supercritical CO₂ is injected into a well located at the west boundary. Dirichlet pressure boundary conditions are imposed at the east boundary. The problem definition with associated parameters is given in Table 7-9.

Table 7-9: Problem definition and parameters used in the 1D CO₂ sequestration example.

Description	Symbol	Value
Domain	l	100 m
Permeability	k	10^{-15} m^2
Porosity	φ	0.12
Tortuosity	τ	1
Injection Rate	Q_{CO_2}	$5 \times 10^{-5} \text{ kg/s}$, duration 0.4 y
Characteristic Curves	modified van Genuchten	[see Eqns.(7-34)-(7-38)]
	λ	0.6
	α	$1.9 \times 10^{-5} \text{ Pa}^{-1}$
	s_{rl}	0
	s_{rg}	0
	P_c^{max}	10^7 Pa
Rock Density	ρ_r	2650 kg/m^3
Rock Specific Heat	c_r	1000 J/kg/K
Rock Thermal Conductivity	$\kappa_{\text{wet, dry}}$	0.5 W/m/K

The PFLOTTRAN initial aqueous solution corresponds to a brine with NaCl concentration of 0.5 m. Mineral reactions are not considered. The initial fluid composition taken from `pflotran.out` is listed in Table 7-10.

The defining equations for the saturation and relative permeability functions for the aqueous solution and supercritical CO₂ are given by the van Genuchten -Corey relations. For the aqueous solution van Genuchten curves are used for capillary pressure P_c

$$P_c(s_e) = \frac{1}{\alpha} \left[(s_e)^{-1/\lambda} - 1 \right]^{1-\lambda}, \quad (7-34)$$

and relative permeability k_{rl}

$$k_{rl} = \sqrt{s_e} \left\{ 1 - \left[1 - (s_e)^{1/\lambda} \right]^\lambda \right\}^2, \quad (7-35)$$

Table 7-10: Initial concentration of primary and secondary species. Mineral reactions are not considered.

Transport Condition: Initial				

iterations:	20			
pH:	5.0273			
ionic strength:	4.7915E-01 [mol/L]			
charge balance:	1.1102E-16			
pressure:	1.6450E+07 [Pa]			
temperature:	54.50 [C]			
density H2O:	992.99 [kg/m^3]			
ln / activity H2O:	0.0000E+00	1.0000E+00	[---]	
mole fraction H2O:	9.8093E-01	[---]		
mass fraction H2O:	9.7160E-01	[---]		

primary	free	total		
species	molal	molal	act coef	constraint

H+	1.1727E-05	2.5844E-17	8.0079E-01	chrg
Na+	4.7913E-01	5.0000E-01	6.8288E-01	total aq
Cl-	4.7913E-01	5.0000E-01	6.4459E-01	total aq
CO2 (aq)	1.1380E-04	1.2551E-04	1.1053E+00	CO2 (g)

complex	molality	act coef	logK	

NaCl (aq)	2.0866E-02	1.0000E+00	6.8511E-01	
HCO3-	1.1713E-05	6.8288E-01	6.2239E+00	
OH-	1.2056E-08	6.6467E-01	1.3123E+01	
NaOH (aq)	1.6487E-09	1.0000E+00	1.3325E+01	
CO3--	3.2433E-10	2.0899E-01	1.6323E+01	

with effective saturation s_e defined by

$$s_e = \frac{s_l - s_{lr}}{1 - s_{lr}}. \quad (7-36)$$

For the supercritical CO₂ phase the Corey curve is used defined by

$$k_{rg} = (1 - s')^2 [1 - (s')^2], \quad (7-37)$$

with

$$s' = \frac{s_l - s_{lr}}{1 - s_{lr} - s_{gr}}. \quad (7-38)$$

Shown in Figure 7-9 is a comparison of PFLOTTRAN with TOUGHREACT (TOUGHREACT results provided by Alt-Epping and Wanner, private comm.). The same thermodynamic database is used for both codes. Only slight differences can be seen. The CO₂ aqueous and total concentrations are essentially identical for PFLOTTRAN in the low pH region where supercritical CO₂ is present, with slight differences for TOUGHREACT. Note that the CO₂ aqueous concentration (and mole fraction X_{CO_2} although not visible in the figure) obtained from PFLOTTRAN is not exactly constant. This is caused, presumably, by a change in pressure as shown in Figure 7-10 for the liquid and CO₂ pressures in addition to the CO₂ saturation s_{CO_2} .

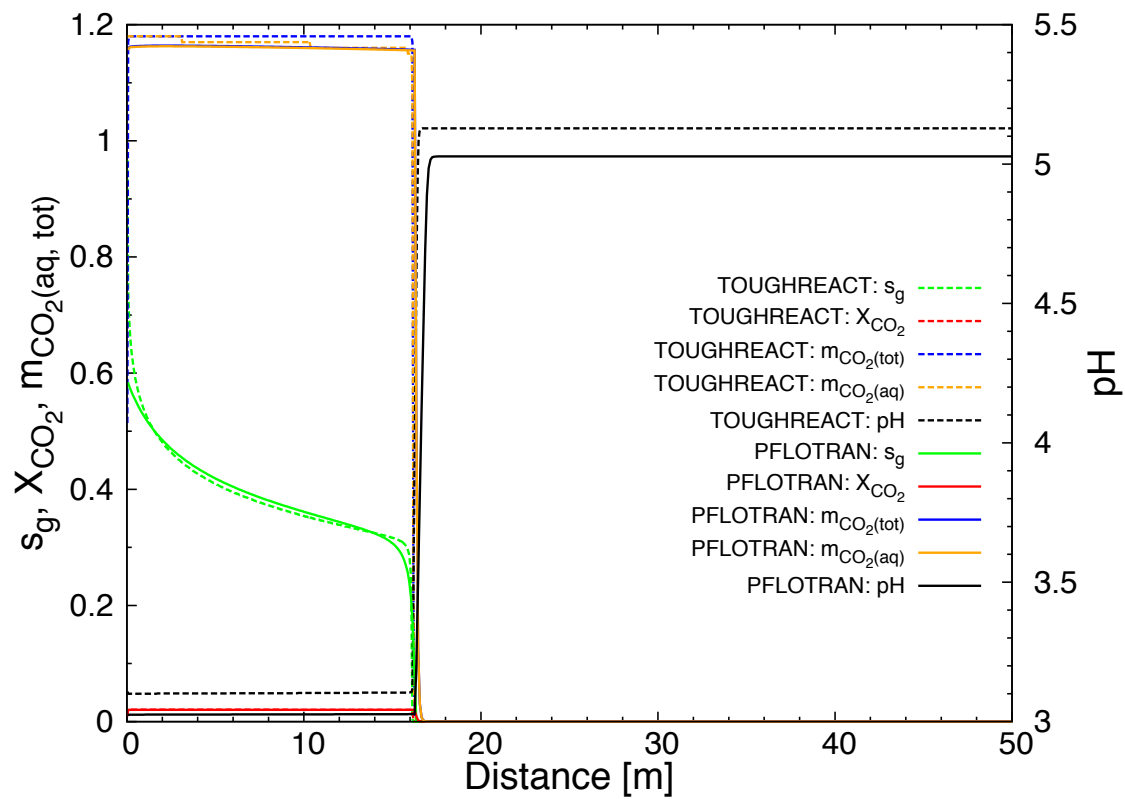


Figure 7-9: Comparison with TOUGHREACT (dashed curves) and PFLOTTRAN (solid curves) after an elapsed time of 0.4 y corresponding to the end of injection. Reasonable agreement is obtained between the two codes.

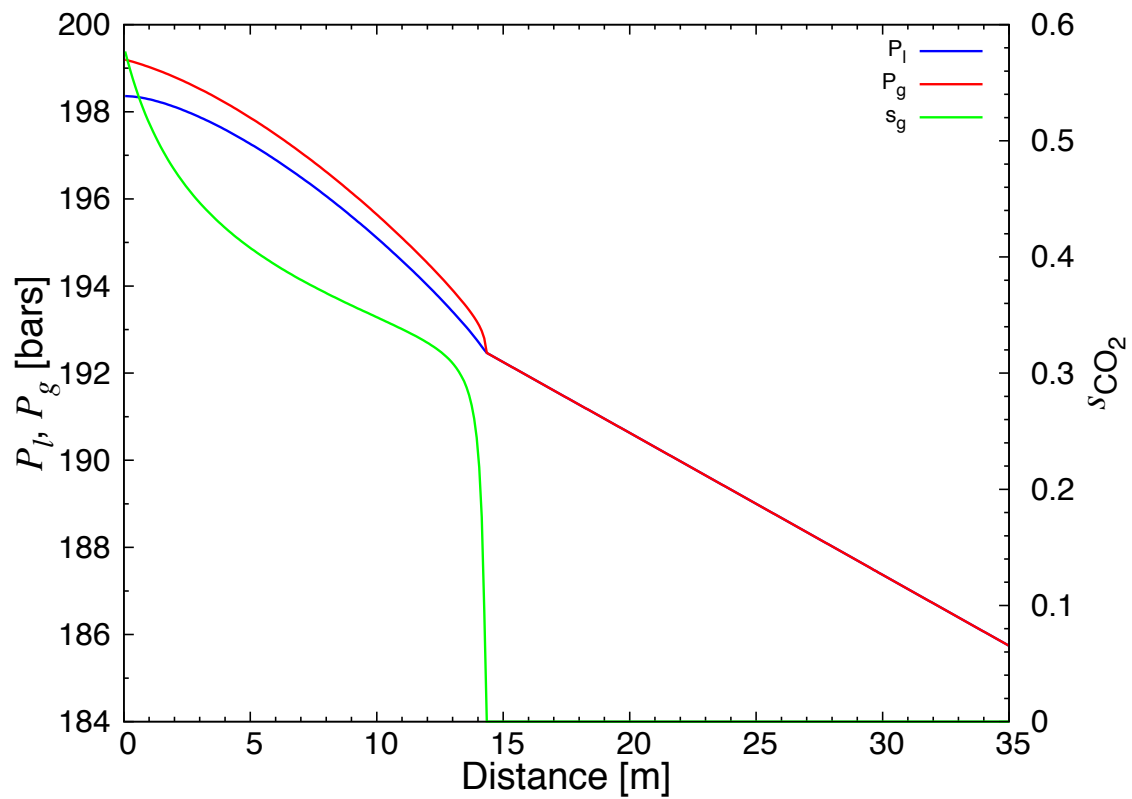


Figure 7-10: Liquid (blue curve) and supercritical CO₂ (red curve) pressures predicted by PFLOTTRAN after an elapsed time of 0.4 y corresponding to the end of injection. Also shown is the CO₂ saturation (green curve).

7.4.2 CO₂ Sequestration in the Presence of a Leaky Well

The simulation domain has a lateral extent of $1,000 \times 1,000$ m and vertical height of 160 m. The leaky well is at the center of the domain and the injection well is 100 m east. There are two aquifers at the top and bottom of the domain, each 30 m thick, and an aquitard with thickness of 100 m sandwiched between the two aquifers. The leaky well is modeled as a porous medium with a higher permeability compared to the formation. Parameter values used in the simulation are listed in Table 7-11. Other parameters used for characteristic curves, heat conduction, etc. may be found in the input file listing (see Table 7.4.2).

The initial conditions consist of hydrostatic pressure, and isothermal temperature of 34°C. The initial pressure at the bottom of the domain is 3.086×10^7 Pa (at 3,000 m depth). At the lateral boundaries, hydrostatic boundary conditions are imposed on the system. The boundaries at the top and bottom of the domain are no flow boundary conditions. CO₂ is injected at a constant rate of 8.87 kg/s for the duration of the simulation of 1000 days and at a constant temperature of 33.6°C.

The computational domain was discretized into $200 \times 200 \times 32$ grid blocks with spacing $\Delta x = \Delta y = 5$ m, and $\Delta z = 5$ m. The total number of degrees of freedom are 3,840,000. The problem was run on 512 processes on the supercomputer Yellowstone at the NCAR-Wyoming Supercomputing Center.

Table 7-11: Model parameters.

Unit	Permeability [m ²]	Porosity [—]	Depth [m]
Aquifer	2×10^{-14}	0.15	0–30, 130–160
Aquitard	1×10^{-18}	0.15	30–130
Leaky well	1×10^{-12}	0.15	0–160

Results of the simulation for an elapsed time of 250 days are shown in Figure 7-11 for liquid pressure and saturation of supercritical CO₂. Supercritical CO₂ proceeds up the leaky well until it ponds at the top of the domain where a closed boundary is imposed.

The leakage of CO₂ through the leaky well as a function of time is shown in Figure 7-12. This is defined as the CO₂ mass flow midway between the top and bottom domain divided by the injection rate. The maximum value in the leak occurs at approximately 800 d. The leak begins at approximately 50 d. The results can be compared to Ebigo et al. (2007), Figure 8. It should be noted that the leakage rate is highly sensitive to the lateral grid spacing.

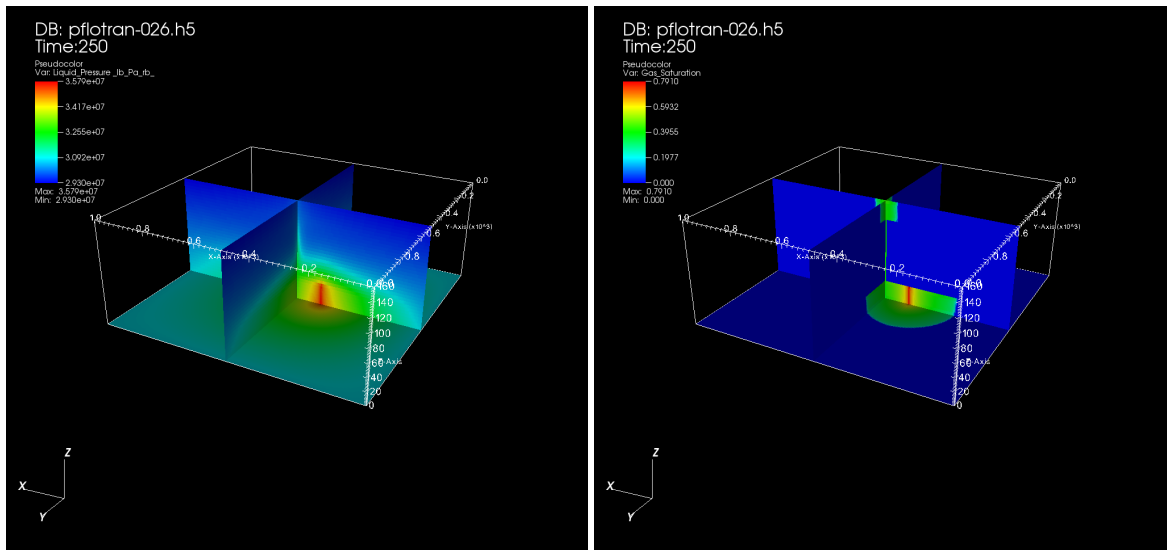


Figure 7-11: Left: pressure; right: supercritical CO₂ saturation; for an elapsed time of 250 days.

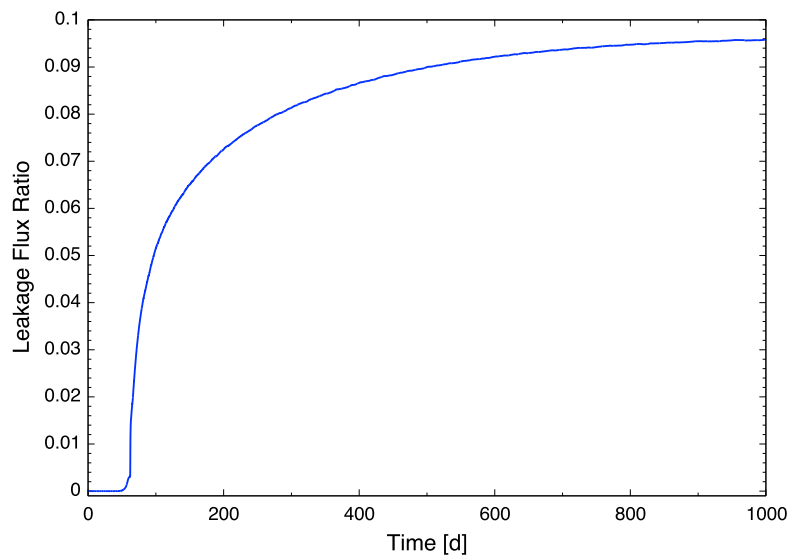


Figure 7-12: Leakage rate relative to injection rate.

Table 7-12: Input file for 3D CO₂ sequestration example problem with a leaky well.

```

SIMULATION
  SIMULATION_TYPE SUBSURFACE
  PROCESS_MODELS
    SUBSURFACE_FLOW flow
    MODE MPHASE
  /
/
END

SUBSURFACE

#=====
#CO2_DATABASE /Users/lichtner/pfplotran/pfplotran-dev/database/co2data0.dat
CO2_DATABASE ./co2data0.dat

#===== define the grid =====
GRID
  TYPE structured
  ORIGIN 0.d0 0.d0 0.d0
  NXYZ 200 200 32
  BOUNDS
    0.d0    0.d0    0.d0
    1000.d0 1000.d0 160.d0
  /
END

#===== solver options =====
TIMESTEPPER FLOW
  TS_ACCELERATION 8
  MAX_PRESSURE_CHANGE 5.e6
  MAX_TEMPERATURE_CHANGE 2.
  MAX_CONCENTRATION_CHANGE 1.e-3
  MAX_SATURATION_CHANGE 0.1
# MAX_STEPS 1
/

NEWTON_SOLVER FLOW
  ATOL 1D-8
  RTOL 1D-8
  STOL 1D-30
  DTOL 1D15
  ITOL 1D-8
  MAXIT 25
  MAXF 100
  MAX_NORM 1d6
END

#===== times =====
TIME
  FINAL_TIME 1000.d0 d
  INITIAL_TIMESTEP_SIZE 1.d-6 d
  MAXIMUM_TIMESTEP_SIZE 1.0 d
  MAXIMUM_TIMESTEP_SIZE 10.0 d at 100 d
/

#===== output options =====
OUTPUT
  TIMES d 0.15 0.25 0.5 0.75 1.0 2.5 5.0 10.0 15. 20.0 \
  25. 30.0 35. 40.0 45. 50.0 60. 70. 80. 90. \
  100. 125. 150. 175 200. 250. 300. 350. 400. 450. \
  500. 550. 600. 650. 700. 750. 800. 850. 900. 950. \
  1000.
# PERIODIC TIME 0.1 d
# SCREEN PERIODIC 50
FORMAT HDF5 MULTIPLE_FILES
FORMAT TECPLOT BLOCK
FLUXES

```

```

    MASS_BALANCE
    PRINT_COLUMN_IDS
    PERIODIC_OBSERVATION TIMESTEP 1
/

#===== fluid properties =====
FLUID_PROPERTY
    LIQUID_DIFFUSION_COEFFICIENT 1.d-9
    GAS_DIFFUSION_COEFFICIENT 2.13e-5
/

!===== intergral flux =====
INTEGRAL_FLUX leakage_plane
    NAME flx
    COORDINATES
        0.d0 0.d0 80.d0
        1000.d0 1000.d0 80.d0
    /
/

#===== material properties=====
MATERIAL_PROPERTY aquifer
    ID 1
    POROSITY 0.15d0
    TORTUOSITY 1.d0
    ROCK_DENSITY 2.84E3
    SPECIFIC_HEAT 1E3
    THERMAL_CONDUCTIVITY_DRY 0.5
    THERMAL_CONDUCTIVITY_WET 0.5
    SATURATION_FUNCTION sf
    PERMEABILITY
        PERM_X 2.d-14
        PERM_Y 2.d-14
        PERM_Z 2.d-14
    /
/

MATERIAL_PROPERTY aquitard
    ID 2
    POROSITY 0.15d0
    TORTUOSITY 1.d0
    ROCK_DENSITY 2.84E3
    SPECIFIC_HEAT 1E3
    THERMAL_CONDUCTIVITY_DRY 0.5
    THERMAL_CONDUCTIVITY_WET 0.5
    SATURATION_FUNCTION sf
    PERMEABILITY
        PERM_X 1.d-18
        PERM_Y 1.d-18
        PERM_Z 1.d-18
    /
/

MATERIAL_PROPERTY well
    ID 3
    POROSITY 0.15d0
    TORTUOSITY 1.d0
    ROCK_DENSITY 2.84E3
    SPECIFIC_HEAT 1E3
    THERMAL_CONDUCTIVITY_DRY 0.5
    THERMAL_CONDUCTIVITY_WET 0.5
    SATURATION_FUNCTION sf
    PERMEABILITY
        PERM_X 1.d-12
        PERM_Y 1.d-12
        PERM_Z 1.d-12
    /
/

#===== saturation functions =====
SATURATION_FUNCTION sf

```

```

    SATURATION_FUNCTION_TYPE BROOKS_COREY
    RESIDUAL_SATURATION LIQUID_PHASE 0.2
    RESIDUAL_SATURATION GAS_PHASE 0.05
    LAMBDA 2.0
    ALPHA 1d-4
    MAX_CAPILLARY_PRESSURE 5.0d5
/

```

```

#===== regions ===

```

```

REGION all
  COORDINATES
    0.d0    0.d0    0.d0
    1000.d0 1000.d0 160.d0
  /
END

```

```

REGION upper
  COORDINATES
    0.d0    0.d0    130.d0
    1000.d0 1000.d0 160.d0
  /
END

```

```

REGION middle
  COORDINATES
    0.d0    0.d0    30.d0
    1000.d0 1000.d0 130.d0
  /
END

```

```

REGION lower
  COORDINATES
    0.d0    0.d0    0.d0
    1000.d0 1000.d0 30.d0
  /
END

```

```

REGION top
  FACE TOP
  COORDINATES
    0.d0    0.d0    160.d0
    1000.d0 1000.d0 160.d0
  /
END

```

```

REGION bottom
  FACE BOTTOM
  COORDINATES
    0.d0    0.d0    0.d0
    1000.d0 1000.d0 0.d0
  /
END

```

```

REGION west
  FACE WEST
  COORDINATES
    0.d0 0.d0    0.d0
    0.d0 1000.d0 160.d0
  /
END

```

```

REGION east
  FACE EAST
  COORDINATES
    1000.d0 0.d0    0.d0
    1000.d0 1000.d0 160.d0
  /
END

```

```

REGION south
  FACE SOUTH

```

```

COORDINATES
  0.d0    0.d0 0.d0
  1000.d0 0.d0 160.d0
/
END

REGION north
FACE NORTH
COORDINATES
  0.d0    1000.d0 0.d0
  1000.d0 1000.d0 160.d0
/
END

REGION well
COORDINATES
  500.d0 500.d0 0.d0
  500.d0 500.d0 160.d0
/
END

REGION inj_well
COORDINATES
  400.d0 500.d0 0.d0
  400.d0 500.d0 30.d0
/
END

#==== flow conditions ====
FLOW_CONDITION initial
  UNITS Pa,C,M,yr
  TYPE
    PRESSURE hydrostatic
    TEMPERATURE dirichlet
    CONCENTRATION dirichlet
    ENTHALPY dirichlet
  /
  IPHASE 1
  # DATUM 0.d0 0.d0 200.d0
  PRESSURE 3.086d7 3.086d7
  TEMPERATURE 33.6d0
  CONCENTRATION 0.d0
  ENTHALPY 0.d0 0.d0
/

FLOW_CONDITION side
  UNITS Pa,C,M,yr
  TYPE
    PRESSURE hydrostatic
    TEMPERATURE zero_gradient
    CONCENTRATION zero_gradient
    ENTHALPY dirichlet
  /
  IPHASE 1
  PRESSURE 3.086d7 3.086d7
  TEMPERATURE 33.6d0
  CONCENTRATION 0.d0
  ENTHALPY 0.d0 0.d0
/

FLOW_CONDITION source
  UNITS Pa,C,M,yr
  TYPE
    RATE mass_rate
    TEMPERATURE dirichlet
    CONCENTRATION dirichlet
    ENTHALPY dirichlet
  /
  RATE LIST
    TIME_UNITS d
    DATA_UNITS kg/s

```

```
0.    0. 1.47833 != 8.87 kg/s / 6
1000. 0. 0.
/
TEMPERATURE 33.6d0
CONCENTRATION 0.d0
ENTHALPY 0.d0 0.d0
/

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

BOUNDARY_CONDITION north
  FLOW_CONDITION side
  REGION north
END

BOUNDARY_CONDITION south
  FLOW_CONDITION side
  REGION south
END

BOUNDARY_CONDITION east
  FLOW_CONDITION side
  REGION east
END

BOUNDARY_CONDITION west
  FLOW_CONDITION side
  REGION west
END

SOURCE_SINK
  FLOW_CONDITION source
  REGION inj_well
END

#==== stratigraphy couplers ==
STRATA
  REGION upper
  MATERIAL aquifer
END

STRATA
  REGION middle
  MATERIAL aquitard
END

STRATA
  REGION lower
  MATERIAL aquifer
END

STRATA
  REGION well
  MATERIAL well
END

END_SUBSURFACE
```


8 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 (Birkhäuser 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.
- Ebigbo, A., Holger Class, H., Helmig, R. (2007) CO₂ leakage through an abandoned well: problem-oriented benchmarks, *Comput Geosciences* 11:103–115 DOI 10.1007/s10596-006-9033-7.
- 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) PFLOTTRAN: 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. (1996a) Continuum Formulation of Multicomponent-Multiphase Reactive Transport, In: *Reactive Transport in Porous Media* (eds. P. C. Lichtner, C. I. Steefel, and E. H. Oelkers), *Reviews in Mineralogy*, **34**, 1–81.
- Lichtner P.C. (1996b) Modeling Reactive Flow and Transport in Natural Systems, Proceedings of the Rome Seminar on Environmental Geochemistry, Eds. G. Ottonello and L. Marini, Castelnovo di Porto, May 22–26, Pacini Editore, Pisa, Italy, 5–72.
- 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.

Appendix A: Governing Equations

A-1 Mode: RICHARDS

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

$$\frac{\partial}{\partial t} (\varphi s \rho) + \nabla \cdot (\rho \mathbf{q}) = Q_w, \quad (\text{A-1})$$

with Darcy flux \mathbf{q} defined as

$$\mathbf{q} = -\frac{k k_r(s)}{\mu} \nabla (P - W_w \rho g z). \quad (\text{A-2})$$

Here, φ denotes porosity [-], s saturation [$\text{m}^3 \text{m}^{-3}$], ρ water density [kmol m^{-3}], \mathbf{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, Brooks-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 [$\text{kmol m}^{-3} \text{s}^{-1}$] has the form

$$Q_w = \frac{q_M}{W_w} \delta(\mathbf{r} - \mathbf{r}_{ss}), \quad (\text{A-3})$$

where q_M denotes a mass rate in $\text{kg/m}^3/\text{s}$, and \mathbf{r}_{ss} denotes the location of the source/sink.

A-2 Capillary Pressure Relations

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

$$s_e = \left[1 + \left(\frac{p_c}{p_c^0} \right)^n \right]^{-m}, \quad (\text{A-4})$$

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

$$s_e = \frac{s - s_r}{s_0 - s_r}, \quad (\text{A-5})$$

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

$$p_c = p_c^0 (s_e^{-1/m} - 1)^{1/n}. \quad (\text{A-6})$$

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

A-2.1 Brooks-Corey Saturation Function

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

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

with $\lambda = mn$ and inverse relation

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

A-2.2 Relative Permeability

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

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

for the Mualem formulation and by

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

for the Burdine formulation.

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

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

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

$$k_r = (s_e)^{5/2+2/\lambda}, \quad (\text{A-12})$$

$$= (p_c/p_c^0)^{-(5\lambda/2+2)}. \quad (\text{A-13})$$

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

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

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

$$k_r = (s_e)^{2+3/\lambda}, \quad (\text{A-15})$$

$$= \left(\frac{p_c}{p_c^0} \right)^{-(2+3\lambda)}. \quad (\text{A-16})$$

A-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

$$a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3 = f_1, \quad (\text{A-17a})$$

$$a_0 + a_1x_2 + a_2x_2^2 + a_3x_2^3 = f_2, \quad (\text{A-17b})$$

$$a_1x_1 + 2a_2x_1 + 3a_3x_1^2 = f'_1, \quad (\text{A-17c})$$

$$a_1x_2 + 2a_2x_2 + 3a_3x_2^2 = f'_2, \quad (\text{A-17d})$$

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

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 0 & 1 & 2x_1 & 3x_1^2 \\ 0 & 1 & 2x_2 & 3x_2^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f'_1 \\ f'_2 \end{bmatrix}. \quad (\text{A-18})$$

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 $f = k_r(s_e)$, $x_1 = 1$, $x_2 = 0.99$, $f_1 = 1$, $f_2 = (k_r)_2$, $f'_1 = 0$, $f'_2 = (k'_r)_2$.

A-3 Mode: MPHASE

The mode MPHASE solves the two-phase system of water and supercritical CO₂. It may also be coupled to chemistry using the CHEMISTRY keyword and its various associated optional and required keywords for selecting the primary and secondary aqueous species and setting up initial and boundary conditions and source/sinks. MPHASE requires that the species CO₂(aq) be used as primary species. In addition, for pure aqueous and supercritical CO₂ phases, the input to MPHASE requires specifying the mole fraction of CO₂. When coupled to chemistry, the CO₂ mole fraction is calculated internally directly from the aqueous concentrations specified in the CONSTRAINT keyword.

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

$$\frac{\partial}{\partial t} \left(\varphi \sum_{\alpha} s_{\alpha} \eta_{\alpha} x_i^{\alpha} \right) + \nabla \cdot \sum_{\alpha} \mathbf{F}_i^{\alpha} = Q_i, \quad (\text{A-19a})$$

for the i th component where the flux \mathbf{F}_i^{α} is given by

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

and

$$\frac{\partial}{\partial t} \left(\varphi \sum_{\alpha} s_{\alpha} \eta_{\alpha} U_{\alpha} + (1 - \varphi) \rho_r c_r T \right) + \nabla \cdot \sum_{\alpha} \left[\mathbf{q}_{\alpha} \eta_{\alpha} H_{\alpha} - \kappa \nabla T \right] = Q_e, \quad (\text{A-19c})$$

for energy. In these equations α designates a fluid phase ($\alpha = l, \text{sc}$) 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 ($i = \text{H}_2\text{O}, \text{CO}_2$); 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

$$\sum_i x_i^{\alpha} = 1, \quad (\text{A-20})$$

the quantities η_{α} , H_{α} , U_{α} refer to the molar density, enthalpy, and internal energy of each fluid phase, respectively; and \mathbf{q}_{α} denotes the Darcy flow rate for phase α defined by

$$\mathbf{q}_{\alpha} = -\frac{k k_{\alpha}}{\mu_{\alpha}} \nabla (P_{\alpha} - \rho_{\alpha} g z), \quad (\text{A-21})$$

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

$$\rho_{\alpha} = W_{\alpha} \eta_{\alpha}, \quad (\text{A-22})$$

where the formula weight W_{α} is a function of composition according to the relation

$$W_{\alpha} = \frac{\rho_{\alpha}}{\eta_{\alpha}} = \sum_i W_i x_i^{\alpha}. \quad (\text{A-23})$$

The quantities ρ_r , c_r , and κ refer to the mass density, heat capacity, and thermal conductivity of the porous rock.

A-3.1 Source/Sink Terms

The source/sink terms, Q_i and Q_e , describe injection and extraction of mass and heat, respectively, for various well models. Several different well models are available. The simplest is a volume or mass rate injection/production well given by

$$Q_i = \sum_n \sum_{\alpha} q_{\alpha}^V \eta_{\alpha} x_i^{\alpha} \delta(\mathbf{r} - \mathbf{r}_n), \quad (\text{A-24a})$$

$$= \sum_n \sum_{\alpha} \frac{\eta_{\alpha}}{\rho_{\alpha}} q_{\alpha}^M x_i^{\alpha} \delta(\mathbf{r} - \mathbf{r}_n), \quad (\text{A-24b})$$

$$= \sum_n \sum_{\alpha} W_{\alpha}^{-1} q_{\alpha}^M x_i^{\alpha} \delta(\mathbf{r} - \mathbf{r}_n), \quad (\text{A-24c})$$

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

$$q_{\alpha}^M = \rho_{\alpha} q_{\alpha}^V. \quad (\text{A-25})$$

Table A-1: Choice of primary variables.

State	X_1	X_2	X_3
Liquid	p_l	T	$x_{\text{CO}_2}^l$
Gas	p_g	T	$x_{\text{CO}_2}^g$
Two-Phase	p_g	T	s_g

The position vector \mathbf{r}_n refers to the location of the n th source/sink.

A less simplistic approach is to specify the bottom well pressure to regulate the flow rate in the well. In this approach the mass flow rate is determined from the expression

$$q_\alpha^M = \Gamma \rho_\alpha \frac{k_\alpha}{\mu_\alpha} (p_\alpha - p_\alpha^{\text{bw}}), \quad (\text{A-26})$$

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

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

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

$$r_e = \sqrt{A/\pi}. \quad (\text{A-28})$$

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

A-3.2 Variable Switching

In PFLOTTRAN 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 $N_{\text{int}} = N_C - N_P + 2$, where N_P denotes the number of phases. The extensive degrees of freedom equals $N_{\text{ext}} = N_P - 1$. This gives a total number of degrees of freedom $N_{\text{dof}} = N_{\text{int}} + N_{\text{ext}} = N_C + 1$, independent of the number of phases N_P in the system. Primary variables for liquid, gas and two-phase systems are listed in Table A-1. The conditions for phase changes to occur are considered in detail below.

A-3.2.1 Gas: $(p_g, T, x_{\text{CO}_2}^g) \rightarrow$ Two-Phase: (p_g, T, s_g)

$$\bullet \text{ gas} \rightarrow \text{2-ph: } x_{\text{CO}_2}^g \leq 1 - \frac{P_{\text{sat}}(T)}{p_g}, \text{ or equivalently: } x_{\text{H}_2\text{O}}^g \geq \frac{P_{\text{sat}}(T)}{p_g}$$

A-3.2.2 Liquid: $(p_l, T, x_{\text{CO}_2}^l) \rightarrow$ **Two-phase:** (p_g, T, s_g)

- liq \rightarrow 2-ph: $x_{\text{CO}_2}^l \geq x_{\text{CO}_2}^{eq}$

The equilibrium mole fraction $x_{\text{CO}_2}^{eq}$ is given by

$$x_{\text{CO}_2}^{eq} = \frac{m_{\text{CO}_2}}{W_{\text{H}_2\text{O}}^{-1} + m_{\text{CO}_2} + \sum_{l \neq \text{H}_2\text{O}, \text{CO}_2} m_l}, \quad (\text{A-29})$$

where the molality at equilibrium is given by

$$m_{\text{CO}_2}^{eq} = \left(1 - \frac{P_{\text{sat}}(T)}{p}\right) \frac{\phi_{\text{CO}_2} p}{K_{\text{CO}_2} \gamma_{\text{CO}_2}}, \quad (\text{A-30})$$

where it is assumed that

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

A-3.2.3 Two-Phase: $(p_g, T, s_g) \rightarrow$ **Liquid** $(p_l, T, x_{\text{CO}_2}^l)$ **or Gas** $(p_g, T, x_{\text{CO}_2}^g)$

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

$$f_{\text{CO}_2} = y_{\text{CO}_2} \phi_{\text{CO}_2} p_g = K_{\text{CO}_2} (\gamma_{\text{CO}_2} m_{\text{CO}_2}), \quad (\text{A-32})$$

where

$$y_{\text{CO}_2} = x_{\text{CO}_2}^g, \quad (\text{A-33})$$

$$x_{\text{H}_2\text{O}}^g = \frac{P_{\text{sat}}(T)}{p_g}, \quad (\text{A-34})$$

and

$$y_{\text{CO}_2} = 1 - x_{\text{H}_2\text{O}}^g = 1 - \frac{P_{\text{sat}}(T)}{p_g}. \quad (\text{A-35})$$

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

$$y_{\text{CO}_2} = \tilde{K}_{\text{CO}_2} x_{\text{CO}_2}, \quad (\text{A-36})$$

where

$$x_{\text{CO}_2} = x_{\text{CO}_2}^l, \quad (\text{A-37})$$

$$\tilde{K}_{\text{CO}_2} = \frac{\gamma_{\text{CO}_2} K_{\text{CO}_2}}{\phi_{\text{CO}_2} p_g} \frac{m_{\text{CO}_2}}{x_{\text{CO}_2}}. \quad (\text{A-38})$$

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

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

Molality–mole fraction:

$$m_i = \frac{n_i}{M_{\text{H}_2\text{O}}} = \frac{n_i}{W_{\text{H}_2\text{O}} n_{\text{H}_2\text{O}}} = \frac{x_i}{W_{\text{H}_2\text{O}} x_{\text{H}_2\text{O}}} = \frac{x_i}{W_{\text{H}_2\text{O}} (1 - \sum_{l \neq \text{H}_2\text{O}} x_l)} \quad (\text{A-39})$$

Mole fraction–molality:

$$x_i = \frac{n_i}{N} = \frac{n_i}{M_{\text{H}_2\text{O}}} \frac{M_{\text{H}_2\text{O}}}{N} = \frac{m_i}{\sum m_l} = \frac{W_{\text{H}_2\text{O}} m_i}{1 + W_{\text{H}_2\text{O}} \sum_{l \neq \text{H}_2\text{O}} m_l} \quad (\text{A-40})$$

Mole fraction–mass fraction:

$$x_i = \frac{n_i}{N} = \frac{W_i^{-1} W_i n_i}{\sum W_l^{-1} W_l n_l} = \frac{W_i^{-1} w_i}{\sum W_l^{-1} w_l} \quad (\text{A-41})$$

Mass fraction–mole fraction:

$$w_i = \frac{M_i}{M} = \frac{W_i n_i}{\sum W_l n_l} = \frac{W_i x_i}{\sum W_l x_l} \quad (\text{A-42})$$

A-3.3 Sequentially Coupling MPHASE and CHEMISTRY

MPHASE and CHEMISTRY may be sequentially coupled to one another by including the CHEMISTRY keyword in the MPHASE input file and adding the requisite associated keywords. At the end of an MPHASE time step the quantities p , T , s_g , q_l and q_g are passed to the reactive transport equations. These quantities are interpolated between the current time t_{MPH} and the new time $t_{\text{MPH}} + \Delta t_{\text{MPH}}$. The reactive transport equations may need to sub-step the MPHASE time step, i.e. $\Delta t_{\text{RT}} \leq \Delta t_{\text{MPH}}$. Coupling also occurs from the reactive transport equations back to MPHASE. This is through changes in material properties such as porosity, tortuosity and permeability caused by mineral precipitation and dissolution reactions (see §A-9.1.2). In addition, coupling occurs through consumption and production of H_2O and CO_2 by mineral precipitation/dissolution reactions occurring in the reactive transport equations. This effect is accounted for by passing the reaction rates $R_{\text{H}_2\text{O}}$ and R_{CO_2} given by

$$R_j = - \sum_m \nu_{jm} I_m, \quad (\text{A-43})$$

back to the MPHASE conservation equations.

A further constraint on the reactive transport equations for aqueous CO_2 is that it must be in equilibrium with supercritical CO_2 in regions where $0 < s_g < 1$. This is accomplished by replacing the CO_2 mass conservation equations in those regions with the constraint $m_{\text{CO}_2(\text{aq})} = m_{\text{CO}_2}^{\text{eq}}$.

A-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

$$\frac{\partial}{\partial t}(\varphi \rho_\alpha s_\alpha) + \nabla \cdot (\rho_\alpha \mathbf{q}_\alpha) = Q_\alpha, \quad (\text{A-44})$$

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

$$\mathbf{q}_\alpha = -\frac{k k_\alpha}{\mu_\alpha} (\nabla p - \rho_\alpha g \hat{\mathbf{z}}), \quad (\text{A-45})$$

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_α , $\alpha = 1, \dots, n-1$ with

$$\sum_{\alpha=1}^n s_\alpha = 1. \quad (\text{A-46})$$

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

$$\frac{\partial}{\partial t} \left(\varphi \sum_{\alpha} s_\alpha \rho_\alpha U_\alpha + (1 - \varphi) \rho_r C_r T \right) + \nabla \cdot \left(\sum_{\alpha} \rho_\alpha \mathbf{q}_\alpha H_\alpha - \kappa \nabla T \right) = Q_e, \quad (\text{A-47})$$

where U_α , H_α denote the internal energy and enthalpy of the α th 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: $(p, T, s_1, \dots, s_{n-1})$.

A-5 Mode: MISCIBLE

The miscible mode applies to a mixture of water and propylene 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

$$\frac{\partial}{\partial t} \varphi \eta x_i + \nabla \cdot [\mathbf{q} \eta x_i - \varphi D \eta \nabla x_i] = Q_i, \quad (\text{A-48})$$

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

$$\eta = W^{-1} \rho, \quad (\text{A-49})$$

and

$$W_i \eta x_i = \rho y_i. \quad (\text{A-50})$$

It follows that

$$W_i \eta \nabla x_i = \rho \nabla y_i + \rho y_i \nabla \ln W. \quad (\text{A-51})$$

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

Simple equations of state are provided for density [g/cm³], viscosity [Pa s], and diffusivity [m²/s]. The density is a function of both composition and pressure with the form

$$\rho(y_1, p) = \rho(y_1, p_0) + \left. \frac{\partial \rho}{\partial p} \right|_{p=p_0} (p - p_0), \quad (\text{A-52})$$

$$= \rho(y_1, p_0) (1 + \beta(p - p_0)), \quad (\text{A-53})$$

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

$$\beta = \left. \frac{1}{\rho} \frac{\partial \rho}{\partial p} \right|_{p=p_0}, \quad (\text{A-54})$$

$$= 4.49758 \times 10^{-10} y_1 + 5 \times 10^{-10} (1 - y_1), \quad (\text{A-55})$$

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

$$\rho(y_1, p_0) = \left(((0.0806 y_1 - 0.203) y_1 + 0.0873) y_1 + 1.0341 \right) 10^3, \quad (\text{A-56})$$

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

$$\mu(y_1) = 10^{(1.6743(1-y_1)-0.0758)} 10^{-3}, \quad (\text{A-57})$$

and

$$D(y_1) = \left((((-4.021 y_1 + 9.1181) y_1 - 5.9703) y_1 + 0.4043) y_1 + 0.5687 \right) 10^{-9}, \quad (\text{A-58})$$

The mass fraction is related to mole fraction according to

$$y_1 = \frac{x_1 W_{\text{H}_2\text{O}}}{W}, \quad (\text{A-59})$$

where the mean formula weight W is given by

$$W = x_1 W_{\text{H}_2\text{O}} + x_2 W_{\text{PPG}}, \quad (\text{A-60})$$

with formula weights for water and propylene glycol equal to $W_{\text{H}_2\text{O}} = 18.01534$ and $W_{\text{PPG}} = 76.09$ [kg/kmol].

Global mass conservation satisfies the relation

$$\frac{d}{dt} M_i = - \int \mathbf{F}_i \cdot d\mathbf{S} + \int Q_i dV, \quad (\text{A-61})$$

with

$$M_i = \int \varphi \eta x_i dV. \quad (\text{A-62})$$

In terms of mass fractions and mass density

$$M_i^m = W_i M_i = \int \varphi \rho y_i dV. \quad (\text{A-63})$$

A-6 Mode: Air-Water

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

$$\frac{\partial}{\partial t} \varphi \left(s_l \rho_l x_i^l + s_g \rho_g x_i^g \right) + \nabla \cdot \left(\mathbf{q}_l \rho_l x_i^l + \mathbf{q}_g \rho_g x_i^g - \varphi s_l D_l \rho_l \nabla x_i^l - \varphi s_g D_g \rho_g \nabla x_i^g \right) = Q_i, \quad (\text{A-64})$$

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

$$\sum_{\alpha=l,g} \left\{ \frac{\partial}{\partial t} (\varphi s_\alpha \rho_\alpha U_\alpha) + \nabla \cdot (\mathbf{q}_\alpha \rho_\alpha H_\alpha) \right\} + \frac{\partial}{\partial t} \left((1 - \varphi) \rho_r C_p T \right) - \nabla \cdot (\kappa \nabla T) = Q, \quad (\text{A-65})$$

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

$$U_\alpha = H_\alpha - \frac{P_\alpha}{\rho_\alpha}. \quad (\text{A-66})$$

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

$$\kappa = \kappa_{\text{dry}} + \sqrt{s_l} (\kappa_{\text{sat}} - \kappa_{\text{dry}}), \quad (\text{A-67})$$

where κ_{dry} and κ_{sat} are dry and fully saturated rock thermal conductivities.

A-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 A-9).

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

$$\frac{\partial}{\partial t} (\varphi s \rho) + \nabla \cdot (\rho \mathbf{q}) = Q_w, \quad (\text{A-68})$$

and

$$\frac{\partial}{\partial t} (\varphi s \rho U + (1 - \varphi) \rho_p c_p T) + \nabla \cdot (\rho \mathbf{q} H - \kappa \nabla T) = Q_e, \quad (\text{A-69})$$

The Darcy flow velocity \mathbf{q} is given by

$$\mathbf{q} = -\frac{k k_r}{\mu} \nabla (P - W \rho g z). \quad (\text{A-70})$$

Here, φ denotes porosity, s saturation, ρ mixture density of the brine, \mathbf{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, Brooks-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

$$U = H - \frac{P}{\rho}. \quad (\text{A-71})$$

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

$$\kappa = \kappa_{\text{dry}} + \sqrt{s_l}(\kappa_{\text{sat}} - \kappa_{\text{dry}}), \quad (\text{A-72})$$

where κ_{dry} and κ_{sat} are dry and fully saturated rock thermal conductivities.

A-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

$$\begin{aligned} \frac{\partial}{\partial t} [\phi (s_l \eta_l X_w^l + s_g \eta_g X_w^g + s_i \eta_i X_w^i)] + \nabla \cdot [X_w^l \mathbf{v}_l \eta_l + X_w^g \eta_g \mathbf{v}_g] \\ - \nabla \cdot [\phi s_g \tau_g \eta_g D_g \nabla X_w^g] = Q_w, \end{aligned} \quad (\text{A-73a})$$

$$\begin{aligned} \frac{\partial}{\partial t} [\phi (s_l \eta_l U_l + s_g \eta_g U_g + s_i \eta_i U_i) + (1 - \phi) \rho_r c_r T] + \nabla \cdot [\mathbf{v}_l \eta_l H_l + \mathbf{v}_g \eta_g H_g] \\ - \nabla \cdot [\kappa \nabla T] = Q_e, \end{aligned} \quad (\text{A-73b})$$

where the subscripts l , i , g denote the liquid, ice and gas phases, respectively; ϕ is the porosity; s_α ($\alpha = i, l, g$) is the saturation of the α -th phase; η_α ($\alpha = i, l, g$) 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^α ($\alpha = i, l, g$) 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_α ($\alpha = i, l, g$) is the molar internal energy of the α -th phase; H_α ($\alpha = l, g$) is the molar enthalpy of the α -th phase; Q_e is the heat source; $\nabla (\cdot)$ is the gradient operator; $\nabla \cdot (\cdot)$ is the divergence operator.

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

$$\mathbf{v}_g = -\frac{k_{rg}k}{\mu_g} \nabla [p_g - \rho_g g z], \quad (\text{A-74a})$$

$$\mathbf{v}_l = -\frac{k_{rl}k}{\mu_l} \nabla [p_l - \rho_l g z], \quad (\text{A-74b})$$

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

$$s_l + s_g + s_i = 1. \quad (\text{A-75})$$

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

$$X_a^l = 0, X_a^i = 0 \Rightarrow X_w^l = 1, X_w^i = 1, \quad (\text{A-76})$$

and so Eqns. (A-73), (A-74), based on the assumption that p_g is hydrostatic (i.e., $p_g = (p_g)_0 + \rho_g g z$), reduce to

$$\frac{\partial}{\partial t} [\phi (s_g \eta_g X_w^g + s_l \eta_l + s_i \eta_i)] + \nabla \cdot [\mathbf{v}_l \eta_l] - \nabla \cdot [\phi s_g \tau_g \eta_g D_g \nabla X_w^g] = Q_w, \quad (\text{A-77a})$$

$$\frac{\partial}{\partial t} [\phi (s_l \eta_l U_l + s_g \eta_g U_g + s_i \eta_i U_i) + (1 - \phi) \rho_r c_r T] + \nabla \cdot [\mathbf{v}_l \eta_l H_l] - \nabla \cdot [\kappa \nabla T] = Q_e, \quad (\text{A-77b})$$

$$\mathbf{v}_l = -\frac{k_{rl} k}{\mu_l} \nabla [p_l - \rho_l g z]. \quad (\text{A-77c})$$

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,

$$X_w^g = \frac{p_v}{p_g}, \quad (\text{A-78})$$

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

$$p_v = P_{\text{sat}}(T) \exp \left[\frac{P_{cgl}}{\eta_l R (T + 273.15)} \right], \quad (\text{A-79})$$

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):

$$\frac{s_l}{s_l + s_g} = S_*(P_{cgl}), \quad (\text{A-80a})$$

$$\frac{s_l}{s_l + s_i} = S_* \left[\frac{\sigma_{gl}}{\sigma_{il}} P_{cil} \right], \quad (\text{A-80b})$$

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_{iw}^0 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:

$$S_* = \begin{cases} [1 + (\alpha P_c)^\gamma]^{-\lambda}, & P_c > 0 \\ 1, & P_c \leq 0 \end{cases} \quad (\text{A-81})$$

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

$$k_{rl} = (s_l)^{\frac{1}{2}} \left[1 - \left(1 - (s_l)^{\frac{1}{\lambda}} \right)^\lambda \right]^2, \quad (\text{A-82})$$

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

The thermal conductivity for the frozen soil is chosen to be

$$\kappa = K e_f \kappa_{\text{wet},f} + K e_u \kappa_{\text{wet},u} + (1 - K e_u - K e_f) \kappa_{\text{dry}}, \quad (\text{A-83})$$

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

$$K e_f = (s_i)^{\alpha_f}, \quad (\text{A-84a})$$

$$K e_u = (s_l)^{\alpha_u}, \quad (\text{A-84b})$$

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 depend on temperature and pressure as follows:

$$D_g = D_g^0 \left(\frac{P_{\text{ref}}}{P} \right) \left(\frac{T}{T_{\text{ref}}} \right)^{1.8}, \quad (\text{A-85})$$

where D_g^0 is the reference diffusion coefficient at some reference temperature, T_{ref} , and pressure P_{ref} .

A-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 an integrated finite volume approach to develop equations for fracture (primary continuum) and matrix (secondary continua) temperatures T_α and T_β , respectively, with fracture volume fraction denoted by ϵ_α . 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.

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

$$\begin{aligned} \frac{\partial}{\partial t} \epsilon_\alpha \left[\varphi_\alpha \rho_\alpha U_\alpha + (1 - \varphi_\alpha) \rho_r C_r T_\alpha \right] + \nabla \cdot \left(\mathbf{q}_\alpha \rho_\alpha H_\alpha - \kappa_a \nabla T_\alpha \right) \\ = -A_{\alpha\beta} \kappa_{\alpha\beta} \frac{\partial T_\beta}{\partial n}, \end{aligned} \quad (\text{A-86a})$$

and

$$\frac{\partial}{\partial t} \rho_r C_r T_\beta + \frac{\partial}{\partial \xi} \left(-\kappa_\beta \frac{\partial T_\beta}{\partial \xi} \right) = 0, \quad (\text{A-86b})$$

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

$$T_\beta(\xi_M, t | \mathbf{r}) = T_\alpha(\mathbf{r}, t), \quad (\text{A-86c})$$

is imposed at the fracture-matrix interface, where ξ_M denotes the outer boundary of the matrix.

Using the control volume configuration shown in Figure A-2, 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 n th fracture control volume is given by

$$\begin{aligned} \left[\varphi_\alpha \left((\rho_\alpha U_\alpha)_n^{k+1} - (\rho_\alpha U_\alpha)_n^k \right) + (1 - \varphi_\alpha) \left((\rho_r C_r T_\alpha)_n^{k+1} - (\rho_r C_r T_\alpha)_n^k \right) \right] \frac{V_n^\alpha}{\Delta t} \\ + \sum_{n'} \left[(q_\alpha \rho_\alpha H_\alpha)_{nn'} + \frac{\kappa_{nn'}^\alpha}{d_n + d_{n'}} (T_{\alpha n} - T_{\alpha n'}) \right] A_{nn'}^\alpha \\ + \sum_{l=1}^{N_\beta} \frac{\kappa_{nM}^{\alpha\beta_l}}{d_n + d_M} (T_{\alpha n} - T_{\beta_l M}) A_{nM}^{\beta_l} = 0, \end{aligned} \quad (\text{A-87a})$$

where V_n^α denotes the fracture volume, and

$$\begin{aligned} \left((\rho_r C_r T_\beta)_m^{k+1} - (\rho_r C_r T_\beta)_m^k \right) \frac{V_m^\beta}{\Delta t} + \sum_{m'} \frac{\kappa_{mm'}^\beta}{d_m + d_{m'}} (T_{\beta m} - T_{\beta m'}) A_{mm'}^\beta \\ + \delta_{mM} \frac{\kappa_{nM}^{\alpha\beta}}{d_n + d_M} (T_{\alpha n} - T_{\beta M}) A_{nM}^\beta = 0, \end{aligned} \quad (\text{A-87b})$$

for the m th matrix node with volume V_m^β . The matrix node designated by M refers to the outer most node in contact with the fracture (see Figure A-1). More than one type of matrix geometry is included in the above equations as indicated by the sum over l in Eqn.(A-87a), where N_β denotes the number of different secondary continua. However, it should be noted that the current implementation in PFLOTRAN only allows coupling to a single secondary continuum ($N_\beta = 1$). The fracture volume V_n^α is related to the REV volume V_n by

$$\epsilon_\alpha = \frac{V_n^\alpha}{V_n}. \quad (\text{A-88})$$

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

$$\kappa_{ll'} = \frac{\kappa_l \kappa_{l'} (d_l + d_{l'})}{d_l \kappa_{l'} + d_{l'} \kappa_l}. \quad (\text{A-89})$$



Figure A-1: Node configuration for secondary (β) continua (nodes $1, \dots, l, \dots, M$), associated with the n th primary (α) continuum node.

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

$$\Delta \xi_m = \rho \Delta \xi_{m-1}, \quad (\text{A-90a})$$

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

$$l_M = 2 \sum_{m=1}^M \Delta \xi_m, \quad (\text{A-90b})$$

which requires that ρ satisfy the equation

$$\frac{l_M}{2\Delta \xi_1} = \frac{\rho^M - 1}{\rho - 1}, \quad (\text{A-90c})$$

with the inner and outer grid spacing related by

$$\Delta \xi_M = \rho^{M-1} \Delta \xi_1. \quad (\text{A-90d})$$

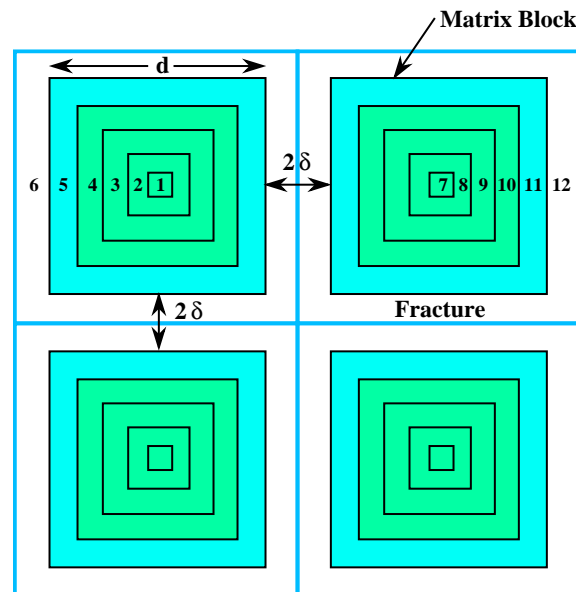


Figure A-2: Control volumes in DCDM multiple continuum model with fracture aperture 2δ and matrix block size d .

According to the geometry in Figure A-2 assuming a 3D orthogonal set of fractures,

$$V_n = (d + 2\delta)^3, \quad (\text{A-91})$$

Table A-2: DCDM geometric parameters.

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

and

$$V_n^\alpha = (d + 2\delta)^3 - d^3, \quad (\text{A-92})$$

giving

$$\epsilon_\alpha = 1 - \frac{d^3}{(d + 2\delta)^3} = 1 - \left(\frac{1}{1 + \frac{2\delta}{d}} \right)^3, \quad (\text{A-93a})$$

$$\simeq \frac{6\delta}{d}. \quad (\text{A-93b})$$

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

$$2\delta = d \left(\frac{1}{(1 - \epsilon_\alpha)^{1/3}} - 1 \right). \quad (\text{A-94})$$

A list of different sub-continua geometries and parameters implemented in PFLOTTRAN is given in Table A-2. Different independent and dependent parameters for the nested cube geometry are listed in Table A-3. 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 $(\epsilon_\alpha, 2\delta, l_m, l_f)$, where l_m denotes the matrix block size and l_f refers to the fracture spacing, two of which are independent.

The fracture-matrix interfacial area A_{nM} per unit volume is equal to

$$A_{nM}^\beta = \frac{\mathcal{N}_\beta}{V} A_\beta^0, \quad (\text{A-95})$$

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

$$\frac{\mathcal{N}_\beta}{V} = \frac{1}{V} \frac{V_\beta}{V_\beta^0} = \frac{\epsilon_\beta}{V_\beta^0}, \quad (\text{A-96})$$

and A_β^0 and V_β^0 refer to the area and volume of each geometric type as listed in Table A-2. The primary-secondary coupling term can then be written in the form

$$\sum_\beta \frac{\kappa_{nM}^{\alpha\beta}}{d_n + d_M} (T_n^\alpha - T_M^\beta) A_{nM}^\beta = V_n \sum_\beta \frac{\epsilon_\beta \kappa_{nM}^{\alpha\beta}}{d_n + d_M} (T_n^\alpha - T_M^\beta) \frac{A_\beta^0}{V_\beta^0}. \quad (\text{A-97})$$

Table A-3: Independent and dependent nested cube parameters.

Independent		Dependent	
ϵ_α	l_f	$2\delta = l_f - l_m$	$l_m = l_f(1 - \epsilon_\alpha)^{1/3}$
ϵ_α	l_m	$2\delta = l_f - l_m$	$l_f = l_m(1 - \epsilon_\alpha)^{-1/3}$
2δ	l_f	$\epsilon_\alpha = 1 - (l_m/l_f)^3$	$l_m = l_f - 2\delta$
2δ	l_m	$\epsilon_\alpha = 1 - (l_m/l_f)^3$	$l_f = l_m + 2\delta$
2δ	ϵ_α	$l_m = 2\delta \left(\frac{1}{(1 - \epsilon_\alpha)^{1/3}} - 1 \right)^{-1}$	$l_m = l - 2\delta$

A-9 Mode: Reactive Transport (Keyword CHEMISTRY)

The governing mass conservation equations for the geochemical transport mode for a multiphase system written in terms of a set of independent aqueous primary or basis species with the form

$$\frac{\partial}{\partial t} \left(\varphi \sum_{\alpha} s_{\alpha} \Psi_j^{\alpha} \right) + \nabla \cdot \sum_{\alpha} \Omega_j^{\alpha} = Q_j - \sum_m \nu_{jm} I_m - \frac{\partial S_j}{\partial t}, \quad (\text{A-98})$$

and

$$\frac{\partial \varphi_m}{\partial t} = \bar{V}_m I_m, \quad (\text{A-99})$$

for minerals with molar volume \bar{V}_m , mineral reaction rate I_m and mineral volume fraction φ_m referenced to an REV. Sums over α in Eqn.(A-98) are over all fluid phases in the system. The quantity Ψ_j^{α} denotes the total concentration of the j th primary species $\mathcal{A}_j^{\text{pri}}$ in the α th fluid phase defined by

$$\Psi_j^{\alpha} = \delta_{l\alpha} C_j^l + \sum_{i=1}^{N_{\text{sec}}} \nu_{ji}^{\alpha} C_i^{\alpha}, \quad (\text{A-100})$$

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

$$\sum_j \nu_{ji}^{\alpha} \mathcal{A}_j^l \rightleftharpoons \mathcal{A}_i^{\alpha}, \quad (\text{A-101})$$

yielding the mass action equations

$$C_i^{\alpha} = \frac{K_i^{\alpha}}{\gamma_i^{\alpha}} \prod_j \left(\gamma_j^l C_j^l \right)^{\nu_{ji}^{\alpha}}, \quad (\text{A-102})$$

with equilibrium constant K_i^{α} , and activity coefficients γ_k^{α} . For the molality of the k th aqueous species, the Debye-Hückel activity coefficient algorithm is given by

$$\log \gamma_k = -\frac{z_k^2 A \sqrt{I}}{1 + B \overset{\circ}{a}_k \sqrt{I}} + bI, \quad (\text{A-103})$$

and the Davies algorithm by the expression

$$\log \gamma_k = -\frac{z_k^2}{2} \left[\frac{\sqrt{I}}{1 + \sqrt{I}} - 0.3I \right]. \quad (\text{A-104})$$

with valence z_k , Debye-Hückel parameters A , B , and ionic radius $\overset{\circ}{a}_k$, and ionic strength I defined as

$$I = \frac{1}{2} \sum_{j=1}^{N_c} m_j^2 + \frac{1}{2} \sum_{i=1}^{N_{\text{sec}}} m_i^2, \quad (\text{A-105})$$

for molality m_j and m_i of primary and secondary species, respectively (note: $C_i^l = \rho_l y_w^l m_i \simeq \rho_l m_i$, ρ_l = fluid density, y_w^l = mass fraction of H₂O). For high-ionic strength solutions (approximately above 0.1 M) the Pitzer model should be used. Currently, however, only the Debye-Hückel algorithm is implemented in PFLOTTRAN.

The total flux Ω_j^α for species-independent diffusion is given by

$$\Omega_j^\alpha = (\mathbf{q}_\alpha - \varphi s_\alpha \mathbf{D}_\alpha \nabla) \Psi_j^\alpha. \quad (\text{A-106})$$

The diffusion/dispersion coefficient \mathbf{D}_α 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 \mathbf{q}_α for phase α is given by

$$\mathbf{q}_\alpha = -\frac{k k_\alpha}{\mu_\alpha} \nabla (p_\alpha - \rho_\alpha g z), \quad (\text{A-107})$$

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 \mathbf{D}_α is the sum of contributions from molecular diffusion and dispersion which for an isotropic medium has the form

$$\mathbf{D}_\alpha = \tau D_m \mathbf{I} + a_T v \mathbf{I} + (a_L - a_T) \frac{\mathbf{v} \mathbf{v}}{v}, \quad (\text{A-108})$$

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 PFLOTTRAN.

The porosity may be calculated from the mineral volume fractions according to the relation

$$\varphi = 1 - \sum_m \varphi_m. \quad (\text{A-109})$$

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

$$D_m(T) = D_m^\circ \exp \left[\frac{A_D}{R} \left(\frac{1}{T_0} - \frac{1}{T} \right) \right], \quad (\text{A-110})$$

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

$$Q_j = \sum_n \frac{q_M}{\rho} \Psi_j \delta(\mathbf{r} - \mathbf{r}_n), \quad (\text{A-111})$$

where q_M denotes a mass rate in units of kg/s, ρ denotes the fluid density in kg/m³, and \mathbf{r}_n refers to the location of the n th source/sink. The quantity S_j represents the sorbed concentration of the j th primary species considered in more detail in the next section.

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

$$C_i = \rho_w m_i. \quad (\text{A-112})$$

The activity of water is calculated from the approximate relation

$$a_{\text{H}_2\text{O}} = 1 - 0.017 \sum_i m_i. \quad (\text{A-113})$$

A-9.1 Mineral Precipitation and Dissolution

The reaction rate I_m is based on transition state theory taken as positive for precipitation and negative for dissolution, with the form

$$I_m = -a_m \left(\sum_l k_{ml}(T) \mathcal{P}_{ml} \right) \left| 1 - (K_m Q_m)^{1/\sigma_m} \right|^{\beta_m} \text{sign}(1 - K_m Q_m), \quad (\text{A-114})$$

where the sum over l represents contributions from parallel reaction mechanisms such as pH dependence etc., and where K_m denotes the equilibrium constant, σ_m refers to Temkin's constant which is defined as the average stoichiometric coefficient of the overall reaction (Lichtner, 1996b; see also Section A-11), β_m denotes the affinity power, a_m refers to the specific mineral surface area, and the ion activity product Q_m is defined as

$$Q_m = \prod_j (\gamma_j m_j)^{\nu_{jm}}, \quad (\text{A-115})$$

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

$$k_{ml}(T) = k_{ml}^0 \exp \left[\frac{E_{ml}}{R} \left(\frac{1}{T_0} - \frac{1}{T} \right) \right], \quad (\text{A-116})$$

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 l th parallel reaction with the form

$$\mathcal{P}_{ml} = \prod_i \frac{(\gamma_i m_i)^{\alpha_{il}^m}}{1 + K_{ml} (\gamma_i m_i)^{\beta_{il}^m}}, \quad (\text{A-117})$$

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).

A-9.1.1 Rate Limiter A rate-limited form of the mineral kinetic rate law can be devised according to the expression

$$\hat{I}_m = -a_m \left(\sum_l \mathcal{P}_{ml} k_{ml} \right) \left[\frac{1 - (K_m Q_m)^{1/\sigma_m}}{1 + \frac{k_{ml}}{k_{ml}^{\text{lim}}} (K_m Q_m)^{1/\sigma_m}} \right], \quad (\text{A-118})$$

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

$$\lim_{K_m Q_m \rightarrow \infty} \hat{I}_m = k_{ml}^{\text{lim}} a_m \sum_l \mathcal{P}_{ml}. \quad (\text{A-119})$$

Defining the affinity factor

$$\Omega_m = 1 - (K_m Q_m)^{1/\sigma_m}, \quad (\text{A-120})$$

or

$$K_m Q_m = \left(1 - \Omega_m \right)^{\sigma_m}, \quad (\text{A-121})$$

the rate may be expressed alternatively as

$$\hat{I}_m = -a_m \sum_l \mathcal{P}_{ml} k_{ml} \frac{\Omega_m}{1 + \frac{k_{ml}}{k_{ml}^{\text{lim}}} (1 - \Omega_m)}. \quad (\text{A-122})$$

A-9.1.2 Changes in Material Properties Porosity, permeability, tortuosity and mineral surface area may be updated optionally due to mineral precipitation and dissolution reactions according to the relations

$$\varphi = 1 - \sum_m \varphi_m, \quad (\text{A-123})$$

$$k = k_0 f(\varphi, \varphi_0, \varphi_c, a), \quad (\text{A-124})$$

with

$$f = \left(\frac{\varphi - \varphi_c}{\varphi_0 - \varphi_c} \right)^a, \quad (\text{A-125})$$

$$= f_{\text{min}} \quad \text{if} \quad \varphi \leq \varphi_c, \quad (\text{A-126})$$

$$\tau = \tau_0 \left(\frac{\varphi}{\varphi_0} \right)^b, \quad (\text{A-127})$$

and

$$a_m = a_m^0 \left(\frac{\varphi_m}{\varphi_m^0} \right)^n \left(\frac{1 - \varphi}{1 - \varphi_0} \right)^{n'}, \quad (\text{A-128})$$

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_{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.(A-123) 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

$$\varphi_m = (1 - \varphi) \frac{x_m \bar{V}_m}{\sum_{m'} x_{m'} \bar{V}_{m'}}, \quad (\text{A-129a})$$

$$= (1 - \varphi) \frac{y_m \rho_m^{-1}}{\sum_{m'} y_{m'} \rho_{m'}^{-1}}, \quad (\text{A-129b})$$

with inverse relation

$$x_m = \frac{\varphi_m}{\bar{V}_m \eta_s (1 - \varphi)}, \quad (\text{A-130})$$

and similarly for the mass fraction, where

$$\rho_m = W_m \bar{V}_m^{-1}, \quad (\text{A-131})$$

and the solid molar density η_s is given by

$$\eta_s = \frac{1}{\sum_m x_m \bar{V}_m}. \quad (\text{A-132})$$

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

$$\rho_s = W_s \eta_s, \quad (\text{A-133})$$

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

$$W_s = \sum_m x_m W_m = \frac{1}{\sum_m W_m^{-1} y_m}. \quad (\text{A-134})$$

Mass and mole fractions are related by the expression

$$W_m x_m = W_s y_m. \quad (\text{A-135})$$

A-9.1.3 Affinity Threshold An affinity threshold f for precipitation may be introduced which only allows precipitation to occur if $K_m Q_m > f > 1$.

A-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

$$a_m(t) = a_m^0 \left(\frac{\varphi_m}{\varphi_m^0} \right)^n \left(\frac{1 - \varphi}{1 - \varphi_0} \right)^{n'} \left(\frac{\varphi_{m'}^c - \varphi_{m'}}{\varphi_{m'}^c} \right)^{n''}, \quad (\text{A-136})$$

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

$$a_m = 0, \quad (\text{A-137})$$

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 .

A-9.2 Sorption

Sorption reactions incorporated into PFLOTTRAN consist of ion exchange and surface complexation reactions for both equilibrium and multirate formulations.

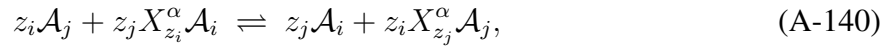
A-9.2.1 Ion Exchange Ion exchange reactions may be represented either in terms of bulk- or mineral-specific rock properties. Changes in bulk sorption properties can be expected as a result of mineral reactions. However, only the mineral-based formulation enables these effects to be captured in the model. 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

$$\omega_\alpha = \frac{N_{\text{site}}}{V} = \frac{N_{\text{site}}}{M_s} \frac{M_s}{V_s} \frac{V_s}{V} = Q_\alpha \rho_s (1 - \phi). \quad (\text{A-138})$$

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

$$\omega_m^{\text{CEC}} = \frac{N_m}{V} = \frac{N_m}{M_m} \frac{M_m}{V_m} \frac{V_m}{V} = Q_m^{\text{CEC}} \rho_m \phi_m. \quad (\text{A-139})$$

In PFLOTTRAN ion exchange reactions are expressed in the form



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

$$K_{ji}^\alpha = \frac{(k_j^\alpha)^{z_i}}{(k_i^\alpha)^{z_j}} = \left(\frac{X_j^\alpha}{a_j} \right)^{z_i} \left(\frac{a_i}{X_i^\alpha} \right)^{z_j}. \quad (\text{A-141})$$

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

$$X_k^\alpha = \frac{z_k S_k^\alpha}{\sum_l z_l S_l^\alpha} = \frac{z_k}{\omega_\alpha} S_k^\alpha, \quad (\text{A-142})$$

with

$$\sum_k X_k^\alpha = 1. \quad (\text{A-143})$$

The site concentration ω_α is defined by

$$\omega_\alpha = \sum_k z_k S_k^\alpha, \quad (\text{A-144})$$

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

$$\omega_\alpha = (1 - \varphi) \rho_s Q_\alpha, \quad (\text{A-145})$$

with solid density ρ_s and porosity φ .

An alternative form of reactions A-140 often found in the literature is

$$\frac{1}{z_j} \mathcal{A}_j + \frac{1}{z_i} X_{z_i}^\alpha \mathcal{A}_i \rightleftharpoons \frac{1}{z_i} \mathcal{A}_i + \frac{1}{z_j} X_{z_j}^\alpha \mathcal{A}_j, \quad (\text{A-146})$$

obtained by dividing reaction A-140 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 A-146 have the form

$$K_{ji}^{\prime\alpha} = \frac{(k_j^{\prime\alpha})^{1/z_j}}{(k_i^{\prime\alpha})^{1/z_i}} = \left(\frac{X_j^\alpha}{a_j} \right)^{1/z_j} \left(\frac{a_i}{X_i^\alpha} \right)^{1/z_i}. \quad (\text{A-147})$$

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

$$K_{ji}^\alpha = (K_{ji}^{\prime\alpha})^{z_i z_j}, \quad (\text{A-148})$$

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

$$S_j^\alpha = \frac{\omega_\alpha}{z} \frac{k_j^\alpha \gamma_j m_j}{\sum_l k_l^\alpha \gamma_l m_l}, \quad (\text{A-149})$$

where m_k denotes the k th cation molality. This expression follows directly from the mass action equations and conservation of exchange sites.

In the more general case ($z_i \neq z_j$) it is necessary to solve the nonlinear equation

$$X_j^\alpha + \sum_{i \neq j} X_i^\alpha = 1, \quad (\text{A-150})$$

for the reference cation mole fraction X_j . From the mass action equation Eqn.(A-141) it follows that

$$X_i^\alpha = k_i^\alpha a_i \left(\frac{X_j^\alpha}{k_j^\alpha a_j} \right)^{z_i/z_j}. \quad (\text{A-151})$$

Defining the function

$$f(X_j^\alpha) = X_j^\alpha + \sum_{i \neq j} X_i^\alpha(X_j^\alpha) - 1, \quad (\text{A-152})$$

its derivative is given by

$$\frac{df}{dX_j^\alpha} = 1 - \frac{1}{z_j X_j^\alpha} \sum_{i \neq j} z_i k_i^\alpha a_i \left(\frac{X_j^\alpha}{k_j^\alpha a_j} \right)^{z_i/z_j}. \quad (\text{A-153})$$

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

$$(X_j^\alpha)^{k+1} = (X_j^\alpha)^k - \frac{f[(X_j^\alpha)^k]}{\frac{df[(X_j^\alpha)^k]}{dX_j^\alpha}}. \quad (\text{A-154})$$

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

$$S_j^\alpha = \frac{\omega_\alpha}{z_j} X_j^\alpha, \quad (\text{A-155})$$

with the derivatives for $j \neq l$

$$\frac{\partial S_j^\alpha}{\partial m_l} = -\frac{\omega_\alpha}{m_l} \frac{X_j^\alpha X_l^\alpha}{\sum_l z_l X_l^\alpha}, \quad (\text{A-156a})$$

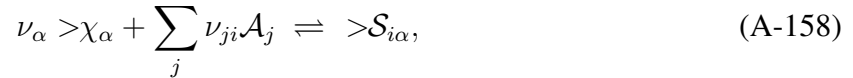
$$= -\frac{1}{m_l} \frac{z_j z_l S_j^\alpha S_l^\alpha}{\sum_l z_l^2 S_l^\alpha}, \quad (\text{A-156b})$$

and for $j = l$

$$\frac{\partial S_j^\alpha}{\partial m_j} = \frac{\omega_\alpha X_j^\alpha}{z_j m_j} \left(1 - \frac{z_j X_j^\alpha}{\sum_l z_l X_l^\alpha} \right), \quad (\text{A-157a})$$

$$= \frac{S_j^\alpha}{m_j} \left(1 - \frac{z_j^2 S_j^\alpha}{\sum_l z_l^2 S_l^\alpha} \right). \quad (\text{A-157b})$$

A-9.2.2 Surface Complexation Surface complexation reactions are assumed to have the form



for the i th 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 $S_{i\alpha}^{\text{eq}}$ is given by

$$S_{i\alpha}^{\text{eq}} = \frac{\omega_\alpha K_i Q_i}{1 + \sum_l K_l Q_l}, \quad (\text{A-159})$$

and the empty site concentration by

$$S_\alpha^{\text{eq}} = \frac{\omega_\alpha}{1 + \sum_l K_l Q_l}, \quad (\text{A-160})$$

where the ion activity product Q_i is defined by

$$Q_i = \prod_j (\gamma_j C_j)^{\nu_{ji}}. \quad (\text{A-161})$$

The site concentration ω_α satisfies the relation

$$\omega_\alpha = S_\alpha + \sum_i S_{i\alpha}, \quad (\text{A-162})$$

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

$$S_{j\alpha}^{\text{eq}} = \sum_i \nu_{ji} S_{i\alpha}^{\text{eq}} = \frac{\omega_\alpha}{1 + \sum_l K_l Q_l} \sum_i \nu_{ji} K_i Q_i. \quad (\text{A-163})$$

A-9.2.3 Multirate Sorption In the multirate model the rates of sorption reactions are described through a kinetic relation given by

$$\frac{\partial S_{i\alpha}}{\partial t} = k_{\alpha}(S_{i\alpha}^{\text{eq}} - S_{i\alpha}), \quad (\text{A-164})$$

for surface complexes, and

$$\frac{\partial S_{\alpha}}{\partial t} = - \sum_i k_{\alpha}(S_{i\alpha}^{\text{eq}} - S_{i\alpha}), \quad (\text{A-165})$$

$$= k_{\alpha}(S_{\alpha}^{\text{eq}} - S_{\alpha}), \quad (\text{A-166})$$

for empty sites, where S_{α}^{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

$$\int_0^{\infty} \wp(k_{\alpha}) dk_{\alpha} = 1. \quad (\text{A-167})$$

The fraction of sites f_{α} belonging to site α is determined from the relation

$$f_{\alpha} = \int_{k_{\alpha}-\Delta k_{\alpha}/2}^{k_{\alpha}+\Delta k_{\alpha}/2} \wp(k_{\alpha}) dk_{\alpha} \simeq \wp(k_{\alpha}) \Delta k_{\alpha}, \quad (\text{A-168})$$

with the property that

$$\sum_{\alpha} f_{\alpha} = 1. \quad (\text{A-169})$$

Given that the total site concentration is ω , then the site concentration ω_{α} associated with site α is equal to

$$\omega_{\alpha} = f_{\alpha} \omega. \quad (\text{A-170})$$

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

$$S_{j\alpha} = \sum_i \nu_{ji} S_{i\alpha}. \quad (\text{A-171})$$

Then the transport equations become

$$\frac{\partial}{\partial t} \left(\varphi \Psi_j + \sum_{\alpha} S_{j\alpha} \right) + \nabla \cdot \Omega_j = - \sum_m \nu_{jm} I_m. \quad (\text{A-172})$$

The total sorbed concentrations are obtained from the equations

$$\frac{\partial S_{j\alpha}}{\partial t} = k_{\alpha}(S_{j\alpha}^{\text{eq}} - S_{j\alpha}). \quad (\text{A-173})$$

A-9.3 Sorption Isotherm <Under Revision>

The distribution coefficient \tilde{K}_j^D [$\text{m}^3 \text{kg}^{-1}$] is customarily defined as the ratio of sorbed to aqueous concentrations with the sorbed concentration referenced to the mass of solid as given by

$$\tilde{K}_j^D = \frac{M_j^s/M_s}{M_j^{\text{aq}}/V_l}, \quad (\text{A-174a})$$

$$= \frac{N_j^s/M_s}{N_j^{\text{aq}}/V_l}, \quad (\text{A-174b})$$

$$= \frac{\tilde{S}_j}{C_j} = \frac{1}{\rho_w} \frac{\tilde{S}_j}{m_j}, \quad (\text{A-174c})$$

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

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

$$K_j^D = \frac{N_i^s}{N_i^{\text{aq}}} = \frac{N_i^s/V}{N_i^{\text{aq}}/V} = \frac{1}{\varphi_{sl}} \frac{S_j}{C_j}, \quad (\text{A-175})$$

by writing

$$K_j^D = \frac{N_i^s}{M_s} \frac{M_s}{V_s} \frac{V_s}{V_p} \frac{V_p}{V_l} \frac{V_l}{N_i^{\text{aq}}}, \quad (\text{A-176})$$

$$= \rho_s \frac{1-\varphi}{\varphi_{sl}} \tilde{K}_j^D = \frac{\rho_b}{\varphi_{sl}} \tilde{K}_j^D, \quad (\text{A-177})$$

with grain density $\rho_s = M_s/V_s$, bulk density $\rho_b = (1-\varphi)\rho_s$, porosity $\varphi = V_p/V$, and saturation $s_l = V_l/V_p$.

An alternative definition of the distribution coefficient denoted by \hat{K}_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

$$\hat{K}_j^D = \frac{N_j^s/V}{N_j^{\text{aq}}/M_w} = \frac{S_j}{m_j}. \quad (\text{A-178})$$

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

$$\frac{\partial}{\partial t} \varphi_{sl} C_j + \nabla \cdot \mathbf{F}_j = -\frac{\partial S_j}{\partial t}, \quad (\text{A-179})$$

for a partially saturated medium. Substituting $S_j = \varphi_{sl} K_j^D C_j$ from Eqn.(A-175) and introducing the retardation \mathcal{R}_j gives

$$\frac{\partial}{\partial t} \mathcal{R}_j \varphi_{sl} C_j + \nabla \cdot \mathbf{F}_j = 0, \quad (\text{A-180})$$

with the retardation given by the alternative forms

$$R_j = 1 + K_j^D, \quad (\text{dimensionless}), \quad (\text{A-181a})$$

$$= 1 + \frac{\rho_b}{\varphi_{sl}} \tilde{K}_j^D, \quad (\text{conventional}), \quad (\text{A-181b})$$

$$= 1 + \frac{1}{\varphi_{sl}\rho_w} \hat{K}_j^D, \quad (\text{molality-based}). \quad (\text{A-181c})$$

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

- linear K_D model:

$$S_j = \varphi_{sl} K_j^D C_j = \hat{K}_j^D m_j, \quad (\text{A-182})$$

with distribution coefficient \hat{K}_j^D ,

- Langmuir isotherm:

$$S_j = \frac{K_j^L b_j^L C_j / \rho_w}{1 + K_j^L C_j / \rho_w} = \frac{K_j^L b_j^L m_j}{1 + K_j^L m_j}, \quad (\text{A-183})$$

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

- Freundlich isotherm:

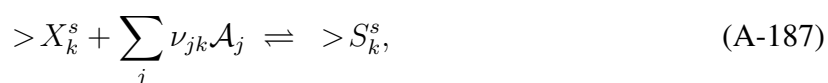
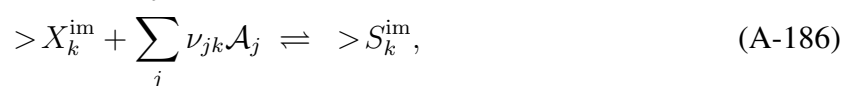
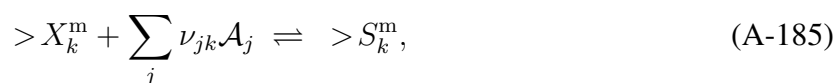
$$S_j = K_j^F \left(\frac{C_j}{\rho_w} \right)^{(1/n_j^F)} = K_j^F (m_j)^{(1/n_j^F)}, \quad (\text{A-184})$$

with coefficients K_j^F and n_j^F .

A-9.4 Colloid-Facilitated Transport

Colloid-facilitated transport is implemented into PFLOTTRAN 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 PFLOTTRAN. 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



with corresponding reaction rates I_k^m , I_k^{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

$$\sum_j \nu_{js} \mathcal{A}_j \rightleftharpoons \mathcal{M}_s. \quad (\text{A-188})$$

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

$$\frac{\partial}{\partial t} \varphi_{sl} \Psi_j^l + \nabla \cdot \Omega_j^l = - \sum_k \nu_{jk} (I_k^m + I_k^{im} + \sum_s I_k^s) - \sum_s \nu_{js} I_s, \quad (\text{A-189})$$

$$\frac{\partial}{\partial t} \varphi_{sl} S_k^m + \nabla \cdot \mathbf{q}_c S_k^m = I_k^m, \quad (\text{A-190})$$

$$\frac{\partial}{\partial t} S_k^{im} = I_k^{im}, \quad (\text{A-191})$$

$$\frac{\partial}{\partial t} S_k^s = I_k^s, \quad (\text{A-192})$$

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

$$\frac{\partial}{\partial t} \left[\varphi_{sl} \Psi_j^l + \sum_k \nu_{jk} (\varphi_{sl} S_k^m + S_k^{im} + \sum_s S_k^s) \right] + \nabla \cdot \left(\Omega_j^l + \mathbf{q}_c \sum_k \nu_{jk} S_k^m \right) = - \sum_s \nu_{js} I_s. \quad (\text{A-193})$$

In the kinetic case either form of the primary species transport equations given by Eqn.(A-189) or (A-193) can be used provided it is coupled with the appropriate kinetic equations Eqns.(A-190)–(A-192). 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

$$I_k^m = k_k (S_k^m - S_{km}^{\text{eq}}), \quad (\text{A-194})$$

with rate constant k_k and where S_{km}^{eq} is a known function of the solute concentrations. In this case, Eqn.(A-190) 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.

A-9.5 Microbial Reactions

Kinetic rate law:

$$\mathcal{R}_i = k_i X_i \frac{C_a}{(K_a + C_a)} \frac{C_d}{(K_d + C_d)} I_i \quad (\text{A-195})$$

$$\mathcal{R}_i = k_i (X_i - X_i^{th}) \left[\frac{(C_a - C_a^{th})}{(K_a + C_a - C_a^{th})} \right] \left[\frac{(C_d - C_d^{th})}{(K_d + C_d - C_d^{th})} \right] I_i \quad (\text{A-196})$$

Biomass creation and degradation:

$$\frac{dX_i}{dt} = \mathcal{Y}_i \mathcal{R}_i - k_{\text{decay},i} X_i, \quad (\text{A-197})$$

Table A-4: Definition of symbols used in modeling microbial reactions.

Symbol	Definition
\mathcal{R}_i	reaction rate for i th biomass reaction
k_i	rate constant
X_i	biomass concentration
I_i	inhibition factor
C_a	electron acceptor concentration
C_d	electron donor concentration
C_a^{th}	electron acceptor threshold concentration
C_d^{th}	electron donor threshold concentration

where \mathcal{Y}_i denotes the yield and $k_{\text{decay},i}$ refers to the biomass decay rate constant. Other parameters are listed in Table A-4.

- Monod inhibition below a concentration $[K_i]$: $I_i = C/(K_i + C)$
- Inverse monod inhibition above a concentration $[K_i]$: $I_i = K_i/(K_i + C)$
- Kinzelbach inhibition: $I_i = 1/2 \pm \arctan[(C - C_i^{th}) f_i]/\pi$ where $f_i \simeq 10^8$
- Thermodynamic inhibition: $1 - Q_i/K_i$

A-10 Tracer Mean Age

PFLOTTRAN implements the Eulerian formulation of solute age for a nonreactive tracer following Goode (1996). PFLOTTRAN solves the advection-diffusion/dispersion equation for the mean age given by

$$\frac{\partial}{\partial t} \varphi s A C + \nabla \cdot (\mathbf{q} A C - \varphi s D \nabla (A C)) = \varphi s C, \quad (\text{A-198})$$

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 $\alpha = A C$ 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 PFLOTTRAN 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 $A = \alpha/C$.

A-11 Thermodynamic Database

PFLOTTRAN 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 PFLOTTRAN 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

$$\log K = c_{-1} \ln T + c_0 + c_1 T + \frac{c_2}{T} + \frac{c_3}{T^2}, \quad (\text{A-199})$$

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 \bar{V}_m , and formula weight w_i) used in PFLOTRAN. The database is divided into 5 blocks as listed in Table A-5, consisting of database primary species, aqueous complex reactions, gaseous reactions, mineral reactions, and surface complexation reactions. Each block is terminated by a line beginning with 'null'. The quantity N_{temp} refers to the number of temperatures at which $\log K$ values are stored in the database. In the `hanford.dat` database $N_{\text{temp}} = 8$ 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

$$\mathcal{A}_r \rightleftharpoons \sum_{i=1}^{\text{nspec}} \nu_{ir} \mathcal{A}_i, \quad (\text{A-200})$$

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 $\text{O}_{2(g)}$. Complexation reactions involving redox sensitive species are written in such a manner as to preserve the redox state.

Table A-5: Format of thermodynamic database.

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

Note that chemical reactions are not unique. For example, given a particular mineral reaction

$$\sum_j \nu_{jm} \mathcal{A}_j \rightleftharpoons \mathcal{M}_m, \quad (\text{A-201})$$

and equally acceptable reaction is the scaled reaction

$$\sum_j f_m \nu_{jm} \mathcal{A}_j \rightleftharpoons f_m \mathcal{M}_m, \quad (\text{A-202})$$

with scale factor f_m corresponding to a different choice of formula unit. A different scale factor may be used for each mineral. The scaled reaction corresponds to

$$\sum_j \nu'_{jm} \mathcal{A}_j \rightleftharpoons \mathcal{M}'_m, \quad (\text{A-203})$$

with $\mathcal{M}'_m = f_m \mathcal{M}_m$, $\nu'_{jm} = f_m \nu_{jm}$. In addition, mineral molar volume \bar{V}_m , formula weight W_m , equilibrium constant K_m , and the kinetic rate constant k_m are scaled according to

$$\bar{V}'_m = f_m \bar{V}_m, \quad (\text{A-204a})$$

$$W'_m = f_m W_m, \quad (\text{A-204b})$$

$$\log K'_m = f_m \log K_m. \quad (\text{A-204c})$$

The saturation index SI_m transforms according to

$$SI'_m = K'_m Q'_m = (K_m Q_m)^{f_m} = (SI_m)^{f_m}. \quad (\text{A-205})$$

Consequently, equilibrium is not affected as is to be expected. However, a more general form for the reaction rate is needed involving Temkin's constant [see Eqn.(A-114)], in order to ensure that the identical solution to the reactive transport equations is obtained using the scaled reaction. Thus it is necessary that the following conditions hold

$$\bar{V}'_m I'_m = \bar{V}_m I_m, \quad (\text{A-206})$$

$$\nu'_{jm} I'_m = \nu_{jm} I_m. \quad (\text{A-207})$$

This requires that the reaction rate I_m transform as

$$I'_m = \frac{1}{f_m} I_m, \quad (\text{A-208})$$

which guarantees that mineral volume fractions and solute concentrations are identical as obtained from solving the reactive transport equations. For this relation to hold it is necessary that the kinetic rate constant transform as

$$k'_m = f_m^{-1} k_m. \quad (\text{A-209})$$

From the above relations it is found that the reaction rate transforms according to

$$I'_m = -k'_m A_m (1 - (K'_m Q'_m)^{1/\sigma'_m}), \quad (\text{A-210a})$$

$$= -\frac{1}{f_m} k_m A_m (1 - (K_m Q_m)^{f_m/\sigma'_m}), \quad (\text{A-210b})$$

$$= \frac{1}{f_m} I_m, \quad (\text{A-210c})$$

where the last result is obtained by scaling Temkin's constant according to

$$\sigma'_m = f_m \sigma_m. \quad (\text{A-211})$$

It should be noted that the mineral concentration ($C'_m = (\bar{V}_m^{-1})' \phi_m = f_m^{-1} C_m$), differs in the two formulations; however, mass density ($\rho_m = W_m \bar{V}_m^{-1}$) is an invariant, unlike molar density ($\eta_m = \bar{V}_m^{-1}$).

A-12 Eh, pe

Output for Eh and pe is calculated from the half-cell reaction



with the corresponding equilibrium constant fit to the Maier-Kelly expansion Eqn.(A-199). The fit coefficients are listed in Table A-6.

Table A-6: Fit coefficients for log K of reaction A-212.

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

A-13 Geomechanics

In PFLOTTRAN, 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:

$$\nabla \cdot [\boldsymbol{\sigma}] + \rho \mathbf{b} = 0 \quad \text{in } \Omega, \quad (\text{A-213a})$$

$$\boldsymbol{\sigma} = \lambda \text{tr}(\boldsymbol{\varepsilon}) + 2\mu \boldsymbol{\varepsilon} - \beta(p - p_0)\mathbf{I} - \alpha(T - T_0)\mathbf{I}, \quad (\text{A-213b})$$

$$\boldsymbol{\varepsilon} = \frac{1}{2} (\nabla \mathbf{u}(\mathbf{x}) + [\nabla \mathbf{u}(\mathbf{x})]^T), \quad (\text{A-213c})$$

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}^p(\mathbf{x}) \quad \text{on } \Gamma^D, \quad (\text{A-213d})$$

$$\boldsymbol{\sigma} \mathbf{n}(\mathbf{x}) = \mathbf{t}^p(\mathbf{x}) \quad \text{on } \Gamma^N, \quad (\text{A-213e})$$

where \mathbf{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), \mathbf{b} is the specific body force (which is gravity in most cases), \mathbf{n} is the outward normal to the boundary Γ^N . Also, \mathbf{u}^p is the prescribed values of \mathbf{u} on the Dirichlet part of the boundary Γ^D , and \mathbf{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 \mathbf{I} is the identity tensor. Note that stress is assumed *positive under tension*. The effect of deformation on the pore structure is accounted for via

$$\phi = \frac{\phi_0}{1 + (1 - \phi_0)\text{tr}(\boldsymbol{\varepsilon})}. \quad (\text{A-214})$$

Note that the above equations are solved using the finite element method (Galerkin finite element) with the displacements solved for at the vertices. Since, the flow equations are solved via the 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, the geomechanics grid must be read in as an unstructured grid. Even if one needs to work with a structured grid, the grid must be set up in the unstructured grid format.

Appendix B: 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.

B-1 Integrated Finite Volume Discretization

The governing partial differential equations for mass conservation can be written in the general form

$$\frac{\partial}{\partial t} A_j + \nabla \cdot \mathbf{F}_j = Q_j, \quad (\text{B-1})$$

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

$$\frac{d}{dt} \int_{V_n} A_j dV + \int_{V_n} \nabla \cdot \mathbf{F}_j = \int_{V_n} Q_j dV. \quad (\text{B-2})$$

The accumulation term has the finite volume form

$$\frac{d}{dt} \int_{V_n} A_j dV = \frac{A_{jn}^{t+\Delta t} - A_{jn}^t}{\Delta t} V_n, \quad (\text{B-3})$$

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

$$\int_{V_n} \nabla \cdot \mathbf{F}_j = \int_{\partial V_n} \mathbf{F}_j \cdot d\mathbf{S}, \quad (\text{B-4})$$

$$= \sum_{n'} F_{j,nn'} A_{nn'}, \quad (\text{B-5})$$

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

$$F_{j,nn'}^\alpha = (q_\alpha X_\alpha)_{nn'} - (\varphi s_\alpha \tau_\alpha D_\alpha)_{nn'} \frac{X_{n'}^\alpha - X_n^\alpha}{d_{n'} + d_n}, \quad (\text{B-6})$$

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

$$(q_\alpha X_\alpha)_{nn'} = \begin{cases} q_{nn'}^\alpha X_{n'}, & q_{nn'} > 0 \\ q_{nn'}^\alpha X_n, & q_{nn'} < 0 \end{cases}. \quad (\text{B-7})$$

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

$$\int_{V_n} Q_j dV = Q_{jn} V_n, \quad (\text{B-8})$$

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

$$\int_{V_n} Q_j dV = \hat{Q}_{jn}. \quad (\text{B-9})$$

B-2 Fully Implicit Newton-Raphson Iteration with Linear and Logarithm Update

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

$$\mathbf{R}(\mathbf{x}) = \mathbf{0}, \quad (\text{B-10})$$

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

$$\mathbf{J}^{(i)} \delta \mathbf{x}^{(i+1)} = -\mathbf{R}^{(i)}, \quad (\text{B-11})$$

at the i th iteration. Iteration stops when

$$\left| \mathbf{R}^{(i+1)} \right| < \epsilon, \quad (\text{B-12})$$

or if

$$\left| \delta \mathbf{x}^{(i+1)} \right| < \delta. \quad (\text{B-13})$$

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

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \delta \mathbf{x}^{(i+1)}. \quad (\text{B-14})$$

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

$$\ln \mathbf{y}^{(i+1)} = \ln \mathbf{y}^{(i)} + \delta \ln \mathbf{y}^{(i+1)}, \quad (\text{B-15a})$$

or

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} e^{\delta \ln \mathbf{y}^{(i+1)}}. \quad (\text{B-15b})$$

B-2.1 Example

To illustrate the logarithmic update formulation the simple linear equation

$$x = x_0, \quad (\text{B-16})$$

is considered. The residual function is given by

$$R = x - x_0, \quad (\text{B-17})$$

with Jacobian

$$J = \frac{\partial R}{\partial x} = I. \quad (\text{B-18})$$

In the linear formulation the Newton-Raphson equations are given by

$$J \delta x = -R, \quad (\text{B-19})$$

$$\delta x = -(x - x_0) \quad (\text{B-20})$$

$$x' = x + \delta x = x_0. \quad (\text{B-21})$$

In the logarithmic formulation the Jacobian is given by

$$J = \frac{\partial R}{\partial \ln x} = x, \quad (\text{B-22})$$

and the Newton-Raphson equations are now nonlinear becoming

$$J^i \delta \ln x^{i+1} = -R^i, \quad (\text{B-23})$$

with the solution update

$$\ln x^{i+1} = \ln x^i + \delta \ln x^{i+1}, \quad (\text{B-24})$$

or

$$x^{i+1} = x^i e^{\delta \ln x^{i+1}}. \quad (\text{B-25})$$

It follow that

$$x^i \delta \ln x^{i+1} = -(x^i - x_0), \quad (\text{B-26})$$

with the solution

$$\delta \ln x^{i+1} = \frac{x_0 - x^i}{x^i}, \quad (\text{B-27})$$

and thus

$$x^{i+1} = x^i \exp\left(\frac{x_0 - x^i}{x^i}\right). \quad (\text{B-28})$$

Given that a solution x exists it follows that

$$\lim_{i \rightarrow \infty} x^i \rightarrow x, \quad (\text{B-29})$$

$$\lim_{i \rightarrow \infty} \frac{x^{i+1}}{x^i} \rightarrow 1, \quad (\text{B-30})$$

$$\lim_{i \rightarrow \infty} \exp\left(\frac{x_0 - x^i}{x^i}\right) \rightarrow 1, \quad (\text{B-31})$$

$$\lim_{i \rightarrow \infty} x^i \rightarrow x_0. \quad (\text{B-32})$$

B-2.2 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.(A-173) in time using the fully implicit backward Euler method gives

$$\frac{S_{j\alpha}^{t+\Delta t} - S_{j\alpha}^t}{\Delta t} = k_\alpha (f_\alpha S_{j\alpha}^{\text{eq}} - S_{j\alpha}^{t+\Delta t}). \quad (\text{B-33})$$

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

$$S_{j\alpha}^{t+\Delta t} = \frac{S_{j\alpha}^t + k_\alpha \Delta t f_\alpha S_{j\alpha}^{\text{eq}}}{1 + k_\alpha \Delta t}. \quad (\text{B-34})$$

From this expression the reaction rate can be calculated as

$$\frac{S_{j\alpha}^{t+\Delta t} - S_{j\alpha}^t}{\Delta t} = \frac{k_\alpha}{1 + k_\alpha \Delta t} (f_\alpha S_{j\alpha}^{\text{eq}} - S_{j\alpha}^t). \quad (\text{B-35})$$

The right-hand side of this equation is a known function of the solute concentration and thus by substituting into Eqn.(A-172) 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.(B-34).

B-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.(A-98) as the two coupled equations

$$\frac{\partial}{\partial t}(\varphi \sum_{\alpha} s_{\alpha} \Psi_j^{\alpha}) + \nabla \cdot \sum_{\alpha} (\mathbf{q}_{\alpha} - \varphi s_{\alpha} \mathbf{D}_{\alpha} \nabla) \Psi_j^{\alpha} = Q_j, \quad (\text{B-36})$$

and

$$\frac{d}{dt}(\varphi \sum_{\alpha} s_{\alpha} \Psi_j^{\alpha}) = - \sum_m \nu_{jm} I_m - \frac{\partial S_j}{\partial t}, \quad (\text{B-37})$$

The first set of equations are linear in Ψ_j (for species-independent diffusion coefficients) 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

$$\Psi_j^* = C_j^* + \sum_i \nu_{ji} C_i^*, \quad (\text{B-38})$$

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 .

B-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

$$S = K_d C. \quad (\text{B-39})$$

The governing equation is given by

$$\frac{\partial}{\partial t} \varphi C + \nabla \cdot (\mathbf{q} C - \varphi \mathbf{D} \nabla C) = - \frac{\partial S}{\partial t}. \quad (\text{B-40})$$

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

$$R = 1 + \frac{1}{\varphi} K_d. \quad (\text{B-41})$$

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

In operator splitting form this equation becomes

$$\frac{\partial}{\partial t} \varphi C + \nabla \cdot (\mathbf{q}C - \varphi D \nabla C) = 0, \quad (\text{B-42})$$

and

$$\frac{d}{dt} \varphi C = -\frac{dS}{dt}. \quad (\text{B-43})$$

The solution to the latter equation is given by

$$\varphi C^{t+\Delta t} - \varphi C^* = -(S^{t+\Delta t} - S^t), \quad (\text{B-44})$$

where C^* is the solution to the nonreactive transport equation. Using Eqn.(B-39), this result can be written as

$$C^{t+\Delta t} = \frac{1}{R} C^* + \left(1 - \frac{1}{R}\right) C^t. \quad (\text{B-45})$$

Thus for $R = 1$, $C^{t+\Delta t} = C^*$ and the solution advances unretarded. As $R \rightarrow \infty$, $C^{t+\Delta t} \rightarrow C^t$ and the front is fully retarded.

Appendix C: Reaction Sandbox

C-1 Background

Researchers often have a suite of reactions tailored to a unique problem scenario, but these reaction networks only exist in their respective research codes. The “reaction sandbox” provides these researchers with a venue for implementing user-defined reactions within PFLOTRAN. Reaction networks developed within the reaction sandbox can leverage existing biogeochemical capability within PFLOTRAN (e.g. equilibrium aqueous complexation, mineral precipitation–dissolution, etc.) or function independently. Please note that although the reaction sandbox facilitates the integration of user-defined reactions, the process still requires a basic understanding of PFLOTRAN and its approach to solving reaction through the Newton-Raphson method. For instance, One must understand the purpose and function of the `rt_auxvar` and `global_auxvar` objects.

C-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()
```