File Information

File state： [ ] Manuscript [x] Revision [ ] Released	File tag：	
	Version：	V2.5.1.875
	Authors：	Danyang Su, Mingliang Xie
	Date：	2024-01-21

File Version Logs

Version	Commit Author	Date	Remarks
V2.5.1.875	Danayng Su	21/01/2024	Implementation of noble gas ingrowth in general format, implementation of intermittent reactions for minerals and adding ph2 to define initial and boundary conditions.
V2.4.8.869	Danayng Su	18/08/2023	Implementation of new solvers for local chemistry, database backup, 'millington-quirk adjustable' tortuosity correction, linear interpolation of ice sheet loading/unloading factors and boundary pore water pressure, mass balance check for surface site, domain decomposition for ArchiSimple code.
V2.3.7.850	Danayng Su	27/01/2022	Implementation and/or updating of noble gas ingrowth, root water related module, simplified icesheet freeze/thaw processes, and output of dilution index and spatial moment.
V2.2.6.826	Danayng Su	24/03/2022	Add ice sheet modeling over 3D domain; add extensive zone selection methods for both structured grid and unstructured grid code; add anisotropic flow and solute transport for MCD code; add latest root water/solute uptake, root respiration and leaf recycle models; port latest ArchiSimple code; fix bugs and update the code.
V2.1.26.786	Danayng Su	06/01/2021	Add new features to unstructured grid code; add supports for external file IO; add supports for freeze/thaw/ponding cycling; add full tensor diffusion coefficients/dispersion coefficients/hydraulic conductivity; fix bugs and update the code.
V2.0.0.726	Danyang Su	07/08/2019	Add new features of parallel I/O related to unstructured grid code, fix bugs and update the code. New naming convention is used.
V1.0.690	Danyang Su	03/02/2019	Transient output resuming when restart; periodic maximum time step control; structured grid simulation in unstructured method.
V1.0.675	Danyang Su	22/01/2019	ArchiSimple code porting; code updates and bug

			fixes.
V1.0.633	Danyang Su	07/12/2018	New boundary conditions, unstructured code optimization
V1.0.609	Danyang Su	14/09/2018	MIN3P-HPC: unstructured grid capabilities and parallelization; code update and bug fixes.
V1.0.598	Mingliang Xie	20/07/2018	Add SIT model and pore clogging function. Bug fix in hydraulic conductivity and porosity output in *.gsv.
V1.0.553	Danyang Su	30/01/2018	Bug fix in mass balance output for ‘atmospheric’ boundary.
V1.0.549	Danyang Su	10/01/2018	Add prefix_o.aqt output before batch reaction.
V1.0.546	Danyang Su	29/11/2017	Space and tab delimiters in database. Update in albite, appelo and shlomo cases.
V1.0.542	Danyang Su	16/11/2017	Bug fix and output format update.
V1.0.535	Danyang Su	24/10/2017	Boundary condition updating during ice melting stage.
V1.0.533	Danyang Su	16/10/2017	Bug fix and code updates in mixed boundary condition, transient boundary condition and race condition in OpenMP
V1.0.530	Danyang Su	10/10/2017	First order gas decay
V1.0.527	Danyang Su	03/10/2017	LIS parallel solver
V1.0.524	Mingliang Xie	18/09/2017	New option for SDSR model, bug fix
V1.0.483	Danyang Su	26/06/2017	Bug fix in DGM, solver check
V1.0.474	Danyang Su	05/06/2017	Parallel configuration, bug fix
V1.0.464	Danyang Su	14/04/2017	PETSc 3.7.x support, bug fix
V1.0.459*	Mingliang Xie	21/03/2017	Add verification examples
V1.0.433	Mingliang Xie	28/07/2016	Salinity dependent SRB
V1.0.402	Danyang Su	05/02/2016	Direct solver, ice loading boundary
V1.0.382	Danyang Su	14/11/2015	SuperLU, MUMPS support, bug fix

V1.0.377	Danyang Su	23/10/2015	Water freezing
V1.0.371*	Danyang Su	09/10/2015	PETSc 3.6.x, bug fix, USG initial
V1.0.303	Danyang Su	04/05/2015	Isotope, gas bubble, full DGM
V1.0.268	Danyang Su	09/01/2015	Unix code porting, binary output
V1.0.221*	Danyang Su	05/08/2014	Make declaration of all variables
V1.0.216	Danyang Su	14/07/2014	Domain decomposition method
V1.0.203*	Mingliang Xie	17/06/2014	Add hMCD model
V1.0.191*	Mingliang Xie	28/04/2014	Add optional unit for SCM
V1.0.184*	Danyang Su	17/04/2014	Modify thread-shared variables
V1.0.177	Danyang Su	10/04/2014	Gas advection implementation
V1.0.167*	Danyang Su	12/03/2014	Code development, Parallel
V1.0.144	Danyang Su	20/11/2013	Code development, Parallel
V1.0.137*	Mingliang Xie	01/11/2013	Code development
V1.0.129*	Mingliang Xie	24/10/2013	Code development
V1.0.107	Danyang Su	21/08/2013	Code development
V1.0.94	Danyang Su	11/06/2013	Shared-memory parallel version
V1.0.67	Danyang Su	15/04/2013	Code development
V1.0.63*	Mingliang Xie	08/04/2013	Code development
V1.0.56	Danyang Su	08/03/2013	Code development
V1.0.37	Danyang Su	31/01/2013	Code development
V1.0.31	Mingliang Xie	29/01/2013	Code development
V1.0.7	Danyang Su	10/12/2012	Project setting and version control
V1.0.6*	Danyang Su	10/12/2012	Project setting and version control
V1.0.3*	Danyang Su	09/12/2012	Project setting and version control
V1.0.2*	Danyang Su	09/12/2012	Initial repository uploaded

-	VisualSVN Server	29/11/2012	Initial repository set

* This is the internal version, not the release version

Contents

Revision 875

Revision Number	875	**Version Number**	V2.5.1.875
Commit Date	2024-01-21	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel Compiler, GCC Compiler with C++11 or PGI compiler. 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32; WINDOWS; USG; CGAL; 2.3 Preprocessor definition for parallel version: RELEASE_X64; WINDOWS; PETSC; MPI; PARDISO; OPENMP; USG; CGAL; LIS; SCHEDULE_STATIC; 2.4 Preprocessor definition for resources: RELEASE_X64| RELEASE_X64_P 2.5 Additional Options: /fp:strict 3. External Library (Optional) 3.1 PETSc: PETSc V3.4 to V3.18 3.2 LIS: LIS V1.7.36 to V2.0.34 3.3 CGAL: CGAL V4.7	Starting from this version, code will be compiled by Intel oneAPI (version 2021.11.0) on new Windows workstation Biot-udc. Old fortran compiler is still compatible, but will not be used in the official MIN3P release.

System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable	

library can be downloaded via the URL https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html#hpc-kit	
Summary of this version	**Remarks**
Key changes in MIN3P-THCm V2.5.0.875 include implementation of noble gas ingrowth in general format, implementation of intermittent reactions for minerals, and adding ph2 to define initial and boundary conditions.	
New Features	**Remarks**
1. Add solver setting 'coefficient of factored connections'. This parameter is used to estimate the number of symbolic factorization. This value is 3 by default in the previous version. However, a larger number may be needed in some scenarios for reactive transport problem. 2. Add intermittent reactions for minerals. The reaction rate can be adjusted accordingly during the specified periods. 3. Add ph2 to define boundary and initial conditions. Please note that ph2 only work for h2(aq), which have to be specified as a component in the input file. 4. Add noble gas ingrowth based on general formulation, including neutron capture, spontaneous fission, first order decay and nucleogenic. 5. Add capability of reading noble gas ingrowth related parameters from external file, including 'solid phase density', 'concentration of radioelements', 'weight fraction of elements' and 'neutron capture probability'.	
Usage of New Features	**Remarks**
1. add 'coefficient of factored connections' followed by a user specified coefficient (e.g., 3 or 4) into data block 'control parameters - variably saturated flow', 'control parameters - energy balance' and 'control parameters - reactive transport'. An example is shown below: 'coefficient of factored connections' 4 2. Add keyword 'intermittent reactions' into data block 'initial condition - reactive transport', followed by mineral name, time and rate coefficient. This feature is zone dependent. An example is shown below: 'initial condition - reactive transport' 2 ;number of zones ! --- 'number and name of zone'	

```
1
'background chemistry 1'

'intermittent reactions'
2                        !number of intermittent reactions
'calcite'                !name of reaction
'pyrite'
2                        !number of target read times
2.0   0.0    1e-8        !active from T = 2.0 or initial time whichever is
                         !bigger, the second and third parameters are
5.0   1e-8    0.0        !the area/effective rate coefficient parameters for
                         !the two intermittent reactions.
                         !In order, from T = initial to 2.0, the rate
                         !parameters set in ‘mineral input’ are used.
                         !From T = 2.0 to 5.0, the rate coefficient 0.0 and
                         !1e-8 are used for 'calcite' and 'pyrite'.
                         !From T = 5.0 until end of simulation, the rate
                         !coefficient 1e-8 and 0.0 are used for 'calcite' and
                         !'pyrite'

'extent of zone'
0.0 1.0  0.0 0.0  0.0 2.5

'end of zone'

! -----------------------------------------------------------------------
'number and name of zone'
2
'background chemistry 2'

'intermittent reactions'
3                         !number of intermittent reactions
'siderite(d)'             !name of reaction
'gypsum'
'gibbsite(c)'
4                         !number of target read times
-1.0  0.0  1e-8  1.0e-4 !active from T = -1.0 or initial time, whichever is
                          !bigger, the second and third parameters are
2.0  1e-8  0.0  1.0e-1  !the area/effective rate coefficient parameters for
                         !the three intermittent reactions.
6.0  1e-5  0.5  1.0e-2  !In order, from T = -1.0 to 2.0, the rate
                         !coefficient 0.0, 1e-8 and 1e-4 are used for
                         !'siderite(d)','gypsum' and 'gibbsite(c)'.
8.0  1.0  1.0  1.0       !From T = 2.0 to 6.0, the rate coefficient 1e-8, 0.0
                         !and 1.0e-1 are used for 'siderite(d)','gypsum'
                         !and 'gibbsite(c)'.
                         !From T = 6.0 to 8.0, the rate coefficient 1e-5, 0.5
                         !and 1.0e-2 are used for 'siderite(d)','gypsum'
                         !and 'gibbsite(c)'.
                         !From T = 8.0 until end of simulation, the rate
                         !coefficient 1.0, 1.0 and 1.0 are used for
```

```
                                  !'siderite(d)','gypsum' and 'gibbsite(c)'.

   'extent of zone'
   0.0 1.0  0.0 0.0  2.5 5.0

   'end of zone'

   'done'
```

3. No extra keyword is required. Just use 'ph2' as the type of initial or boundary condiiton for aqueous concentration input.

4. Database 'noblegases.dbs' is required to be added into database folder. User needs to define noble gas ingrowth, related radioelements and neutron capture components in block 'geochemical system' and 'initial condition - reactive transport'.

Example input in block 'geochemical system':

```
   noble gas ingrowth'   !new keyword to activate noble gas ingrowth
   16                    !number of noble gas isotopes to be considered
   '4he ingrowth'        !names of noble gas isotopes to be considered
   '3he ingrowth'        !
   '20ne ingrowth'
   '21ne ingrowth'
   '22ne ingrowth'
   '36ar ingrowth'
   '38ar ingrowth'
   '40ar ingrowth'
   '83kr ingrowth'
   '84kr ingrowth'
   '86kr ingrowth'
   '129xe ingrowth'
   '131xe ingrowth'
   '132xe ingrowth'
   '134xe ingrowth'
   '136xe ingrowth'

   'noble gas ingrowth - radioelements'
   4
   '235u'
   '238u'
   '232th'
   '40k'

   'noble gas ingrowth - elements'
   8
   'na'
   'mg'
   'al'
   'si'
   'c'
   'o'
   'cl'
   'f'

   'noble gas ingrowth - neutron capture elements'
   4
   '6li'
   '35cl'
   '37cl'
   '235u'
```

Example input in block 'initial condition - reactive transport':

```
   'direct release to aqueous phase'
   !alternative option: 'mass transfer to aqueous phase'
   !the alternative option will also require the implementation of
   !a mass transfer model, to be considered at a later time

   'solid phase density'
   2.65                              !density in kg solid dm-3 solid

   !The following parameters are for the full set of noble gas ingrowth
   !reactions it would be best to code this up in a way that it is possible to
   !specify only the parameters for the ingrowth reactions specified, e.g. if
   !only noble gas ingrowth of 4He is specified, only the concentrations of
   !235U, 238U and 232Th are required.
   !Generally, values for upper crust from Ballentine and Burnard (2002)
   !were used below

   'constant radioelement concentration'

   'noble gas ingrowth - concentration of radioelements'
   !input in units of mg RE kg-1 solid or ppm
   0.02      ;'235u'                      !0.7% of U
   2.78      ;'238u'                      !99.3% of U
   10.7      ;'232th'
   3.0       ;'40k'                       !0.011% of total K

   'noble gas ingrowth - weight fraction of elements'
   !input in units of kg E kg-1 solid or weight fraction
   2.89e-2   ;'na'
   1.33e-2   ;'mg'
   8.04e-2   ;'al'
   3.09e-1   ;'si'
   3.24e-3   ;'c'
   4.75e-1   ;'o'
   2.32e-4   ;'cl'
   1.00e-6   ;

   'noble gas ingrowth - neutron capture probability'     !unitless
```

```
  2.05e-4   ;'6li'        ! the fraction of neutrons captured by 6Li [-]
  1.67e-4   ;'35cl'       !76% 35Cl
  0.53e-4   ;'37cl'       !24% 37Cl
  5.03e-8   ;'235u'
```

5. To read 'solid phase density' from external file for structured grid version, please name the external file prefix.ngirho and use keyword 'read solid phase density from file' in block 'initial condition - reactive transport'.

To read 'solid phase density' from external file for unstructured grid version, please name the external file prefix.ngirho.vtk (variable name rho_solid) and use keyword 'read solid phase density from vtk file' in block 'initial condition - reactive transport'.

To read 'concentration of radioelements' from external file for structured grid version, please name the external file prefix.ngiconc and use keyword 'read concentration of radioelements from file' in block 'initial condition - reactive transport'.

To read 'solid phase density' from external file for unstructured grid version, please name the external file prefix.ngiconc.vtk (variable name is same as solid phase name) and use keyword 'read concentration of radioelements from vtk file' in block 'initial condition - reactive transport'.

To read 'weight fraction of elements' from external file for structured grid version, please name the external file prefix.ngiwfe and use keyword 'read weight fraction of elements from file' in block 'initial condition - reactive transport'.

To read 'weight fraction of elements' from external file for unstructured grid version, please name the external file prefix.ngiwfe.vtk (variable name is same as element name) and use keyword 'read weight fraction of elements from vtk file' in block 'initial condition - reactive transport'.

To read 'neutron capture probability' from external file for structured grid version, please name the external file prefix.ngincp and use keyword 'read neutron capture probability from file' in block 'initial condition - reactive transport'.

To read 'neutron capture probability' from external file for unstructured grid version, please name the external file prefix.ngincp.vtk and use keyword 'read neutron capture probability from vtk file' (variable name is same as neutron capture element name) in block 'initial condition - reactive transport'.

Codes Update	**Remarks**
1. Update all the tiny_time used in the code to 1% of minimum timestep or 1.0d-12, whichever is bigger. This parameter is not consistent in the code that it ranges from 1.0d-20 to 1.0d-12 in different functions. This	

change may cause small difference in time step. 2. Format of mass related output (*.mac) is changed due to the change of aqueous component concentration contributed by noble gas ingrowth. Two new columns have been added. 3. Update of gradient reconstruction calling format. Algorithm remains the same but the code has been reorganized to make it easier to maintain.	
Benchmarks Update	**Remarks**
1. benchmarks_new_add\intermittent-reaction-V2.5 2. benchmarks_new_add\noble-gas-ingrowth-V2.5 There is mineral molecular weight problem in mineral database. This part was fixed in the Benchchmarks V2.5.1.875 on Biot-udc workstation, as shown below: Currently used in Benchchmarks V2.5.1.875 on Biot-udc and after: Na-Jarosite (484.6978) K-Jarosite (500.81) (Reference: GWB/PHREEQC MinteqV4) Previously used in Benchchmarks V2.5.1.875 on Stargazer and earlier: Jarositena (478.6978) jarositek (494.8100).	
Documentations Update	**Remarks**
1. Version report.	
Bugs Fixed	**Remarks**
1. Fix bug in flush() function in the code. The standard flush function requires passing the file unit number, for example, ‘call flush(1000)’. The non-standard version is supported in the GNU compiler and old intel compiler. For the new intel compiler (e.g., version 2021.11.0), the code can still be compiled without error but crashes after this function is called.	
Notes	**Remarks**
Summary of code verification (test running)	**Remarks**
The verification of this branch has been tested against Benchmarks_V2.4.8.869. The final results between the latest version and the reference version V2.4.8.869 are shown in Diff_Benchmarks_V2.5.1.875_vs_V2.4.8.869. The difference in the results is from code updates and new features implemented. The difference is consistent with the changes in the code. Starting from this version, compilation and verification of MIN3P code will	The test running has been completed on Stargazer2012 workstation. The same benchmarks

be done on Biot-udc workstation for Windows users. The following benchmarks are tested during the work handover from Stargazer to Biot-udc workstation. Benchmarks_V2.5.1.875 : Database with molecular weight problem fixed. Code was compiled on Biot-udc and run on Biot-udc. Benchmarks_V2.5.1.875_Old_Database: Database without molecular weight problem fixed. Code was compiled on Biot-udc and run on Biot-udc. Benchmarks_V2.5.1.875_Old_Database_Stargazer: Database without molecular weight problem fixed. Code was compiled on Stargazer and run on Stargazer. For the next verification, we will use Benchmarks_V2.5.1.875 as the reference. All the future benchmark verification will be done on Biot-udc workstation.	will be run on Biot-udc workstation and the results will be used for the future MIN3P release.

Revision 869

Revision Number	869	**Version Number**	V2.4.8.869
Commit Date	2023-08-18	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel Compiler, GCC Compiler with C++11 or PGI compiler. 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32; WINDOWS; USG; CGAL; 2.3 Preprocessor definition for parallel version: RELEASE_X64; WINDOWS; PETSC; MPI; PARDISO; OPENMP; USG; CGAL; LIS; SCHEDULE_STATIC; 2.4 Preprocessor definition for resources: RELEASE_X64| RELEASE_X64_P 2.5 Additional Options: /fp:strict 3. External Library (Optional) 3.1 PETSc: PETSc V3.4 to V3.18 3.2 LIS: LIS V1.7.36 to V2.0.34 3.3 CGAL: CGAL V4.7	

System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL	

https://www.intel.com/content/www/us/en/developer/articles/tool/compilers-redistributable-libraries-by-version.html	
Summary of this version	**Remarks**
Key changes in MIN3P-THCm V2.4.8.869 mainly include new solvers for local chemistry, database backup to simulation folder, 'millington-quirk adjustable' tortuosity correction, linear interpolation of ice sheet loading/unloading factors and boundary pore water pressure, mass balance check for surface site, domain decomposition for ArchiSimple code.	
New Features	**Remarks**
1. Add an underrelaxation factor to local chemistry. 2. Add the solver settings to the local chemistry solver. The previous code uses hardwired parameters in this part. 3. Add an extra linear solver for local geochemistry. Least squares method based on QR factorization and singular value decomposition (SVD) were implemented. 4. Add database backup feature by saving all the items used in file prefix_o.dbs. This backup database allows the user to rebuild the database for the MIN3P code (new feature 5). 5. Add a feature to use the backup database. 6. Add 'millington-quirk adjustable' to represent Def = D_0 * sat^(b+1) * phi ^(a+1) so that tortuosity is expressed as tau=phi^a * sat^b. 7. Add non-physical aqueous concentration check. When the aqueous concentration exceeds the specified value, the simulation will output error information and terminate. By default, the maximum value is 1.0d300, so this feature is not activated. 8. Add a mass balance check for surface site in case of non-physical solution due to the failure of the local geochemistry solver. 9. Add linear interpolation of the ice sheet loading/unloading factor pw and pice. 10. Add linear interpolation of the boundary pore pressure factor due to ice sheet loading/unloading. In the previous version, before the ice sheet advances, the boundary condition of the ground surface is set to the first type; during the advance of the ice sheet, the boundary condition at the base of the ice sheet basal is closed and the pore pressure underneath builds up; however, during the retreat of the ice sheet, the boundary at the base of the ice sheet is set back to the boundary condition before the advance of the ice sheet, plus the extra pressure head (pw) caused by	

the ice sheet. When linear interpolation of the ice sheet loading/unloading factor pw is considered, pw might be small at the beginning of the ice sheet retreat process, making a significant drop in pressure head. By applying the accumulated pore pressure back as part of the boundary condition during ice sheet retreat, the pressure change is smooth, avoiding a significant drop in pressure head.

11. Add 'transient ice sheet factors' for the unstructured grid version, including the scaling factor of the pressure head (pw), the pore stress (pice), and the boundary pore pressure (pp) due to ice sheet loading/unloading. By default, these parameters are constants during the ice sheet advance and the ice sheet retreat, respectively. By applying transient ice sheet factors, these variables can be changed over time.

12. Add domain decomposition method for ArchiSimple code (root-related model). The user can now use the MPI parallel version when ArchiSimple code is used. Please note there are some random functions inside ArchiSimple code that will cause a small difference due to this effect.

13. Add support for heat transport under steady-state condition.

14. Add support for skipping transient data output during the specified time range.

15. Add support for the transient boundary condition for boundary types 'pch4', 'pco2x' and 'par'

Usage of New Features	**Remarks**

1. Add the keyword 'user specified underrelaxation factor' followed by parameters into the data block 'control parameters - local geochemistry'. An example is shown below.

```
   'user specified underrelaxation factor'
   0.2              ;underrelaxation factor
```

2. Add the keyword 'solver settings' followed by parameters in the data block 'control parameters - local geochemistry'. An example is shown below.

```
   'solver settings'
   2000             ;maximum iteration number
   1                ;information level
```

3. Add the keyword 'linear solver type' followed by 'gaussian', 'qr', or 'svd'. By default, the 'gaussian' method is used. An example is shown below.

```
   'linear solver type'
   'svd'
```

4. No extra keyword is required.

5. User can either copy the database items from the backup database prefix_o.dbs to create the database files (e.g., comp.dbs, mineral.dbs, ...) or specify the keyword 'use database backup' into the data block 'geochemical system'. The latter requires that the backup database file be

renamed as prefix.dbs. Once the code is successfully launched, it will create a folder 'database.backup' and save all the database items inside this folder in separated database files. Important note: The database backup does not include any unused items in the input file prefix.dat, which might differ from the possible secondary species and minerals listed in prefix_o.psp file using the original database.

6. Add 'millington-quirk adjustable' followed by parameters a and b into the data block 'control parameters - reactive transport'. An example is shown below:

```
   'millington-quirk adjustable'
   1.0          ;parameter a
   2.0          ;parameter b
```

7. To overwrite the default value, use the following keyword in block 'control parameters - local geochemistry'

```
'maximum component concentration'
1.0d5          ;change maximum allowed aqueous concentration to 1.0d5.
```

8. No keyword is required.

9. Add the keyword 'linear interpolation of factor for pw and pice' and the range of pw and pice into the data block 'ice sheet loading/unloading'. An example is shown below.

```
! Data Block 17: Ice Sheet loading/unloading
! -------------------------------------------------------------------------
!
'ice sheet loading/unloading'

'linear interpolation of factor for pw and pice'

.false.
440000.0d0
4000.0d0
2.5d0                         ! a
1.0d0                         ! b
0.0d0                         ! logkxx
0.0d0                         ! logkyy
0.0d0                         ! logkzz
0.0d0                         ! temperature in the permafrost zone
1.0d3                         ! ice density
1.0d3                         ! fresh water density
3                             ! number of stages
1                             ! istage
.false.
0.0d0 0.0d0                   ! Factor for pw
1.0d0 1.0d0                   ! Factor for pice
0.0d0   12500.0d0             ! time(1,i),time(2,i)
0.0d0   439950.0d0            ! l(1,i),l(2,i)
```

```
0.0d0   2000.0d0             ! h(1,i),h(2,i)
140000.0d0  540000.0d0  ! l1perm(1,i),l1perm(2,i)
0.0d0   200000.0d0         ! l2perm(1,i),l2perm(2,i)
100.0d0    100.0d0           ! thickperm(1,i),thickperm(2,i)
2                                  ! istage
.false.
0.0d0 0.0d0                   ! Factor for pw
1.0d0 1.0d0                   ! Factor for pice
12500.0d0  17500.0d0     ! time(1,i),time(2,i)
439950.0d0   439950.0d0 ! l(1,i),l(2,i)
2000.0d0   2000.0d0        ! h(1,i),h(2,i)
540000.0d0  540000.0d0  ! l1perm(1,i),l1perm(2,i)
200000.0d0   200000.0d0 ! l2perm(1,i),l2perm(2,i)
100.0d0    100.0d0           ! thickperm(1,i),thickperm(2,i)
3                                  ! istage
.true.
0.0d0  0.95d0                 ! Factor for pw
1.0d0  1.0d0                   ! Factor for pice
17500.0d0  22500.0d0     ! time(1,i),time(2,i)
439950.0d0 0.0d0           ! l(1,i),l(2,i)
2000.0d0   0.0d0             ! h(1,i),h(2,i)
0.0d0      0.0d0               ! l1perm(1,i),l1perm(2,i)
0.0d0      0.0d0               ! l2perm(1,i),l2perm(2,i)
0.0d0      0.0d0               ! thickperm(1,i),thickperm(2,i)

'concentration input'         ! rain water composition equilibrated with
pco2=0.00034 [atm]
0.2915E-02      'charge'     ;cl
0.4000E-04      'free'         ;br
0.5220E-03      'free'         ;na
0.6400E-04      'free'         ;k
6.0                  'ph'           ;h
0.1722E-04      'free'         ;hco3
0.6240E-03      'free'         ;ca
0.8230E-03      'free'         ;mg
0.2600E-03      'free'         ;so4
1.0d0               'po2'         ;o2(aq)

'done'
```

10. Add the keyword 'linear interpolation of ice sheet factors' and range of the factor to the data block 'ice sheet loading/unloading'. An example is shown below.

```
! Data Block 17: Ice Sheet loading/unloading
! --------------------------------------------------------------------------
!
!
'ice sheet loading/unloading'
'linear interpolation of ice sheet factors'
```

```
.false.
440000.0d0
4000.0d0
2.5d0                    ! a
1.0d0                    ! b
0.0d0                    ! logkxx
0.0d0                    ! logkyy
0.0d0                    ! logkzz
0.0d0                    ! temperature in the permafrost zone
1.0d3                    ! ice density
1.0d3                    ! fresh water density
4                        ! number of stages
1                        ! istage
.false.
0.0d0 0.0d0              ! Factor for pw
1.0d0 1.0d0              ! Factor for pice
0.0d0 0.0d0              ! Factor for boundary pore pressure, no pore
pressure is passed back as a boundary condition.
0.0d0   12500.0d0        ! time(1,i),time(2,i)
0.0d0   439950.0d0       ! l(1,i),l(2,i)
0.0d0   2000.0d0         ! h(1,i),h(2,i)
140000.0d0  540000.0d0 ! l1perm(1,i),l1perm(2,i)
0.0d0   200000.0d0       ! l2perm(1,i),l2perm(2,i)
100.0d0    100.0d0       ! thickperm(1,i),thickperm(2,i)
2                        ! istage
.false.
0.0d0 0.0d0              ! Factor for pw
1.0d0 1.0d0              ! Factor for pice
0.0d0 0.0d0              ! Factor for boundary pore pressure, no pore
pressure is passed back as a boundary condition.
12500.0d0  17500.0d0     ! time(1,i),time(2,i)
439950.0d0  439950.0d0 ! l(1,i),l(2,i)
2000.0d0   2000.0d0      ! h(1,i),h(2,i)
540000.0d0  540000.0d0 ! l1perm(1,i),l1perm(2,i)
200000.0d0  200000.0d0 ! l2perm(1,i),l2perm(2,i)
100.0d0    100.0d0       ! thickperm(1,i),thickperm(2,i)
3                        ! istage
.true.
0.0d0 0.95d0             ! Factor for pw
1.0d0 1.0d0              ! Factor for pice
1.0d0 0.0d0              ! Factor for boundary pore pressure, 100% pore
pressure is passed back as part of boundary condition and then
decreased linearly to zero.
17500.0d0  18000.0d0     ! time(1,i),time(2,i)
439950.0d0 395955.0d0  ! l(1,i),l(2,i)
2000.0d0   1990.0d0      ! h(1,i),h(2,i)
0.0d0  0.0d0             ! l1perm(1,i),l1perm(2,i)
0.0d0  0.0d0             ! l2perm(1,i),l2perm(2,i)
0.0d0  0.0d0             ! thickperm(1,i),thickperm(2,i)
4                        ! istage
.true.
```

```
0.95d0 0.95d0            ! Factor for pw
1.0d0 1.0d0              ! Factor for pice
0.0d0 0.0d0              ! Factor for boundary pore pressure
18000.0d0  22500.0d0     ! time(1,i),time(2,i)
395955.0d0  0.0d0        ! l(1,i),l(2,i)
1990.0d0   0.0d0         ! h(1,i),h(2,i)
0.0d0  0.0d0             ! l1perm(1,i),l1perm(2,i)
0.0d0  0.0d0             ! l2perm(1,i),l2perm(2,i)
0.0d0  0.0d0             ! thickperm(1,i),thickperm(2,i)

!'compute permafrost'

'done'
```

11. Add the keyword 'transient ice sheet factors' into the data block 'control parameters - ice sheet loading/unloading' and then provide the file (prefix.isf) should be provided with the following input.

```
!c file format of prefix.isf (ice sheet factors)
time_1 pw_grow pw_decay pice_grow pice_decay pp_grow pp_decay
time_2 pw_grow pw_decay pice_grow pice_decay pp_grow pp_decay
...
time_n pw_grow pw_decay pice_grow pice_decay pp_grow pp_decay
```

12. No extra keyword is required.

13. No extra keyword is required.

14. Add the keyword 'skip time of transient data' into data block 'output control', an example is shown below.

```
   'skip time of transient data'
   -1.0d3  0.0d0        ;transient output between -1.0d3 and 0.0 is skipped.
```

15. No extra keyword is required.

Codes Update | **Remarks**

1. Update 'millington-quirk experimental' formula from Def = D_0 * sat^(b) * phi ^(a +1) to Def = D_0 * sat^(b+1) * phi ^(a+1) so that tortuosity is expressed as tau=phi^a * sat^b.

2. Revert 'millington-quirk experimental' formula to Def = D_0 * sat^(b) * phi ^(a +1). Add 'millington-quirk adjustable' to represent Def = D_0 * sat^(b+1) * phi ^(a+1) so that tortuosity is expressed as tau=phi^a * sat^b.

3. Change the file name from prefix.evap to prefix_o.evap for the evaporation information output.

4. Disable spatial output in terms of depth. For structured grid version, spatial output in terms of depth is available to non-vector output, but not available to vector output such as velocity. For unstructured grid version, this feature is disabled for all spatial outputs to avoid confusion. This

update will cause a difference in spatial output in the unstructured grid version. 5. Add extra linear solvers for local geochemistry in computing surface composition based on the equilibrated solution (surfcomp.F90). 6. Update the spatial output. By default, output is enabled whenever the output control is used or not. In the new version, spatial output is enabled when 'output control' block is used, and vice versa. 7. Update some root-related parameters to quadratic precision (e.g., rewm_quad, rsum_vprop_quad, rsum_vprop_inc_quad). The roundoff error in the collective operator (e.g., reduction in OpenMP and MPI_Allgather in MPI) causes some difference in the results. However, since MPI_Allgather does not well support quadratic precision in Fortran, it is not used in MPI parallel version. The latest Intel Fortran compiler and Intel MPI support quadratic precision in Fortran. 8. Update the format of the storage change code by moving time step parameter to the storage calculation function (storvs and storfs). By doing this, all the storage term functions have the same format. Before, storvs and storfs do not include time-step parameters that this term should be divided by time step in jacvs and jacfs. Theoretically, the results are the same, but may cause small differences due to the round-off error (e.g., Bemidji in gas-bubble-V1.0.303/Amos and Mayer 2006). The difference is not identifiable. 9. Remove the keyword 'restart at maximum timestep'. The projected timestep saved in the restart file will be used automatically.	
Benchmarks Update	**Remarks**
1. benchmarks_new_add/Blaise-V2.4.0.860/A1 2. benchmarks_new_add/Blaise-V2.4.0.860/A2 3. benchmarks_new_add/Blaise-V2.4.0.860/A3	Demonstration of QR and SVD solver.
Documentations Update	**Remarks**
1. Version report.	
Bugs Fixed	**Remarks**
1. Fix a bug in passing linear solver convergence parameters from PETSc command options. 2. Fix a bug in writing 'The strong inhibition is owing to the component ' in MPI parallel version. This information should be called by master processor only, not other processors. 3. Fix a bug in index of 'sorbed species' to 'non-aqueous components' in outputlc.F90. Value of isites varies over sorbed species, as shown below. do isb = 1,nsb_surf	

```
          !cff fix up for isites > 1
          !isites = 1

!cdsu this is not right, value of isites depends on 'sorbed species'
          isites = isurf2isite(isb)

          write(igen,'(a12,3(3x,1pe15.6e3))')                     &
               namesb_surf(isb),csb_surf(isb,tid),               &
               csb_surf(isb,tid)/site_mass(isites)/              &
               site_area(isites),                                &
               csb_surf(isb,tid)/totsitec*100.0d0
        end do
      end if
```

4. Fix a bug in mineral mass balance.

5. Fix a bug in jacrt when dealing with tinyrate. This bug is only in revision 860.

6. Fix a bug in spatial output when the number of spatial output is zero. The spatial output is not correctly assigned when the number of output is zero.

7. Fix a bug in PETSc functions ISLocalToGlobalMappingRestoreIndices and DMDAGetBoundingBox for PETSc new version 3.18 and later. The returned value from DMDAGetBoundingBox is not strict and will cause some differences in the MPI parallel version compared to the sequential version.

8. Fix a bug in calling the rew function. Should pass address instead of value in the first parameter.

9. Fix a bug in the heat transport related function under steady-state condition.

10. Fix a bug in NaN in the precipitation/dissolution rate in function modrate.F90. Set the ratemp to zero when the ratem is zero.

11. Fix a bug in the selection through coordinates for transient output. In the MPI parallel version, there might be an index shift in which the actual output location is different from the specified coordinates. This bug only exists in structured grid version using MPI parallelization.

12. Fix a bug in the update of the total aqueous concentration in tprfrtlc.F90. The function ‘comptotc’ should be placed before data output. The output of total aqueous concentration in the previous version does not consider equilibrium reactions.

13. Fix a bug in reading the 'salinity dependent reaction rate of minerals'. The previous code only allows one salinity dependent mineral. It will also cause a problem in MPI parallel version since only the master processor

is returned. This bug does not affect simulation of one salinity dependent mineral.

  previous code

```
        do i= 1, nmin
          ierrcd = 9
          read(icnv,*,err=999,end=999) nametemp
          read(icnv,*,err=999,end=999) sdtype(i), nfact

          if (sdtype(i) .eq. 'equation') then
            icount = 0
            do im=1,nm
              if (nametemp .eq. namem(im)) then
                salinity_dependent(im) = .true.
                nfac(im) = nfact
                icount=im
                iciunt1 = iciunt1 +1
              end if
            end do
            if (icount .eq. 0 .or. iciunt1 .lt. nmin) then
              if (rank == 0 .and. b_enable_output) then
                write(ilog,*) 'error reading salinity dependent factors of
minerals'
                exit
              end if
            end if

            ...

          end do
```

  new code

```
        do i= 1, nmin
          ierrcd = 9
          read(icnv,*,err=999,end=999) nametemp
          read(icnv,*,err=999,end=999) sdtype(i), nfact

          if (sdtype(i) .eq. 'equation') then
            icount = 0
            do im=1,nm
              if (nametemp .eq. namem(im)) then
                salinity_dependent(im) = .true.
                nfac(im) = nfact
                icount=im
              end if
            end do
            if (icount .eq. 0) then
              if (rank == 0 .and. b_enable_output) then
```

write(ilog,*) 'error reading salinity dependent factors of minerals' end if goto 999 end if ... end do 14. Fix a bug in applying the restart time and projected timestep in the restart file. The project timestep is saved in the restart file but never used during restart. The restart time now can support negative initial time.	
Notes	**Remarks**
Summary of code verification (test running)	**Remarks**
The verification of this branch has been tested against Benchmarks_ V2.3.7.850. The final results between the latest version and the reference version V2.3.7.850 are shown in Diff_Benchmarks_V2.4.8.869_vs_V2.3.7.850. The difference in the results is from bugs being fixed, code updates and new features. The difference is consistent with the changes in the code. Tests for different revisions have also been made during code development/maintance. These tests indicate that the difference is from bugs being fixed, code updates and new features being added that changes the output file. Detail information is shown below: 1. Differences from bugs 1-5 and code updates 1-5 are shown in Diff_Benchmarks_V2.4.4.861_vs_V2.3.7.850 folder. 2. Differences from other bugs and code updates are shown in Diff_Benchmarks_V2.4.8.869_vs_V2.4.4.861.	

Revision 850

Revision Number	850	**Version Number**	V2.3.7.850
Commit Date	2023-01-27	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel Compiler, GCC Compiler with C++11 or PGI compiler. 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32; WINDOWS; USG; CGAL; 2.3 Preprocessor definition for parallel version: RELEASE_X64; WINDOWS; PETSC; MPI; PARDISO; OPENMP; USG; CGAL; LIS; SCHEDULE_STATIC; 2.4 Preprocessor definition for resources: RELEASE_X64| RELEASE_X64_P 2.5 Additional Options: /fp:strict 3. External Library (Optional) 3.1 PETSc: PETSc V3.4 to V3.18 3.2 LIS: LIS V1.7.36 to V2.0.23 3.3 CGAL: CGAL V4.7			Add support for PETSc 3.17 and 3.18 version;
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL			

http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
Key changes in MIN3P-THCm V2.3.7.850 include implementation and/or updating of noble gas ingrowth, root water related module, simplified icesheet freeze/thaw processes, and output of dilution index and spatial moment.	
New Features	**Remarks**
1. Add support of restart and parallelization for solute uptake, root respiration and mineral return related modules. Please note ArchiSimple module might not work for MPI parallel version since no detail information/algorithm is provided by original developers. 2. Add nodes/cells selection constraint by distance to plane(s). The plane is described as $ax+by+cz+d = 0$. The distance from (x_0,y_0,z_0) to the plane is $\vert ax_0+by_0+cz_0+d\vert/\mathrm{sqrt}(a^2+b^2+c^2)$. 3. Add first order noble gas ingrowth function. The reaction rate in described by rate = -k[R] where k is rate constant and [R] is the concentration of mineral phase in unit mol/L bulk. The current function assumes the noble gas produced will be released to pore fluid immediately. 4. Add 'double layer basal temperature model' to estimate ice sheet basal temperature based on 1D steady state heat equation (ODE) for double layer medium. After the basal temperature is estimated, it is then passed back to MIN3P code as a boundary condition. This is a simplified model to computer temperature distribution between two materials. 5. Add pressure melting point function to freeze/thaw processes. The freeze/thaw temperature is adjusted based on the calculated pressure value. When this feature is used, the relative conductivity is adjusted accordingly. 6. Add 'read battaglia reduction function from file' for plant transpiration and passive/rejective uptake. 7. Add 'read root water uptake from file' for plant transpiration and passive/rejective uptake. 8. Add 'calculation dilution index' that represents the progressive distribution of solute mass. The calculation follows Kitanidis (1994) paper with porosity and saturation included for heterogeneous material. The output file is prefix_o.dix. 9. Add 'calculate spatial moment' (first order and second order) that represents the plume centre and transport distance. The calculation follows publication of Goltz and Huang (1987), Oware and Moysey	

(2014). The output file is prefix_o.spm.

10. Add 'linear transient maximum time step' to control maximum timestep change over time. Different from 'transient maximum time step' which uses constant maximum time step during a specified period, the new function uses a linear function to calculate maximum time step during a specified period.

11. Add output of root water uptake related parameters (e.g., reduction factor, wilting point saturation, pressure ...) into gsroot and gbroot files.

Reference:

Goltz, M. N. & Roberts, P. V. Using the method of moments to analyze three-dimensional diffusion-limited solute transport from temporal and spatial perspectives. Water Resource Research, 1987, 23, 1575-1585.

Oware, E. K. & Moysey, S. M. J. Geophysical evaluation of solute plume spatial moments using an adaptive POD algorithm for electrical resistivity imaging. Journal of Hydrology, 2014, 517, 471-480.

Usage of New Features	Remarks

1. No specific keywords are required. To use restart file of mineral return related function, please change the file name of restart.append.tmp* to restart.append.dat, similar as change the file name of restart.tmp* to restart.dat.

2. To use constraint to one plane, add keyword 'extent of zone: distance to plane constraint' in zone selection followed by parameter a, b, c, d and distance range. An example is shown below:

```
'extent of zone: distance to plane constraint'
1.0  2.0 3.0 4.0 10.0 100.0
!distance to plane 1.0x+2.0y+3.0z+4.0=0 should be between 10.0 and 100.0.
```

To use constraint to multiple planes, add keyword 'extent of zone: distance to planes constraint' in zone selection followed by number of places and parameters for each plane. An example is shown below:

```
'extent of zone: distance to planes constraint'
3
0 0 1 -1.0 0.2 0.5
!distance to plane z-1.0=0 should be between 0.2 and 0.5.
1 1 0 -1 0.1 0.2
!distance to plane x+y-1.0=0 should be between 0.1 and 0.2.
1 -1 0 0 0.0 0.4
!distance to plane x-y=0 should be between 0.1 and 0.4.
```

3. To use first order noble gas ingrowth, a fractional term describing the concentration of mineral phase should be added into mineral.dbs. An example of ^{238}U to ^{4}He is shown below.

```
!
!'238u-yield-8he'
!
'238u-yield-8he'
'surface'
238.0    19.1                 ;to be updated accordingly for other reactions
1  '4he(aq)'  8               ; 2  '206Pb(aq)'  1.0  '4he(aq)'  8, ignore Pb here
```

```
    'irreversible dissolution'
    'fractional C^m'   1          ;function of solid or mineral phase concentration
    '238u-yield-8he'   1
```

4. Add keyword 'double layer basal temperature model' into Data Block 17: control parameter of ice Sheet loading/unloading followed by ice sheet conductivity (W/mK), porous media conductivity (W/mK), and temperature change per meter elevation increase. An example is shown below:

```
    !use 1D steady state heat equation for double layer medium to estimate ice sheet
    basal temperature
    'double layer basal temperature model'
    2.22            ;W/mK, 1D thermal conductivity of ice sheet
    3.50            ;W/mK, 1D thermal conductivity of porous media (sedimentary rock)
    -0.006          ;oC per meter, temperature change per meter elevation increase
```

5. Add keyword 'compute pressure melting point' into Data Block 6: control parameters - variably saturated flow.

6. Add keyword 'read battaglia reduction function from file' into data block 'plant transpiration and passive/rejective uptake'. External file name is prefix.brf and the file should be in tecplot format, with first three columns represent x, y and z, followed by fitting parameters rew0 and p1.

7. Add keyword 'read root water uptake parameters from file' into data block 'plant transpiration and passive/rejective uptake'. External file name is prefix.rwu and the file should be in tecplot format, with first three columns represent x, y and z, followed by parameters h1lim, h1field, h1opt, root length density and uptakefactor. Please note if root length density is read from external file prefix.rld, then the root length density is ignored from rwu file.

8. Add keyword 'calculation dilution index' into data block 'control parameters - reactive transport'.

9. Add keyword 'calculate spatial moment' into data block 'control parameters - reactive transport'.

10. Add 'linear transient maximum time step' into data block 'time step control - global system' followed by number of time step controls and detail time. An example is shown below:

```
    'linear transient maximum time step'
    2
    1.0d1  1.0d2  1.0d-2  1.0d-1    ;tstart, tend, max_delt_start, max_delt_end
    !maximum time step ranges from 1.0d-2 to 1.0d-1 during period 1.0d1 to 1.0d2
    1.0d2  1.0d4  1.0d-1  1.0d0    ;tstart, tend, max_delt_start, max_delt_end
    !maximum time step ranges from 1.0d-1 to 1.0d0 during period 1.0d2 to 1.0d4
```

11. No extra keyword is required.

Codes Update	**Remarks**
1. After revision 845, Perot's method () for velocity reconstruction in unstructured grid is set as default method. This method works better for	

heterogeneous problem. Before revision 844, averaged method, which is estimated based averaged gradient and hydraulic conductivity, is used by default for velocity reconstruction. To use the averaged method in the new version, please specify keyword 'use averaged velocity reconstruction' into data block 'spatial discretization'. Since the method used to estimate node velocity is changed, Courant number is also changed in density dependent flow, resulting in the change of time step estimation. For density dependent flow problem, this update changes velocity output as well as some other output when unstructured grid is used (due to the change in timestep). For other problems, this update changes the velocity output only when unstructured grid is used. Reference: *Blair Perot, 2000, Conservation properties of unstructured staggered mesh schemes, Journal of Computational Physics, 159, 58-89, doi:10.1006/jcph.2000.6424.*	
Benchmarks Update	**Remarks**
1. Add noble gas ingrowth benchmark into folder ‘Benchmarks\benchmarks_new_add\noble-gas-ingrowth-V2.3.7.850’. 2. Add demonstration example for 'double layer basal temperature model' into ‘Benchmarks\benchmarks_new_add\ basin-2d-polythermal-icesheet-V2.3.7.850’.	
Documentations Update	**Remarks**
1. Version report.	
Bugs Fixed	**Remarks**
1. Fix bug in restart file for solute uptake and mineral return function. Extra restart data are required and these data are placed in restart.append.tmp* file. When restart file is used, the name of the extra restart file should be changed from restart.append.tmp* to restart.append.dat, similar as changing restart.tmp* to restart.dat. 2. Fix bug in updating root length density from file. There is small time difference that may cause root lengh density be updated one timestep earlier. 3. Fix bug in OpenMP parallel version when solute uptake and mineral return function is used. 4. Fix bug in spatial data output using binary format (MPI parallel version). The number of variables in the head may be inconsistent with the actual output. This bug makes some variables unaccessable in postprocessing. 5. Fix bug in time step calculation due to truncation error. For example, the original code delt = delt + gs_tout(igstime) - time may not return accurate value if there is very big difference between delt and gs_tout	

(time). e.g., delt = 1.0e-12, gs_tout = 1000000.999999, time = 1000000.999998, 'delt + gs_tout(igstime) - time' will return 0.000001 while 'delt + (gs_tout(igstime) - time)' will return 0.000001000001. This small difference causes some difference in time step and the final results. This bug is benign and the difference is very small.

6. Fix bug in transient boundary condition when restart file is used. The boundary condition update after restart is not strict due to the small timestep tolerance, as indicated in previous bug.

7. Fix bug in NaN output in prefix_0.gsp file when plant transpiration and solute uptake is used.

8. Fix bug in memory allocation for unstructured grid code output. The size of variable totcflux_vols does not match variable totcflux. The error only occurs on Linux machine and being ignored on Windows and Mac.

9. Fix optional argument compatibility issue in Intel Fortran compiler. Explicit interface is required for Intel Fortran Compiler with support to Fortran 2003.

10. Fix bug in 'third' and 'third-evap' boundary conditions with when diffusive flux is considered. The diffusive flux for third and third-evap boundary conditions can be turned on by keyword 'include diffusive flux for third/mixed boundary condition'. By default, third and third-evap boundary condition do not consider diffusive flux at the boundary. When diffusive flux is considered for the third boundary condition, it is same as mixed boundary condition for the aqueous phase. This bug does not affect any benchmarks since this feature is never used before.

11. Fix bug in velocity reconstruction and output when velocity average method is used. The previous code skips some extreme value when material properties are highly different (e.g., matrix and fracture). This bug causes some difference in velocity output using unstructured grid. For density dependent flow, using Perot velocity output may cause small difference due to the change in Courant number estimation. If Courant number is used in tilmestep control, the final results may not be identical, but the results are very close and the difference are acceptable.

12. Fix bug in Perot velocity reconstruction. Operator in fluxfs_cvol_usg function is not correctly placed.

13. Fix bug in dummy argument of ‘typetotbc’ in ice sheet model. The length of character should be big enough or system decided (len=*) when used as a dummy argument. This bug causes error if boundary type is switched during ice sheet modeling when restart file is used. The bug was introduced in revision 826 but did not exist before revision 825. The bug was fixed after porting the branch to trunk version.

Notes	**Remarks**

Summary of code verification (test running)	**Remarks**
The verification of this branch has been tested against Benchmarks_V2.2.6.826. The final results between the latest version and the reference version V2.2.6.826 are shown in Diff_Benchmarks_V2.3.7.850_vs_V2.2.6.826. The difference in the results is from bugs being fixed, code updates and new features. The difference is consistent with the changes in the code. Tests for different revisions have also been made during the code development/maintance. These tests indicate that the difference is either from bugs being fixed, code updates and new features being added that changes the output file. Detail information are shown below: 3. Differences from bugs 1-5 are shown in Diff_Benchmarks_V2.3.1.830_vs_V2.2.6.826 folder. 4. Differences from bug 6 are shown in Diff_Benchmarks_V2.3.1.834_vs_V2.3.1.830. 5. Differences from bug 7 to 12 and code update 1 are shown in Diff_Benchmarks_V2.3.7.845_vs_V2.3.1.834. 6. Differences from new feature 11 are shown in Diff_Benchmarks_V2.3.7.846_vs_V2.3.7.845.	

Revision 826

Revision Number	826	**Version Number**	V2.2.6.826
Commit Date	2022-03-24	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel Compiler, GCC Compiler with C++11 or PGI compiler. 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32; WINDOWS; USG; CGAL; 2.3 Preprocessor definition for parallel version: RELEASE_X64; WINDOWS; PETSC; MPI; PARDISO; OPENMP; USG; CGAL; LIS; SCHEDULE_STATIC; 2.4 Preprocessor definition for resources: RELEASE_X64| RELEASE_X64_P 2.5 Additional Options: /fp:strict 3. External Library (Optional) 3.1 PETSc: PETSc V3.4 to V3.16 3.2 LIS: LIS V1.7.36 to V2.0.23 3.3 CGAL: CGAL V4.7			Add support for latest PETSc 3.16 version;
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL			

http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
MIN3P-THCm V2.2.6.824 has many new features added/updated and bugs fixed. Specifically, this version added ice sheet modeling over 3D domain; added extensive zone selection methods for both structured grid and unstructured grid code; added anisotropic flow and solute transport for MCD code; added latest root water/solute uptake, root respiration and leaf recycle models; ported latest ArchiSimple code.	
New Features	**Remarks**
1. Add output of ice saturation and water saturation when water freezing/thawing is considered. Variable name in the output file are s_ice and s_water, respectively. 2. Add support of 'extent of zone: distance constraint' for node/cell selection. 3. Add support of heat parameters update for ice when freezing/thawing is considered. 4. Add buffer for zone selection using polygon. The nodes/cells located within the buffer are considered as selected. 5. Add zone selection using node/cell layer id. This feature is for layered mesh only. If nodes/cells are stored from top to bottom, then the top layer id is one and the bottom layer id is number of node/cell layers, and vice versa. 6. Add scaling factor for ice thickness. 7. Add scaling factor mineral reactivity. 8. Add 'extent of zone: top layer coordinates constraint' and 'extent of zone: bottom layer coordinates constraint'. 9. Add 'input units of mineral volume fractions' to use volume fraction of solid phase as mineral input unit. By default, the mineral input unit is volume fraction of bulk. 10. Add support of double precision output in ascii format. By default, the output in ascii format is single precision. 11. Add support of 'anticipated number of newton iterations' for flow and heat transport solver. The previous versions only have maximum number of newton iterations for flow and heat transport solver and use a constant to increase timestep. Sometimes this causes aggressive time step increasing which finally results into a lot of failed iterations.	

12. Add 'ellipsoid-fitting model' for ice sheet. The thickness of ice sheet is calculated based on ellipsoid function (x-x0)^2/a^2 + (y-y0)^2/b^2 + (z-z0)^2/c^2 = 1 and rotation angle alpha along Z axis.

13. Add support to convert pre-exist lake/river pressure head to ice sheet thickness during ice sheet advance stage.

14. Add 'gradient-radius' type initial/boundary condition. This is similar to 'gradient' type but use a radius distance instead of x (or y, z) coordinate.

15. Add node and cell selection by distance range to a specified point.

16. Add node and cell selection by distance range to a specified line segment.

17. Add parallel spatial data input (e.g., read material properties and/or initial conditions from files) using master processor and then broadcast to all other processors. In the previous versions, the data is read by all the processors. The performance depends on the system structure and networking bandwidth. There is no significant difference in performance between these two methods so far based on the tested case with 1 million control volumes.

18. Add feature to set memory limitation per processor so that when code crashes because of memory limitation, it reports error to the screen output and log file.

19. Add error information to screen output and log file when database file is not found or error in reading.

20. Add switch to turn off the respiration/exudation if the concentration is too low.

21. Add spatial and transient output of root density. The suffix is gsroot for spatial output and gbroot for transient output.

22. Add option to include diffusive flux for aqueous phase for third type bounary condition. The previous version only has advective flux in the third type boundary condition.

23. Add option to limite output of transient data. The previous versions only have limited spatial output.

24. Add output (prefix_o.rup) for total solute uptake, including passive uptake and respiration.

25. Add leaf recycling module. This module takes solute uptake to corresponding minerals and distribute them in the specified zone and during the specified time period. Two output files are added, prefix_o.rup and prefix_o.rupcm, respectively. File prefix_o.rup contains the solute uptake over time including passive uptake and respiration uptake. File

prefix_o.rupcm contains the recyclable solute uptake and returned mineral into specified zone over time. The latter includes total recyclable solute uptake and returned mineral in the current cycle as well as the past cycles. For both files, positive value means solute uptake from the porous media into the root system while negative value means solute exudes from root system into the porous media. 26. Add function to exclude diffusive flux from mixed boundary condition. 27. Add anistropy and unstructured grid capability for multi-component diffusion. 28. Add zone dependent loading factor. Loading factor can now be calculated based on mechanical parameters (e.g., Poisson's module) or specified values. 29. Add spatial weighting for gradient average at control volume interface. This feature is only available for Green-Gauss and Least Square gradient reconstruction method. For high order least square method, no spatial weighting is required since the gradient is directly calculated at the control volume interface. 30. Port Fred's root related code to latest MIN3P version and fixed bugs. 31. Add legacy root uptake method, which is the same as old version root water uptake formula before porting Fred's root related code. 32. Add root upscaling factors for ArchiSimple code in X, Y and Z directions. Please note these factors should be positive. 33. Assign default 'harmonic' spatial averaging for multicomponent diffusion if the spatial averaging for MCD is not specified. This is a required keyword to make MCD code work. The previous code does not check this and results are not correct if no spatial averaging is specified. 34. Add first order decay for aqueous species.	
Usage of New Features	**Remarks**
1. No extra keyword is required. When water freezing/thawing is activated, this feature available automatically. 2. Add keyword 'extent of zone: distance constraint' followed by a reference point coordinate (x, y, z) and distance range (min, max). Please note this keyword should be used together with general zone selection. 'extent of zone: all nodes in box' -1.0d10 1.0d10 0.0d0 0.0d0 -1.0d10 1.0d10 'extent of zone: distance constraint' 250.0 0.0 300.0 286.9 300.1	

3. Add keyword 'update energy balance parameters for ice' into data block 'control parameters - energy balance' and add ice heat capacity and ice thermal conductivity into data block 'physical parameters - energy balance'.

```
   !data block 'control parameters - energy balance'
   'update energy balance parameters for ice'

   !data block 'physical parameters - energy balance'
   'specific heat of ice'
   2108

   'ice thermal conductivity in x-direction'
   1.6d0

   'ice thermal conductivity in y-direction'
   1.6d0

   'ice thermal conductivity in z-direction'
   1.6d0
```

4. Add keyword 'polygon zone selection buffer' followed by buffer value into data block 'spatial discretization'. By default, the buffer value is 1.0d-4. The unit is same as the input mesh unit.

```
   'polygon zone selection buffer'
   1.0d-6      ; default value is 1.0d-4 is the keyword is not specified.
```

5. Add keyword 'extent of zone: boundary nodes by layer id', or 'extent of zone: internal nodes by layer id' or 'extent of zone: all nodes by layer id' followed by layer id or layer id range. For cell selection, change 'nodes' to 'cells'. An example is shown below.

```
   'extent of zone: boundary nodes by layer id'
   'read data'
   2          ; number of layer id to read
   3          ; first layer id to read
   7          ; second layer id to read

   'extent of zone: boundary nodes by layer id'
   'read data range'
   2          ; number of layer id ranges to read
   3, 5       ; first layer id range to read
   7, 9       ; second layer id range to read
```

6. Add keyword 'scaling factor of ice thickness' followed by scaling factor into data block 'control parameters - ice sheet loading/unloading'. An example is shown below.

```
   'scaling factor of ice thickness'
   2.0      ; ice thickness will be multiplied by a factor of 2.0.
```

7. Add keyword 'scaling factors of mineral reactivity' followed by scaling factor of each mineral reactivity into data block 'geochemical system'. An example is shown below.

```
   'scaling factors of mineral reactivity'
   1.0      ; mineral 1
   1.0      ; mineral 2
   5.0      ; mineral 3
   10.0    ; mineral 4
   0.1      ; mineral 5
```

8. Add keyword 'extent of zone: top layer coordinates constraint' and/or 'extent of zone: bottom layer coordinates constraint' followed by elevation range. Please note this keyword should be used together with general zone selection.

```
   'extent of zone: all nodes in box'
   -1.0d10  1.0d10  0.0d0  0.0d0  -1.0d10  1.0d10

   'extent of zone: top layer coordinates constraint'
   100.0  200.0        ;only the nodes whose top layer elevation between
100.0 and 200.0 are selected
```

9. Add keyword 'input units of mineral volume fractions' followed by unit into data block 'control parameters - reactive transport'. An example is shown below.

```
   'input units of mineral volume fractions'
   'volume fraction of solid phase'
```

10. Add keyword 'use double precision' into data block 'output control'.

```
   'output control'
   …
   'use double precision'
```

11. Add keyword 'anticipated number of newton iterations' into data block 'control parameters - variably saturated flow', followed by the number of newton iterations. An example is shown below.

```
   'control parameters - variably saturated flow'
   ...
   'anticipated number of newton iterations'
   20
```

Alternatively, user can put this new parameter inside newton iteration settings with the new keyword 'newton iteration settings plus'. An example is shown below.

```
   'newton iteration settings plus'
   1.0e-5                    ; increment for numerical differentiation
   20                        ; anticipated number of newton iterations
   60                        ; maxit_vs, max.  number of newton iterations
   1.0e-6                    ; tol_vs, convergence tolerance
   0.1                       ; sw_star
```

12. Add 'ellipsoid fitting ice sheet model' followed by seven parameters describing the model into data block 'initial condition - ice sheet loading/unloading' and 'boundary conditions - ice sheet loading/unloading'. An example is shown below:

```
   'ellipsoid fitting ice sheet model'
   582882.0            ; center of ellipsoid, X0, same unit as mesh
   5033160.0           ; center of ellipsoid, Y0, same unit as mesh
   0.0                 ; center of ellipsoid, Z0, same unit as mesh
   10000.0             ; principal semiaxes in X axis, same unit as mesh
   2000.0              ; principal semiaxes in Y axis, same unit as mesh
   1000.0              ; principal semiaxes in Z axis, same unit as mesh
   60.0                ; counterclockwise rotation angle ° (along Z axis)
```

For transient ice boundary conditions, add time series of these parameters into prefix.bcice file. An example is shown below:

```
   ! 1stt column: time
   ! 2nd column: center of ellipsoid, Y0, same unit as mesh
   ! 3rd column: center of ellipsoid, Z0, same unit as mesh
   ! 4th column: principal semiaxes in X axis, same unit as mesh
   ! 5th column: principal semiaxes in Y axis, same unit as mesh
   ! 6th column: principal semiaxes in Z axis, same unit as mesh
   ! 7th column: counterclockwise rotation angle (along Z axis) in degree

   0.0E+0 6.0E+5 5.0E+6 0.0E+0 1.0E+3 1.0E+3 1.0E+2 4.5E+1
   6.5E+2 6.0E+5 5.0E+6 0.0E+0 1.6E+4 2.6E+3 1.0E+2 4.5E+1
   1.2E+3 6.0E+5 5.0E+6 0.0E+0 2.9E+4 4.6E+3 1.0E+2 4.5E+1
   1.5E+3 6.0E+5 5.0E+6 0.0E+0 3.6E+4 5.8E+3 1.0E+2 4.5E+1
```

13. Add keyword 'transform pre-exist surface water to ice' into data block 'control parameters - ice sheet loading/unloading'.

14. Add keyword 'gradient-radius' and set direction character to 'r' (default if not set to 'r'). The direction character line should include the radius centre in x, y, z format. An example is shown below:

```
   'boundary type'
   'gradient-radius' 0.0d0
   'r' 0.0  0.0  0.0
   4000.0e0
   5.0e-4
```

15. Add keyword 'extent of zone: boundary (internal or all) nodes (or cells) within distance to point' followed by point coordinates line and distance range line. An example is shown below:

```
   'extent of zone: all nodes within distance to point'
   100.0   0.0   200.0    ; coordinates of a specified point
   50.0    500.0          ; minimum and maximum distance
```

16. Add keyword 'extent of zone: boundary (internal or all) nodes (or cells) within distance to line' followed by coordinates of two points and distance range. An example is shown below:

```
   'extent of zone: all nodes within distance to line'
   100.0   0.0   200.0    1000.0 0.0 200.0    ; coordinates of line  segment
   50.0    500.0  .true.  .false.             ; minimum and maximum
distance, and flags to indicate sphere extension for end points
```

17. Add keyword 'read spatial data using master processor' in to data block 'global control parameters'. Without this keyword, the data is read by all the processors.

18. Add keyword 'memory limitation per processor' followed by memory size and unit. An example is shown below.

```
   'memory limitation per processor'
   1000.0 MB      ; or 1.0 GB or 1.0e-3 TB or 1.0e-6 PB.
```

19. No extra keyword is required to use this feature.

20. Add keyword 'minimum aqueous concentration for root respiration' followed by value of minimum total aqueous concentration of each component into data block 'plant transpiration and passive/rejective uptake'.

21. No extra keyword is required to use this feature.

22. Add keyword 'include diffusive flux for third/mixed boundary condition' into section of boundary condition setting. This parameters is zone dependent.

23. Add keyword 'limited output of transient data' into data block 'output control'.

24. No extra keyword is required to use this feature.

25. The related keywords of component-mineral recycle should be added into data block 'plant transpiration and passive/rejective uptake'. There are two types of parameters: global parameters and zone dependent parameters.

The following keywords are for global parameters.

To define recyclable component-mineral pair, add keyword 'recycle components and minerals' followed by list of component-mineral pairs. An example is shown below.

```
   'recycle components and minerals'
   4
   'ca+2' 'CaOxalate_ox'     ; ca+2 is recycled as CaOxalate_ox
   'ca+2' 'calcite'          ; ca+2 is recycled as calcite
   'al+3' 'kaolinite'        ; al+3 is recycled as kaolinite
   'ca+2' 'kaolinite'        ; no recycle happens as kaolinite has no ca+2
```

To define return time, add keyword 'recycle and return times'. An example is shown below.

```
   ! Define mineral recycle and return times:
   ! time1: start time of solute uptake for recycle in current cycle.
   ! time2: start time of mineral return in current cycle.
   ! time3: end time of mineral return in current cycle.
   ! solute uptake before time1 or after time3 are not considered.
   ! Between time1 and time2 (stage1), only solute uptake is considered.
   ! Between time2 and time3 (stage2), the total solute uptake in stage1
   and solute uptake in stage2 are returned as mineral
   !
   ! format of 'recycle and return times'
   ! ntimes: number of cycles
   ! time1, time2, time3
   ! time1, time2, time3
   ! ...
   ! time1, time2, time3

   'recycle and return times'
   6
   0  0.9  1
   1  1.9  2
   2  2.9  3
   3  3.9  4
   4  4.9  5
   5  5.9  6
```

When overlap is allowed in mineral return zones, the same return zone can accept mineral return from solute uptake/respiration from different regions. To enable this feature, add keyword 'allow overlap in return zones'.

```
   'allow overlap in return zones'
```

By default, solute uptake during mineral return time is also considered as returned mass. To exclude solute uptake in leaf-mineral recycling during return time, add keyword 'exclude uptake during return time'.

```
   'exclude uptake during return time'
```

The following keywords are for zone dependent parameters.

To define the amount of component recycled as specific mineral, add keyword 'return component to mineral coefficients' followed by list of coefficients.

```
   'return component to mineral coefficients'
   0.9    ; 'ca+2' 'CaOxalate_ox', 90% ca+2 is recycled as CaOxalate_ox
   0.1    ; 'ca+2' 'calcite'       10% ca+2 is recycled as calcite
   1.0    ; 'al+3' 'kaolinite'     100% al+3 is recycled as kaolinite
   0.1    ; 'ca+2' 'kaolinite'     no recycle happens as kaolinite has no ca+2
```

To define return zones, add keyword 'return zones: box' or 'return zones: cylinder' followed by list of specified zones. An example is shown below.

```
   ! format 1: mineral return in box zone
   ! xmin, xmax, ymin, ymax, zmin, zmax
   ! xmin, xmax, ymin, ymax, zmin, zmax
   ! ...
   ! xmin, xmax, ymin, ymax, zmin, zmax

   ! format 2: mineral return in cylinder zone
   ! xcenter, ycenter, zmin, zmax, radius_min, radius_max
   ! xcenter, ycenter, zmin, zmax, radius_min, radius_max
   ! ...
   ! xcenter, ycenter, zmin, zmax, radius_min, radius_max

   ! 'return zones: box'
   ! 0.0  1.0  0.0  1.0  0.0 1.0
   ! 0.0  3.0  0.0  2.0  0.0 1.0
   ! 0.0  5.0  0.0  3.0  0.0 1.0

   'return zones: cylinder'
   0  2.5  0  1  0  1
   0  2.5  0  1  0  1
   0  2.5  0  1  0  1
   0  2.5  0  1  0  1
   0  2.5  0  1  0  1
   0  2.5  0  1  0  1
   ...
```

26. Add keyword 'exclude diffusive flux for third/mixed boundary condition' or 'exclude diffusive flux for mixed boundary condition' into each zone in data block 'boundary conditions - reactive transport'.

27. No extra keyword is required.

28. Add keyword 'user specified loading factor' followed by specified loading factor data block 'physical parameters - porous medium'. An example is shown below.

```
   'user specified loading factor'
   0.95d0
```

29. Add keyword 'gradient average spatial weighting' into data block 'spatial discretization' followed by spatial weighting method. An example is shown below.

```
   'gradient average spatial weighting'
   'centered'    ; default
   !'harmonic'
   !'minimum'
```

30. Add keyword 'pure evaporation' into data block 'global control parameters' when 'root uptake' is not activated. Detail documentation for root uptake is to be added.

31. Add keyword 'legacy root uptake method' into data block 'plant transpiration and passive/rejective uptake'.

32. Add keyword 'root upscale factor - archisimple' followed by three factors into data block 'plant transpiration and passive/rejective uptake'. An example is shown below:

```
   'root upscale factor - archisimple'
   8.0 1.0 8.0       ; x, y, z upscale factor

   or

   'root upscale factor - AS'
   8.0 1.0 8.0       ; x, y, z upscale factor
```

33. No extra keyword is required.

34. Add 'components decay half time' followed by half time in time unit second or add 'components decay rate constant' followed by rate constant s^-1 into data block . T_half = 0.693/rate_constant. An example is shown below.

```
   'components'
   1                    ;number of components (nc-1)
   'hto'

   'components decay half time'
   387892800       ;12.3 years
```

Codes Update

Remarks

1. Set all PI value to 3.141592653589793d0. The precision of PI in the previous versions are not the same in some functions.

2. Replace tab with space in the source code for alignment.

3. Disable water saturation update when initial condition is read from external file and flow velocity is fixed to zero.

4. Update the tiny time parameters to 1.0d-12. The tiny time ranges from 1.0d-20 to 1.0d-5 in different modules in previous versions.

5. Add courant number calculation in unstructured grid code and enable courant number calculation only when courant target number is less than 1.0d30.

6. Increase the maximum number of components that can be used in a mineral, secondary specie, gas and etc from 10 to 100.

7. Update domain decomposition option for hexahedral mesh. Interpolate option is turned off for hexahedral mesh to have better decomposition.

8. Update in 'reading material properties/initial condition ... from vtk file' when structured grid is used. These features are initially implemented for unstructured grid and it is now also available in structured grid version.

9. Update default setting for corss diffusion term. This term is now included by default for multi-point flux approximation.

10. Update the sign in por_stress (porosity change due to mechanical loading stress). Use positive value when mechanical loading is released and negative value when mechanical loading is added. No effect in simulation results except that the output of por_stress is reversed.

11. Update the default gradient reconstruction method of unstructured grid code from 'least square' to 'least square second order' for better convergence rate.

12. Update parameters of 'root respiration and exudation of aqueous phase' and 'minimum aqueous concentration for root respiration' to zone dependent parameters.

13. Remove output of porosity change due to mechanical loading. Porosity value is already availabe and the change can be calculated based on the porosity value.

14. Update 'read initial aqueous concentration from vtk file' or 'read initial aqueous concentration from hdf5 file'. When a component does not exist in the file, assign a small value 1.0d-20 to the component.

15. Update in reading data from external vtk file. The old version calls rewind before reading a variable which significantly slows down the performance for ordered output (list of components have the same order in the input file and external data file).

16. Update in tpfrtlc. The update in total aqueous concentration and secondary components cause small difference when transient output is used. The update is disabled inside the transient output except for batch reactions.

17. Update in respiration output (*.resp). Replace the last half columns with accumulative total respiration over the elapsed time.

18. Update keyword 'calculate loading efficiency coefficient' to 'calculate loading factor'.

19. Important update for keyword 'root water uptake'. Water pressure optimal value is added under 'root water uptake'.

```
Old:
'root water uptake'
160.0          ; water pressure wilting point to further reduc the uptake
1.0            ; water pressure field capacity
0.0            ;root density (optional when density field is specified)

New:
'root water uptake'
160.0          ; water pressure wilting point to further reduc the uptake
1.0            ; water pressure field capacity
10.0           ; water pressure optimal value (used in Feddes and VG)
0.0            ; root density (optional when density field is specified)
```

20. Important update for keyword 'soil hydraulic function parameters'. A 6th optional parameter 'air dried aqueous pressure head' is required to turn on evaporation function. If not provided, this parameter is zero thus no evaporation is used.

```
'soil hydraulic function parameters'
0.15                ; residual saturation
1.20                ; van genuchten - alpha
1.38                ; van genuchten - n
0.5                 ; expn
0.0                 ; air entry pressure
1600.0              ; psidry (air-dry pressure head)
```

21. Update format of 'read anisotropic correction angles from file', angles are provided in (alpha(Z-axis), beta(Y-axis), gamma(X-axis)) format.

22. Remove keyword for porosity and tortuosity correction for MCD diffusion, the keyword does not take any effect. Similar keywords for hMCD work.

23. Update information in the transient output file (*.gb*) with coordinates (x, y, z). This changes the head information of transient output.

24. Add support for latest PETSc-3.16. Please note the option '--download-hdf5-fortran-bindings' should be used in PETSc configuration starting this version.

Benchmarks Update **Remarks**

1. stripf_usg: use 'two-point flux approximation', or reduce ratio of horizontal dispersivity and transverse dispersivity. The previous version has a bug in gradient reconstruction.

2. d45_salt_dome_problem_d133_b3_usg: use 'least square second order' and 'multi-point flux approximation' method, change maxit_sia to 6 and iter_target to 3, and Courant to 1.0 for better convergence rate.

3. Other unstructured grid examples: use 'least square second order' as default gradient reconstruction method.

4. Delete demonstration examples 'aniso-diff-disp-cond-tensor-V2.0.26.783', replace with 'anisotropy-diff-mcd-V2.1' where analytical solution for anisotropic diffusion is available. 5. Add benchmark anisotropy-diff-mcd-V2.2. 6. Added benchmark root-uptake-leaf-recycle-V2.2.	
Documentations Update	**Remarks**
1. Version report. 2. User manual (TBD)	The user manual is to be updated, after the revision of official user manual is completed
Bugs Fixed	**Remarks**
1. Fix bug in 'zero flow velocity' when specific storage coefficient is zero. All the matrix entries and rhs are zero. No solver is required for flow. 2. Fix bug in boundary condition output in gen file. The output information of 'volume boundary type hydraulic head' should be replaced by 'volume boundary type pressure head' for non-density-dependent flow and 'volume boundary type pressure' for density-dependent flow. 3. Fix bug in 'output of transient data for interface fluxes' in MPI parallel version. Only the local owned interface fluxes are considered. The limitation is, if the interface pair ivol-jvol belongs to different domain, it is not considered. 4. Fix bug in 'zero flow velocity'. No saturation update should be called when reading initial condition from file. 5. Fix bug in global variable num_cells_per_layer for MPI parallel version. This variable is not broadcasted to all processors. It only affects layered mesh with ice loading/unloading when cells are selected by layer id. 6. Fix bug in porosity output in the gsv file. The calculation of porosity output is only correct when all the solids are considered in the mineral input section, including those "inert" minerals. Now all the porosity output uses pornew variables, which is calculated based on the user specified porosity value in the input file and mineral precipitation/dissolution if update porosity/permeability is turned on. 7. Fix bug in running ice sheet model for 2D unstructured version. Quadrilateral mesh is not considered in the previous version. 8. Fix bug in density dependent flow. Parameter tds_old and densold_pitzer are not set to the new value during Picard iteration. Only	

the variable densold = density is there.

9. Fix bug in dfluxvg function in jacrt. The gpivol_ivol, gpivol_jvol should be used instead of gpivol_ivol, gpivol_ivol pair. This affects convergence rate of gas advection cases.

10. Fix bug in reading legacy vtk format. Please note, for legacy vtk format, mixed cell types are NOT supported.

11. Fix bug in mineral mass balance output format. When number of parallel reaction pathways are considered, the tecplot header is not consistent with columns of data output.

12. Fix bug in binary output for transient data (.gb* files). Some of the unnecessary written is activated in the previous version.

13. Fix bug in harmonic mean for reactive transport problem using unstructured grid method.

14. Fix bug in binary format for transient output when MPI parallel version is used as binary format does not allow rewind operation.

15. Fix bug in surface mesh related output (e.g., icesheet) for MPI parallel version.

16. Fix bug in stencil width setting for MPI parallel version. The old version skip defaul solver parameter setting when root.dat file exists, causing zero stencil width problem which cannot be supported in domain decomposition.

17. Fix bug in MPI parallel version realted to the noncollective memory monitor.

18. Fix bug in 'limited output of spatial data'. Logical Unit of 6 (screen output) is turned off when 'limited output of spatial data' is used, resulting screen output to fort.6.

19. Fix bug in gradient reconstruction for component concentration. This part affects reactive transport with anisotropic material properties. It also reduces numerical diffusion when multipoint flux approximation is used.

20. Fix bug in mass output through specified boundaries for MPI parallel version. The array jatmsb is non-collective.

21. Fix bug in diffusion coefficient calculation in MCD code. The variables mdiff_ic_cvol and mdiff_ix_cvol should not be accumulated overtime. This bug does not affect steady state flow since the function is called only once. But for unsteady state flow, the function is called at each timestep and the diffusion coefficient after first timestep is not correct.

22. Fix bug in passing increment to gradient reconstruction for aqueous concentration. 23. Fix bug in bubble model for MPI version. Gas_pair setting is put inside output block which is not correct. 24. Fix bug in high order gradient reconstruction for MPI parallel version. The memory monitor inside building control volume interface is not collective. 25. Fix bug in ary-dry aqueous pressure parameter. The index of this parameter that has been used in root related part is not correctly set in the code. 26. Fix bug in NaN (not a number) in charge balance error output for multicomponent diffusion. 27. Fix bug in 'read soil hydraulic function parameters from file'. The previous version does not open this feature for general flow problem, but only for root uptake related code.	
Notes	**Remarks**
Summary of code verification (test running)	**Remarks**
Due to the extensive new features and updates to the code, as well as some format change in the output file (e.g., new columns added, header information change), the comparison needs to be checked manually. Comparison of the results for all examples obtained by the current and the previous MIN3P-HPC version (V2.1.26.768) shows the results are the consistent, but there are some differences in the numerical values. The postprocessing of these results shows no distinguishable difference except for those affected by the bug fixed and code udpates.	

Revision 786

Revision Number	786	**Version Number**	V2.1.26.786
Commit Date	2021-01-06	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel Compiler, GCC Compiler with C++11 or PGI compiler. 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32; WINDOWS; USG; CGAL; 2.3 Preprocessor definition for parallel version: RELEASE_X64| RELEASE_WIN32; WINDOWS; PETSC; MPI; PARDISO; OPENMP; USG; CGAL; LIS; SCHEDULE_STATIC; 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library (Optional) 3.1 PETSc: PETSc V3.4 to V3.14 3.2 LIS: LIS V1.7.36 to V2.0.12 3.3 CGAL: CGAL V4.7			Add support for PGI compiler; Add support for latest PETSc 3.14 version;
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable			

library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
Add new features to unstructured grid code; add supports for external file IO; add supports for freeze/thaw/ponding cycling; add full tensor diffusion coefficients/dispersion coefficients/hydraulic conductivity; fix bugs and update the code.	
New Features	**Remarks**
1. Add autorevision.sh script for Linux/Unix OS. This is an alternative way to get revision information which is similar to subwcrev on Windows OS. To use it, go to directory src/version and run the script. The revision information in Version.F90 file will be updated. 2. Add support of overlayed boundary flux calculation and output. The previous version does not allow overlay in the specified boundary nodes that the latter selection replaces the ealier selection if duplicated is found. 3. Add chunk size and compression to HDF5 result data files. Previous chunk size is 1 which is low efficient and results into much larger data size. Chunk size has been set to min(nvols,1024). By default, compression is not used. User need to specify compression level to use this feature. To use compressed output, HDF5 should be configured with related library such as szip. 4. Add support to read soil hydraulic function parameters from external file. With this feature, each node (control volume) can have its own soil hydraulic function parameters. 5. Add incomplete factorization level control in PETSc solver configuration. 6. Add output of soil hydraulic function parameters to the external file _o.hyc (_o.hyc.vtk, _o.hyc.h5). 7. Add memory consumption report to the log file. Please note the memory consumption estimation is based on the allocatable arrays used in the code. It does not include memory used by external library. 8. Add RCM (reverse Cuthill-McKee) mesh ordering for unstructured mesh. The node ordering of the original mesh, generated by Gmsh or other software, may be not reordered. RCM mesh ordering can be applied to reduce the bandwidth and improve convergence rate in linear solver. Please note, the sequential version using Watsov sovler uses RCM ordering for matrix by default. This method is adopted here for the RCM mesh ordering.	

9. Add the output of detailed memory usage based on allocatable arrays.

10. Add support of partitioner types for domain decomposition such as 'chaco', 'ptscotch', 'simple'. By default, PETSc uses parmetis partitioner for unstructured mesh. Please note the specified external library should be configured in PETSc installation.

11. Add support for PGI compiler. Note: pgfortran does not support mixed characters and numbers by implicit string with quotes, user need to explicitly specify 'use space delimiter into Data Block 2: geochemical system. PGI compiler also added a carriage return (CR) after string trim that CR needs to be ignored.

12. Add support for density change over temperature for variably saturated flow.

13. Add derivative for storage change in density dependent flow (for both fully saturated and variably saturated flow) due to compressibility.

Old code: term1=densnew*satnew*(pornew-porold)/delt1. The old code uses derivative of porosity directly, which is not correct when specific storage coefficient is used (e.g., pressure change without porosity change).

New code: term1 = densnew*satnew*spstor*(pwnew-pwold)/delt1. The new code applies chain rule to the derivative of porosity, d(porosity)/dt = d(porosity)/d(pressure)*d(pressure)/dt. When 'specific storage coefficient' is non-zero, the old code generates much larger mass balance error than the new code and it encounters significant convergence problem. When ice sheet model is used, porosity and pressure are decoupled, in which porosity is modified due to ice loading/unloading but pressure is not modified accordingly (porosity change without pressure change). In this case, the old code based on porosity derivative is used.

14. Add central weighting for density dependent flow under unsaturated condition. The previous version always uses upstream weighting even if central weighting is specified in the Data Block 6: 'control parameters - variably saturated flow'.

15. Add support to disable density change (drho_dc = 0, drho_dt = 0) when run steady state flow in transient mode.

16. Add support to read intial condition of density dependent flow from results obtained by variably saturated flow. The code will automatically convert the pressure head to pressure. It supports keywords 'read initial condition from vtk file' and 'read initial condition from hdf5 file'.

17. Add support to use either pressure or water head as Newton increment and Newton convergence tolerance for density dependent flow. The input unit was treated as pressure for the density dependent flow in the previous versions.

18. Add support to decouple density change over temperature in storage

term calculation. By default, the coupled formula is used.

19. Add support to decouple flow and heat equations (df(Pa)/dT = 0, df(T)/dPa = 0).

20. Add support to use underrelaxation factor when the maximum update is in the specified range. This feature is available in flow and heat transport.

21. Add support to read data range in node/cell selection by material id.

22. Add support to allow maximum density update at first timestep. In some circumstances, the density update may be large at the first timestep when the initial condition is not good. To enable this feature, the density update at the first timestep is not checked.

23. Add support for transient temperature boundary conditions.

24. Add support to disable frozen boundary connection. When this feature is activated, connection is disabled between boundary and frozen node.

25. Add support to disable flux when boundary is ponding. If the ponding depth is bigger than the maximum allowed ponding depth, the influx is stopped. This feature can be applied to influx boundary condition (second or point type boundary condition).

26. Add support to provide relative permeability for frozen region by specifying temperature-coefficient pairs.

27. Add/update support of internal boundary conditions for transport equations. This is an updated version of 'tracer' boundary condition. Please note this new feature is compatible to the 'tracer' boundary condition but new keywords should be used.

28. Add support of transient 'seepage' boundary boundary.

29. Add new keyword 'face-based flux approximation'. This keyword is better known. It is the same as 'use separated interface centers' and 'use control volume half face center'. By default, edge-based flux approximation is used.

30. Add support of dynamic seepage and second type mixed boundary condition for flow. When seepage node is unsaturated, it uses second boundary condition (e.g., zero flux or precipitation influx), when seepage node is saturated, it uses atmospheric boundary condition (e.g., zero pressure).

31. Add support of legacy vtk input file. For legacy format, each line of coordinate contains three points. The vtk file saved by Paraview is legacy format.

32. Add root respiration (solute uptake/release) to the code.

33. Add 'maximum number of seepage face iterations' for seepage face boundary conditions.

34. Add 'interpolate' option for unstructured mesh partition in PETSc. For the prism/wedge mesh, without 'interpolate', the parition quality is not good and there are much more ghost cells/nodes compared to the partition with 'interpolate'. For other cell types, there is no big difference in partition with or without interpolate.

35. Add 3D ice sheet implementation.

36. Add 'reverse enthalpy change' for the database in which enthalpy is reversed.

37. Add 'compatible bubble gas database' where temperature dependent solubility coefficients are placed after gram formula weight so that when gas bubble is deactivated, no modification in the database is required.

38. Add linear interpolation of transient boundary conditions. This function can be used to make the transient boundary condition update smoothly when necessary.

39. Add control of external file I/O flush control. Some computer system or compiler may not support external flush. Use this keyword to disable external flush and use intrinsic flush only.

40. Add control of 'maximum iterations of temperature correction'. For the previous version, there is no limitation on the number of temperature corrections, which may cause infinite loop.

41. Add mineral rate correction over saturation change(ported and modified from Xueying Yi's WRP simulation).

42. Add 'reduction ratio of frozen diffusion coefficients' update in the code.

43. Add method to specify number of node layers for unstructured grid mesh. The previous version calculates number of node layers (for layered mesh only) based on the material id properties. This requires the material id for each physical layer is identical and consecutive. For mesh with layer refinement, the same material id can be assigned to different physical refined layer, thus the previous method cannot work. User needs to calculate number of node layers by coordinate or specified manually.

44. Add node/cell selection using id range constraint and layer thickness constraint.

45. Add general 'gradient' type of initial condition setting for heat transport.

46. Add termination keyword when transient steady state or maximum simulation time is reached.

47. Add 'linear sorption decay rate constant' for adsorbed components. To be further checked. This feature is not verified yet.

48. Add fully decoupled flow and heat equation. There are three coupling methods for flow and heat equation: fully coupled, semi-coupled and decoupled. If fully coupled, both df(Pa)/dT and df(T)/dPa are considered; if loosely coupled, neither df(Pa)/dT nor df(T)/dPa is considered but heat and flow equation is solved at the same time; and if decoupled, neither df(Pa)/dT nor df(T)/dPa is considered and heat and flow equation is solved separately.

49. Add zone dependent user specified mechanical parameters. The previous version only allows 1 zone to define mechanical parameters. If the mechanical parameters are different in the simulation domain, the parameters should be read from file. There is no such limitation in the current version.

50. Add the calculation of related parameters based on the mechanical parameters. These calculated parameters include porosity, skempton coefficient, specific storage coefficient and loading efficiency coefficient.

51. Add transient dispersivity for reactive transport.

52. Add node/cell selection using depth constraint.

53. Add zone dependent component concentration input for ice sheet boundary conditions using unstructured mesh.

54. Add transient temperature output when heat transport or temperature field is specified. The previous version only outputs temperature when evaporation is specified. No extra keyword is needed.

55. Add function to adjust hydraulic conductivity/permeability over depth. Here we use a similar permeability relationship over depth reported by (Stefano Delfino Normani, 2009, Paleoevolution of Pore Fluids in Glaciated Geologic Settings, PhD thesis presented to the University of Waterloo, pp 71-72). The function used is K(depth) = K(surface) * 10**(a*exp(-b*depth)-c) where a, b, and c are user specified parameters.

56. Add timestep control before and after update boundary conditions. In some cases, very small timestep may lead to convergence problem.

57. Add 'combined output of aqueous and sorbed concentrations'. When this feature is used, the concentration of aqueous concentration and sorbed species are combined and written into prefix.gstgsb file.

58. Add feature to transform ice sheet thickness based on log function. The gradient based or curve fitting based ice thickness is linear

distribution. These parameters are used to modifiy the ice thickness based on log function h'=a*(log(h*b+1))^c where h is the original ice thickness, a, b, and c are the specified parameters.

59. Add full diffusion tensor.

60. Add full dispersion tensor.

61. Add full hydraulic conductivity/permeability tensor.

62. Add zero flow velocity feature that disable the flow in the simulation domain.

63. Add 'water freezing relative conductivity function' to calculate relative conductivity. This function is similar to 'water freezing relative conductivity curve' but has less input parameters. The formula used is rel = 0.5*(relcond_max+relcond_min)*(sin(pi/(tempmax-tempmin)*(T-0.5*(tempmax+tempmin)))+1.0) where tempmin, tempmax, relcond_min and relcond_max are the four input parameters.

64. Add 'anisotropic correction' for anisotropic material properties such as effective diffusion coefficient (by tortuosity correction factor), dispersivity, hydraulic conductivity, thermal conductivity and thermal dispersivity. When anisotropic correction is used, the material properties are transformed from principle terms to full tensor. The input parameters accept the Tait–Bryan angles (alpha, beta, gamma in rotation) to define the angle from the xyz axis in the current coordinate to the principal xyz axis of material properties in the current coordinate. The commonly used strike direction and dip angle can be easily converted to Tait-Bryan angles by setting the strike direction to alpha and dip angle to gamma. All the rotation should obey right-hand-thumb rule.

Usage of New Features

Remarks

1. Run autorevision.sh in directory src/version.

2. Add 'allow overlap in specified boundary' into data block 'output control', e.g.,

```
  'output of mass through specified boundary'
  6                        ;total number of specified boundary segments

  'allow overlap in specified boundary'

  'number and name of zone'   ;keyword for zone
  1                            ;number of specified boundary segments
  'Flux-top_full'              ;name of specified boundary segments
  'extent of zone: boundary nodes only'   ;keyword for zone selection
(other keywords are available)
  0.0 1.0  0.0 1.0  1.0 1.0    ;coordinates for zone selection
  'end of zone'                ;end of current zone
  'number and name of zone'   ;keyword for zone
  2                           ;number of specified boundary segments
  'Flux-top_zone_1'           ;name of specified boundary segments
  'extent of zone: boundary nodes only'   ;keyword for zone selection
(other keywords are available)
  0.0 0.40  0.0 1.0  1.0 1.0  ;coordinates for zone selection
  'end of zone'               ;end of current zone
  ... ...
```

3. Add following keyword into data block 'output control' to set compression level. 0 means no compression; 1 means best compression speed but least compression; 9 means best compression ratio but slowest speed; 2 through 8 means compression improves but speed degrades.

```
  'compression level of result data'
  6                            ;compression level
```

4. Add 'read soil hydraulic function parameters from file' or 'read soil hydraulic function parameters from vtk file' into Data Block 10: physical parameters - variably saturated flow.
If 'read soil hydraulic function parameters from file' is used, please provide the data in tecplot format with file name prefix.shfp. The variables from left to right are:
x y z swr spalpha spbeta expn aentry h1dry(if root uptake is included) and sgr_imbi(if trap bubble is included)
If 'read soil hydraulic function parameters from vtk file' is used, please provide data in vtk format with file name prefix.shfp.vtk. The variable tpye used in vtk file should be scalar and variable name is 'swr', 'spalpha', 'spbeta', 'expn', 'aentry', 'h1dry'(if root uptake is included) and 'sgr_imbi'(if trap bubble is included).

5. Add "petsc: factorization level in flow" and "petsc: factorization level in reactive transport" to solver configuration file. Alternatively, user can use command argument "-pcfactor_level_flow" and "-pcfactor_level_react", respectively.

6. No keyword is needed. When soil hydraulic function parameters are read from external file, it will export to _o.hyc (_o.hyc.vtk, _o.hyc.h5) in corresponding format.

7. No keyword is needed. Code should be compiled with preprocessing "MEMORY_MONITOR". The memory consumption will be written back to the log file after simulation is done.

8. By default, natural mesh ordering is used for the unstructured mesh. To use RCM node ordering, please add 'rcm mesh ordering' into data block 'spatial discretization'.

9. Add keyword 'enable output of detailed memory usage' into data block 'global control parameters'.

10. Add following keywords into solver configuration file solver.cfg, or use command options '-partitioner_type name'.

```
!> PETSc partitioner type
!> PETSc supports partioning using different libraries.
!> Parmetis is used as the default partioner type if not specified.
!> The following partitioner type can be used.
!> Please make sure the specified external library is installed
!>   PETSCPARTITIONERCHACO    "chaco"
!>   PETSCPARTITIONERPARMETIS "parmetis"
!>   PETSCPARTITIONERPTSCOTCH "ptscotch"
!>   PETSCPARTITIONERSHELL    "shell"
!>   PETSCPARTITIONERSIMPLE   "simple"
!>   PETSCPARTITIONERGATHER   "gather"
PETSC: PARTITIONER TYPE
parmetis
```

11. When PGI compiler is used, the database needs to be in fixed form in which the lengh of string in the database should be exactly the same as defined in the variables. Alternatively, user can use free format that supports space delimiter. Please add 'use space delimiter into Data Block 2: geochemical system.

12. Add the following keywords into Data Block 'control parameters - variably saturated flow' followed by a coefficient. By default, zero value is used.

```
  'specific density change over temperature'
  0.0d0    !unit, [M L-3 C-1]
```

13. Add keyword 'use old storage derivative' into Data Block: 'control parameters - variably saturated flow' to use the old formula for storage term derivative, which is buggy. When this keyword is not used, the new formula for storage derivative is used. Convergence and mass balance error are significantly improved with the new formula when 'specific storage coefficient' is non-zero.

When ice sheet model is used, porosity and pressure is decoupled and the old formula is used automatically. In other circumstance, please add keyword 'decouple porosity pressure change' into Data Block: 'control parameters - variably saturated flow' to use the old formula for porosity change over time.

14. Use same keywords as general variably saturated flow.

15. Add keyword 'disable density change in transient mode' into data block 'control parameters - variably saturated flow'.

16. No extra keyword is required. Use the same keyword 'read initial condition from vtk file' and 'read initial condition from hdf5 file'.

17. Add keyword 'input units for newton increment and tolerance' into data block 'control parameters - variably saturated flow'. By default, the

input units is pressure (Pa) for density dependent flow.

```
   'input units for newton increment and tolerance'
   'pressure'               ;use pressure (Pa) as input units
or
   'freshwater head'     ;use water head (m water) as input units
```

18. Add keyword 'decouple density temperature change' into data block 'control parameters - variably saturated flow'.

19. Add keyword 'decouple flow and heat equations' into data block 'control parameters - energy balance'.

20. Add keyword 'user specified underrelaxation range' into data block 'control parameters - variably saturated flow' and 'control parameters - energy balance'.

```
   'user specified underrelaxation range'
   0.0d0  1.0d4
```

21. Add keyword 'read data range' below specified node/cell selection keywords. An example is shown below:

```
   'extent of zone: boundary nodes by material id'
   'read data range'
    2          ;number of data ranges
    1,4        ;first range, material id 1 to 4
    8,11       ;second range, material id 8 to 11
```

22. Add keyword 'allow maximum density update at first timestep' into data block 'control parameters - variably saturated flow'.

23. Add keyword 'transient boundary conditions' or 'transient temperature boundary conditions' into data block 'boundary conditions - energy balance'. If the boundary types and zones remain the same, only the values need to be updated. User can use keyword 'transient boundary conditions: values only' or 'transient temperature boundary conditions: values only' to get better performance.

24. Add keyword 'disable frozen boundary connection' into data block 'control parameters - variably saturated flow'.

```
   'disable frozen boundary connection'
```

25. Add keyword 'apply pressure head tolerance of ponding boundary' into data block 'control parameters - variably saturated flow'.

```
   'apply pressure head tolerance of ponding boundary'
```

And then add 'pressure head tolerance of ponding boundary' followed by tolerance value into data block 'boundary conditions - variably saturated flow'.

```
   'pressure head tolerance of ponding boundary'
   1.0d-3                    ;meter water
```

26. Add keyword 'water freezing relative conductivity points' followed by number of pairs and temperature-coefficient pairs into data block 'control

parameters - variably saturated flow'.

```
    'water freezing relative conductivity points'
    21              ; number of pairs
    -1.0  1.0d-6
    -0.9  1.0d-5
    -0.8  1.0d-4
    -0.7  2.0d-4
    -0.6  3.0d-4
    -0.5  4.0d-4
    -0.4  5.0d-4
    -0.3  6.0d-4
    -0.2  7.0d-4
    -0.1  8.0d-4
    -0.0  9.0d-4
    0.1   1.0d-3
    0.2   2.0d-3
    0.3   3.0d-3
    0.4   4.0d-3
    0.5   5.0d-3
    0.6   5.0d-2
    0.7   5.0d-1
    0.8   8.0d-1
    0.9   9.0d-1
    1.0   1.0d0
```

27. Specify keyword 'idle' for the flow boundary condition and 'first' and/or 'third' boundary condition for the transport boundary condition. Please note this new feature is compatible to the 'tracer' boundary condition but new keywords ('idle' for flow and 'third' for transport) should be used.

28. No extra keyword is needed. This feature is used only if transient boundary condition is turned on. The transient boundary condition should be specified through *.bcvs or *.bcheat.

29. Add 'face-based flux approximation' into data block 'spatial discretization'

30. Change the boundary type to 'seepage-second', followed by the influx rate, e.g., 'seepage-second' 1.0e-8 m/s.

31. Add keyword 'use legacy vtk format' into Data Block 3: spatial discretization.

32. Add keyword 'root respiration and exudation of aqueous phase' followed by rates into Data Block 17: plant transpiration and passive/rejective uptake. The input is the respiration rate mol/(root density * cvol)/second for each component.

```
    'root respiration and exudation of aqueous phase'
    -1.0d-10        !h+1, respiration rate mol/(root density * cvol)/second
    -5.0d-11        !co3-2
```

```
  0.0              !ca+2
  5.0d-11          !o2(aq)
  0.0              !al+3
  0.0              !h4sio4
  0.0              !na+1
  0.0              !CaOxalate_aq
  0.0              !cl-1
  0.0              !fe+2
  0.0              !fe+3
  0.0              !oxalate-2B
  -1.0d-7           !ch3cooh(aq)
```

33. Add keyword 'maximum number of seepage face iterations' followed by number into data block 'control parameters - variably saturated flow'. The default value is 15.

```
'maximum number of seepage face iterations'
10            ;maximum number of iterations
```

34. No keyword is required to use this feature. This is a default option in the new version. User can use the following section in the parallel solver configuration file (Block: Domain decomposition setting) to turn on or off this option.

```
!> Mesh interpolate
!> For domain decomposition, the shared cells/vertices are automatically
!> created, but shared faces and edges is not created by default.
!> To create faces and edges in the mesh, user need to set interpolate
!> to true. The code actually does not need to use the shared faces and
!> edges from PETSc. However, the current PETSc version (<PETSc
3.13.x)
!> does not generate best partition for prism mesh without interpolate.
!> For the either cell types, there is no big difference with or without
!> interpolation. By default, interpolation is set to true.
PETSC: CREATE FACES AND EDGES IN THE MESH
false [true]
```

35. Definition of ice boundary condition. Each boundary type has five parameters. The first column is time, followed by 5 parameters for the boundary type in each zone.

For 'gradient' ice boundary type, the columns represent time, thick_grad, thick_slope_grad, rdummy, rdummy, rdummy.

For 'curve-fitting' ice boundary type, the columns represent time, a, b, c, d, e where a is sea level [m], b is sea level change per temperature change [m/T], c is ice thickness change per sea level change [m/m], d is the threshold of sea level change to form ice sheet for the specified elevation [m], and e is the threshold of elevation to form ice sheet. r100 means temperature drop 1C per 100 meter elevation gain [m/T].

The 'formula' used is: ice thickness = max(min(a-d,0)*c-(z-e)/r100*b*c,0)

For 'first' ice boundary type, the columns represent time, ice thickness, rdummy, rdummy, rdummy, rdummy. An example based on curve-fitting ice boundary condition is shown below:

!c Sea level and temperature change in the past 120ka, reference Gordan Stuhne and W. R. Peltier, 2016,
!c Sensitivity Analyses of Surface Boundary Conditions During Long-Term Climate Change, 2016, pp3, figure 1.
!c Suppose there are boundary conditions defined for two zones, the example input is shown below. The first a to e columns are the input parameters for the boundary condition in the first zone and the second a to e are the input parameters for the boundary condition in the second zone.

```
!c Year a      b     c     d     e     a      b     c     d     e
0       4.4159  4.4  -10  -5  1000  4.4159  4.4  -10  -5  1000^M
645     5.2369  4.4  -10  -5  1000  5.2369  4.4  -10  -5  1000^M
1160    4.3886  4.4  -10  -5  1000  4.3886  4.4  -10  -5  1000^M
```

36. The current MIN3P-THCm version uses reversed enthalpy change while the old MIN3P ebullition and isotope code do not. To use the current code to simulate ebullition and isotope problems but using the same database in the old format, please specify the following keyword into data block 'geochemical system'.

```
'reverse enthalpy change'
```

37. Add keyword 'compatible bubble gas database' into data block 'geochemical system'.

38. Add keyword 'linear interpolation of boundary conditions' after keyword 'transient boundary conditions: values only'. This keyword only works when keyword 'transient [flow, temperature, ice sheet] boundary conditions: values only' is specified.

39. Add keyword 'disable external flush' into Data block 'output control'.

40. Add keyword 'maximum iterations of temperature correction' into Data block 'control parameters - energy balance' followed by number of maximum iterations. If this keyword is not specified, the maximum number is 10.

```
   'maximum iterations of temperature correction'
   20
```

41. Add curve for rate correction over saturation into data block 'control parameters - reactive transport'.

```
   'saturation correction'
   1.273d-5  27.6  3.4085    35     ;parameter a, b, c and d for rate correction curve satcorr_coef=d+(a-d)/(1.0+((3.0-saturation)/c)^b)
   0.7       0.95  1.0d-10   1.0       ;minimum saturation, maximum saturation, minimum coefficient, maximum coefficient
```

Add keyword 'rate correction over saturation' into mineral database.

```
   'o2source-xy'
   'surface'
```

```
    32.0000    1000.000
    1  'o2(aq)'   1.000
    'irreversible dissolution'   2.898  1.844
    'log rate constant' -8.0d0
    'rate correction over saturation'
    !'no freezing rate update'          !c this is old keyword, the rate is
actually update when this keyword is specified,  it is deprecated due to
misleading.
```

42. Add keyword 'reduction ratio of frozen diffusion coefficients' into data block 'physical parameters - reactive transport'. By default, the diffusion coefficient is reduced by 6 orders of magnitude in water and 1 order of magnitude in air. This part can be further optimized by using temperature dependent diffusion coefficient (need code implementation).

```
    'diffusion coefficients'
    2.38d-9              ;free diffusion coefficient in water, unit m^2/s
    2.07d-5              ;free diffusion coefficient in air

    'reduction ratio of frozen diffusion coefficients'
    1.0d-6              ;reduction ratio of free diffusion coefficient in water
    1.0d-1              ;reduction ratio of free diffusion coefficient in air
```

43. Add the following keyword to data block 'spatial discretization'

```
    'method to calculate number of node layers'
    !'material id'                 ;only if material id is consecutive and identical
for each layer
    'coordinate'                   ;only if coordinate is layered in one direction,
either x or y or z.
    !'user specified'   123    ;user specified value
```

44. Add 'extent of zone: layer thickness constraint' and 'extent of zone: id range constraint' into zone selection.

```
   'extent of zone: id range constraint'
   100  200     ;node/cell id range constraint is between 100 and 200

   'extent of zone: layer thickness constraint'
   0.1  1.0       ;layer thickness constraint is between 0.1 and 1.0.
```

45. Same as flow, use 'gradient', 'gradient-top' and/or 'gradient-bottom' as the keyword.

46. Add keyword 'stop after transient steady state' and/or 'maximum time of transient steady state' into data block 'control parameters - variably saturated flow'. The simulation time here is defined in the time control block.

```
    'stop after transient steady state'

    'maximum time of transient steady state'
    1.0d2                    ;exit after 1.0d2 [time unit]
```

47. Add 'linear sorption decay rate constant' followed by decay rate constant [s^-1] into data block 'geochemical system'.

```
   'linear sorption decay rate constant'
   2.1d-6
```

48. Add following keyword into data block 'global control parameters'. By default, the flow and heat equations are fully coupled.

```
   'type of coupling for flow and heat equations'
   'coupled' or 'semi-coupled' or 'decoupled'       ;type of coupling
```

49. Add 'mechanical parameters' followed by seven parameters into data block 'physical parameters - porous medium'. The previous version only allows 1 zone to define mechanical parameters. If the mechanical parameters are different in the simulation domain, the parameters should be read from file. There is no such limitation in the current version. An example is shown below.

```
   'number and name of zone'
   1
   'all layers'
   0.10                       ;porosity
   'mechanical parameters'
   7.7d9        !Young's modulus E in the unit of [Pa]
   1.0d300      !Mineral grain modulus Ks for the rock formation [Pa]
   3.0d9        !Pore fluid modulus Kf for the rock formation [Pa]
   0.18         !Poisson's ratio v in the unit of [-]
   1.0d-8       !coefficient of vertical compressibility beta [Pa^-1]
   2.65d3       !density of rock in the unit of [kg/m^3]
   1.0d3        !density of fluid in the unit of [kg/m^3]
   'extent of zone'
   400000.0 600000.0  4000000.0 6000000.0  -2000.0 1000.0
```

50. Add the following keyword and parameters into data block 'physical parameters - porous medium'.

```
   'number and name of zone'
   1
   'all layers'
   0.10                       ;porosity
   'mechanical parameters'
   7.7d9        !Young's modulus E in the unit of [Pa]
   1.0d300      !Mineral grain modulus Ks for the rock formation [Pa]
   3.0d9        !Pore fluid modulus Kf for the rock formation [Pa]
   0.18         !Poisson's ratio v in the unit of [-]
   1.0d-8       !coefficient of vertical compressibility beta [Pa^-1]
   2.65d3       !density of rock in the unit of [kg/m^3]
   1.0d3        !density of fluid in the unit of [kg/m^3]

   'calculate porosity'
   !recalculate porosity change over depth based on Bahr et al 2001
model.
```

```
   !The following material properties (skempton coefficient, specific
storage coefficient and loading efficiency coefficient) can be calculated
based on hydromechanical parameters.
   !Reference: J.F. Sykes, S.D. Normani, and Y. Yin, Hydrogeologic
Modelling, NWMO DGR-TR-2011-16.

   'calculate skempton coefficient'
   'calculate specific storage coefficient'
   'calculate loading efficiency coefficient'

   'extent of zone'
   400000.0 600000.0  4000000.0 6000000.0  -2000.0 1000.0
```

51. Add the following keywords into data block 'physical parameters - reactive transport' and put the time dependent dispersivity into file prefix.disprt.

```
   'transient dispersivity'
   'linear interpolation of dispersivity'
   example of prefix.disprt file
   !transient dispersivity for different zones
   !format:
   !time disx disy disz disx disy disz disx disy disz
   !       |   zone 1    |    zone 2      |    zone 3    |
   0.0     1.0  0.0  0.0  2.0  0.0  0.0  3.0  0.0  0.0
   60.0    1.5  0.0  0.0  2.5  0.0  0.0  3.5  0.0  0.0
   120.0 2.0  0.0  0.0  3.0  0.0  0.0  4.0  0.0  0.0
   240.0 1.0  0.0  0.0  2.0  0.0  0.0  3.0  0.0  0.0
```

52. Add the following keywords into zone selection.

```
   'extent of zone: layer depth constraint'
   0.0d0  40.0d0         ;minimum depth allowed, maximum depth allowed,
only the nodes/cells with the specified depth range are selected.
```

53. Add the following keyword and parameter for each zone into data block 'boundary conditions - ice sheet loading/unloading'.

```
   ! Data Block 20: boundary conditions of ice sheet loading/unloading
   'boundary conditions - ice sheet loading/unloading'
   1
   ! ------------------------------------------------------------------------
   'number and name of zone'
   1
   'lake-ice'

   'concentration input'
   0.2915E-02      'charge'   ;cl
   0.4000E-04      'free'        ;br
   0.5220E-03      'free'        ;na
   0.6400E-04      'free'        ;k
              6.0       'ph'          ;h
   0.1722E-04      'free'        ;hco3
   0.6240E-03     'free'       ;ca
   0.8230E-03     'free'       ;mg
   0.2600E-03     'free'       ;so4
          1.0d0    'po2'       ;o2(aq)
   'boundary type'
   'curve fitting ice sheet model'
   5.0          ;[m] sea level change
   4.4          ;[m/T] sea level change per temperature change
   -10.0        ;[m/m] ice thickness change per sea level change
   -5.0         ;[m], threshold of sea level change to form ice sheet
   1000.0       ;[m], threshold of elevation to form ice sheet

   'extent of zone: boundary nodes by material id'
   'read data'
   1
   39

   'extent of zone: coordinates constraint'
   -1.0E10 1.0E10 -1.0E10 1.0E10 -1.0E10 176.0

   'end of zone'

   'done'
```

54. No extra keyword is required.

55. Add following keyword and parameters for each zone into data block 'physical parameters - variably saturated flow'

```
   'hydraulic conductivity change over depth'
   4.5  0.002469  4.5    ;parameter a, b and c
```

56. Add the following keywords into data block 'time step control - global system'.
To use minimum timestep after updating boundary condition, please use the keyword below.

```
   'minimum timestep after updating boundary condition'
```

To adjust timestep before updating boundary condition, please use the keyword below. The timestep will be reduced by half so the next timestep is close to the previous one.

```
   'relaxed timestep before updating boundary condition'
```

57. Add keyword 'combined output of aqueous and sorbed concentrations' into data block 'output control'.

58. Add keyword 'transformation parameters of ice thickness' into data block 'control parameters - ice sheet loading/unloading' followed by three parameters.

```
   'transformation parameters of ice thickness'
   1000.0  5.0  0.75
```

59. To use difusion coefficient tensor, please add 'principle diffusion coefficients' or 'diffusion coefficients tensor' into data block 'physical parameters - reactive transport'. For principle diffusion coefficients, three values should be provided. For full diffusion coefficients tensor, nine values (3x3) should be provided. An example is shown below.

```
   !'principle diffusion coefficients'
   !1.0d-9  0.0d0    5.0d-9
   !tensor after rotation of principle axis for 30 degree using right-hand-thumb rule.
   !  Dxx' = 1.0d-9, Dzz' = 5.0d-9
   !  θ = 30°
   ! |Dxx Dxz|   |cosθ  -sinθ| |Dxx'  0   | |cosθ   sinθ|
   ! |           | = |                | |           | |               |
   ! |Dxz Dzz|   |sinθ  cosθ | |0     Dzz'| |-sinθ  cosθ|

   'diffusion coefficients tensor'
   2.0d-9      0.0d0    1.732d-9
   0.0d0       0.0d0    0.0d0
   1.732d-9   0.0d0    4.0d-9
```

60. To use full dispersion tensor, please add keyword 'use full dispersion tensor' into data block 'physical parameters - reactive transport'. The BF form dispersion tensor proposed by Burnett and Frind (1987) is used.

61. To use full hydraulic conductivity tensor, please add keyword 'use full hydraulic conductivity tensor' into data block 'control parameters - variably saturated flow' and then add hydraulic conductivity tensor into data block 'physical parameters - variably saturated flow'.

```
   'control parameters - variably saturated flow'
   ...
   'use full hydraulic conductivity tensor'
   ...

   'physical parameters - variably saturated flow'
   ...
   !'hydraulic conductivity in x-direction'
   !1.16d-4                    ;K_xx

   !'hydraulic conductivity in z-direction'
   !1.16d-5                    ;K_zz

   !'hydraulic conductivity tensor'
   !1.16d-4  0.0d0    0.0d0
   !0.0d0    0.0d0    0.0d0
   !0.0d0    0.0d0    1.16d-5

   !after 30 degree rotation
   'hydraulic conductivity tensor'
   8.99d-5  0.0d0    4.52d-5
   0.0d0    0.0d0    0.0d0
   4.52d-5  0.0d0    3.77d-5
```

62. Add keyword 'zero flow velocity' into data block 'initial condition - variably saturated flow'.

63. Add 'water freezing relative conductivity function' into data block 'control parameters - variably saturated flow'. The formula used is rel = 0.5*(relcond_max+relcond_min)*(sin(pi/(tempmax-tempmin)*(T-0.5*(tempmax+tempmin)))+1.0) where tempmin, tempmax, relcond_min and relcond_max are the four input parameters as shown below.

```
    'water freezing relative conductivity function'
    -1.0     1.0    1.0d-6   1.0    ;minimum teperature, maximum temperature, minimum relcond, maximum relcond
```

64. To use this feature, add 'anisotropic correction' into data block 'physical parameters - porous medium' and then add 'anisotropic correction angles' into each zone. The parameters for 'anisotropic correction angles' are Tait–Bryan angles, namely alpha, beta and gamma, corresponding to yaw (about z axis), pitch (about y axis) and roll (about x axis), respectively. The anisotropic correction angles can also be read from file. The keyword is 'read anisotropic correction angles from file' or 'read anisotropic correction angles from vtk file', depending on the file format. In geologic maps, strike direction and dip angle are commonly used. The strike direction (about z axis) is alpha and dip angle is gamma (about x axis), respectively. All the rotation should obey right-hand-thumb rule.

To apply anisotropic correction for hydraulic conductivity or permeability, please add 'anisotropic hydraulic conductivity correction' or 'anisotropic permeability correction' into data block 'physical parameters - variably saturated flow'.

To apply anisotropic correction for dispersivity, please add 'anisotropic dispersivity correction' into data block 'physical parameters - reactive transport'.

To apply anisotropic correction for effective diffusion coefficient, please add 'anisotropic tortuosity correction' into data block 'physical parameters - reactive transport' and then add correction factor for longitudinal, transverse horizontal, transverse vertical for each zone. The keywords are 'longitudinal anisotropic tortuosity factor', 'transverse horizontal anisotropic tortuosity factor' and 'transverse vertical anisotropic tortuosity factor', respectively.

To apply anisotropic correction for thermal conductivity, please add 'anisotropic thermal conductivity correction' into data block 'physical parameters - energy balance'.

To apply anisotropic correction for thermal dispersivity, please add 'anisotropic dispersivity correction' into data block 'physical parameters - energy balance'.

Example input:

```
    !Block 'physical parameters - porous medium'

    'anisotropic correction'

    !Read correction angles from file
```

'read anisotropic correction angles from file' ;prefix.aca file, tecplot format, columns x y z alpha beta gamma, unit of angle ° 'read anisotropic correction angles from vtk file' ;vtk format, keyword 'vector aca double', unit of angle ° !Read correction angles from each zone 'anisotropic correction angles' 30.0 0.0 45.0 ;alpha (about z axis, yaw), beta (about y axis, pitch), gamma (about x axis, roll) !Block 'physical parameters - variably saturated flow' 'anisotropic hydraulic conductivity correction' 'anisotropic permeability correction' !Block 'physical parameters - reactive transport' 'anisotropic tortuosity correction' 'anisotropic dispersivity correction' !Read anisotropic tortuosity correction factor from each zone 'longitudinal anisotropic tortuosity factor' 1.0d0 'transverse horizontal anisotropic tortuosity factor' 1.0d-1 'transverse vertical anisotropic tortuosity factor' 1.0d-1 !Block 'physical parameters - energy balance' 'anisotropic thermal conductivity correction' 'anisotropic dispersivity correction'	
Codes Update	**Remarks**
1. Change the description "mass flux across top boundary [mol/d]" in the mgc file to "mass flux across boundary [mol/d]" as this is confusing when there is gas flux across other boundaries. The boundary can be side boundary or even bottom boundary. The mgc file is system mass in/out/change. 2. Update pre-processing definition MPI to PETSC since the code use MPI from PETSC library. 3. Update HDF5 result data compression setting. Data compression enabled for PETSc-V3.11 dev and PETSc-V3.12+ versions.	

4. Remove reduntant code for root_uptake functions. Variables such as BINev and BINT are used during the simulation. Need further check on these function as some varialbes are defined but not used or properly set.

5. Update convergence information for MPI parallel version. Exclude ghost nodes from total number of unconverged nodes.

6. Update label value setting based on 32-bit operation for node_to_layer_node.

7. Optimize memory consumption in unstructured grid code. This causes very small difference in benchmarks_usg/react/clem3d_usg, patchf_usg and pile_hexa. The difference is not distinguishable in plots.

8. Optimize unstructured grid data reading by master processor only.

9. Optimize memory usage for variables that are used only during initialization (e.g., nodes_gbl, cells_gbl, node_idx_g2lg, cell_idx_g2lg)

10. Optimize transient boundary condition update without reading boundary type and zone area from file if the boundary type and zone type are not changed. To use this feature, use keyword 'transient boundary conditions: values only' instead of 'transient boundary conditions'.

11. Update configuration in makefile.

12. Update code to allow underrelaxation for full saturated flow.

13. Update the estimated maximum number of non-zero entries. This may change the behaviour of solver and cause very small numerical difference, e.g., benchmarks_new_add/uraninite-reoxidation-V1.0.56.

14. Update solver parameters (e.g., dinc_vs, tol_vs) for density dependent flow. These solver parameters are in the unit of hydraulic head and should be converted to fluid pressure for density dependent flow.

15. Update aqueous concentration and wate content symbol from s_w and theta_w to s_a and theta_a. Update gas concentration and gas content from s_a and theta_a to s_g and theta_g. It was contradictory that s_a represents air saturation (gas saturation) or aqueous saturation in different input/output. Now all the water phase is represented by _a, gas phase by _g, and napl phase by _n.

16. Update name of variable p_w in the output. For variably saturated flow of the previous version, p_w represents pressure head (meter water). For density dependent flow of the previous version, p_w represents pressure head (meter water) with flow verification but represents pressure (Pa) without flow verification. From this version, p_w represents pressure (Pa, for dependent flow output only), h_w represent pressure head (meter water, for variably saturated flow only), ph_w

represents pressure head (meter water, for density dependent flow only), fh_w represents freshwater hydraulic head (meter water, for density dependent flow only).

17. Update 'decouple flow and heat equations' to 'loosely couple flow and heat equations'. The previous keyword 'decouple flow and heat equations' still works.

18. Update of 'free-drainage' boundary condition with user specified gradient of total hydraulic head. The previous version always uses 1.0.

19. Set tkel to system default temperature (298K) when geochemistry is not activated.This is initialized for freezing/thawing model without geochemistry system.

20. For the bubble flow simulation, MIN3P latest version allows to place temperature dependent solubility coefficients after gram formula weight so that when gas bubble is deactivated, no modification in the database is required. For example, the last three parameters -55.66578, 82.0262 and 22.5929 (shown below) are temperature dependent solubility coefficients. When dhc is between 9.975e2 .and. 9.995e2, this part is used for bubble model which is kind of hardwired. Should be used together with keyword 'compatible bubble gas database'.

```
  ar(g)    998.0000   2.8534     39.9480  -55.66578   82.0262   22.5929
  1   ar(aq)       1.000
```

The original format in the bubble code is shown below, which is not compatible with gases database when bubble flow is deactivated.

```
  ar(g)   998.0000   2.8534              39.9480
  1   ar(aq)       1.000
  -55.66578   82.0262   22.5929
```

21. Parallel section for OpenMP is deprecated in the code as there is no benefit in the current implementation.

22. Keyword for decouple flow and heat transport is updated and moved to the data block 'global control parameters'

23. Add PETSc convergence reason check. When linear solver is not converged, user can know the reason why it fails to converge. Look at PETSc user manusl for detail.

24. Update the maximum norm of linear solver to 1.0d100 for parallel solver. This large value will deactive norm check since parallel solver (e.g., PETSc) may use estimated norm, not the actual norm.

25. Remove restriction of number of zones when read intial condition from file. The previous code requires the number of zones to one only in some variables (e.g., aqueous concentration).

26. Update component concentration input for ice boundary. The

component concentration input has now been moved into the block of 'boundary conditions - ice sheet loading/unloading'.

27. Update keyword for 'porosity threshold', both minimum and maximum value should be provided.

```
   'porosity threshold'
   0.000001d0  0.999999d0   ;minimum  maximum
```

28. Update condition for permeability factor update using Kozeny-Carman equation. NaN value is calculated when porosity is zero or one. The old version only check the pornew value without checking porold value which is actually a bug in the code.

29. Add temperature output when temperature field is specified. The previous version outputs temperature only when heat transport is activated.

30. Replace tiny_time in time comparison by minimum timestep. The tiny_time varies (e.g., 1.0d-10 and 1.0d-20) in different subroutines, making the time not consistent in next timestep and boundary update. This update causes small difference in the timestepping and the results might not be identical to the previous version from numerical point, but the meaning is the same.

31. Exclude the sorbed fraction of master species in the combined output of total aqueous and sorbed.

32. Update 'bubble iteration settings' parameters. The following parameters are required.

```
   'bubble iteration settings'
   200         ;maximum number of inner bubble iterations
   200         ;maximum number of outer bubble iterations
   20          ;anticipated number of bubble iterations
   1.0d-7      ;convergence tolerance, inner loop
   1.0d-7      ;convergence tolerance, outer loop
   1.0e-4      ;maximum update in gas saturation(for relaxation scheme)
   1.0e-3      ;maximum degassing rate(delSa/day)
```

33. Update in computing rate of degassing, revert back to the code used in isotope model. The code was modified to the code used in ebullition model, but it seems not work well for William bubble simulation case.

34. Update in HDF5 IO. The surface parts have their own communication id, which is separated from the global communication id. The default chunk size is also updated.

35. Update restart read/write for unstructured grid version. The code can now restart with different number of processors. This feature is only available for unstructured grid version with binary format. User should keep the original domain file and rename it to prefix.domain.h5. The code reads node mapping from this file so that different number of processors can be supported.

Benchmarks Update	**Remarks**
1. aniso-diff-disp-cond-tensor-V2.0.26.783 2. basin_3d_coarse_drift_20m_vs_ice1k_po21_po1_e1-V2.0.26.783 3. d1_verification_vertical_stress_3d_usg-V2.0.26.783 4. d43_density_dep_heat_solute_decoupled-V2.0.26.783 5. shlomot-slope-seepage-second-V2.0.26.783 6. tranddvs_rt_freeze_thaw_pond_heat_rel-V2.0.26.783 7. trans-internal-bd-first-V2.0.26.783	
Documentations Update	**Remarks**
1. Version report. 2. User manual (TBD)	The user manual is to be updated, after the revision of official user manual is completed
Bugs Fixed	**Remarks**
1. Fix bug in the size of gasdecayratetot array. Array size should be n instead of ng. 2. Fix bug in newton convergence check in density dependent flow under saturated condition. The newton iteration is not checked when keyword 'iterative solver' is not specified (default). In this case, the flow is not converged in newton iteration and results are still returned. 3. Fix bug in mesh generation when MPI parallel version is used. Only the master mpi rank needs to be called since mesh generation module is sequential. 4. Fix bug in pre-processing syntax in sptldisc.F90. 5. Fix bug in RCM mesh ordering. Some of the variables are not released after RCM mesh ordering. This bug is in the previous version only. 6. Fix bug in running structured grid case using MIN3P-HPC USG version. The bug is activated when external initial condition/material properties are provided and MPI parallelization is used. For the previous version, please remove USG preprocessing definition in the makefile if the compiled code is used for the structured grid cases using MPI parallelization. 7. Fix bug in reading zone information from file when 'transient boundary conditions' is used. The external file of the 'zone information' is not closed/reopened or rewinded properly in the previous versions. 8. Fix temerature correction for batch reaction. 9. Fix temperature correction for general reactive transport simulation	

when 'define temperature' is used. Temperature correction in the previous versions only work when 'define temperature field' or 'heat transport' is used.
This causes differences in the following benchmarks due to temperature change. The difference is insignificant in the plot.

gas-bubble-V1.0.303\Amos and Mayer 2006\Williams,
mine-waste-attenuation-V2.0.0.726

It also causes very small difference in the following benchmarks due to the round-off error or calling of temperature correction. The difference is irrecognizable in the plot.
isotope-V1.0.303\Jamieson-Hanes et al 2012\Cr_columns - 3
isotope-V1.0.303\Wanner et al 2014\Problem 1
transient-boundary-V1.0.303\co2-seq-gacc
transient-boundary-V1.0.303\co2-seq-sine

10. Fix bug in reading 'concentration input' in 'ice sheet loading/unloading' block. When redox couples are specified, reorder components is required in reading concentration and type value. The previous version uses natural ordering which does not allow redox couples. At the same time, format of 'concentration input' in 'ice sheet loading/unloading' block is updated (number of components line is not required). Please use the same input format as 'boundary conditions - reactive transport'. User can double check the concentration input in the gen file. An example is shown below:

```
'concentration input'
 0.2915E-02     'charge'  ;cl
 0.4000E-04     'free'    ;br
 0.5220E-03     'free'    ;na
 0.6400E-04     'free'    ;k
        6.0     'ph'      ;h
 0.1722E-04     'free'    ;hco3
 0.6240E-03     'free'    ;ca
 0.8230E-03     'free'    ;mg
 0.2600E-03     'free'    ;so4
      0.21d0    'po2'     ;o2(aq)
```

11. Fix bug in time step increment in ddfsflow. 'call tstepfs' is missing in this function that delt_vs does not increase accordingly. This does not affect the benchmarks since they do not use delt_vs when reactive tranpsort is turned on. But this affects the steady state flow when it runs in transient mode.

12. Fix bug in hydraulic conductivity output in gsv file when update porosity and permeability are activated. The previous version only works when hydraulic conductivity is assigned from zone selection. The value of hydraulic conductivities are zero in the output file (*.gsv) when hydraulic conductivity is read from external file. Note: this part was added for test purpose only in 1D basin simulation. It's not required to output hydraulic

conductivity in gsv file for general purpose. 13. Fix bug in file unit variable igsa_first in opnpgfls.F90. igsa_first is used in both aqueous phase and gas phase. The previous versions do not initialize this variable when gas is not included, causing potential conflicts in file unit. 14. Fix bug in storage term calculation for both saturated and variably saturated flow. Convergence and mass balance errors are significantly improved with the new format when 'specific storage coefficient' is non-zero. 15. Fix bug in Pitzer density model when heat transport is not used. In the previous versions, Pitzer density model only takes effect when heat transport is activated. The new version allows Pitzer density model to be used without heat transport. This bug fix affects all the simulation cases with Pitzer density model but without heat transport, e.g., benchmarks_new_add/sulfur-V1.0.431/sulfur. 16. Fix bug in updtbcdd.F90. 'uvsnew(ivol) = bcondvs(ibvs)' should be used instead of 'uvsnew(ivol) = rwork(ibz,1)' since rwork does not always equal to fluid pressure. 17. Fix bug in running steady state flow in transient mode for density dependent flow. Density dependent flow should be restored after steady state (with or without density effect) is reached. 18. Fix bug in reading unsaturated initial condition for density dependent flow. The reading format has an extra rdummy variable in the old version, which is not right in reading initial condition values. 19. Fix bug in changing upstream point in variably saturated density dependent flow. The upstream point (if used) should remain unchanged due to the pressure (head) inrement during jacobi assembly. Otherwise, it may cause convergence problem if the pressure increment reverses the upstream point. 20. Fix bug in calculating mininum and maximum value of local mesh and global mesh for USG code. This part (e.g., zlmaxgbl) is missing in sptldisc and affects the initial condition setting of energy balance where zlmaxgbl is used. 21. Fix bug in density dependent and heat transport functions. The variables densold2, saold2 and tds_old2 (previous old tilmestep) are deprecated. There is convergence problem when previous old timestep is considered. Need further check here. 22. Fix bug in mass balance calculation for density dependent flow. The arguments passing to function stordd(fs) are not corrected that might causs large mass balance error.	

23. Fix local bug in mass balace calculation (revision 750-754).

24. Fix bug in transient boundary condition for density dependent flow under unsaturated condition, variable uvsnew applies the boundary condition without unit conversion from hydraulic head to pressure head in the previous version. This bug only affects unsaturated density dependent flow.

25. Fix bug in freeze/thaw model for density dependent flow and heat transport scenario. The previous version does not apply the reduced hydraulic conductivity to the freezing zone.

26. Fix bug in readint_new.F90 for isotope model. The bug was first fixed in Richard Amos's branch ramos_isotope_bubble r592 on June 26, 2018.

27. Fix bug in 'apply pressure head tolerance of ponding boundary' for USG code. This feature is not included for the USG code in the previous version and causes crash. The bug is in the local branch only. The trunk version is not affected.

28. Fix bug in cell_idx_g2lg in solver_ddmethod.F90. This variable is not allocated when cell based hydraulic conductivity/permeability is not used, causing crash in the previous version. The bug is in the local branch only. The trunk version is not affected.

29. Fix bug in 'transient boundary/temperature conditions: values only' for the MPI parallel version. The subdomain without boundary condition applied does not work properly.

30. Fix bug in seepage boundary condition when heat transport is used. In the boundary condition setting of heat transport, it turns off seepage boundary condition for no reason. This is, I guess, caused by copy/paste when heat transport code was developed.

31. Fix bug in solute mass balance calculation for density dependent flow. The boundary flux is not correctly calculated when density dependent flow is used.

32. Fix bug in 'appending results' when last record is earlier than restart time. This is possible when 'timestep between transient output' is large. Extra backspace is needed in handling the file.

33. Fix bug in reporting total number of Newton iteration to log file when heat transport is used.

34. Fix bug in seepage iteration of density dependent flow for MPI parallel version.

35. Fix bug in gas bubble model. The 'bubble iteration settings' in the specified input files are not consistent with the implemented code. For the new version, please use the following 'bubble iteration settings'.

```
  'bubble iteration settings'
  200         ;maximum number of inner bubble iterations
  200         ;maximum number of outer bubble iterations
  20          ;anticipated number of bubble iterations
  1.0d-7      ;convergence tolerance, inner loop
  1.0d-7      ;convergence tolerance, outer loop
  1.0d-3      ;maximum update in gas saturation (for relaxation scheme)
  1.0d-3      ;maximum degassing rate (delSa/day)
```

36. Fix bug for 'temperature correction for solution' for heat transport. This correction may cause infinite loop.

37. Fix bug in confining pressure calculation for density dependent flow. The passing parameter is pressure in unit Pa, not pressure head in unit m.

38. Fix bug in transient boundary condition update. When time step is reduced just after the boundary condition is updated, the flow and heat boundary variables (e.g., tempnew, uvsnew) are reset to the value of old time step, making the boundary condition inconsistent with the updated boundary condition.
Below is how the bug is triggered:
 If you have following boundary condition update in the bcvs file (same for bcheat file).

```
 10    0.00E+00     0.00E+00     2.95E+02
 15    0.00E+00     6.85E-09     2.95E+02
 20    0.00E+00     9.09E-09     2.95E+02
 25    0.00E+00     1.13E-08     2.95E+02
```

 And you have following simulation run time
 timestep k-1 at time 14.0
 timestep k   at time 15.0
 timestep k+1 at time 15.2 -> call updtbcvs -> uvsnew is set to new bcondvs
 1) if timestep k+1 fails, then timestep k+1 uses reduced time at 15.1, uvsnew is set to old uvsold from previous timestep, the bug is triggered because updtbcvs will not be called until the next update time after 20.0.
 2) If timestep k+1 converges, then go to next timestep k+2 at 15.5. If timestep k+2 fails, the bug is not triggered since the old uvsold already has the value with updated boundary condition.

39. Fix bug in the number of Newton of iterations inside seepage iteration (ddvsflow). The Newton iteration number should be reset to zero for each seepage iteration, otherwise, there may be infinite loop.

40. Fix bug in porz value setting for batch reaction problem in initpppm.F90 file. This array is not allocated for batch reaction but exist in the batch reaction code. Actually this array is not required for batch reaction, so it is removed.

41. Fix bug in variable coepsil when SIT model is implemented. This

variable is not allocated for batch reaction.

42. Fix bug in reporting the total number of newton iterations and linear solver iterations when energy balance is used.

43. Fix bug in setting increment in jacddfs.F90. The following part is buggy. When uvsnew is very small or very large, this gives very small or large increment, causing zero diagonal entry in the matrix or numerical unstable. The keyword 'iterative solver' is disabled to turn off iterative_solver_flow because there is no iterative solver implemented for this feature. This keyword did nothing except changing the increment, as shown below.

```
    !c old code
    !if (iterative_solver_flow.and.uvsnew(ivol)/=r0) then
    !  dinc_vs_loc = uvsnew(ivol)*dinc_vs
    !else
    !  dinc_vs_loc = dinc_vs
    !end if

    !c new code
    dinc_vs_loc = dinc_vs
```

44. Fix bug in calculating ice thickness based on curve fitting model when restart file is used. This bug is local when the ice thickness model for unstructured grid model is added.

45. Fix bug in atmospheric boundary condition check. The previous version requires all the reactive transport boundary nodes are set in the flow boundary condition when atmospheric boundary condition is used. This is not always necessary.

46. Fix bug in the evaporation model, including data output (.evap) and compute_sensible function. The previous version output evaporation data only if transient output is specified. When there is no transient output, it only creates the file and writes the head information without detail evaporation data. It should be noted that some evaporation parameters used in this function are not initialized or control volume dependent. Use this with caution. The correct output should be used for specified boundary nodes only.

47. Fix bug in the evaporation calculation for the OpenMP parallel version. Some variables in the jacbevap should be local instead of global.

48. Fix bug in component concentration setting for ice sheet boundary condition using unstructured mesh. This bug is local.

49. Fix bug in flux_calc.F90. Some functions in this file changes the aqueous concentrations that should not be modified. It should be noted that adding new output cannot change the output of other results. This bug affects all simulations with keyword 'output fluxes' in the input file.

50. Fix bug in restart_r.F90 when initial start time is negative and update boundary condition is used. Replace the time label from zero to initial start time. This bug causes error in reading boundary condition. 51. Fix bug in boundary condition interpolation for transient boundary conditions. 52. Fix bug in 'calculate specific storage coefficient' based on mechanical properties. For density dependent flow, there is internal conversion of storage coefficient since density dependent flow uses pressure [Pa] as the unit of unknowns and non-density dependent flow uses head [m] as the unit of unknowns. This bug is local. 53. Fix bug in storage coefficient output when density dependent flow is used. The unit should be converted back to m^-1 as the internal unit is Pa^-1 when density dependent flow is used. 54. Fix bug in reading transient boundary condition file. This bug is local when linear interpolation of boundary condition is implemented. 55. Fix bug in velocity output in density dependent flow for unstructured grid version. Use darcy velocity instead of control volume interface flux.	
Notes	**Remarks**
1. The head of mgc file of the benchmarking case is changed due to updates 1. 2. Results of density dependent flow is changed accordingly due to fixed bug 2. There is no obvious change in the plot. The newton convergence tolerance is changed to 1.0e-4 for case benchmarking_nwmo_report\nwmo_verification_examples\d1_verification_vertical_stress. 3. The estimated Picard iteration number needs to be adjusted for some density dependent flow due to fixed bug 14 and 15. For example, the Picard iteration target (estimated picard iteration number) of basin simulation should be increased from 5 to 8 so that larger timestep can be used. 4. The density dependent related code has been tested for both saturated and variably saturated simulation cases and both work without convergence problem for the tested cases. The density dependent flow with heat transport under variably saturated condition may need further check. The heat transport model was coded for fully saturated flow and some functions may be missing in the variably saturated flow. 5. Due to the code updated and bugs fixed, parameters of some density dependent flow cases and heat transport cases (e.g., d32_elder_problem, d43_density_dep_heat_solute) needs to be adjusted to have better performance. For example, increase the estimated and	

maximum picard iteration for d32_elder_problem case; reduce the minimum timestep to 1.0d-8 for d43_density_dep_heat_solute case. 6. Static link option is now available in the makefile. Please note on some systems, some basic shared libraries may still required.	
Summary of code verification (test running)	**Remarks**
Due to the extensive new features and updates to the code, as well as some format change in the output file (e.g., new columns added), the comparison needs to be checked manually. Comparison of the results for all examples obtained by the current and the previous MIN3P-HPC version (V2.0.0.726) shows the results are the consistent, but there are some differences in the numerical values. The postprocessing of these results shows no distinguishable difference except for those affected by the bug fixed and code udpates. Please note some variables in the output (e.g., s_w, s_a) have been updated to make sure these variables are consistent in different modules. User needs to update the postprocessing layout accordingly if the variables used in the layout are changed.	The differences of the following benchmarks are bigger due to bug fixes: 1. henry-hilleke 2. salt_dome 3. verif_evap 4. co2seq-water-freezing-V1.0.377 5. sul2d (sulfur-V1.0.431\2D-sal)

Revision 726

Revision Number	726	**Version Number**	V2.0.0.726
Commit Date	2019-08-07	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel Compiler or GCC Compiler with C++11 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32; WINDOWS; USG; CGAL; 2.3 Preprocessor definition for parallel version: RELEASE_X64| RELEASE_WIN32; WINDOWS; PETSC; MPI; PARDISO; OPENMP; USG; CGAL; LIS; SCHEDULE_STATIC; 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library (Optional) 3.1 PETSc: PETSc V3.4 to V3.11 3.2 LIS: LIS V1.7.36 to V2.0.12 3.3 CGAL: CGAL V4.7			PETSc V3.9+ versionis required for unstructured grid parallel version. CGAL is required if node/cell selection by polyhedral in 3D is used. Configuration of LIS solver is required to be compatible with MIN3P configuration, e.g., LIS OpenMP version for MIN3P OpenMP version.
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable			Update this library accordingly if new

library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	compiler is used.
Summary of this version	**Remarks**
Add new features of parallel I/O related to unstructured grid code, fix bugs and update the code. New naming convention is used.	
New Features	**Remarks**
1. Add global to natural ordering by using label for parallel unstructured grid version. This feature supports reading spatial data from external files for MPI parallel version. The PETSc package comes with default labeling feature, but it is very low efficient when a large number of different labels are used. Here we improved this feature by using two step labeling method. First the 32-bit integer label is extracted to 32 bits before distribution and then the 32 bits are reset to a 32-bit integer after distribution during domain decomposition in MPI parallel version. By using this, we only need 32 labels to actually represent 4 billion labels that would significantly improve the performance. 2. Add feature of reading cell based hydraulic conductivity/permeability from external file for MPI parallel version using unstructured mesh. 3. Add feature of scaled mesh coordinates for spatial output. This is optional as user can use postprocessing software to do mesh scaling. This feature is only available for unstructured grid version. 4. Add feature of reading initial condition from hdf5 file. This feature is available in unstructured grid, MPI parallel version where binary format data output is used. The number of processors used to read external file can be different from the number of processors used to generate those external files. 5. Add feature of writing domain information to the xmf file for spatial output. 6. Add feature of PETSc V3.11.x. 7. Add feature of 'trace' type boundary condition. For flow problem, 'tracer' type boundary condition marks the flow through the control volume. It does not change affect the flow in the domain. For reactive tranpsort, 'tracer' type boundary condition is used as a 'third' type boundary condition. The total mass transported is calculated based on the specified tracer concentration and the flux through the control volume. 8. Add freature of running steady state flow simulation in transient mode. For some steady state flow, because of transient effect, the simulation may encounter convergence problem if using steady state flow directly. In this case, transient flow can be used until flow reaches steady state condition. 9. Add feature of writing mass through specified boundary, including water (*_i_b.mvc), aqueous phase (*_ic_i_b.mac) and gas phase (*_ig_i_b.mgc). Here i is the index of specified boundary, ic is the index of aqueous component and ig is the index of gas. The format of these three files are similar to *_o.mvc, *_ic.mac, *_ig.mgc except that the storage terms and other source sink terms are not included. Please look	

at gen file and fls file for detail. 10. Add feature of activity update alias. Previous code allows to use 'no update' instead of 'no_update' and 'double update' instead of 'double_update'. The new code add support '_',' ', and '-' as delimiters. Now the code supports the following activity update alias: 'no_update', 'no update', 'no-update'; 'double_update', 'double update', 'double-update'; 'time_lagged', 'time lagged', 'time-lagged'. 11. Add feature of automatic cell-node re-ordering if the ordering is not counterclockwise for 2D plane in 3D space. MIN3P-HPC requires cell-node ordering in counterclockwise that obeys RHT rules. 12. Add feature of 'update root length density from file'.	
Usage of New Features	**Remarks**
1. No extra keyword is required. 2. No extra keyword is required. 3. Add the following keyword and parameter to data block "spatial discretization". 'scale factor for mesh output' 1.0 1.0 10.0 ; scale factor for x, y and z axis. 4. This feature require external file prefix.domain.h5 and prefix.ivs.h5. These two files can be renamed from prefix_domain.h5 and prefix_*.gsp.h5. Add the following keyword to data block 'initial condition - variably saturated flow'. 'read initial condition from hdf5 file' 5. No extra keyword is required. 6. No extra keyword is required. 7. Add the following keyword to the block 'boundary conditions - variably saturated flow' 'boundary type' 'tracer' rdummy ;tracer type and the following keyword to the block 'boundary conditions - reactive transport'. 'boundary type' 'tracer' 8. To use this feature, please set flow to transient flow and add the following keywords to data block 'control parameters - variably saturated flow'. 'run steady state flow in transient mode' To determine whether the flow has reached steady state, user can specify the tolerance for steady state by using relative difference value and absolute difference value divided by time step. Please add the following keyword and parameters after keyword 'run steady state flow in transient mode'. By default, these tolerance is preset to 1.0d-6 for both relative difference and absolute difference. 'tolerance for steady state' 1.0d-6 ! relative difference divided by time step (day) for hydraulic head between current time step and previous time step 1.0d-6 ! absolute difference divided by time step (day) for hydraulic head between current time step and previous time step	The feature of 'update root length density from file' is part of further development and will be documented in the future release.

User can also control if the heat transport has reached steady state by adding following keyword to data block 'control parameters - energy balance'.

```
   'tolerance for steady state'
   1.0d-6    !  relative difference divided by time step (day) for temperature between current time step and previous time step
   1.0d-6    !  absolute difference divided by time step (day) for temperature between current time step and previous time step
```

9. Add the keyword 'output of mass through specified boundary' followed by number of specified boundary and boundary zone definition into 'Data Block 8: output control'. An example is shown below

```
   'output of mass through specified boundary'
   2                                       ;total number of specified boundary

   'number and name of zone'   ;keyword for zone
   1                                        ;number of specified boundary
   'boundary-1'                        ;name of specified boundary
   'extent of zone: boundary nodes only'      ;keyword for zone selection (other keywords are available)
   0.0 0.1  0.0 0.0  0.0 0.0          ;coordinates for zone selection
   'end of zone'                         ;end of current zone

   'number and name of zone'     ;keyword for zone
   2                                          ;number of specified boundary
   'boundary-2'                          ;name of specified boundary
   'extent of zone: boundary nodes only'      ;keyword for zone selection (other keywords are available)
   0.1 0.1  0.0 0.0  0.15 0.20        ;coordinates for zone selection
   'end of zone'                           ;end of current zone
```

10. As mentioned in new features.

11. This feature is turned on by default for 2D simulation. To disable this feature, please add following keyword into data block 'spatial discretization'.

```
   'disable automatic cell-node ordering'
```

12. Add keyword 'update root length density from file' followed by number of times and time list into data block 'plant transpiration and passive/rejective uptake'. An example is shown below:

```
   'update root length density from file'
   4                 ;number of times to update root density
   30               ;first time to update root density
   60               ;second time to update root density
   90               ;third time to update root density
   120             ;fourth time to update root density
```

Please note that data sharing of coordinates is used in the rld file to save memory in the file. An example of the data format is shown below:

title = "dataset transp-updrld" variables = "x", "y", "z", "rld" zone t = "Initial root length density" i = 30, j = 30, k = 1, f=point 0.000000E+00 0.000000E+00 0.000000E+00 1.000000E-01 3.448276E-02 0.000000E+00 0.000000E+00 1.000000E-01 6.896552E-02 0.000000E+00 0.000000E+00 1.000000E-01 ... zone t = "root length density on day 3" i = 30, j = 30, k = 1, f=point, varsharelist=([1,2,3]=1) 1.000000E-01 1.000000E-01 ... zone t = "root length density on day 6" i = 30, j = 30, k = 1, f=point, varsharelist=([1,2,3]=1) 1.000000E-01 1.000000E-01 ...	
Codes Update	**Remarks**
1. Update 'reverse cell-node ordering' for prism and hexahedral cell. The cell node should obey right hand thumb rule and the outward normal of the bottom face (first three nodes) should point to the cell center. 2. Add support of tab delimiter in vtk file. 3. Change the output format (e.g., e12.4, e16.8) to 1pe15.6e3 to avoid converting the data 1.23456E-123 to 1.23456-123 when written to the file. For fortran, it is the default feature to write very small value in the format 1.23456-123. This is mainly caused by the variables without initialization. 4. Update cell-node ordering of 2D coordinates in 3D. The cell-node ordering should be obey RHT (right hand thumb) rule. For example, for coordinate in XZ plane, the node-cell is in counterclockwise where the outward normal points to the Y direction. This update affect all xz coordinates since the previous code does not obey RHT rule for XZ plane. 5. Separated source/sink term from oxidation/reduction reactions and intra-aqueous kinetic reactions. The previous version output the two terms together as intra-aqueous reactions. The output of mac file is changed and is not backward compatible with previous version.	
Benchmarks Update	**Remarks**
1. benchmarks_new_add\ mine-waste-attenuation-V2.0.0.726 2. benchmarks_new_add\segment-output-V2.0.0.726 3. benchmarks_new_add\tranvs-steady-V2.0.0.726	Demonstration case for new feature added and bug fixed
Documentations Update	**Remarks**
1. MIN3P_THCm_UserManual 2. MIN3P_THCm_USG_UserManual	
Bugs Fixed	**Remarks**
1. Fix bug in storage term calculation (storvs.F90) for flow problem and retrieving physical parameters for scaling (rtrvpprm.F90). The variable	

por in storvs and rtrvpprm is a global scalar that it saves the porosity of the last zone/node only, causing the inconsistence when these functions are called. This bug affects all the problems with heterogeneous porosity distribution. 2. Fix bug in direct solver configuration (SuperLU, MUMPS) for PETSc release version 3.9+. 3. Fix bug in mass balance error output for root water uptake (e.g., .mve file). 4. Fix bug in mass balance output when 'append results' is used (e.g., .mac file). Some code is missing in the previous version. 5. Fix bug in periodic time step control. Small time step range (e.g., time range < global maximum time step) may be skipped 6. Fix bug in reading enthalpy change (dhc) for secondary aqueous sepcies, should use 'dhcx(ix) = -dhc' instead of 'dhcx(ix) = dhc'. 7. Fix bug in allocating iamdisoa array. Size of iamdisoa should be (naq*2+1). 8. Fix bug in 'reassign rates based on combined mineralogical parameters'. This part was ported from MIN3P isotope model but may cause significant convergence problem for other cases. This section is kind of hardwired and it is now modified. This section is used only if isotope pair is set, e.g., nip > 0. 9. Fix possible bug in setting tortuosity, porosity, saturation et al. in infcrt_g and infcrt_mcd. These variables should be added as each of the same iisav is looped more than once. For example, given mesh with 17x1x25 nodes, for such a 2D problem, ix=15, iy=1, iz=13, idim=1, ipair=1 ==> iisav = 1050 and ix=15, iy=1, iz=14, idim=1, ipair=2 ==> iisav = 1050. The old code set gsatij(iisav) = satav and gsatij(isymm(iisav)) = satav which always uses the second node pair while the new code uses the average value. Need further check here. 10. Fix bug in velocity output for DGM model. The variable dgflux_cd_dgm and neflux is not initialized and should be set to zero when there is no gas phase. 11. Fix bug in OpenMP parallel version when alias of activity update is used. The previous OpenMP code can only work with activity update of 'no_update', 'double_update' and 'time_lagged'. Now it works with 'no_update', 'no update', 'no-update', 'double_update', 'double update', 'double-update', 'time_lagged', 'time lagged' and 'time-lagged'. 12. Fix bug in 'mixed' boundary type condition when flow is out of domain. Should use free advective mass outflux. 13. Fix bug in updating reactive transport boundary conditions. It was called one time step earlier than it should be. The method to check time is also updated accordingly. Now both flow and reactive transport can update boundary conditions at the same time. 14. Fix bug in transient output for unstructured grid code using MPI parallel version. The outupt is duplicated in each processor.	
Notes	**Remarks**
1. From this version, MIN3P use ‘Major.Minor.Patch.Revision’ format as version number. For example, if the previous version is MIN3P-HPC-V1.0.0.724, if there is bug fix, the version number may be changed to MIN3P-HPC-V2.0.1.725 where the major version is 2, minor version is 0,	

patch version is 1 due to the bug fix and revision is 725 when commit to the trunk. The major version number is updated manually when major development is completed, e.g., change in software architecture. The minor version number is updated when backwards-compatible features are added and committed to the trunk of repository. The patch version number is updated manually when patch for the previous version is needed. The revision number is updated automatically when code is committed. A pre-release version MAY be denoted by appending a hyphen and a series of dot separated identifiers immediately following the revision number, e.g., MIN3P-HPC-V2.0.0.724-alpha.	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-HPC version (V690) shows no difference except those affected by the bug fixed and code udpates. Please note the output format has been changed. Users who use hardwired code to postprocessing the output need to modify the code if necessary.	

Revision 690

Revision Number	690	**Version Number**	V1.0.690
Commit Date	2019-02-03	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel Compiler or GCC Compiler with C++11 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32; WINDOWS; USG; CGAL; 2.3 Preprocessor definition for parallel version: RELEASE_X64| RELEASE_WIN32; WINDOWS; PETSC; MPI; PARDISO; OPENMP; USG; CGAL; LIS; SCHEDULE_STATIC; 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library (Optional) 3.1 PETSc: PETSc V3.4 to V3.10 3.2 LIS: LIS V1.7.36 to V2.0.12 3.3 CGAL: CGAL V4.7			PETSc V3.9+ versionis required for unstructured grid parallel version. CGAL is required if node/cell selection by polyhedral in 3D is used. Configuration of LIS solver is required to be compatible with MIN3P configuration, e.g., LIS OpenMP version for MIN3P OpenMP version.
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable			Update this library accordingly if new

library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	compiler is used.
Summary of this version	**Remarks**
Resume ransient output when restart, without overwriting the previous results; support different maximum time steps by using periodic maximum time step control; run structured grid simulation using unstructured method.	
New Features	**Remarks**
1. Add feature to resume output from breaking point when simulation is restarted, with support to dynamic check if the file exists and meet the data appending requirement. There are two restart files generated during MIN3P simulation, with breaking point t1 and t2. When simulation is restarted, user can choose one restart file to restart simulation. The results can be appended in two different methods: the normal results appending and the legacy results appending. The former one appends results immediately after the breaking point wheile the latter one appends results after it reaches the second breaking point. For example, assuming t1 < t2, if results are appended in normal mode and t1 is chosen as restart time, the code removes the old transient output data after t1 and appends the new results after t1; if results are appended in legacy mode, user can only choose the earlier restart file (t1) and the code keeps the old transient output before t2 and appends the new results after t2. When simulation is restarted, there is loss of significance that may cause blips in the transient output at the first time step (e.g., storage change over time). In this case, user can use legacy mode to avoid such a problem. 2. Add feature to run simulation based on structured grid using unstructured grid method and generate results of spatial output in vtk format. This feature works for 2D and 3D and does not support 1D. 3. Add feature to force the buffer data back to external file when restart point (time to create restart file) is reached. Writing to disk is slow so that the code actually write the data to a buffer in memory until the buffer is filled or the file is closed (default setting by most compilers). When simulation does not exit normally, the output data in the buffer is lost. In this circumstance, appending the transient data is impossible when simulation is restarted. In the updated version, every time when restart point is reached, the buffer is flushed to make sure all the output data is saved back to the external file in time. 4. Add periodic maximum time step control. This feature support to use different maximum time steps in the specified time ranges.	
Usage of New Features	**Remarks**
1. For the normal results appending, add the keyword 'append results' into data block 'global control parameters'. This keyword should be used	User manual of ArchiSimple code

together with keyword 'restart'. For the legacy mode results appending, please use the earlier restart file as restart condition and add the keyword 'append results in legacy mode' into data block 'global control parameters'. The input keywords are shown below. 'restart' !'append results' ;Append results after the first restart time 'append results in legacy mode' ;Append results after the second restart time 'backup frequency' 100 2. Add keywords 'use unstructured grid method' followed by 'structured spatial discretization' into data block 'spatial discretization', as shown below. 'spatial discretization' 'use unstructured grid method' 'structured spatial discretization' 1 ;number of discretization intervals in x 21 ;number of control volumes in x 0. 1.0 ;xmin,xmax 1 ;number of discretization intervals in y 1 ;number of control volumes in y 0. 1.0 ;ymin,ymax 1 ;number of discretization intervals in z 41 ;number of control volumes in z 0. 2.00 ;zmin,zmax 3. Default feature, no keyword is required. 4. Add keyword 'periodic maximum time step' into data block 'time step control - global system', followed by a list of time ranges and maximum time steps. An example is shown below 'periodic maximum time step' 3 ;number of time step list 0.1 0.2 0.001 ;time start, time end, maximum time step 0.5 0.6 0.0001 ;time start, time end, maximum time step 0.8 1.0 0.01 ;time start, time end, maximum time step	has been added. User manual of the new added features have been added.
Codes Update	**Remarks**
1. Use timestep saved in the restart file (delt_t0/r10) as initial timestep when simulation is restarted with results appending feature. In the previous versions, when restart file is used, it starts with the minimum timestep to run the simulation.	
Benchmarks Update	**Remarks**

None	
Documentations Update	**Remarks**
3. MIN3P_THCm_UserManual 4. MIN3P_THCm_USG_UserManual	
Bugs Fix	**Remarks**
1. Fix bug in the tecplot header in gbm and lbm file where vairable "charge bal [%]" is in a separated new line in the previous versions.	
Notes	**Remarks**
1. When 'append results' or 'append results in legacy mode' is used, user will find the following information in the prefix_o.fls file, indicating the results afterward are appended results to the previous simulations. The simulation can be restarted consecutively. +++ output of appended results +++ or +++ output of appended results in legacy mode +++ 2. If the results files are generated by the old code, the transient output data may be not completed (last record earlier than the last breaking point). In this case, the last record is before restarting time, an warning information is given in prefix_o.fls file, as shown below. warning - last data record in file prefix_x.abc is before restarted time 3. When 'append results' is used but the results of transient output in the previous simulation do not exist, the outputs of initial condition (time = 0.0) in the transient results are skipped. Other than that, the results output are the same as normal 'restart'. At the restart point, the time steps between output is ignored. Compared to the previous code, this will cause some difference in transient output if 'number of skipped output times' is larger than 1, due to extra output at restart point. 4. The accumulative mass balance and accumulative relative mass balance error are recalculated based on the breaking point (restarted time) saved in the transient output file. If the last record is before breaking point, last record is used instead of breaking point record and there will be numerical error in the acculative relative error. 5. There may be a bug in gas bubble model (case 506-4) that there are	

very small difference when different 'number of skipped output times' is used. This problem occurrs in all MIN3P versions with gas bubble model. Since the new code changes transient output frequency (e.g., force output at restart point), it causes some difference in the results because of the potential bug in gas bubble model. If 'number of skipped output times' is 1 (no skip), there is no difference between old code and new code. The difference is very small and can be ignored. Further check is needed to debug this problem in gas bubble model.	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm version (V675) shows no difference except the following: 1. the transient output file when 'number of skipped output times' is larger than 1, e.g., benchmarking_nwmo_report\nwmo_verification_examples_D5\d51_dedolomitization\min3p-nwmo\; 2. the gbm and lbm files in all benchmarks due to the bug in tecplot header. There is vey small difference in benchmark: benchmarks_new_add\gas-bubble-V1.0.303\Jeen et al 2012\506-4 caused by potential bug in gas bubble model as mentioned in Notes section.	

Revision 675

<table>
<tr><td><b>Revision Number</b></td><td>675</td><td><b>Version Number</b></td><td>V1.0.675</td></tr>
<tr><td><b>Commit Date</b></td><td>2019-01-22</td><td><b>Commit Author</b></td><td>Danyang Su</td></tr>
<tr><td><b>Software Grade</b></td><td></td><td>Signature</td><td>Date</td></tr>
<tr><td><b>Verified by</b></td><td>Mingliang Xie</td><td></td><td></td></tr>
<tr><td><b>Approved by</b></td><td>Uli Mayer</td><td></td><td></td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="3"><b>Compiler and configurations</b></td><td><b>Remarks</b></td></tr>
<tr><td colspan="3">1. Compiler: Intel Compiler or GCC Compiler with C++11
2. Configurations
2.1 Preprocess source file: Yes (/fpp)
2.2 Preprocessor definitions for sequential version:
RELEASE_X64|RELEASE_WIN32; WINDOWS; USG; CGAL;
2.3 Preprocessor definition for parallel version:
RELEASE_X64| RELEASE_WIN32; WINDOWS; PETSC; MPI; PARDISO; OPENMP; USG; CGAL; LIS; SCHEDULE_STATIC;
2.4 Preprocessor definition for resources:
RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P
2.5 Additional Options: /fp:strict
3. External Library (Optional)
3.1 PETSc: PETSc V3.4 to V3.10
3.2 LIS: LIS V1.7.36 to V2.0.12
3.3 CGAL: CGAL V4.7</td><td>PETSc V3.9+ versionis required for unstructured grid parallel version.
CGAL is required if node/cell selection by polyhedral in 3D is used.
Configuration of LIS solver is required to be compatible with MIN3P configuration, e.g., LIS OpenMP version for MIN3P OpenMP version.</td></tr>
<tr><td colspan="3"><b>System Requirements</b></td><td><b>Remarks</b></td></tr>
<tr><td colspan="3">Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable</td><td>Update this library accordingly if new</td></tr>
</table>

library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	compiler is used.
Summary of this version	**Remarks**
Port ArchiSimple code (root water uptake and evaporation) to MIN3P; add support for node velocity reconstruction using Perot’s method; update maximum number of columns and variables in the result file from 50 to 1000; update 'twothird-mix' surface area update; fix bug in gas tortuosity correction for default option 'same as aqueous' in revision 633; fix bug in cell center dual volume method for unstructured grid version.	ArchiSimple code is not documented.
New Features	**Remarks**
1. Merge ArchiSimple code (root water uptake and evaporation) to MIN3P. Thanks to Céline Blitz Frayret and Frederic Gerard for providing ArchiSimple code and initial code porting to MIN3P. 2. Add support for node velocity reconstruction using Perot's method.	
Usage of New Features	**Remarks**
1. Usage of ArchiSimple code is to be added, an example of ArchiSimple block is shown below. ! Data Block 1: global control parameters 'global control parameters' 'transpiration 2D couplage rld' .true. ;varsat_flow .false. ;steady_flow .false. ;fully_saturated .false. ;reactive_transport .true. ;plant_uptake !Data Block 17: plant transpiration and passive/rejective uptake ! --- ! 'plant transpiration and passive/rejective uptake' ! --- !NB :If none of these options is activated, the root length densities !NB considered are those specified in the 'root water uptake' section. !NB In the 'update-internal' option, ivol refers to the control volume ID. !'read root length density from file' ; 1D !'update - internal' ; 1D model: !rld(ivol)=exp(a+b*sqrt(z(ivol))+c*time+d*sqrt(z(ivol))*time) !-12.48 1.725 0.0017 0.0003961 !; a, b, c, d parameters of the model for 'update - internal' !'update - coupled - AS' ; 1D/2D (automatic detection) 'update - coupled - RT' ; 2D	Theoretical model, verification and user manual of ArchiSimple code are to be added.

```
!If seed < 0, use srand((unsigned) time(null))
!If seed >= 0, use srand((unsigned) seed)
!Default (without keyword 'random generator seed - *'), seed = -1
!
!'random generator seed - AS'
!123456

'random generator seed - RT'
654321

'transpiration global parameters'
2.0                                         ;rew_0
-2.0                                        ;p1
0.1                                         ;f_i
'correction method for water stress'
1

'A'

'root water uptake'
160.0                                       ;water pressure wilting point
1.0                                         ;water pressure field capacity
0.1                                         ;root length density

'passive solute uptake'
1.0
'end of zone'

'B'

'root water uptake'
160.0                                       ;water pressure wilting point
1.0                                         ;water pressure field capacity
1.0                                         ;root length density

'passive solute uptake'
1.0
'end of zone'

'done'
```

2. Add 'use perot velocity reconstruction' into Data Block 3: spatial discretization.

Codes Update	**Remarks**
1. Update the maximum number of columns and variables in the result file from 50 to 1000. In the previous version, the maximum number of columns (variables) is 50 while the maximum number of variables ranges from 50 to 1000, which is inconsistent. When number of output columns exceeds 50, it will be continued in the next line. There is no problem in loading the data into tecplot. However, for those who use external	

software, this may cause trouble in loading data, e.g., the gsd files in benchmarks_new_add\sit-clogging-V1.0.598\clogging. 2. Update 'twothird-mix' surface area update method.This feature updates reactive surface area using twothirds power relationship considering reaction direction and porosity change. For the primary mineral (those initially present with a volume fraction > phinucthrd), it uses twothird power relation for dissolution, but initial area for precipitation. Formula of dissolution reaction: Area = Area_init*(phim(im)/phim_init(im))^(r2/r3) Formula of precipitation reaction: Area = Area_init For secondary mineral (those initially present with a volume fraction <= phinucthrd), it uses twothird power relation for dissolution, but initial area for precipitation. Formula of dissolution reaction: Area = Area_init*(phim(im))^(r2/r3) Formula of precipitation reaction: Area = Area_init When porosity update is used, an additional term (pornew/porinit)^(r2/r3) is used in multipling surface area. 3. Update velocity reconstruction for variably saturated flow velocity output using Perot's method (Blair Perot, 2000, Conservation properties of unstructured staggered mesh schemes, Journal of Computational Physics, 159, 58-89). It was observed that this method does not always preduce good reconstruction, e.g., for stript_usg case, the reconstruction velocity is not as good as previous version which is based on the gradient reconstruction of the node. Also, the advective and diffusive velocity in reactive transport need further check. 4. Update of random generator in ArchiSimple code. By default, ArchiSimple code use time dependent random generator. This will make the result not identical. User can now use constant seed in random generator to keep the result identical for the same input file.	
Benchmarks Update	**Remarks**
1. benchmarks_new_add\root-uptake-ArchiSimple-V1.0.704	
Documentations Update	**Remarks**
5. MIN3P_THCm_UserManual 6. MIN3P_THCm_USG_UserManual	
Bugs Fix	**Remarks**

1. Fix bug in default gas tortuosity correction in Revison V633. When gas tortuosity correction is not specified, the default option is 'same as aqueous'. But in Revision V633, this default option is missing. 2. Fix bug in cell center dual control volume method. The control volume interface area is not accurate when the control volume interface does not cross the edge at the middle point of the edge. This cause small changes in \benchmarks_usg\react\d43_density_dep_heat_solute_usg_cc.	
Notes	**Remarks**
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm version (V633) shows no difference except the aforementioned benchmarks: benchmarks_new_add\sit-clogging-V1.0.598\clogging \benchmarks_usg\react\d43_density_dep_heat_solute_usg_cc The difference is caused by code update and bug fix.	

Revision 633

<table>
<tr><td><b>Revision Number</b></td><td>633</td><td><b>Version Number</b></td><td>V1.0.633</td></tr>
<tr><td><b>Commit Date</b></td><td>2018-12-07</td><td><b>Commit Author</b></td><td>Danyang Su</td></tr>
<tr><td><b>Software Grade</b></td><td></td><td>Signature</td><td>Date</td></tr>
<tr><td><b>Verified by</b></td><td>Mingliang Xie</td><td></td><td></td></tr>
<tr><td><b>Approved by</b></td><td>Uli Mayer</td><td></td><td></td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="3"><b>Compiler and configurations</b></td><td><b>Remarks</b></td></tr>
<tr><td colspan="3">1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089
2. Configurations
2.1 Preprocess source file: Yes (/fpp)
2.2 Preprocessor definitions for sequential version:
RELEASE_X64|RELEASE_WIN32; WINDOWS; USG; CGAL;
2.3 Preprocessor definition for parallel version:
RELEASE_X64| RELEASE_WIN32; WINDOWS; PETSC; MPI; PARDISO; OPENMP; USG; CGAL; LIS; SCHEDULE_STATIC;
2.4 Preprocessor definition for resources:
RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P
2.5 Additional Options: /fp:strict
3. External Library (Optional)
3.1 PETSc: PETSc V3.4 to V3.10
3.2 LIS: LIS V1.7.36 to V2.0.12
3.3 CGAL: CGAL V4.7</td><td>PETSc V3.9+ versionis required for unstructured grid parallel version.
CGAL is required if node/cell selection by polyhedral in 3D is used.
Configuration of LIS solver is required to be compatible with MIN3P configuration, e.g., LIS OpenMP version for MIN3P OpenMP version.</td></tr>
<tr><td colspan="3"><b>System Requirements</b></td><td><b>Remarks</b></td></tr>
<tr><td colspan="3">Intel Visual Fortran Redistributable library is required to run the parallel</td><td>Update this library</td></tr>
</table>

version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	accordingly if new compiler is used.
Summary of this version	**Remarks**
Add new boundary conditions (third-evap, mixed-evap, free-drainage) types; add activity coefficients output; add multi-point flux approximation (MPFA) and other optimization.	
New Features	**Remarks**
1. Add support of activity coefficients output. New files .gsac, .gbac and .lbac are created when activity coefficient output is activated. 2. Add support for tortuosity correction using 'millington-quirk experimental' or 'fix diffusion' based on the following formula. case('millington-quirk experimental') diffcoff = diff_p * (sat_p**b_mq) * (por**(a_mq+r1)) case('fix diffusion') diffcoff = diff_p 3. Add support to reverse cell-node ordering for unstructured mesh. For MIN3P unstructured grid version, counterclockwise cell-node ordering is required. In case the cell-node ordering is in clockwise ordering, this keyword can be used to maker internal conversion for cell-node ordering. 4. Add support to read initial condition and other parameters from vtk file. By default, the unstructured grid version uses tecplot data format when reading these variables from the external file. User now can use 'read ... from vtk file' to activate this feature. The file name of external data is changed accordingly by adding suffix '.vtk'. 5. Add support for 'minutes' time unit. 6. Add support for 'multi-point upstream weighting' for unstructured grid code. In structured grid code, upstream is based on two-point flux approximation. For unstructured grid, upstream can be based on multi-point flux aproximation that the upstream point can be defined by the gradient direction, not just by the hydraulic head. This features provide better convergence for highly nonlinear problem. When ‘upstream’ weighting is used, 'multi-point upstream weighting' is used as a default option in simulation using unstructured mesh. 7. Add keywords 'multi-point flux approximation ...' and 'two-point flux approximation ...' to include or exclude cross-diffusion term. These keywords have the same effect as 'include cross diffusion term ...' and 'exclude cross diffusion term ...'	New features work for both structured and unstructured version have been documented in MIN3P_THCm_UserManual. New features work for USG only have been documented in MIN3P_THCm_USG_UserManual. Consider to merge all these user manuals in the future. Features 11 to 14 have been verified against HP1 in the benmarking paper to be submitted to Comput. Geosci.

8. Add keyword 'use control volume half face center'. This keyword has the same effect as 'use separated interface centers'.

9. Add keyword 'threshold of failed timestep ratio'. This keyword is used to terminate simulation when the ratio of failed timesteps over total timesteps exceeds the specified threshold.

10. Add support for 'gas tortuosity correction' using 'assigned tau'. Gas tortuosity value can be read from zone input or from external file.

11. Add exponential decay function for kinetic reaction.

12. Add 'third-evap' boundary condition for transport boundary. Compared to 'third' type boundary condition, 'third-evap' allows inflow mass flux but no outflow mass flux, which means the mass is retained in the domain, e.g., water evaporation but solute is retained.

13. Add support for 'free-drainage' boundary type for flow. For free drainage, the gradient of total head is equal to 1 and the gradient of the pressure head is equal to 0.

14. Add 'mixed-evap' boundary condition for reactive transport. Different from 'mixed' boundary type, 'mixed-evap' boundary type does not allow mass out when flow flux is negative, like evaporation.

Usage of New Features

Remarks

1. Add 'output activity coefficients' into Data Block 'output control'.

2. Add the following part into Data Block 'control parameters - reactive transport'

```
   'tortuosity correction'
   'millington-quirk experimental'
   2.0d0                    ; a_mq
   4.2d0                    ; b_mq

   or

   'tortuosity correction'
   'fix diffusion'
```

3. Add 'reverse cell-node ordering' into Data Block 'spatial discretization'.

4. Change 'read ... from file' to 'read ... from vtk file' to read data from vtk file format. For example, when 'read initial mineral volume fractions from file' is used, the system will read in prefix.min as the external data file. To use vtk file format, user need to use 'read initial mineral volume fractions from vtk file' and change the external data file name to prefix.min.vtk.

5. Change the time unit in Data Block 4: 'time step control - global system' to 'minutes'. An example is shown below.

```
   'time step control - global system'
```

```
'minutes'              ;time unit
0.0                    ;time at start of solution
60.0                   ;final solution time
1.0                    ;maximum time step
1.0E-10                 ;minimum time step
```

6. Add 'multi-point upstream weighting' into Data Block 6: 'control parameters - variably saturated flow'. Note, this feature is only available when 'upstream weighting' is used in the flow system. It's not avaiable in 'centered weighting'. This is a default option for unstructured grid code. To use standard upstream weighting, please add the keywords to 'two-point upstream weighting' into Data Block 6.

7. Add 'multi-point flux approximation', 'multi-point flux approximation for flow', 'two-point flux approximation for flow' ... into data block 'spatial discretization'.

8. Add 'use control volume half face center' into data block 'spatial discretization'.

9. Add 'threshold of failed timestep ratio' followed by threshold value into data block 'time step control - global system'. As shown below, when more than 50% of timesteps fail, the simulation will be terminated.

```
'threshold of failed timestep ratio'
0.5d0
```

10. If 'gas tortuosity correction' is set to 'manual', the old prefix.gvs file is used as gas tortuosity.

If 'gas tortuosity correction' is set to 'assigned tau', user needs to specify gas tortuosity in the block 'physical parameters - porous medium', example input is

```
! -------------------------------------------------------------------------
'number and name of zone'
1
'background1'
0.25                 ;porosity
0.50                 ;tortuosity for aqueous phase
0.50                 ;tortuosity for gas phase

'extent of zone'
0.0 0.205  0.0 1.0  0.0  1.0

'end of zone'
```

If 'gas tortuosity correction' is set to 'assigned tau' and external file (prefix.torgas) is used as gas tortuosity, user needs to add keyword 'read gas tortuosity field from file' in the block 'physical parameters - porous medium', example input is

```
   'physical parameters - porous medium'
   1                          ;number of property zones

   'read tortuosity field from file'
   'read gas tortuosity field from file'
```

11. Modify redox database to add 'exponential T^a' term. Example is shown below.

```
   'exponential T^a'  2
   'biomass1'     0.5
   'biomass1'     0.3
```

12. Change the boundary type to 'third-evap'.

```
   'number and name of zone'
   1
   'inflow boundary'

   'boundary type'
   'third-evap'

   'concentration input'
   1.0         'free'          ;co3-2
   2.0         'free'          ;h+1

   'extent of zone'
   0.0 1.0  0.0 1.0  100.00 100.00

   'end of zone'
```

13. Add the following keywords to the block of 'boundary conditions - variably saturated flow'. Note: the parameter following 'free-drainage' is a dummy parameter. The data 0.0 is ignored after reading. This is used only to meet the requirement of input data format.

```
   'boundary type'
   'free-drainage'  0.0     ;flux
```

14. Use 'mixed-evap' instead of 'mixed' in Data Block 'boundary conditions - reactive transport'.

Codes Update	**Remarks**

1. Add support for PETSc V3.10 version.

2. Set default unit to atm for keyword 'partial gas pressure'. To use Pascal as input gas unit, please use keyword 'partial gas pressure (pa)'. The following keywords are equal.

```
   'partial gas pressure'
   0.3947694E-03              ;atm
```

'partial gas pressure (pa)' 40.0 ;Pa 2. Update column alignment for species and units in fls file. 3. Update the code to be compatible with latest Intel Compiler XE 2019 when OpenMP is used. The problem has been reported to Intel (Request 03681401) as there is bug in Intel Compiler that does not parse the OpenMP code correctly when unstructured grid capability is used. 4. Update of output information when default solver configuration is used. 5. Replace '\' with '/' for 'parallel solver configuration file' path as '\' only works for Windows. 6. Disable '.masmcd' and '.mssmcd' file creating for MCD case as the output for '.masmcd' and '.mssmcd' is not implemented in mbal_mcd.F90. 7. Length of btypezn is changed from 12 to 16 to account for the length of 'free-drainage'	
Benchmarks Update	**Remarks**
1. Benchmarks_new_add/ SOM_V1.0.632	
Documentations Update	**Remarks**
1. MIN3P_THCm_UserManual 2. MIN3P_THCm_ValidationReport 3. MIN3P_THCm_Parallel_UserManual 4. MIN3P_THCm_USG_UserManual	
Bug Fixes	**Remarks**
1. Fix bug in screen output of dbs_dir check. 2. Fix bug in velocity output for unstructured grid under unsaturated condition. Spatial weighting (e.g., upstream) is missing in calculate velocity. This bug affects velocity output file for unstructured grid version only. 3. Fix bug in totviscnew calculation. This variable is global, however, in the previous version, totviscnew is changed when loop over control volumes and connections in jacrt and then passed directly to calculate mass balance, causing error in calculating mass balance. The difference in mcd, mac, mae may be not significant. 4. Fix bug in mineral volume fraction assignment when read from file. In the previous code, the volume fraction read from file is compared to the value phic(im) from the zone and assign phic(im) instead if the value is below phic(im). In the updated code, the volume fraction read from file is compared to phimin(im) instead. The bug fix will affect the results in sulfur-V1.0.431.	

Notes	Remarks
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current (Benchmarks_V1.0.632) and the previous (Benchmarks_V1.0.609) MIN3P-THCm version shows the differences in Diff_Benchmarks_V1.0.632_vs_V1.0.609. These differences are caused by the code updates or bug fixes. The results are manually checked through BeyondCompare and tecplot postprocessing. The differences are comparatively small and generally unrecognizable in the visualization. Difference in the following benchmarks are detected. Benchmarks_new_add/sulfur-V1.0.431 Benchmarks_standard/MCD-2 Benchmarks_standard/MCD-2-1 Benchmarks_standard/MCD-2-advection Benchmarks_standard/MCD-2-advection-rc Benchmarks_standard/MCD-2-high-pH Benchmarks_standard/MCD-2-rc Benchmarks_usg/ velocity output (*.vel.vtk)	

Revision 609

Revision Number	609	**Version Number**	V1.0.609
Commit Date	2018-09-14	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32; WINDOWS; USG; CGAL; 2.3 Preprocessor definition for parallel version: RELEASE_X64| RELEASE_WIN32; WINDOWS; PETSC; MPI; PARDISO; OPENMP; USG; CGAL; LIS; SCHEDULE_STATIC; 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library (Optional) 3.1 PETSc: PETSc V3.4 to V3.9 3.2 LIS: LIS V1.7.36 to V2.0.12 3.3 CGAL: CGAL V4.7			PETSc V3.9 is required for unstructured grid parallel version. CGAL is required if node/cell selection by polyhedral in 3D is used. Configuration of LIS solver is required to be compatible with MIN3P configuration, e.g., LIS OpenMP version for MIN3P OpenMP version.
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable			Update this library accordingly if new

library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	compiler is used.
Summary of this version	**Remarks**
1. Add unstructured grid capabilities; 2. Add HPC (OpenMP, MPI, hybrid MPI-OpenMP) to unstructured grid code. 3. Update parallel code to support the latest PETSc library. 4. Bug fixes and other updates.	Name: MIN3P-HPC HPC stands for **h**igh **p**erformance **c**omputing as well as **h**igh **p**erformance code for **c**omplex geometries
New Features	**Remarks**
1. Implementation of unstructured grid capabilities and parallelization to the code. 2. Add 'fixed flow velocity' for unstructured mesh. For unstructured mesh, it is hard to generate strictly one-direction flow unless the mesh is fully orthogonal. In this case, the flow velocity can be forced to one direction with specified value. 3. Add constraints for node/cell selection. This can be combined used for node/cell selection by material id, node/cell id or by polygon/polyhedral selection. If the node or cell center is within this box (xmin xmax ymin ymax zmin zmax) or provided polygon/polyhedral, the node/cell is selected in the zone selection. 4. Add support for 'gradient' or 'gradient-top' or 'gradient-bottom' initial condition type for variably saturated flow. Type 'gradient-top' or 'gradient-bottom' is only applicable to unstructured grid with layered mesh. If 'gradient-top' is used, it will use the top surface node above the target node to calculate gradient, if 'gradient-bottom' is used, the bottom node below the target node is used to calculate gradient. 5. Add support for control of boundary flux/flow direction for the second boundary type in flow. Without specification, all flux direction is perpendicular to the boundary face. 6. Add control for velocity output averaging for flow and density dependent flow problem in the unstructured mesh. By default, the velocity of control volume ivol is averaged based on the velocity of control volume interfaces linked to ivol. For the nodes at the boundary of different zones with highly different material properties (e.g., conductivity), the velocity distribution of control volume interfaces may quite different for the unstructured mesh, depending on the provided mesh, making the velocity interpolation not accurate. In this case, a simplified method, based directly on the gradient of this node and minimum conductivity of this control volume is used to calculate the velocity. Note: unstructured grid code estimates velocity output at	The unstructured grid related commands are documented in MIN3P_THCm_USG_UserManual.

<table>
<tr><td>control volume but structured grid code calculates velocity at the control volume interface.
7. Add keyword to read skempton coefficients from zone selection commands.
8. Add support of nonlinear solver divergence check for Newton iteration in the code. For the previous code, if the solver diverges, it will continue until the maximum Newton iteration is reached. With divergence check, the Newton solver skips the unnecessary computing and reduce timestep when divergence in Newton iteration is found. For highly nonlinear problem, do not use this option.
9. Add water freezing/thawing process for general flow and reactive transport problems. For the previous version, it is only available when heat transport is considered.
10. Add water freezing relative conductivity fitting curve to smooth the conductivity change when water freezing/thawing process is considered. If this option is used, the relative conductivity is estimated based on the provided curve when temperature is between the provided minimum temperature and maximum temperature.
11. Add support of reading initial conditions from vtk file for flow, density dependent flow and heat transport. File name is prefix.ivs.vtk. This is only available for unstructured grid code.</td><td></td></tr>
<tr><td><b>Usage of New Features</b></td><td><b>Remarks</b></td></tr>
<tr><td>1. Add the following commands and keywords into Data Block 3: spatial discretization. The code will read in mesh prefix.vtk as the input mesh file and do spatial discretization based on given options. An example is shown below. Please look at MIN3P_THCm_USG_UserManual for more instructions.
'spatial discretization'
'read unstructured grid from file'
'gradient reconstruction method'
'least square' or 'green gauss' or 'least square second[third|fourth] order'
'control volume method'
'voronoi diagram' or 'median dual' or 'cell center'
'allow obtuse cells'
'done'
2. Add the following keywords and data into Data Block 'control parameters - variably saturated flow'
'fixed flow velocity'
0.0d0 0.0d0 0.0d0 ;fixed flow velocity in (x,y,z) direction in m/day
3. Modify the zone selection accordingly, as shown below. Please look at MIN3P_THCm_USG_UserManual for more instructions.
'extent of zone: coordinates constraint'
xmin xmax ymin ymax zmin zmax
4. When 'gradient' is used as initial condition definition, if dir_grad == 'x', then pos_grad=xg(ivol), else if dir_grad == 'y', then pos_grad=yg(ivol), else if dir_grad == 'z', then pos_grad=zg(ivol). The initial condition is calculated as hinitial = h0_grad + slope_grad*pos_grad.
'initial condition'</td><td></td></tr>
</table>

```
  'gradient' dummy           !dummy value is ignored
  'dir_grad'
  h0_grad
  slope_grad
  'extent of zone'
  400.0 600.0  400.0 600.0  -200.0 100.0
```

When 'gradient-top' or 'gradient-bottom' is used as initial condition definition, if dir_grad == 'x', then pos_grad=xg(ivol-top), else if dir_grad == 'y', then pos_grad=yg(ivol-top), else if dir_grad == 'z', then pos_grad=zg(ivol-top). The initial condition is calculated as hinitial = h0_grad + slope_grad*pos_grad.

```
  'initial condition'
  'gradient-top' dummy           !dummy value is ignored
  'dir_grad'
  h0_grad
  slope_grad
  'extent of zone'
  400.0 600.0  400.0 600.0  -200.0 100.0
```

5. Add keyword 'boundary flow direction' when setting the second boundary type for flow/evaporation boundary.

```
  'number and name of zone'
  1
  'inflow boundary'
  'boundary type'
  'second'   1.11d-7            ;flux
  'boundary flow direction'
  0.0 0.0 1.0
  'extent of zone'
  0.0 100.0  0.0 50.0  20.0 20.0
  'extent of zone: boundary nodes only'
  'end of zone'
```

6. Add the following command into Data Block 'spatial discretization'. If the coefficient of variation of velocity distribution is bigger than this value, then use simplified method, otherwise, use normal velocity interpolation method. By default, the threshold is 0.5.

```
  'variation coefficient for velocity average'
  0.5d0
```

7. Add following command line and data into Data Block 'physical parameters - variably saturated flow'

```
  'specific skempton coefficient'
  1.0
```

8. Add 'enable divergence check' in into Data Block 'control parameters – variably saturated flow' and 'control parameters – reactive transport'.
9. Add the following commands into Data Block 'control parameters - variably saturated flow' to activate water freezing/thawing module for flow problem.

```
  'water freezing temperature'
  -1.0
  'water freezing conductivity'
```

1.0d-30 Add the following command into Data Block 'control parameters - reactive transport' to set reaction rate to zero or very small value when water is freezing. 'water freezing reaction rate' 1.0d-30 10. Add the following keywords and parameters into Data Block 'control parameters - variably saturated flow'. Make sure keyword 'water freezing conductivity' is removed, otherwise, it will use fixed 'water freezing conductivity' instead. The output of temperature - fitted relative conductivity is exported in the gen file. 'water freezing relative conductivity curve' 0.01273 22.47 2.002 1.002 ;parameter a, b, c and d for relative ;conductivity curve relcond=d+(a-d)/ ;(1.0+((temperatureC+2.0)/c)^b) -1.0 1.0 0.01 1.0 ;minimum and maximum temperature, ;minimum and maximum relcond 11. Use vtk file format named prefix.ivs.vtk as the initial condition file. Add the following keywords to related block. 'read initial condition from vtk file'	
Codes Update	**Remarks**
1. Update convergence information output of Picard iteration.The previous picard convergence output is not of the current timestep, but of the previous timestep. 2. Update in output format for real*8, use 1.0E-123 format instead of 1.0-123. The later format is not supported by some software such as paraview. 3. Modify makefile for PETSc version preprocessing definition. User does not need to specify PETSc version. The current active PETSc version ($PETSC_DIR) will be used. 4. Remove initial guess of solution vector norm check for parallel version. This part is not required and the time spent in norm calculation does not worth. 5. Update support for sytem enviornment variable OMP_NUM_THREADS. By default, if number of threads is specified in the solver configuration file, the code will use this value, otherwise, it will use sytem enviornment variable OMP_NUM_THREADS. 6. Update default lis solver option. The new default lis solver option is " -p ilu -tol 1.0E-20 -maxiter 5000". This new default option works better in MIN3P code than the default LIS solver option where no preconditioner is used. 7. Disable output of hydraulic conductivities (.hycx). The output is actually the scaling factor for permeability (frac). 8. Update activities guess in guess.F90. Previous code use actv(ic) = r10**(-2.8980d0)*totco(ic) when DGM model is ported. This function is now switched back to actv(ic) = vhoff(-2.7945d0,1.2647d-3,tempkel,tempks,rgascal) and actv(ic) = actv(ic)*totco(ic) for non-DGM cases. The difference caused by this change is small. The following cases are affected. benchmarks_new_add\edbasin-base-V1.0.473	

benchmarks_new_add\co2seq-water-freezing-V1.0.377 benchmarks_new_add\dusty-gas-model-V1.0.303\deviss139 benchmarks_standard\reactran\amd_ex benchmarks_standard\reactran\amd_ex2 benchmarks_standard\reactran\degas benchmarks_standard\reactran\perm3_new 9. Update the maximum line length in the input file from 72 to 256. Please note the maximum keywords length is still 72 while the maximum data length, e.g., database path, can be up to 256. For the previous code, if the length of database path or other keyword exceeds length 72, the code will crash. 10. Disable file creating when there is no output in the file, e.g., gbm file in benchmark fen22 to fen24.	
Benchmarks Update	**Remarks**
Added the following unstructured grid benchmarks: 1. Benchmarks_usg/flow/clem3d_hexa 2. Benchmarks_usg/flow/clem3d_prism 3. Benchmarks_usg/flow/clem3d_usg 4. Benchmarks_usg/flow/clement_usg 5. Benchmarks_usg/flow/shlomot_usg 6. Benchmarks_usg/flow/stedfs_usg 7. Benchmarks_usg/flow/stedvs_usg 8. Benchmarks_usg/flow/tranfs_usg 9. Benchmarks_usg/flow/tranvs_quad 10. Benchmarks_usg/flow/tranvs_usg 11. Benchmarks_usg/react/clem3d_usg 12. Benchmarks_usg/react/d32_elder_problem_usg 13. Benchmarks_usg/react/d33_modified_henry_usg 14. Benchmarks_usg/react/d43_density_dep_heat_solute_usg 15. Benchmarks_usg/react/d43_density_dep_heat_solute_usg_cc 16. Benchmarks_usg/react/d43_density_dep_heat_solute_usg_ls3md 17. Benchmarks_usg/react/d44_thermo_haline_convection_usg 18. Benchmarks_usg/react/d45_salt_dome_problem_d80_b5_usg 19. Benchmarks_usg/react/d45_salt_dome_problem_d133_b3_usg 20. Benchmarks_usg/react/d45_salt_dome_problem_d200_b2_usg 21. Benchmarks_usg/react/d311_hydrostatic_box_test_usg 22. Benchmarks_usg/react/d312_hydrodynamic_box_problem_usg 23. Benchmarks_usg/react/diff_h_cc_2D_C_usg 24. Benchmarks_usg/react/het_2d_complex_usg 25. Benchmarks_usg/react/het_2d_usg 26. Benchmarks_usg/react/hetero_frac_slope_usg 27. Benchmarks_usg/react/patchf_usg 28. Benchmarks_usg/react/pile_hexa 29. Benchmarks_usg/react/ssbench4_sand_usg 30. Benchmarks_usg/react/stripf_usg 31. Benchmarks_usg/react/tranrt_quad 32. Benchmarks_new_add/perm3-clogging-V1.0.609	Modified based on the benchmarks of structured grid cases. The boundary conditions may be different as the meshes are also different. These cases are for demonstration purpose.
Documentations Update	**Remarks**

5. MIN3P_THCm_USG_UserManual.pdf 6. MIN3P_THCm_Parallel_UserManual.pdf	MIN3P_THCm is continued to be used in documentation.
Bug Fixes	**Remarks**
1. Add convergence check and fix bug in ddfsflow, fsflow when newton iteration is not converged. Previous code returns the unconverged results. 2. Fix bug in valovapour, use ups_heat instead of ups_flow for diff_energybal. The previous code use spatial weighting of flow for heat transport. 3. Fix bug in timeloop for picard iteration, comment out "not_converged_sia=.false.", otherwise, picard iteration may return results without convergence check. 4. Fix bug in updating delt_tds value when timestep needs to be reduced. The previous code in timeloop.F90 does not include this part when timestep is reduced. If this value is not reduced accordingly, the convergence is not improved when timestep is reduced, which may cause iteration failure. 5. Fix bugs in storage term calculation in density dependent flow. This bug is caused in pressure difference calculation when 'average density in z' is used. Bug is benign and does not show significant difference. 6. Fix bug in using uninitialized variable in jacddfs_energybal. This may cause extreme value in calculating matrix entry. 7. Fix bug in gasdecay mass balance calculation. 8. Fix bug in OpenMP related race condition. 9. Fix bug in PETSc KSP solver when initial solution meet the convergence results (iteration=0). Previous code will treat this as non-converged situation. 12. Fix bug in passing solver parameter from solver.cfg (For PETSc and LIS solver type) back to MIN3P code. Some convergence checking need these parameters to be set. By default, these parameters are from input file. However, if solver type is specified to PETSc or LIS, these parameters should be updated accordingly. 13. Fix bug in solver 'overflow' problem, timestep needs to be reduced instead of termination for transient flow in full saturated condition (fsflow). 14. Fix bug in NaN check for linear solver ws209. The previous code does not check this and still return 'converged' with zero solution. 15. Fix bug in calculation of Jacobian matrix entries when 'clogging' is used. The previous code returns zero diagonal entry when it is clogged, causing NaN after ILU factoration. The solver in the previous version does not check this error and returns zero solutions. 16. Fix bug in 'read hydraulic conductivity field from file'. If this command is used, van Genuchten soil parameters and/or specific storage coefficient are ignored. This bug forces the code to work for saturated flow only when hydraulic conducitivity field is read from enternal file in	The bug fixes in linear solver and nonlinear solver are generally benign and do not cause sigfinicant difference for the results of the benchmarks we have. This is because the returned solutions are close to the real solutions. However, it could be fatal when simulation conditions are changed.

the previous version.

17. Fix bug in hydraulic conductivity output when hydraulic conductivity or permeability is read from the file. The previous code will only outputs when hydraulic conductivity or permeability is read from zone setting.
18. Fix bug in PETSc residual output for user-defined convergence, this bug is benign and does not affect the results.
19. Fix bug in result output of gsb file, in which saturation*porosity should be applied for the unit conversion. The previous code lacks the porosity term. This affects the output of gsb when ion-exchange or surface adsorption is used.
20. Fix bug in hydraulic conductivity output in gsv file. The zone pointer in the previous code always point to the last zone. This bug fix will cause difference in sit-clogging-V1.0.598\clogging\benC1a benchmark.
21. Fix bug in total velocity output in gsa file. The total velocity term is not averaged while other (adv, dif, mig) terms are averaged, making tot != adv + dif + mig. This bug fix will affect all the gsa output.
22. Fix bug in the header of gbp file, coordinates in gbp file is not correct.
23. Fix bug in timestep modification when update boundary condition is used. The previous code "if (itsrc.le.ntsrc.and.tsrc(itsrc)-tiny_time.lt.time) then" in timeloop.F90 is buggy that when tsrc is much bigger than tiny_time. This is caused by decimal error when add/minus a very small value to/from a big value. The timestep adjustment may fail as tsrc(itsrc)-tiny_time is equal to time, e.g., tsrc(itsrc)=4380000.00 (12000 years), tiny_time = 1.0d-10, time=4380000.00 (12000 years). Theoretically, tsrc(itsrc)-tiny_time < time, however, for double precision, tsrc(itsrc)-tiny_time = time. To make it work, avoid subtracting a very small value from a very big one due to the precision limit in compute code.
24. Fix bug in DGM model gas flux calculation when Maxwell Stefan module is used. The previous code and MIN3P-DGM-2012 code pass the wrong variable to function ms_fluxdg_s. This will affect all the mgc files.
25. Fix bug in using uninitialized variables. Without initialization, a very small value, e.g., 1.0d-312, is used and may cause unrealistic output when the exact value should be zero.

Notes

Remarks

1. MIN3P-THCm-USG work with PETSc-3.8+ dev version and/or PETSc PETSc-3.9+ release version.
2. HDF5 write in Linux system take large memory space due to the unreleased memory of hdf5 write buffer. When HDF5 is used (USG, Binary), it takes three times more memory than the code uses ascii output or disabled output. This is a system problem, not the HDF5 library, as in Linux, if the kernel needs more memory, it grabs as much as there is physical memory. See detail in https://support.hdfgroup.org/HDF5/faq/linux-mem.html.
3. 'reading *** from file' is not supported for unstructured grid version when the code is running in MPI or hybrid MPI-OpenMP parallelization mode. This is because the current PETSc version does not support node/cell numbering mapping to subdomain for the

general users. Will update this part if this features is open for public use.	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current (Benchmarks_V1.0.609) and the previous (Benchmarks_V1.0.598) MIN3P-THCm version shows the differences in Diff_Benchmarks_V1.0.605_vs_V1.0.598. These differences are caused by the code updates or bug fixes. The results are manually checked through BeyondCompare and tecplot postprocessing. The differences are comparatively small and generally unrecognizable in the visualization. The following cases are affected by bug fixes or code updates: benchmarking_nwmo_report\nwmo_verification_examples\d32_elder_problem benchmarking_nwmo_report\nwmo_verification_examples\d311_hydrostatic_box_test\coupled benchmarking_nwmo_report\nwmo_verification_examples\d312_hydrodynamic_box_problem benchmarking_nwmo_report\nwmo_verification_examples_D4\d42_density_dep_energy\centered\injection benchmarking_nwmo_report\nwmo_verification_examples_D4\d42_density_dep_energy\upstream\injection benchmarking_nwmo_report\nwmo_verification_examples_D4\d44_thermo_haline_convection\negative_buoyancy benchmarking_nwmo_report\nwmo_verification_examples_D4\d44_thermo_haline_convection\positive_buoyancy benchmarking_nwmo_report\nwmo_verification_examples_D5\d51_dedolomitization\min3p-nwmo benchmarking_nwmo_report\nwmo_verification_examples_D5\d52_cation_exchange\min3p benchmarking_nwmo_report\nwmo_verification_examples_D5\d52_cation_exchange\min3p benchmarking_nwmo_report\nwmo_verification_examples_D5\d53_aerobic_degradation\min3p benchmarking_nwmo_report\nwmo_verification_examples_D6\d6_rt_highly_saline\min3p benchmarking_nwmo_report\nwmo_verification_examples_D452\d200_b2 benchmarks_new_add\co2seq-water-freezing-V1.0.377 benchmarks_new_add\dusty-gas-model-V1.0.303\bem20510e-2 benchmarks_new_add\dusty-gas-model-V1.0.303\deviss139 benchmarks_new_add\elder-lis-omp4-V1.0.527 benchmarks_new_add\gas_advection-V1.0.175\amd505dgm1_gas_adv benchmarks_new_add\gas_advection-V1.0.175\amd505dgm1_gas_adv_water_table benchmarks_new_add\gas-bubble-V1.0.303\Amos and Mayer 2006\Williams benchmarks_new_add\gas-bubble-V1.0.303\Jeen et al 2012\506-4 benchmarks_new_add\gas-decay-first-order-V1.0.530\gas-decay-	

napl benchmarks_new_add\gas-decay-first-order-V1.0.530\gas-nodecay-napl benchmarks_new_add\isotope-V1.0.303\Gibson et al 2011\Waybrant_2 – new benchmarks_new_add\isotope-V1.0.303\Jamieson-Hanes et al 2012\Cr_columns – 3 benchmarks_new_add\isotope-V1.0.303\Jamieson-Hanes et al 2014\CrZVI_FTC - 3b benchmarks_new_add\isotope-V1.0.303\Jamieson-Hanes et al 2014\CrZVI_FTC - 3c benchmarks_new_add\isotope-V1.0.303\Wanner et al 2014\Problem 1 benchmarks_new_add\multisite-ionx-V1.0.129\ionx-3sites-2domains benchmarks_new_add\multisite-ionx-V1.0.129\ionx-m-2domains benchmarks_new_add\multisite-ionx-V1.0.129\ionx-m-2domains-gapon benchmarks_new_add\sit-clogging-V1.0.598\1D_sit_verif benchmarks_new_add\sit-clogging-V1.0.598\clogging benchmarks_new_add\sulfur-V1.0.431\2D-sal benchmarks_new_add\sulfur-V1.0.431\2D-sal_ice4 benchmarks_new_add\sulfur-V1.0.431\sulfur benchmarks_new_add\sulfur-V1.0.431\verification\sul2pqc-3Domain benchmarks_new_add\sulfur-V1.0.431\verification\sul2pqc-cos benchmarks_new_add\sulfur-V1.0.431\verification\sul2pqc-n benchmarks_new_add\surfx-ionx-unsat-V1.0.191\surfx-drain1 benchmarks_new_add\surfx-ionx-V1.0.106\ionx-2-crunch-crunch-dbs-cec0.73 benchmarks_new_add\surfx-ionx-V1.0.106\ionx-2-phreeqc-default-dbs-cec0.14 benchmarks_new_add\surfx-ionx-V1.0.106\ionx-2-phreeqc-default-dbs-cec0.73 benchmarks_new_add\surfx-ionx-V1.0.106\ionx-2-phreeqc-default-dbs-ph7-cec0.14 benchmarks_new_add\surfx-ionx-V1.0.106\ionx-2-phreeqc-default-dbs-ph7-cec0.73 benchmarks_new_add\surfx-ionx-V1.0.106\ionx-m benchmarks_new_add\surfx-ionx-V1.0.106\ionx-m-2domains benchmarks_new_add\surfx-ionx-V1.0.106\ionx-multisite-2domain benchmarks_new_add\surfx-ionx-V1.0.106\ionx-multisite-default-dbs-cec0.73 benchmarks_new_add\surfx-ionx-V1.0.106\surfx-2-phreeqc-default-dbs benchmarks_new_add\surfx-ionx-V1.0.106\surfx-2-phreeqc-default-dbs-new benchmarks_new_add\surfx-ionx-V1.0.106\surfx-ionx-phreeqc-default-dbs-cec0.73-eq benchmarks_new_add\surfx-ionx-V1.0.106\surfx-ionx-phreeqc-default-dbs-cec0.73-eq-new benchmarks_new_add\uraninite-reoxidation-V1.0.56\2d-run10 benchmarks_new_add\uraninite-reoxidation-V1.0.56\2d-run16	

benchmarks_standard\flow\shlomo benchmarks_standard\flow\shlomot benchmarks_standard\reactran\amd_ex benchmarks_standard\reactran\amd_ex2 benchmarks_standard\reactran\degas benchmarks_standard\reactran\diff_harmonic\diff_h_cc_2D_H benchmarks_standard\reactran\diff_harmonic\diff_h_cc_3D_H benchmarks_standard\reactran\dissol benchmarks_standard\reactran\ex11 benchmarks_standard\reactran\het_2d benchmarks_standard\reactran\hom_2d benchmarks_standard\reactran\MCD-2-advection\min3p benchmarks_standard\reactran\MCD-2-advection-rc\min3p benchmarks_standard\reactran\perm1 benchmarks_standard\reactran\perm1_new benchmarks_standard\reactran\perm2 benchmarks_standard\reactran\perm2_new benchmarks_standard\reactran\perm3 benchmarks_standard\reactran\perm3_new benchmarks_standard\reactran\prb benchmarks_standard\reactran\retardation benchmarks_standard\reactran\surfx benchmarks_standard\reactran\transrc benchmarks_standard\reactran\weather The following cases are affected by the change in activities guess in guess.F90: benchmarks_new_add\basin-base-V1.0.473 benchmarks_new_add\co2seq-water-freezing-V1.0.377 benchmarks_new_add\dusty-gas-model-V1.0.303\deviss139 benchmarks_standard\reactran\amd_ex benchmarks_standard\reactran\amd_ex2 benchmarks_standard\reactran\degas benchmarks_standard\reactran\perm3_new	

Revision 598

Revision Number	598	**Version Number**	V1.0.598
Commit Date	2018-07-20	**Commit Author**	Mingliang Xie
Software Grade		Signature	Date
Verified by	Danyang Su		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;LIS;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x			LIS in MIN3P_THCm is currently only available on Linux/Unix OS.
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.			
Summary of this version			**Remarks**

1. Added SIT model (based on the Specific ion Interaction Theory). 2. Added pore clogging function. 3. Updated the code for the output for the porosity and hydraulic conductivities in the output file *_*.gsv.	
New Features	**Remarks**
26. SIT model: It is used to estimate single-ion activity coefficients in electrolyte solutions at relatively high concentrations. In comparison to the Pitzer model, the SIT model is not as accurate but used less coefficients. 27. Pore clogging function: this function is added to enable the simulations of cases with pore clogging. The criteria for “fully clogged” cells is when the porosity is less than or equal to the minimum porosity threshold. It is assumed that no chemical reactions occur within fully clogged control volume(s) and no flux between adjacent control volume(s).	
Usage of New Features	**Remarks**
1. SIT model: use the keywords **'use sit model'** in Data Block 2: geochemical system. The ion interaction coefficients are provided by the database SIT.dbs, which should be in the database folder. The format of the SIT.dbs is documented in the user manual. 2. Pore clogging function: use the keywords **'pore clogging**’ in Data Block 7: control parameters - reactive transport. Note: This function works only when ‘update porosity’ and ‘update permeability’ functions are turned on. It is recommend to specify the minimum porosity threshold. Otherwise, the default value is set to 1.0e-6. Example: ! Data Block 7: control parameters - reactive transport ! --- ! 'control parameters - reactive transport' 'dense block matrices' 'mass balance' 'spatial weighting' 'upstream' 'activity update settings' 'time_lagged' 'update porosity' 'update permeability' **'pore clogging'** 'porosity threshold minimum' 1.0d-4	Demonstration purpose only for the pore clogging function, not verified.
Codes Update	**Remarks**
1. Modified code in outputrt.F90 for the output of hydraulic conductivity from the previous kzz only to kxx, kyy and kzz (affected output file is *_*.gsv). 2. Modified the same file for the output of porosity from the previous value of the local parameter por_out – the total bulk volume (i.e. 1.0) minus the	

volume fraction of all specified minerals, to the parameter pornew(ivol) representing the porosity at the current time step that is used for the flow and reactive transport simulations. In the previous version, the porosity calculation is correct if the volume fractions of all solid phases (i.e. reactive and inert ones). Otherwise, this could result in the output of a porosity higher than the actual value. Nevertheless, this is only for output, no impact on the overall simulation results.	
Benchmarks Update	**Remarks**
Added two benchmarks: 33. Benchmark **sit** is to verify the implementation of the SIT model through code comparison to PHREEQC. 34. Benchmark **clogging** is to demonstrate the pore clogging function. 35. Added the folder: .\benchmarks_new_add\sit-clogging-V1.0.598	
Documentations Update	**Remarks**
1. Version report 2. User guide 3. Theory manual 4. Verification report	
Bug Fixes	**Remarks**
none	
Notes	**Remarks**
none	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm version shows no difference except the *.gsv files – different porosity values for most cases, plus two additional hydraulic conductivity values in cases used update permeability functions. The difference is caused by the code update in the output of porosity and hydraulic conductivity values. The values of vertical hydraulic conductivity remains unchanged. The porosity values of some benchmarks that do not consider the volume fractions of all solid phases are different. The affected benchmarks showing difference in porosity are: dedo, polyhal-pitz, amd505, bem, sch, thor, gastest, RnDecayNaplVar, amd, Cr-FTC, Cr, sul2pqc, uo2-oxidation, amd-ex, comptran, degas, dissol, prb, raoult, weather. Nevertheless, all other results such as *.gst, *.gbt show no difference, indicating the difference in porosity value has no impact on the simulation results.	

Revision 553

Revision Number	553	**Version Number**	V1.0.553
Commit Date	2018-01-30	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;LIS;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x			LIS in MIN3P_THCm is currently only available on Linux/Unix OS.
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.			
Summary of this version			**Remarks**

Add aqueous and non-aqueous component concentration output (prefix_o.aqt) before batch reaction.	
New Features	**Remarks**
None.	
Usage of New Features	**Remarks**
None.	
Codes Update	**Remarks**
1. Update the water freezing related code and boundary condition setting when 'water freezing' is used. In the previous code, when water is freezed, hydraulic conductivity is always reduced to a very small value, in the updated code, the hydraulic conductivity change for the freezed water can be specified or turned off.	Demonstration purpose only, not rigorous, not verified.
Benchmarks Update	**Remarks**
None.	
Documentations Update	**Remarks**
1. Version report	
Bug Fixes	**Remarks**
1. Fix bug in mass balance calculation when 'atmospheric' boundary condition type is used. The flow and heat flux on the boundary in mass balance calculation in the previous versions are not updated and always use the last saved values. This does not cause problem in 1D as there is only 1 atmospheric boundary assigned. But for 2D and 3D where flow and heat flux across the boundary can be different, this causes imbalance in the mass output.	
Notes	**Remarks**
1. 'water freezing' feature is for demonstration purpose only. This part is not theoretical rigorous or verified with other code or experiment data.	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm version shows no difference except the demonstration case co2seq-water-freezing-V1.0.377. The difference is caused by the code update in water freezing part. This case is for demonstration purpose only.	

Revision 549

Revision Number	549	**Version Number**	V1.0.549
Commit Date	2018-01-10	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;LIS;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x			LIS in MIN3P_THCm is currently only available on Linux/Unix OS.
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.			
Summary of this version			**Remarks**

Add aqueous and non-aqueous component concentration output (prefix_o.aqt) before batch reaction.	
New Features	**Remarks**
1. Add aqueous and non-aqueous component concentration output (prefix_o.aqt) before batch reaction. This can be further used as initial condition when reading aqueous component concentration from file.	
Usage of New Features	**Remarks**
1. Change prefix_o.aqt to prefix.aqt as the initial aqueous and non-aqueous component concentration.	
Codes Update	**Remarks**
None.	
Benchmarks Update	**Remarks**
None	
Documentations Update	**Remarks**
1. Version report	
Bug Fixes	**Remarks**
None.	
Notes	**Remarks**
2. The initial component concentration in prefix_0.gst file is after batch reaction. When non-aqueous component is excluded, there is no big difference in prefix_o.aqt or prefix_0.gst. However, when non-aqueous components are included, there is significant difference in these two files.	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm version shows no difference except the new generated prefix_o.aqt files.	

Revision 546

Revision Number	546	**Version Number**	V1.0.546
Commit Date	2017-11-29	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;LIS;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x			LIS in MIN3P_THCm is currently only available on Linux/Unix OS.
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.			
Summary of this version			**Remarks**

This version adds support of space and tab delimiters for database files comp.dbs, complex.dbs and gases.dbs.	
New Features	**Remarks**
1. Add support of space and tab delimiters for database files comp.dbs, complex.dbs and gases.dbs. The previous versions need fixed column for these three database files. Duplicated space and tab are treated as one delimiter. This feature is useful for those who need to modify the database but not familiar with fixed column format. Incorrect column index cause either incorrect parameters input or code crash.	
Usage of New Features	**Remarks**
1. Add 'use space delimiter in database' in block 'geochemical system'.	
Codes Update	**Remarks**
None.	
Benchmarks Update	**Remarks**
1. batch\albite\albite.dat. Add 'mineral input' for the second zone. 2. batch\appelo\appelo.dat. Add 'mineral input' for the second zone. For MIN3P-THCm code prior to Revision 303, if 'mineral input' is missing in the second zone, the code will output error information in the log file but continue the simulation. From Revision 303, the code stops if this error information is detected. 3. flow\shlomo\shlomo.dat. Change maximum linear iteration from 100 to 500. From MIN3P-THCm code Revision 382, the code will check the linear solver convergence when maximum linear iteration number is reached. For the code prior to R382, the linear solver will return the solution as 'converged' even if it does not meet the criteria of solver residual and maximum tolerance.	These benchmarks should be updated at the same time when code was updated.
Documentations Update	**Remarks**
1. Version report	
Bug Fixes	**Remarks**
None.	
Notes	**Remarks**
None.	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm version shows no difference except the aforementioned changes in benchmarks update. Folder: MIN3P-THCm-Benchmarks\Diff_Benchmarks_V1.0.546_vs_V1.0.542.	

Revision 542

Revision Number	542	**Version Number**	V1.0.542
Commit Date	2017-11-16	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;LIS;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x	LIS in MIN3P_THCm is currently only available on Linux/Unix OS.

System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	

Summary of this version	**Remarks**
This version includes 1) bug fix in reading .gsv and .gvs file as initial condition for the MPI parallel version and 2) output format from e15.7 to e15.6e3.	
New Features	**Remarks**
None	
Usage of New Features	**Remarks**
none	
Codes Update	**Remarks**
1. Update in reading key word 'van leer' and 'van leer 2', these two keywords will be treated as 'vanleer' and 'vanleer2' respectively. This is used in checking stencil width for the parallel version. All vanleer spatial weighting need stencil width >= 2. The previous version only check keyword 'vanleer' and 'vanleer2' without checking 'van leer' and 'van leer 2'. 2. Format the output from e15.7 to e15.6e3. For very small value, e.g., 1.2345678E-123, e15.7 format exports the value as 1.2345678-123 and e15.6e3 exports the value as 1.234567E-123. The former one is the default format by fortran. Please note the 7th decimal is ignored in order to keep the same data column width in the ascii output.	
Benchmarks Update	**Remarks**
None	
Documentations Update	**Remarks**
1. Version report	
Bug Fixes	**Remarks**
1. Fix bug in reading .gvs and .gsv file for the MPI parallel version, node stride is missing for these two files.	
Notes	**Remarks**
1. The verification tool (tools\MIN3PRunAndResultCompare\bin) is updated, with relative error checking.	MIN3P-THCm-Benchmarks\Benchmarks_V1.0.542_Check
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm version shows no difference except the output format.	

Revision 535

Revision Number	535	**Version Number**	V1.0.535
Commit Date	2017-10-24	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;LIS;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x	LIS in MIN3P_THCm is currently only available on Linux/Unix OS.
System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	

Summary of this version	**Remarks**
This version includes 1) reactive transport boundary condition update during ice melting stage; 2) ascii data format for transient output for the parallel version.	
New Features	**Remarks**
1. Add boundary condition updating during ice melting stage. 2. Add support for ascii data format output for transient output in parallel.	
Usage of New Features	**Remarks**
1. The input file needs to be modified so as to make the reactive transport boundary condition work in the block 'ice sheet loading. Add the boundary condition of chemical components (format shown below) after declariation of ice stages if this is missing. 'concentration input' 10 ! number of chemical components 0.2915E-02 'charge' ;cl 0.4000E-04 'free' ;br 0.5220E-03 'free' ;na 0.6400E-04 'free' ;k 6.0 'ph' ;h 0.1722E-04 'free' ;hco3 0.6240E-03 'free' ;ca 0.8230E-03 'free' ;mg 0.2600E-03 'free' ;so4 1.0d0 'po2' ;o2(aq) 2. By default, if 'use binary format' is specified, transient output will also in binary format. If 'use binary format' is specified and you want to use ascii data format for the transient output, please add 'use ascii data format for transient output' in the output section.	Without 'concentration input' commands, the reactive transport boundary change during ice loading/unloading stages is ignored.
Codes Update	**Remarks**
1. Remove maximum pressure update restriction caused by ice loading/unloading. This code is not strict. Use larger porosity update factor to solve the convergence problem caused by large pressure update.	
Benchmarks Update	**Remarks**
None	
Documentations Update	**Remarks**
1. Version report	
Bug Fixes	**Remarks**
1. Fix bug in the output of gen file for mpi version. The code will output gen file information to the fort.xxxx as the file unit is not defined for porcessors with rank > 0. 2. Fix bug in flow boundary condition adjustment under ice sheet for different	

stages. No flow boundary condition is used for boundary cells under ice sheet during the glaciation time. 90% (or other, from input file) of ice thickness is used as pressure head during deglaciation time.	
Notes	**Remarks**
None	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm version shows no difference except the following ice melting related cases. The difference is caused by the bug fix in the new version. benchmarks_new_add\basin-base-V1.0.473 benchmarks_new_add\sulfur-V1.0.431\2D-sal_ice4	

Revision 533

Revision Number	533	**Version Number**	V1.0.533
Commit Date	2017-10-16	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;LIS;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x	LIS in MIN3P_THCm is currently only available on Linux/Unix OS.
System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	

Summary of this version	**Remarks**
This version includes some bug fix for mixed boundary condition, transient boundary condition and race condition for OpenMP version.	
New Features	**Remarks**
None.	
Usage of New Features	**Remarks**
None.	
Codes Update	**Remarks**
1. Exclude diffusive flux as default for the mixed boundary condition type. To include diffusive flux from mixed boundary condition for reactive transport, please add the following command to the zone using 'mixed' boundary condition type in Block "boundary conditions - reactive transport". 'include diffusive flux from mixed boundary condition' 2. Command of 'spatial weighting of boundary condition interpolation' is removed. 3. Remove tranbcrt_read.F90, restore to old tranbcrt function with transient boundary condition function added.	
Benchmarks Update	**Remarks**
1. benchmarks_standard\reactran\amd_ex 2. benchmarks_standard\reactran\amd_ex2	1. with gas diffusive flux for the mixed bc 2. without gas diffusive flux for the mixed bc
Documentations Update	**Remarks**
None	Not updated in the last version
Bug Fixes	**Remarks**
1. Fix bug in 'mixed' boundary condition type for reactive transport. This bug is caused by diffusive flux in jacbrt.F90, mbalrt.F90 and mbal_mcd.F90. All related code should include the switch of b_fluxd_mixed_bcond(ivol) when applying mixed boundary condition. By default, diffusive flux is NOT used in the mixed boundary condition type. 2. Fix bug in jacbrt.F90 for derivative considering mixed boundary condition when diffusive flux in the aqueous phase is not considered. 3. Fix bug in the transient boundary conditions for reactive transport. It may cause inconsistent boundary condition. This bug is introduced when transient	

boundary condition function and boundary condition interpolation were added. 4. Fix bug in tid related subroutines. The previous code will assign tid = 1 to those outside parallel loop, which may cause thread inconsistent. 5. Fix bug in race condition in jacbrt and mbalrt.	
Notes	**Remarks**
None	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm version shows no difference except the examples with mixed boundary condition and transient boundary condition. Results from the following examples shows small difference due to the bug fixes and code updates. benchmarks_standard\reactran\amd_ex benchmarks_standard\reactran\transrc benchmarks_new_add\co2seq-water-freezing-V1.0.377 benchmarks_new_add\dusty-gas-model-V1.0.303\deviss139 benchmarks_new_add\gas-bubble-V1.0.303\Amos and Mayer 2006\Williams benchmarks_new_add\gas-decay-first-order-V1.0.530\gas-decay-napl benchmarks_new_add\gas-decay-first-order-V1.0.530\gas-nodecay-napl benchmarks_new_add\isotope-V1.0.303\Gibson et al 2011\Waybrant_2 – new benchmarks_new_add\isotope-V1.0.303\Jamieson-Hanes et al 2012\Cr_columns – 3 benchmarks_new_add\transient-boundary-V1.0.303\co2-seq-gacc benchmarks_new_add\uraninite-reoxidation-V1.0.56\2d-run10 benchmarks_new_add\uraninite-reoxidation-V1.0.56\2d-run16 Postprocessing of above examples indicates that the changes in the new results are reasonable.	general-comp.lay in amd_ex example shows difference between new code and old code for mixed boundary condition type

Revision 530

Revision Number	530	**Version Number**	V1.0.530
Commit Date	2017-10-10	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;LIS;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x	LIS in MIN3P_THCm is currently only available on Linux/Unix OS.
System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version for Windows if compiled by intel compiler. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	

Summary of this version	**Remarks**
1. Add first order gas decay to gas component.	
New Features	**Remarks**
1. Add first order gas decay to gas component.	
Usage of New Features	**Remarks**
1. To use first order gas decay, add the following command 'gases decay half time' 1.0e6 ;half time for first gas, unit [s] 1.0e3 ;half time for second gas, unit [s] ... or use decay rate constant instead of half time 'gases decay rate constant' 6.93e-7 ;decay rate constant for first gas, unit [s^-1], equal to ln2/halftime 6.93e-4 ;decay rate constant for first gas, unit [s^-1], equal to ln2/halftime ...	
Codes Update	**Remarks**
None	
Benchmarks Update	**Remarks**
1. benchmarks_new_add\gas-decay-first-order-V1.0.530	Demonstration case, provided by Olivier Atteia
Documentations Update	**Remarks**
Parallelization user manual.	Not updated in the last version
Bug Fixes	**Remarks**
1. Fix bug of dg_lim = r1 in readint_new.F90, should be dg_limaq instead.	
Notes	**Remarks**
None	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm version shows no difference.	

Revision 527

Revision Number	527	**Version Number**	V1.0.527
Commit Date	2017-10-03	**Commit Author**	Danyang Su
Software Grade		Signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;LIS;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x	LIS in MIN3P_THCm is currently only available on Linux/Unix OS.

System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	

Summary of this version	**Remarks**
This version add support for LIS (http://www.ssisc.org/lis/) parallel solver.	
New Features	**Remarks**
1. Add LIS solver to MIN3P, including OpenMP version, MPI version and Hybrid MPI-OpenMP version.	
Usage of New Features	**Remarks**
1. To use LIS solver, set the solver type to 3 in the solver configuration file and then add the LIS solver setting block into the solver configuration file. !> *** !> Block: LIS solver setting !> *** !> Set this parameter if you want to check the result of matrix !> solver with ws209. Only valid if i_solver_type = 0 !> Requirement: Optional if you need to compare the result of !> matrix solver. !LIS: SOLVER TEST WITH WS209 !> Set this command to use the default configuration by LIS. !> If you want to use the default solver configuration or !> if you want to read the configuration from the command line, !> uncomment this command. Otherwise, comment this. LIS: USE DEFAULT CONFIGURATION IN FLOW !> Set LIS solver method for flow problem !> The following methods are supported in this file: !> "bicg" !> "gmres" !> "bicgstab" !> Requirement: Optional if use sequential mode !> Required if use parallel mode !> Input example: !> LIS: KSP TYPE IN FLOW !> gmres !> The default value is bicg LIS: KSP TYPE IN FLOW bicg !> Set LIS preconditioner type for flow problem !> The following methods are supported in this file: !> "none" !> "jacobi" !> "ilu" !> Requirement: Optional if use sequential mode !> Required if use parallel mode !> Input example: !> LIS: PRECONDITIONER TYPE IN FLOW	Set LIS solver parameters in the solver configuration file

```
!>            jacobi
!> The default value is none
LIS: PRECONDITIONER TYPE IN FLOW
none

!> Set LIS solver precision for flow problem
!> the following precision are supported in this file:
!> "double"
!> "quad"
!> Note: double precision operations sometimes require large
!> number of iterations because of the rounding error. Quadruple
!> precision operations can improve this operation. Both
!> matrix and vectors are still double precision.
!> Requirement: Optional if use sequential mode
!>           Required if use parallel mode
!> Input example:
!>            LIS: SOLVER PRECISION IN FLOW
!>            quad
!> The default value is double
LIS: SOLVER PRECISION IN FLOW
double

!> Set the relative convergence tolerance for flow problem
!> The default value is 1.0E-12
LIS: CONVERGENCE TOLERANCE IN FLOW
1.0E-12

!> Set the maximum number of iterations for flow problem
!> The default value is 1000
LIS: MAXIMUM NUMBER OF ITERATIONS IN FLOW
1000

!> Set other options that are not included above
!> Please look into LIS solver manual for detail
!>
!> Requirement: Optional if use sequential mode
!>           Required if use parallel mode
!> Input example:
!>            LIS: SOLVER OPTIONS IN FLOW
!>            -i cg -p ssor
!LIS: SOLVER OPTIONS IN FLOW
!-i cg -p ssor -print mem

!> Set this command to use the default configuration by LIS.
!> If you want to use the default solver configuration or
!> if you want to read the configuration from the command line,
!> uncomment this command. Otherwise, comment this.
LIS: USE DEFAULT CONFIGURATION IN REACTIVE TRANSPORT

!> Set LIS solver method for reactive transport problem
!> The following methods are supported in this file:
```

```
!> "bicg"
!> "gmres"
!> "bicgstab"
!> Requirement: Optional if use sequential mode
!>             Required if use parallel mode
!> Input example:
!>              LIS: KSP TYPE IN REACTIVE TRANSPORT
!>              gmres
!> The default value is bicg
LIS: KSP TYPE IN REACTIVE TRANSPORT
bicg

!> Set LIS preconditioner type for reactive transport problem
!> The following methods are supported in this file:
!> "none"
!> "jacobi"
!> "ilu"
!> Requirement: Optional if use sequential mode
!>             Required if use parallel mode
!> Input example:
!>              LIS: PRECONDITIONER TYPE IN REACTIVE TRANSPORT
!>              jacobi
!> The default value is none
LIS: PRECONDITIONER TYPE IN REACTIVE TRANSPORT
none

!> Set LIS solver precision for reactive transport problem
!> the following precision are supported in this file:
!> "double"
!> "quad"
!> Note: double precision operations sometimes require large
!> number of iterations because of the rounding error. Quadruple
!> precision operations can improve this operation. Both
!> matrix and vectors are still double precision.
!> Requirement: Optional if use sequential mode
!>             Required if use parallel mode
!> Input example:
!>              LIS: SOLVER PRECISION IN REACTIVE TRANSPORT
!>              quad
!> The default value is double
LIS: SOLVER PRECISION IN REACTIVE TRANSPORT
double

!> Set the relative convergence tolerance for reactive transport problem
!> The default value is 1.0E-12
LIS: CONVERGENCE TOLERANCE IN REACTIVE TRANSPORT
1.0E-12

!> Set the maximum number of iterations for reactive transport problem
!> The default value is 1000
LIS: MAXIMUM NUMBER OF ITERATIONS IN REACTIVE TRANSPORT
```

1000 !> Set other options that are not included above !> Please look into LIS solver manual for detail !> !> Requirement: Optional if use sequential mode !> Required if use parallel mode !> Input example: !> LIS: SOLVER OPTIONS IN REACTIVE TRANSPORT !> -i cg -p ssor !LIS: SOLVER OPTIONS IN REACTIVE TRANSPORT !-i cg -p ssor -print mem	
Codes Update	**Remarks**
1. Update in OpenMP preprocessing flags in the code. Put "#ifdef OPENMP" between the whole OpenMP related code. 2. Set value for un-initialized iteration number. 3. Move "Block: Domain decomposition setting" out of PETSc setting. A single block to define domain decomposition is added to the solver configuration file.	
Benchmarks Update	**Remarks**
2. benchmarks_new_add\elder-lis-omp4-V1.0.527	OpenMP parallel version.
Documentations Update	**Remarks**
Parallelization user manual.	
Bug Fixes	**Remarks**
None	
Notes	**Remarks**
None	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference. Comparison of elder-lis-omp4-V1.0.527 to the sequential version shows no difference except round-off error.	

Revision 524

Revision Number	524	**Version Number**	V1.0.524
Commit Date	2018-09-18	**Commit Author**	Mingliang Xie
Software Grade		signature	Date
Verified by	Danyang Su		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x	
System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	

Summary of this version	**Remarks**
This version implemented a new option for the salinity dependent biogenic sulfate reduction (SDSR) model to account the strong dependence of biogenic sulfate reduction reaction on a component (i.e. Cl^{-}) concentration. A new benchmark is added, which is verified through code comparison against PHREEQC.	
New Features	**Remarks**
1. Dependence of biogenic sulfate reduction on the concentration of a component: The model is to simulate the rate of the biogenic sulfate reduction reaction that depends on the salinity (represented by the concentration of Cl^{-}) using the cosine function (i.e. f=cos(a*x+b). The main feature of this this option is that the reaction can be completely stopped when the Cl^{-} concentration is higher than a threshold.	
Usage of New Features	**Remarks**
1. Add the keywords 'salinity dependent reaction rate of minerals' under Data Block 14: initial condition - reactive transport followed by the related parameters: the number of minerals applying this function, name of the mineral(s), 'strong inhibition by one component' - type of relation, number of components (=1), name of the component, the concentration of the component when the reaction rate is the highest (i.e f=1.0) and lowest (i.e. f=0.0), respectively. 2. Example input: 'salinity dependent reaction rate of minerals' 1 ;number of minerals 'ch2o-h2s' ;name of minerals 'strong inhibition by one component' 1 ;type of relation, no. of comp. 'cl-1' ; Name of the component 0.076 ; concentration of the component in mol/L when f=1.0. 1.41 ; concentration of the component in mol/L when f=0.0. This example defines a salinity dependent biogenic sulfate reduction reaction for the mineral ch2o-h2s on the component Cl^{-}, when the concentration of Cl^{-} is ≤0.076 mol/L, the reaction rate for the mineral is the highest; when Cl^{-} concentration is ≥1.41 mol/L, the reaction rate is the lowest.	
Codes Update	**Remarks**
None	
Benchmarks Update	**Remarks**
One benchmark is added under ..\benchmarks_new_add\sulfur-V1.0.431\verification\sul2pqc-cos Note: This benchmark can be run up version V1.0.524, but it is put under the	

folder sulfur-V1.0.431 because the function is just a new option of the salinity dependent sulfate reduction model (SDSR) developed at the version V1.0.431.	
Documentations Update	**Remarks**
The user manual and the verification report	
Bug Fixes	**Remarks**
1. Fix bug in SDSR model. The sarinc should be set to minimum salinity (min_salinity) value in case the calculated salinity value is lower than min_salinity. Otherwise, it could leads to negative sar that should be between 0 and 1, which will cause the termination of the calculation.	
Notes	**Remarks**
none	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference except the following: benchmarks_new_add\sulfur-V1.0.431\2D-sal\ 1. 2D-sal: The difference is caused by the parameter sarinc. The difference in file *_4.gs*, *_4.vel is not big, the maximum error is well below 0.001%.	

Revision 483

Revision Number	483	**Version Number**	V1.0.483
Commit Date	2017-06-26	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x			
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-			

visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
This version fix bug in uninitialized parameter in dusty gas model (DGM) and update the solver convergence check in updatert.	
New Features	**Remarks**
None	
Usage of New Features	**Remarks**
None	
Codes Update	**Remarks**
None	
Benchmarks Update	**Remarks**
None	
Documentations Update	**Remarks**
Parallelization user manual.	
Bug Fixes	**Remarks**
1. Fix bug in DGM model. The variable beta_diff is not initialized in dgm_dfluxdg.F90 when 'pressure dependence' is not used, resulting in incorrect entry in the matrix. 2. Check the solver convergence in updatert.F90 by adding the check the maximum update value. The previous version will check the residual only when maximum linear iteration are reached, if the maximum update meet convergence criteria but the maximum iteration number is used, it will treat this as unconvergenced. Causing unnecessary timestep reducing.	
Notes	**Remarks**
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference except the following: 1. amd505, bem600disp and bem20510e-2 for DGM related cases: The difference is caused by the uninitialized parameter. The difference is not big for the these three cases. 2. d43_density_dep_heat_solute, amd_ex and perm3_new: The difference is caused by the solver convergence check in updatert. Case amd_ex has	

generally recongnizable difference at the beginning as it is more sensible to the solver. Given more strict convergence parameter will diminish difference. The difference in the other two cases is unidentified in the plot.	

Revision 474

Revision Number	474	**Version Number**	V1.0.474
Commit Date	2017-06-05	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x			

System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
The version add support for PC factor shift in the configuration file. Fix bug for very low mineral rate and fix bug in reactive transport code when there is no update in reactive transport equation.	
New Features	**Remarks**
1. Add support for PC factor shift in the code for flow and reactive transport problem, the previous only allowed this option from command line.	
Usage of New Features	**Remarks**

1. Add the following section in the solver configuration file for both flow and reactive transport configuration

```
!>
!> Set PC Factor type for flow problem
!> The following methods are supported in PETSc:
!> MAT_SHIFT_NONE                 "none"
!> MAT_SHIFT_NONZERO              "nonzero"
!> MAT_SHIFT_POSITIVE_DEFINITE    "positive_definite"
!> MAT_SHIFT_INBLOCKS             "inblocks"
!>
PETSC: PC FACTOR SHIFT TYPE IN FLOW
None

!>
!> Set PC Factor type for reactive transport problem
!> The following methods are supported in PETSc:
!> MAT_SHIFT_NONE                 "none"
!> MAT_SHIFT_NONZERO              "nonzero"
!> MAT_SHIFT_POSITIVE_DEFINITE    "positive_definite"
!> MAT_SHIFT_INBLOCKS             "inblocks"
!>
PETSC: PC FACTOR SHIFT TYPE IN REACTIVE TRANSPORT
nonzero
```

Codes Update	**Remarks**

1. Change the default solver parameters in the configuration file to the same as PETSc default, that is 1.0d-50 for absolute residual tolerance and 50000 for the maximum linear iteration. 2. Change the restart timestep back to minimum timestep, unless indicated by user with maximum timestep.	
Benchmarks Update	**Remarks**
none	
Documentations Update	**Remarks**
Parallelization user manual.	
Bug Fixes	**Remarks**
1. Fix bug in low rate < 1.0d-300, previous skip some calculation for this value but this is not strict. 2. Fix bug in updatert.F90. Treat zero update as converged.	
Notes	**Remarks**
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference except the following: 1. Cases starts from restart file; 2. *.gss, *.gsd files in d51_dedolomitization and other cases related to the bug fixed. 3. For those with numerical difference, the plot shows the difference is very small and insignificant in the postprocessing.	

Revision 464

Revision Number	464	**Version Number**	V1.0.464
Commit Date	2017-04-14	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.x, PETSc V3.6.x, PETSc V3.7.x			

System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
This version includes support to PETSc 3.7.x, bug fix in mass balance report of dusty gas model (DGM) and bug fix in the structure of reaction matrix block when redox is considered.	
New Features	**Remarks**
1. Add support to PETSc 3.7.x	
Usage of New Features	**Remarks**
1. Add preprocessing definition PETSC_V3_7_X when compiling the code.	
Codes Update	**Remarks**
none	
Benchmarks Update	**Remarks**
none	
Documentations Update	**Remarks**
Parallelization user manual.	
Bug Fixes	**Remarks**
1. Fix bug in mass balance report when dusty gas model (DGM) is used. The output of advection is put at the end of mac file in the previous version, which cause misunderstanding of mass balance. The header of mac has also been changed to separated gas diffusion and gas advection when DGM is used. This bug fix will results into difference in .mac file and .fls file. 2. Fix bug in reaction matrix block structure when redox is considered. Connection of components to the redox secondary component may be lost as the connection to the redox primary component is not correctly used in the	

previous version. This bug fix will after V1.0.167 and thereafter, if default block matrices is used.	
Notes	**Remarks**
The bug fix in matrix block structure solve the convergence slow problem in perm3_new.	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference except the following: *.mac files in dusty gas model (DGM) related benchmarks: dusty-gas-model-V1.0.303, gas_advection-V1.0.175 and transient-boundary-V1.0.303; all results in benchmark perm3_new, there is no significant difference if viewing the results from post processing software, but the bug will significant slow the convergence.	

Revision 459

Revision Number	459	**Version Number**	V1.0.459
Commit Date	2017-03-21	**Commit Author**	Mingliang Xie
Software Grade		signature	Date
Verified by	Danyang Su		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.2, PETSc V3.6.2	

System Requirements	Remarks
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
This version includes some bug fixes. To 3. This is not a bug. Even if ratemp(ireac,tid) is a local parameter, it corrects the calculated rates to 0.0 if a mineral disappears. MX Added two examples verified against PHREEQC.	
New Features	**Remarks**
none	
Usage of New Features	**Remarks**
none	
Codes Update	**Remarks**
none	
Benchmarks Update	**Remarks**
1. Added two verification examples against PHREEQC under .\benchmarking\benchmarks_new_add\ sulfur-V1.0.431\verification\	
Documentations Update	**Remarks**
1. Updated the verification/validation manual for the added benchmarks	
Bug Fixes	**Remarks**
1. Reverted the bug fix in version v1.0.433 for the output file *.gsd and revised the treatment of mineral dissolution using the parallel mineral approach for the salinity dependent sulfate reduction reaction. Consequently, the *.gsd files will be identical to those by V1.0.414,	

which is verified.	
Notes	**Remarks**
none	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference except the following: All the output files *.gsd are different. This is because of the bug fixed mentioned above. Another comparison against the results by the version V1.0.414 showed no differences of all output files including the *.gsd files, except *.gen and *.log files.	

Revision 433

Revision Number	433	**Version Number**	V1.0.433
Commit Date	2016-07-26	**Commit Author**	Mingliang Xie
Software Grade		signature	Date
Verified by	Danyang Su		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.089 or compatible compiler 2. Configurations for Windows 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict			√ **DS**

<table>
<tr><td>3. Configurations for Linux
3.1 Preprocess source file: Yes
3.2 Preprocessor definitions for sequential version:
LINUX_X64|LINUX
3.3 Preprocessor definition for parallel version:
RELEASE_X64;LINUX;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED
3.4 Additional Options: N/A
4. External Library
4.1 PETSc: PETSc V3.4.4, PETSc V3.5.2, PETSc V3.6.2</td><td></td></tr>
<tr><td><b>System Requirements</b></td><td><b>Remarks</b></td></tr>
<tr><td>For Windows user with Intel Fortran Compiler, Intel Visual Fortran Redistributable library is required to run the parallel version on Windows for the OpenMP version.
The redistributable library can be downloaded via the URL
http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.
For Linux user with Intel Fortran Compiler, Intel Visual Fortran Redistributable library is required to run the parallel version on Linux for the OpenMP version.
The redistributable library can be downloaded via the URL
https://software.intel.com/en-us/articles/redistributable-libraries-for-the-</td><td></td></tr>
</table>

intel-c-and-fortran-composer-xe-2013-for-linux For both Windows and Linux user: Parallel versions of MIN3P-THCm, including OpenMP version, MPI version and hybrid OpenMP-MPI version, are system specific that the executable file may malfunction or cause unexpected results if the code is compiled on one system but run on another system with different software environment or hardware environment. It is highly recommend to compile the parallel code on the local machine that will be used to run the code.	
Summary of this version	**Remarks**
This branch includes the implementation of the salinity dependent sulfur reducing bacteria (SRB) reaction model and some bug fixes.	
New Features	**Remarks**
1. Added the salinity dependent sulfur reducing bacteria (SRB) reaction model	
Usage of New Features	**Remarks**
1. Add the keywords 'salinity dependent reaction rate of minerals' to block 'initial condition - reactive transport' followed by the number of the mineral(s) to be considered for the model; 2. Add the name of the first mineral plus the keyword ‘equation’, and the number and the values of the coefficients (f_i) of a function to calculate the salinity inhibition factor k_{sal} (in [-]) depending on the salinity S (in [g/L]) based on the experimental data: $$k_{sal} = \sum_{i=0}^{n} f_i S^i$$ 3. Add the minimum salinity & related k_{sal}, and the maximum salinity & related k_{sal} for the first mineral to ensure no negative k_{sal} will be calculated according to the above equation; 4. If more than one minerals are considered, follow the steps 2 and 3 for the rest of the minerals; 5. The other properties of the minerals (e.g. kinetics) remain unchanged. 6. Example:	

```
! Data Block 15: initial condition - reactive transport
! ------------------------------------------------------------------------
!
'initial condition - reactive transport'
3                        ;number of zones


! ------------------------------------------------------------------------
'number and name of zone'
1
'inflow'


'concentration input'            ; From Hobbs et al. (2011), Table A-3,
Sample ID: CFN-161
6000.00         'free'    'ca+2'
10900.00        'charge'         'na+1'
2800.00         'free'    'mg+2'
450.00 'free'    'k+1'
31000.00        'free'    'cl-1'
1240.00         'free'    'so4-2'
5.95    'ph'     'h+1'
303.00 'free'    'co3-2'
1.0d-10         'free'    'hs-1'
-200.00         'eh'      'o2(aq)'


'mineral input'                 ;bottom
1.00d-10  .false.   'constant' ;ch2o-h2s
```

```
1.00d-10  6.9d-9     0.00d0    ;
1.00d-10  .false.   'constant' ;calcite
1.00d-10  4.0d-8     0.00d0    ;
0.50d-10  .false.   'constant' ;anhydrite
0.50d-10  4.0d-9     0.00d0    ;
0.50d-10  .false.   'constant' ;halite
0.50d-10  1.16d-8   0.00d0    ;


'salinity dependent reaction rate of minerals'
1                           ;number of minerals
ch2o-h2s                    ;name of minerals
'equation' 4                ;type of relation (=equation), no of parameters
-2.716E-01     2.11E-02        -9.455E-05     -3.20E-08     ;coefficients
15.0     3.0E-02               ; minimum salinity [g/L], k_sal [-]
225.0   0.046e-4            ; maximum salinity [g/L], k_sal [-]


'extent of zone'
0.0 20.0  0.0 1.0  0.0 1.0


'end of zone'
```

This example applied the model for one mineral ch2o-h2s using the function:

$$k_{sal} = -0.2.716 + 0.0211S - 9.455\mathrm{E} - 5S^{2} - 3.20E - 8S^{3}$$

The minimum salinity is 15.0 g/L and the k_{sal} is 0.03, which means if the

salinity lower than 15.0 g/L, 0.03 will be used for the calculation. The maximum salinity is 225.0 g/L, and the k_{sal} is 4.6e-6. When the salinity is higher than 225.0 g/L, the k_{sal} is 4.6e-6.	
Code Updates	**Remarks**
1. Added a file dratemin_salinity.F90 2. Updated all the project configuration files for windows OS. 3. Updated all makefiles for linux/unix OS.	
Benchmarks Update	**Remarks**
1. Added one demonstration example sulfur for the new model under the folder .\benchmarking\benchmarks_new_add\ sulfur-V1.0.431\ 2. Added six validation benchmarks from the Mt Terri projects under the folder .\benchmarking\benchmarks_new_add\mtterri-V1.0.431	Verification examples needed. UM Added two examples verified against PHREEQC MX
Documentations Update	**Remarks**
1. Updated the user manual for the new model 2. Updated the theory manual for the new model 3. Updated the verification/validation manual for the added benchmarks	
Bug Fixes	**Remarks**
1. Fix bug of using restart file for hybrid multicomponent model (hmc). The activity coefficients at the old timestep was not initialized. In the old version, these coefficients will be calculated because of a bug changing the update type from 'time_lagged' to 'double_update' subroutine in file gcreact.F90 but not changing back, which was fixed in version V1.0.303. 2. Fix bug of calculating the secondary species concentrations totcnewf and totcoldf. 3. Fix bug of mineral dissolution rate output in file outputrt.F90. Previous version writing out the ratemp(ireac,tid), which is a parameter for local geochemical system. Now it is replaced with the parameter ratemdp(im,ivol). This change will not influence the calculation, but the results in *.gsd will be different.	To 3. This is not a bug. The parameter ratemp(ireac,tid) considers the parallel reactions. Will be retrieved in the next version. MX
Notes	**Remarks**
The maximum number for the total species for output was 100, which might be not enough under complex geochemical conditions.	

Consequently, only results for the first 100 species will be written in *.gsc file. Fortunately, this has no impact on the calculation. Now it is changed to 1000.	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference except the following: Some of the output files *.gsd are different. This is because of the bug fixed mentioned above.	
Tested Platform and version type	**Remarks**
Microsoft Windows [Version 6.2.9200] on the workstation stargazer2012 MIN3P-THCm_V1.0.433 sequential version X64 against MIN3P-THCm_V1.0.414 sequential version X64	
Tested examples	**Remarks**
The tested examples are listed in the APPENDIX.	
Detailed descriptions of difference (if any)	**Remarks**
1. *.gsd file: Previous version produces the dissolution rates are 0.0 if there is the mineral's volume fraction is 0.0. After the code modification, the rates calculated using the aqueous solution composition are provided.	

Revision 402

Revision Number		**Version Number**	V1.0.402
Commit Date	2016-02-05	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**

<table>
<tr><td>1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3
2. Configurations
2.1 Preprocess source file: Yes (/fpp)
2.2 Preprocessor definitions for sequential version:
RELEASE_X64|RELEASE_WIN32;WINDOWS
2.3 Preprocessor definition for parallel version:
RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED
2.4 Preprocessor definition for resources:
RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P
2.5 Additional Options: /fp:strict
3. External Library
3.1 PETSc: PETSc V3.4.4, PETSc V3.5.2, PETSc V3.6.2</td><td></td></tr>
<tr><td><b>System Requirements</b></td><td><b>Remarks</b></td></tr>
<tr><td>Intel Visual Fortran Redistributable library is required to run the parallel version.
The redistributable library can be downloaded via the URL
http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.</td><td></td></tr>
</table>

Summary of this version	**Remarks**
This branch includes two important updates for the complex reactive transport simulation: 1. Add direct solver ill-conditioned matrices. These cases (e.g., basin simulation with density dependent flow, thermal transport and ice loading/unloading) have features that the matrices are well conditioned in the beginning, but ill-conditioned later. Solving ill-conditioned matrix is difficult, which usually takes a lot of iterations for iterative solver. Direct solver is preferrable for ill-conditioned matrix. In MIN3P-THCm, two direct solver package, MUMPS and SuperLU, are available. 2. Modify ice loading/unloading formulation by changing the boundary	
New Features	**Remarks**
1. 1. Add PETSc KSP runtime options to the parallel version. Now user can use the parallel version without configuration file, just use the PETSc runtime options. 2. Add direct solver norm check (optional). 3. Add timestep control when ice loading/unloading is considered. The previous code does not consider the timeline of ice moving, which may cause timestep spaning two or more ice moving stages. 4. Add maximum boundary pressure update caused by ice sheet loading/unloading for each timestep. This makes ice loading/unloading more gently, using linear interpolation. 5. Add maximum internal pressure update caused by ice sheet loading/unloading for each timestep. This makes ice loading/unloading more gently, using linear interpolation.	
Usage of New Features	**Remarks**
1. PETSc KSP runtime options in MIN3P-THCm are grouped by flow KSP prefix "flow_" and reactive transport KSP "react_". To use PETSc runtime options only, first comment the command "parallel solver configuration file" in the input file. Then use the standard PETSc KSP runtime options together with the prefix. For example, to use PETSc SuperLU solver, the standard PETSc runtime option is: '-pc_type lu -pc_factor_mat_solver_package superlu_dist' or '-pc_type lu -pc_factor_mat_solver_package superlu' For MIN3P-THCm, if you want to use SuperLU for flow problem but default solver for reactive transport problem, the runtime option is:	

'-flow_pc_type lu -flow_pc_factor_mat_solver_package superlu_dist' If you want to use MUMPS solver for flow problem but SuperLU solver for reactive transport problem, the runtime option is: '-flow_pc_type lu -flow_pc_factor_mat_solver_package mumps -react_pc_type lu -react_pc_factor_mat_solver_package superlu_dist' 2. Add "PETSC: CHECK DIRECT SOLVER NORM IN FLOW" or "PETSC: CHECK DIRECT SOLVER NORM IN REACTRAN TRANSPORT" to solver configuration file or use runtime option "-ksp_check_norm_flow" or "-ksp_check_norm_react" for flow problem or reactran transport problem. Example input in solver configuration file is shown below: !> Check direct solver norm in flow problem. If this is set, !> the direct solver will use the value set in !> 'ABSOLUTE CONVERGENCE TOLERANCE IN FLOW' !> to check if the solution meets the convergence criteria. PETSC: CHECK DIRECT SOLVER NORM IN FLOW 4. Add the following commands to block 'ice sheet loading/unloading'. Please be note that the unit is Pa. 'maximum ice loading boundary pressure update' 1.0d5 ;maximum pressure update for each timestep 5. Add the following commands to block 'ice sheet loading/unloading'. Please be note that the unit is Pa. 'maximum ice loading internal pressure update' 1.0d5 ;maximum pressure update for each timestep	
Codes Update	**Remarks**
1. Update PETSc default matrix option to '-mat_superlu_dist_fact SamePattern' when SuperLU is used. The previous default option 'SamePattern_SameRowPattern' maycause SuperLU failure when the matrix	

is ill-conditioned. 2. Some solver configuration has been updated. 3. Add restart file check for parallel version.	
Benchmarks Update	**Remarks**
Documentations Update	**Remarks**
1. N/A	
Bug Fixes	**Remarks**
1. Fix bug of 'call hydrostatic_pitzerdens' for the parallel version. This subroutine calculate pressure from the top nodes to the bottom nodes. For the previous parallel version, it only calculates the pressure from the local top nodes without considerring the global top nodes. The modified version will passing the uper subdomain pressure to the lower subdomain by MPI message communication.	
Notes	**Remarks**
1. Some test code (comment out) are embedded. 2. The previous code adds or release ice loading in one timestep, regardless of the current timestep increment. This causes the boundary value changes sharply (e.g., increase hydraulic head by more than 1000 m) in one single timestep (e.g., 2.5 years). Now the code considers the maximum update in boundary hydraulic head change (e.g., 10 m) caused by ice loading/unloading for each timestep. If the maximum update exceeds the threshold, then timestep should be reduced. In this case, the boundary pressure change is accumulated until it reaches the maximum value. The results may be a little difference before this maximum value is reached (e.g., in the first 5 years when ice loading/unloading takes), but should be no big difference in the long run. This could significantly reduce the condition number of the matrices and the parallel version can successfully run the simulations.	
Summary of code verification (test running)	**Remarks**
There is no difference compared to the previous internal version V1.0.385 except the following: *.mac, *.gsc, *.gsp, *.gst and *.vel. The difference is caused by the code modification in ice loading, as	

mentioned above.	

Revision 382

Revision Number	382	**Version Number**	V1.0.382
Commit Date	2015-11-14	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.2, PETSc V3.6.2			
System Requirements			**Remarks**

Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
New Features	**Remarks**
1. Add support of direct solver (superlu and mumps) for parallel version.	
Usage of New Features	**Remarks**
1. Edit the solver configuration file 'solver.cfg' by commenting out of "PETSC: USE DEFAULT CONFIGURATION IN FLOW (or reactive ...)" and changing the "PETSC: KSP TYPE IN FLOW (or reactive ...)" to "superlu" or "mumps".	
Codes Update	**Remarks**
Benchmarks Update	**Remarks**
The following benchmarks inputs have been updated with modification of maximum iteration and residual norm. benchmarking_nwmo_report\nwmo_verification_examples\d32_elder_problem\elder.dat benchmarking_nwmo_report\nwmo_verification_examples\d311_hydrostatic_box_test\coupled\diersch.dat benchmarks_standard\flow\shlomo\shlomo.dat	
Documentations Update	**Remarks**
1. N/A	
Bug Fixes	**Remarks**

1. Fix bug of temperation correction when energy balance is considered. 2. Fix bug of using inaccurate linear solution. Check the convergence when the maximum linear iteration is reached.	
Notes	**Remarks**
Summary of code verification (test running)	**Remarks**
There is no difference compared to the previous internal version V1.0.381.	

Revision 377

Revision Number	377	**Version Number**	V1.0.377
Commit Date	2015-10-23	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Software Grade		**Verified by**	Mingliang Xie

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library	

3.1 PETSc: PETSc V3.4.4, PETSc V3.5.2	
System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
Summary of branch dsu_freezing (revision 316 to 323, 372 to 374): Add water freezing/thawing processing; add water removal for minerals; add reaction rate control for frozen water.	
New Features	**Remarks**
1. Add water freezing/thawing processing 2. Add water removal for mineral. This will affect the flux calculation. 3. Add reaction rate control for frozen water.	
Usage of New Features	**Remarks**
1. Usage of feature 1: Add the following commands to 'Data Block 6B: control parameters - energy balance' 'water freezing temperature' -1.0d-5 !Temperature below this value is considered frozen 'water freezing conductivity' 1.0d-20 !Conductivity for frozen water 2. Usage of feature 2: Add 'mineral water removal coefficient' together with 'minerals' section in the input file.	

e.g., 'minerals' 3 ;number of minerals ! 'chrysotileeq' 'artinite' !Mg2(CO3)(OH)2·3H2O 'nesquehonite' !Mg(HCO3)(OH)·2H2O 'mineral water removal coefficient' 0.0 3.0 2.0 3. Add the following commands to 'Data Block 6B: control parameters - energy balance' 'water freezing reaction rate' 1.0d-20	
Codes Update	**Remarks**
Benchmarks Update	**Remarks**
benchmarks_new_add\co2seq-water-freezing-V1.0.377	
Documentations Update	**Remarks**
1. N/A	
Bug Fixes	**Remarks**
Notes	**Remarks**
Summary of code verification (test running)	**Remarks**
There is no difference compared to the previous internal version V1.0.373.	

Revision 371

Revision Number	371	**Version Number**	V1.0.371
Commit Date	2015-10-09	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.2			
System Requirements			**Remarks**

Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
Summary of branch dsu_usg_flow (revision 313 to 369): Add support for PETSc 3.6.x; add matrix and rhs output at specific timestep, with support of new format;fix bug of initial conditions (ccnew vector) before start of Newton Raphson process; fix bug of minmaxwd for MPI version.	
New Features	**Remarks**
1. Output matrix and rhs at specific timestep. 2. Set previous solution as initial guess for the next solve and reuse preconditioner. The two features can be used independently. This is usually effective when the linear systems do not change very much between successive steps. By default, this feature is not used.	
Usage of New Features	**Remarks**
1. Add 'output sparse matrix data set and rhs at timestep' followed by the timestep (new line) to the solver configuration file or 'global control parameters' in the input file. For PETSc matrix output, add "output sparse matrix data set and rhs using petsc" to the solver configuration file. 2. Add the following commands to the PETSc solver configuration file. !> Set previous solution as initial guess for the next solve !> for flow problem. This is usually effective when the linear !> systems do not change very much between successive steps. !> This is not controlled by DEFAULT CONFIGURATION. !> The default value is to set initial guess to zero.	

```
PETSC: KSP SET INITIAL GUESS NONZERO IN FLOW


!> Set operators, keeping the identical preconditioner matrix
!> for all linear solves. This approach is often effective when
!> the linear systems do not change very much between successive
!> steps.
!> This is not controlled by DEFAULT CONFIGURATION.
!> The default value is NOT reusing the same preconditioner.
PETSC: KSP REUSE PRECONDITIONER IN FLOW


!> Set previous solution as initial guess for the next solve
!> for reactive transport problem. This is usually effective when the linear
!> systems do not change very much between successive steps.
!> This is not controlled by DEFAULT CONFIGURATION.
!> The default value is to set initial guess to zero.
PETSC: KSP SET INITIAL GUESS NONZERO IN REACTIVE TRANSPORT


!> Set operators, keeping the identical preconditioner matrix
!> for all linear solves. This approach is often effective when
!> the linear systems do not change very much between successive
!> steps.
!> This is not controlled by DEFAULT CONFIGURATION.
!> The default value is NOT reusing the same preconditioner.
PETSC: KSP REUSE PRECONDITIONER IN REACTIVE TRANSPORT
```

Or

use the following commands in the command lines: -ksp_initial_guess_nonzero_flow -ksp_reuse_preconditioner_flow -ksp_initial_guess_nonzero_react -ksp_reuse_preconditioner_react	
Codes Update	**Remarks**
1. Add Metis library for unstructured grid version 2. Add support for PETSC 3.6.x	
Benchmarks Update	**Remarks**
Documentations Update	**Remarks**
1. N/A	
Bug Fixes	**Remarks**
1. Fix bug of initial conditions (ccnew vector) before start of Newton Raphson process. 2. Fix bug of minmaxwd for MPI version. Call minmaxwd_mpi after this procedure	
Notes	**Remarks**
Summary of code verification (test running)	**Remarks**
There is no difference compared to the previous internal version V1.0.367	

Revision 303

Revision Number	303	**Version Number**	V1.0.303
Commit Date	2015-05-04	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.2	
System Requirements	**Remarks**

Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
Merge isotope model and gas bubble model from Richards's standard version; Merge full dusty gas (DGM) model from Molin's standard version; Add timestep update for full saturated flow problem; Improve transient boundary implementation; Fix a lot of bugs.	
New Features	**Remarks**
1. Merge isotope model from Richard's standard version 2. Merge gas bubble model from Richard's standard version 3. Merge full dusty gas model from Molin's standard version. ***Other New Features*** 1. Add timestep update for full saturated flow problem. 2. Add spatial weighting of boundary condition interpolation method for transient boundary condition. This value is between 0 and 1. If 0, use the linear interpolation for the transient boundary. If 1, use the previous updated boundary condition as the current boundary condition. Genearlly, 0.0d0 is prefered and 1.0 is apt to cause occillation. By default, 1 is used as spatial weithting so as to backward compatible. 3. Add transient boundary function method. Transient function is a function of simulation time t and it is a 4-parameter sine function or polynomial function.	

3.1 Sine function: f(t) = A*SIN(B*t+C)+D, where B*t+C is radian 3.2 Polynomial function: f(t) = A*t^3+B*t^2+C*t+D The transient boundary function works for 'concentration input', 'guess for ph' and 'scaling for intra-aqueous kinetic reactions' in Section 15: 'boundary conditions - reactive transport'. If you want to use transient boundary function for the specific zone, add 'use function for transient boundary input' to Section 15. The first parameter is function type, 'polynomial' for polynomial function and 'sine' for sine function. The sectond to fifth parameters are parameters A to D for sine function and polynomial function. The last parameter is component type. If the boundary condition value is constant, set A, B and C to 0 and D to the constant 4. Add linear interpolation for transient boundary conditions. All the transient boundary conditions are read only once (improve performance) when simulation starts and then updated every time step. This is also a bug fix for the transient boundary as the previous version may cause oscillation for transient boundary. 5. Add temperature correction for solution update to avoid extreme small or large temperature. 6. Add mole concentration correction for solution update to avoid extreme small or large concentration. 7. Add dynamic control of log cycle update for reactive and energy solution update to avoid non-physical variables.	
Usage of New Features	**Remarks**
Please see the relative manual for isotope model, gas bubble model and full DGM model. ***Usage of Other New Features***	

2. Example input of Feature 2. Add the following section to 'Section 15: boundary conditions - reactive transport' for each zone.

```
'spatial weighting of boundary condition interpolation'

0.0d0
```

3. Example input of Feature 3. Add the following commands to boundary setting.

```
!!Input format example

'boundary conditions - reactive transport'

2                         ;number of zones

'number and name of zone'

1

'influx boundary'

'boundary type'

'first'


'use function for transient boundary input'


'concentration input'

'polynomial'  0.0d0     0.0d0     0.0d0  1.0d-4      'free'  ;'function type'  A  B  C  D  'ctype'; 'cl-1'

'polynomial'  0.0d0     0.0d0     0.0d0  1.0d-4      'free'  ;'function type'  A  B  C  D  'ctype'; 'na+1'

'polynomial'  0.0d0     0.0d0     0.0d0  5.6d0       'ph'    ;'function type'  A  B  C  D  'ctype'; 'h+1'

'sine'        0.025d0   1.571d0   0.0d0  1.00d0      'pn2'   ;'function type'  A  B  C  D  'ctype'; 'n2(aq)'

'polynomial'  0.0d0     0.0d0     0.0d0  1.248d-4    'free'  ;'function type'  A  B
```

```
C  D  'ctype'; h4sio4


    'guess for ph'

    'polynomial'  0.0d0     0.0d0     0.0d0  7.0


    'extent of zone'

    0.0 1.0  0.0 1.0  20.0  20.0


    'end of zone'


    ! ------------------------------------------------------------------------


    'number and name of zone'

    2

    'bottom'

    'boundary type'

    'second'


    'extent of zone'

    0.0 1.0  0.0 1.0  0.0 0.0


   'end of zone'
```

5. To use this feature, add 'temperature correction for solution' followed by a new line with tempcorr_min and tempcorr_max value (Celsius) in block 'control parameters - energy balance'. By default, this correction is not activated and the above two values are set to extreme small or large.

e.g., 'temperature correction for solution' -273.0 1.0d4 6. To use this feature, add 'concentration correction for solution' followed by a new line with moleconcorr_min, moleconcorr_max in block 'control parameters - reactive transport'. By default, this correction is not activated and the above two values are set to extreme small or large. e.g., 'concentration correction for solution' 1.0d-20 2.0d1 7. Add error check code for MPI and PETSc routines.	
Codes Update	**Remarks**
1. Update in parameters of bulkconc, always use "bulkconc(new parms) - bulkconc(old parms)". This will cause slight difference in some simulations but will get much better mass balance.	
Benchmarks Update	**Remarks**
1. Add benchmarks_new_add\co2-seq-ultramafic-tailing-V1.0.303 2. Add benchmarks_new_add\dusty-gas-model-V1.0.303 3. Add benchmarks_new_add\gas-bubble-V1.0.303 4. Add benchmarks_new_add\isotope-V1.0.303	
Documentations Update	**Remarks**
1. N/A	
Bug Fixes	**Remarks**
1. Fix bug of file description (.gsa, .gsga) 2. Fix bug of mineral area calculation in cliqdisp and its relative subroutine.	

Area of mineral is changed after this calling. 3. Fix bug of mineral database variables order. E.g., 'inhibition phi^m' 1 'mno2-c7h8m' 1.000d-6 1.000 Some of the database will written in 'inhibition phi^m' 1 1.000d-6 1.000 'mno2-c7h8m' 4. Fix bug of update_reactivity change in gcreact.F90. Revert the update_activity type after running initial, boundary condition and batch simulation. The update_activity in gcreact is only used for the calculation of the initial condition and boundary conditions or batch simulations. In any case, in reactive transport, the code should use 'time_lagged', for the initial and boundary condition, it should use 'double update'. 5. Fix bug of extreme large relative permeability for the gas phase that cause extreme value in gas flux. 6. Fix bug in mass balance output when gas advection is included. The order of output variables (only exists when using ascii format) is not correct.	
Notes	**Remarks**
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference except following benchmarks: *.gsc, *.gsp, *.gst, *.mac, *.mve, *.mvs, *.mas and *.vel in the benchmark of d42_density_dep_energy. The difference was caused by the bug fixing as mentioned above.	

Revision 268

Revision Number	268	**Version Number**	V1.0.268
Commit Date	2015-01-09	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4, PETSc V3.5.2			
System Requirements			**Remarks**

Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
Summary of this version	**Remarks**
Port code to linux with compatible to different system and compiler. Add input command arguments, mass balance check, binary output, control of enable/disable output and control of restart file.	
New Features	**Remarks**
1. Add input command arguments, most of the parameters in parallel configuration file solver.cfg now can be read from the command arguments. 2. Add relative mass balance tolerance to convergence check for flow simulation. 3. Add openmp parallel section preprocessing definition 'PARALLEL_SECTION', if this flag is not used in compiling, the OpenMP parallel section is not activated. 4. Add make project for shared-memory version(OpenMP) and hybrid-distributed-shared memory version(MPI+OpenMP) 5. Add thread number option '-threadcomm_nthreads' and '-mpicomm_nthreads' at runtime. Use together with 'GLOBAL: USE NUMBER OF THREADS FROM MPI' in solver configuration file 'solver.cfg' 6. Add error information output when the input command does not meet the requirement, e.g., (1) vanleer spatial weighting requires stencil width 2 or larger, (2) GFortran cannot read non-standard string such as "str"ing. 7. Add binary input and output for restart file. 8. Add 'disable output of all gen files' to disable output of gen information. 9. Add 'enable output of master gen file' to output of gen information for the master thread.	

10. Add binary output for spatial and transient output. 11. Add restart file read and write methods for different data format and storage types. 12. Add support for combining the zones in binary output for spatial data result. This is only valid when 'use binary format' is enabled and 'use separated file for spatial subdomain output' is disabled. 13. Add output precision control. Both double precision and single precision float point are supported in binary output. 14. Port codes to Linux with compatible to Intel Fortran, GFortran and IBM XL Compilers.	
Usage of New Features	**Remarks**
1. All the commands start with "-" and may be followed by an argument. Usage of input command: '-command_name' or '-command_name command_data', e.g., ./program.exe -input_file file_name -solver_type_flow 0 -solver_type_react 0 -numofthreads_global 4 2. Add 'mass balance convergence tolerance settings' followed by relative tolerance and absolute tolerance error to data block 'control parameters - variably saturated flow'. By default, the tolerance is set to a large value. E.g., 'mass balance convergence tolerance settings' 1.0d-6 !relative mass balance convergence tolerance 1.0d-8 !absolute mass balance convergence tolerance 3. Add 'PARALLEL_SECTION' to preprocessing definition. By default, this is not used. 4. N/A 5. Set 'GLOBAL: USE NUMBER OF THREADS FROM MPI' in solver configuration file or use command argument '-use_numofthreads_from_mpi'	

together with arguments '-threadcomm_nthreads' and '-mpicomm_nthreads' E.g., mpiexec -n 4 .\executable.exe -input_file file_name -use_numofthreads_from_mpi -mpicomm_nthreads 6. This will run the program using 4 processors, each processor will use 6 threads. Totally 24 cores will be used. 6. N/A 7. Add the following commands to "Data Block 1: global control parameters" 'restart' !Indicate whether to run from the restart file. 'use distributed restart files to read' !Indicate whether to use distributed restart files to read. 'use distributed restart files to write' !Indicate whether to use distributed restart files to write. 'use binary restart file to read' !Indicate whether to use binary restart file to read. 'use binary restart file to write' !Indicate whether to use binary restart file to write. 'backup frequency' !Indicate the backup frequency. 100 !Backup frequency 100 indicate write restart files every 100 time steps. 8. Add 'disable output of all gen files' to 'Data Block 1: global control parameters' 9. Add 'enable output of master gen file' to 'Data Block 1: global control parameters' 10. Add the following section to the input data file.	

```
! Data Block 8: output control

! ------------------------------------------------------------------------

!

'output control'

'use binary format'
```

11. Add the following section to the input file 'Data Block 1: global control parameters'

```
'restart'

'use distributed restart files to read'    !restart files are distributed stored

'use distributed restart files to write'   !restart files are distributed stored

'use binary restart file to read'          !restart files are in binary format

'use binary restart file to write'         !restart files are in binary format
```

12. By default, the binary output combines the multi-block zones (sub-domains) into one zone for spatial result output.

If you want to use multi-block zones output, add the command 'use multizone data format for spatial output' to 'Data Block 8: output control'.

E.g.,

```
'output control'

'use binary format'                              !use binary format

'use separated file for spatial subdomain output' !files are separated by subdomains

'use multizone data format for spatial output'    !use multi-block zones output
```

13. Add "OUTPUT_DOUBLE" to the preprocessor definition if "double precision output" is needed.

14. Add preprocessor definitions "STANDARD_FORTRAN", "PETSC_V3.5.X" in consistence with the system environment. #ifdef STANDARD_FORTRAN 1 format(15x,100a15) 2 format(e15.5,100e15.7) #else 1 format(15x,<ncol>a15) 2 format(e15.5,<ncol>e15.7) #endif	
Codes Update	**Remarks**
1. Disable binary I/O in sequential version and OpenMP version. Binary output in sequential version and OpenMP version is not perfect. 2. Change output control for runtime. The global output control 'disable output' does not have any effect on runtime statistics output. 3. Change file name solver/solver_results.f90 to solver/solver_results.F90. 4. Update the openmp scheduling method in makefile to static for openmp version and hybrid mpi-openmp version 5. Modify makefile to exclude PETSc variables for sequential version and OpenMP version 6. Modify the number of maximum opening files to 65536, previous value is 4096. 7. Add convergence parameters output to screen and log file for Parallel version. 8. Update the output of gas advection in *.mac file, move the gas advection terms to the end. 9. Update to use MSHPC2012 and "PETSc for Windows 3.5.2" 10. Remove temporary storage (*.cnv, *.tmp) when the program terminates. 11. Replace the tab with four space ' ' in the codes. Tab-format lines are not part of the Fortran standard. It is an extension to Compaq/Intel's compiler. It is	

unlikely that other Fortran will support this format well. Usually, the compiler (e.g., GFortran) will give out "nonconforming tab character" warning.	
Benchmarks Update	**Remarks**
1. N/A	
Documentations Update	**Remarks**
1. N/A	
Bug Fixes	**Remarks**
1. Fix bug of race condition in infcrt_g.F90 and mbalrt.F90. This bug is introduced when gas advection is added. 2. Fix bug of uninitialized variable and array bound problem in OpenMP version. Need further consideration of array length n and nc when gas advection or compositional model is used. 3. Fix bug of debug build error and binary write when the output format is ascii format. 4. Fix bug of gas advection support for density dependent flow. Fix bug of .gsk output (gas advection) in OpenMP version. Change gas advection output extention name from .gsa to gsga. 5. Fix bug of unassigned values in vsflow when energy balance is considered for non-density-dependent flow. This bug exist for all MIN3P-THCm versions. It was first detected in the case co2_seq (CO2 sequestration in ultramafic tailings, Anna Harrison). 6. Fix bug of mass balance output (.mcd) for mcd problems. In code mbal_mcd.F90, the variables cfluxin_diff, cfluxin_mig, cfluxout_diff and cfluxout_mig should only be applied to boundary type first. 7. Fix bug of infinity value when divided by a small value. This bug exists in	

WestGrid Cluster.

8. Fix bug of restart simulation for transient source and updated boundary condition. Restart failed when the restart time is later than the target time of updating boundary condition.

9. Fix bug of gas component connection when 'sparse block matrices' option is used. For the previous version, this connection is considered for gas advection. But for the case 'nickel', this should also be considered.

10. Fix bug of file unit allocation. The previous codes will cause file unit conflicts (ifort.xxx files) when the number of output of transient increases. Form this version, a file unit manage module is included.

11. Fix bug of 'driving force gradient' output in *.gtk files for diffusion problem. The previous output is always zero. This bug occurs when gas advection is used.

12. Fix bug of NAN output in viscosity output in _0.gsy as the mole fraction of gas phase is not calculated for the initial output, use zero instead.

13. Fix bug of uninitialization array when compiled using gfortran.

14. Fix bug of duplicated or missing tecplot header (*.mv*, *.mm*, *.ma*, *.ms*, *.mg*, *.mic, etc.

15. Fix bug of NaN value in vapor transport (.evap)

16. Fix bug of the output of *.hyc for the parallel version. Change to output local nodes only, without ghost nodes.

17. Fix bug of temp_n and visco_n output that are missing in full saturated flow when heat transport is considered in outputdd.F90 . 18. Fix bug of velocity output for density dependent flow, the output includes 9 variables, not 6 variables. 19. Fix bug of velocity output in velocity_a.F90, velocity_g.F90. Zone information is missing in the previous output. 20. Fix bug of initopgs.F90 when 'coordinate output' is used. This bug exists in parallel version only. 21. Fix bug in restart_r.F90. Call secspec(c, cx(ix,ivol), ...) should be call secspec(c(1,ivol), cx(ix,ivol), ...). This bug occurs in nwmo_basin simulations.	
Notes	**Remarks**
Summary of code verification (test running)	**Remarks**
Based on the bug fixes and updates in the code, most of the results in benchmarks are affected. We have compared the results through post processing software (tecplot) and the difference is reasonable.	

Revision 221

Revision Number	221	**Version Number**	V1.0.221
Commit Date	2014-08-05	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4			

System Requirements	Remarks
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
New Features	**Remarks**
1. No new features	
Usage of New Features	**Remarks**
1.	
Codes Update	**Remarks**
1. Make declaration of all the variables. No implicit declaration is used any more.	
Benchmarks Update	**Remarks**
1.	
Documentations Update	**Remarks**
1.	
Bug Fixes	**Remarks**
1. Fix bugs in some un-defined variables.	
Notes	**Remarks**
The following variables are not set in the source codes 1. i2upfind_heat.F90: rhalf, rho_z 2. initcpevaporation.F90: stabfac_atm	
Summary of code verification (test running)	**Remarks**
There is no difference compared to the previous version except the following:	

velocity results *.vel and *.velvap in most of the benchmarks; *.gbc, *.gbt, *.gsc and *.gst files in benchmark diff_harmonic. The difference is caused by the use of undefined /initialized varialbes.	

Revision 216

Revision Number	216	**Version Number**	V1.0.216
Commit Date	2014-07-14	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict 3. External Library 3.1 PETSc: PETSc V3.4.4	
System Requirements	**Remarks**

Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
New Features	**Remarks**
1. Add domain decomposition method for flow and reactive transport simulation. 2. Add control 'disable output' to disable global output for parallel performance test. 3. Add support to read and write restart file for MPI parallel version. 4. Modify dispersive/diffusive flux for mixed boundary condition. By default, dispersive/diffusive flux is applied from the boundary cells across the surface.	
Usage of New Features	**Remarks**
1. Set the solver type to 2 and configure "Block: PETSc solver setting" in solver configuration file. !> *** !> Block: PETSc solver setting !> *** !> Set this parameter if you want to check the result of matrix !> solver with ws209. Only valid if i_solver_type = 0 !> Requirement: Optional if you need to compare the result of !> matrix solver. PETSC: SOLVER TEST WITH WS209	

```
!> Stencil width
!> By default, the stencil width is 1, which can meet the requirements
!> of most cases, but for the case with "vanleer" spatial weighting,
!> it will consider the second upstream point for flux limiter that the
!> stencil width should be 2 or more.
PETSC: STENCIL WIDTH
2

!> Set this command to use the default configuration by PETSc.
!> If you want to use the default solver configuration or
!> if you want to read the configuration from the command line,
!> uncomment this command. Otherwise, comment this.
PETSC: USE DEFAULT CONFIGURATION IN FLOW

!> Set PETSc Krylov method for flow problem
!> The following methods are supported in PETSc:
!> KSPGMRES     "gmres"
!> KSPBCGS      "bcgs"
!> Requirement: Optional if use sequential mode
!>             Required if use parallel mode
!> Input example:
!>              PETSC: KSP TYPE IN FLOW
!>              kspgmres or gmres
!> The default value is gmres
PETSC: KSP TYPE IN FLOW
```

```
kspgmres


!> Set PETSc preconditioner type for flow problem
!> The following methods are supported in PETSc:
!> Algorithm        Name       Parallel
!> PCNONE            "none"        N
!> PCJACOBI          "jacobi"      Y
!> PCLU             "lu"          N
!> PCBJACOBI         "bjacobi"     Y
!> PCILU            "ilu"         N
!> PCASM             "asm"         Y
!> PCKSP             "ksp"         Y
!> PCHYPRE           "hypre"       Y
!> Requirement: Optional if use sequential mode
!>              Required if use parallel mode
!> Input example:
!>               PETSC: PRECONDITIONER TYPE IN FLOW
!>               pcbjacobi or bjacobi
!> The default value is bjacobi
PETSC: PRECONDITIONER TYPE IN FLOW
pcbjacobi


!!> Set norm type
!!>  KSP_NORM_NONE - skips computing the norm, this should only be used if you
!!>              are using the Krylov method as a smoother with a fixed small number
!!>              of  iterations.
!!>              Implicitly sets KSPSkipConverged as KSP convergence test.
!!>              Supported only by CG, Richardson, Bi-CG-stab, CR, and CGS methods.
!!>  KSP_NORM_PRECONDITIONED - the default for left preconditioned solves,
!!>              uses the l2 norm of the preconditioned residual
!!>  KSP_NORM_UNPRECONDITIONED - uses the l2 norm of the true b - Ax
!!>              residual, supported only by CG, CHEBYSHEV, and RICHARDSON,
!!>              automatically true for right (see KSPSetPCSide()) preconditioning..
!!>  KSP_NORM_NATURAL - supported  by KSPCG, KSPCR, KSPCGNE, KSPCGS
!!PETSC: KSP NORM TYPE IN FLOW
!!ksp_norm_preconditioned


!> Set KSP solver convergence criteria
!> By default, KSP solver will reach convergence when
!> rnorm < MAX (rtol * rnorm_0, abstol) or divergence if
!> rnorm > dtol * rnorm_0. The rnorm here is preconditioned
!> residual norm. If '-ksp_norm_type unpreconditioned' is used,
!> rnorm is the true residual norm.
!> If the solver convergence criteria is set to user-defined
!> criteria, then the true residual will be calculated every
!> iteration. The solver will reach convergence when
!> rnorm < MAX (rtol * rnorm_0, abstol).
!> The following methods are supported in PETSc:
!> KSPDEFAULT       "default"
!> KSPUSERDEFINED    "userdefined"
!> Requirement: Optional if use sequential mode
!>              Required if use parallel mode
!> Input example:
!>               PETSC: KSP CONVERGENCE CRITERIA TYPE IN FLOW
!>               kspdefault or default
!> The default value is kspdefault
PETSC: KSP CONVERGENCE CRITERIA TYPE IN FLOW
kspdefault



!> Set the relative convergence tolerance for flow problem
!> The default value is 1.0E-5
PETSC: RELATIVE CONVERGENCE TOLERANCE IN FLOW
1.0E-5
!> Set the absolute convergence tolerance for flow problem
!> The default value is 1.0E-20
PETSC: ABSOLUTE CONVERGENCE TOLERANCE IN FLOW
1.0E-20
!> Set the divergence tolerance for flow problem
!> The default value is 1.0E5
PETSC: DIVERGENCE TOLERANCE IN FLOW
1.0E5
!> Set the maximum number of iterations for flow problem
!> The default value is 1000
PETSC: MAXIMUM NUMBER OF ITERATIONS IN FLOW
100

!> Set this command to use the default configuration by PETSc.
!> If you want to use the default solver configuration or
!> if you want to read the configuration from the command line,
!> uncomment this command. Otherwise, comment this.
PETSC: USE DEFAULT CONFIGURATION IN REACTIVE TRANSPORT

!> Set PETSc Krylov method for reactive transport problem
!> The following methods are supported in PETSc
!> KSPGMRES      "gmres"
!> KSPBCGS       "bcgs"
!> Requirement: Optional if use sequential mode
!>              Required if use parallel mode
!> Input example:
!>               PETSC: KSP TYPE IN REACTIVE TRANSPORT
!>               kspgmres or gmres
!> The default value is gmres
PETSC: KSP TYPE IN REACTIVE TRANSPORT
kspgmres

!> Set PETSc preconditioner type for reactive transport problem
!> The following methods are supported in PETSc:
!> Algorithm          Name       Parallel
!> PCNONE          "none"        N
!> PCJACOBI        "jacobi"      Y
!> PCLU            "lu"          N
!> PCBJACOBI       "bjacobi"     Y
!> PCILU           "ilu"         N
!> PCASM           "asm"         Y
!> PCKSP           "ksp"         Y
!> PCHYPRE         "hypre"       Y
!> Requirement: Optional if use sequential mode
!>              Required if use parallel mode
!> Input example:
!>              PETSC: PRECONDITIONER TYPE IN REACTIVE TRANSPORT
!>              pcbjacobi or bjacobi
!> The default value is bjacobi
PETSC: PRECONDITIONER TYPE IN REACTIVE TRANSPORT
pcbjacobi


!!> Set norm type
!!>  KSP_NORM_NONE - skips computing the norm, this should only be used if you
!!>               are using the Krylov method as a smoother with a fixed small number
!!>               of  iterations.
!!>               Implicitly sets KSPSkipConverged as KSP convergence test.
!!>               Supported only by CG, Richardson, Bi-CG-stab, CR, and CGS methods.
!!>  KSP_NORM_PRECONDITIONED - the default for left preconditioned
```

```
solves,
   !!>              uses the l2 norm of the preconditioned residual
   !!>  KSP_NORM_UNPRECONDITIONED - uses the l2 norm of the true b - Ax
   !!>              residual, supported only by CG, CHEBYSHEV, and RICHARDSON,
   !!>              automatically true for right (see KSPSetPCSide()) preconditioning..
   !!>  KSP_NORM_NATURAL - supported  by KSPCG, KSPCR, KSPCGNE, KSPCGS
   !!PETSC: KSP NORM TYPE IN REACTIVE TRANSPORT
   !!ksp_norm_preconditioned


   !> Set KSP solver convergence criteria
   !> By default, KSP solver will reach convergence when
   !> rnorm < MAX (rtol * rnorm_0, abstol) or divergence if
   !> rnorm > dtol * rnorm_0. The rnorm here is preconditioned
   !> residual norm. If '-ksp_norm_type unpreconditioned' is used,
   !> rnorm is the true residual norm.
   !> If the solver convergence criteria is set to user-defined
   !> criteria, then the true residual will be calculated every
   !> iteration. The solver will reach convergence when
   !> rnorm < MAX (rtol * rnorm_0, abstol).
   !> The following methods are supported in PETSc:
   !> KSPDEFAULT       "default"
   !> KSPUSERDEFINED    "userdefined"
   !> Requirement: Optional if use sequential mode
   !>          Required if use parallel mode

   !> Input example:

   !>          PETSC: KSP CONVERGENCE CRITERIA TYPE IN REACTIVE TRANSPORT

   !>          kspdefault or default

   !> The default value is kspdefault

   PETSC: KSP CONVERGENCE CRITERIA TYPE IN REACTIVE TRANSPORT

   kspdefault


   !> Set the relative convergence tolerance for reactive transport problem

   !> The default value is 1.0E-5

   PETSC: RELATIVE CONVERGENCE TOLERANCE IN REACTIVE TRANSPORT

   1.0E-5

   !> Set the absolute convergence tolerance for reactive transport problem

   !> The default value is 1.0E-20

   PETSC: ABSOLUTE CONVERGENCE TOLERANCE IN REACTIVE TRANSPORT

   1.0E-20

   !> Set the divergence tolerance for reactive transport problem

   !> The default value is 1.0E5

   PETSC: DIVERGENCE TOLERANCE IN REACTIVE TRANSPORT

   1.0E5

   !> Set the maximum number of iterations for reactive transport problem

   !> The default value is 1000

   PETSC: MAXIMUM NUMBER OF ITERATIONS IN REACTIVE TRANSPORT
```

```
  100
```

2. Add 'disable output' to "Data block 1: global control parameters"

3. When domain decomposition method is used, the restart file will be exported by different processors, with '_rank_n' as the indicator of the parallel restart file, e.g., restart_rank_0.tmp1, restart_rank_1.tmp1, ... The number of the restart files (number of processors) is also exported in the header of each restart file.If the restart data is a single file, you can run with the restart file without any change. If the restart data is distributed stored, e.g., restart_rank_0.tmp1, restart_rank_1.tmp1, ..., change the file extension file to .dat. Add the command 'distributed restart' to 'Data Block 1: global control parameters'. The number of processor used should be the same as the number of restart files, otherwise, there will be error showing that the number of restart file does not match.

4. Add 'exclude diffusive flux from mixed boundary condition' to 'Data Block 16: boundary conditions - reactive transport'.

e.g.,

```
 ! Data Block 16: boundary conditions - reactive transport

 ! -------------------------------------------------------------------------

 !

 'boundary conditions - reactive transport'

 2                              ;number of zones


 ! -------------------------------------------------------------------------

 'number and name of zone'

 1

 'inflow boundary'
```

'boundary type' 'mixed' 'exclude diffusive flux from mixed boundary condition' 'concentration input' 1.0000d-3 'free' ;cl-1 1.0000d-3 'free' ;na+1 1.8700d-3 'free' ;o2(aq) ... !Note: When dispersive/diffusive flux is applied from the boundary cells, the results (*.gsc) of third boundary condition and mixed boundary condition cannot match. Exclude dispersive/diffusive flux for mixed boundary condition will solve this problem.	
Codes Update	**Remarks**
1. Change the matrix extension file name from '.mtp' to '.mtx', the latter is the default suffix for MtxView software.	
Benchmarks Update	**Remarks**
1.	
Documentations Update	**Remarks**
1.	
Bug Fixes	**Remarks**
1.	
Notes	**Remarks**
Bad convergence benchmarks benchmarks_new_add\biomass-V1.0.56 batch1 - parallel	

benchmarks_new_add\biomass-V1.0.56 batch2 – parallel ***Deprecated benchmarks*** Benchmarks_new_add\hMCD-V1.0.137. This benchmark is not backward compatible.	
Summary of code verification (test running)	**Remarks**
There is no difference compared to the previous version.	

Revision 203

Revision Number	203	**Version Number**	V1.0.203
Commit Date	2014-06-17	**Commit Author**	Mingliang Xie
Software Grade		signature	Date
Verified by	Danyang Su		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 12.1.5 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict	
System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version.	

The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
New Features	**Remarks**
1. Development of the hybrid multicomponent diffusion (hMCD) model.	
Usage of New Features	**Remarks**
1. Add keyword 'hybrid multicomponent diffusion' in data Block 1: global control parameters to activate hMCD model. 2. The correction factors of effective porosity and tortuosity for each primary and secondary species should be provided to each material group in Data Block 11: physical parameters - reactive transport. If no data is provided, default value of 1.0 will be applied, which means no modification. 3. The correction factors of effective porosity of primary species are provided through the keyword ‘porosity correction factor of primary species for hMCD diffusion’ followed by the factors of each primary species in the same order as the components defined in Data Block 2: geochemical system under 'components'. 4. The correction factors of effective porosity of secondary species are provided through the keyword ‘porosity correction factor of secondary species for hMCD diffusion’ followed by the factors of each secondary species in the same order as the species defined in Data Block 2: geochemical system under 'secondary aqueous species'. 5. The correction factors of effective tortuosity of primary species are provided through the keyword ‘tortuosity correction factor of primary species for hMCD diffusion’ followed by the factors of each primary species in the same order as the components defined in Data Block 2: geochemical system under 'components'. 6. The correction factors of effective tortuosity of secondary species are provided through the keyword ‘tortuosity correction factor of secondary species for hMCD diffusion’ followed by the factors of each secondary species in the same order as the species defined in Data Block 2: geochemical system under 'secondary aqueous species'. 7. Example: The following example showed a 1D diffusion in radial coordinate. There are four materials – the circulation chamber, filter, gap and clay (‘background 1’). The hMCD model is applied only for the clay. The number of primary species is 17, while the number of secondary species is 23.	

```
! Data Block 11: physical parameters - reactive transport
! --------------------------------------------------------------------------
!
'physical parameters - reactive transport'

! --------------------------------------------------------------------------
'circulation chamber'

'longitudinal dispersivity'
0.0
'transverse horizontal dispersivity'
0.0
'transverse vertical dispersivity'
0.0
'end of zone'

! --------------------------------------------------------------------------
!
'filter'
'longitudinal dispersivity'
0.0
'transverse horizontal dispersivity'
0.0
'transverse vertical dispersivity'
0.0
```

```
'end of zone'


! -------------------------------------------------------------------------
!
'gap'
'longitudinal dispersivity'
0.0
'transverse horizontal dispersivity'
0.0
'transverse vertical dispersivity'
0.0
'end of zone'


! -------------------------------------------------------------------------
!
'background 1'
'longitudinal dispersivity'
0.0
'transverse horizontal dispersivity'
0.0
'transverse vertical dispersivity'
0.0


'porosity correction factor of primary species for hMCD diffusion'
  1.0    ;'cs+'
  1.0    ;'na+1'
  1.0    ;'k+1'

  1.0    ;'mg+2'

  1.0    ;'ca+2'

  1.0    ;'sr+2'

  0.667  ;'cl-1'

  1.0    ;'h+1'

  1.0    ;'o2(aq)'

  0.667  ;'so4-2'

  0.667  ;'co3-1'

  0.533  ;'i-1' =0.08/0.15, effective porosity of I- is 0.08

  0.667  ;'br-1' 0.1/0.15, effective porosity of Br- is 0.10

  1.0    ;'HTO'

  1.0    ;'co_60+2'

  1.0    ;'eu+3'

  1.0    ;'sr_85+2'

  1.0    ;'=soh(s)'

  1.0    ;'=soh(w)'

  1.0    ;'=toh'


'tortuosity correction factor of primary species for hMCD diffusion'

  11.1    ;'cs+'

   1.0    ;'na+1'

   1.0    ;'k+1'

   1.0    ;'mg+2'

   1.0    ;'ca+2'

   1.0    ;'sr+2'
   1.0    ;'cl-1'

   1.0    ;'h+1'

   1.0    ;'o2(aq)'

   1.0    ;'so4-2'

   1.0    ;'co3-1'

  0.55    ;'i-1'

  0.38    ;'br-1'

   1.0    ;'HTO'

  3.0     ;'co_60+2'

   1.0    ;'eu+3'

   2.8    ;'sr_85+2'

   1.0    ;'=soh(s)'

   1.0    ;'=soh(w)'

   1.0    ;'=toh'


'porosity correction factor of secondary species for hMCD diffusion'

  1.0             ;'khso4(aq)'

  1.0             ;'h2so4(aq)'

  1.0       ;'oh-'

  1.0             ;'mgoh+'

  1.0             ;'mgco3aq'

  1.0             ;'mghco3+'

  1.0             ;'mgso4aq'

  1.0             ;'caoh+'

  1.0             ;'cahco3+'

  1.0             ;'caco3aq'
```

```
1.0             ;'caso4aq'

1.0             ;'cahso4+'

1.0       ;'naco3-'

1.0             ;'nahco3aq'

1.0       ;'naso4-'

1.0       ;'kso4-'

1.0             ;'sroh+'

1.0       ;'hco3-'

1.0             ;'h2co3aq'

1.0       ;'hso4-'

1.0             ;'cscl(aq)'

1.0             ;'csi(aq)'

1.0             ;'csbr(aq)'


'tortuosity correction factor of secondary species for hMCD diffusion'

1.0             ;'khso4(aq)'

1.0             ;'h2so4(aq)'

1.0             ;'oh-'

1.0             ;'mgoh+'

1.0             ;'mgco3aq'

1.0             ;'mghco3+'

1.0             ;'mgso4aq'

1.0             ;'caoh+'

1.0             ;'cahco3+'

1.0             ;'caco3aq'

1.0             ;'caso4aq'
```

1.0 ;'cahso4+' 1.0 ;'naco3-' 1.0 ;'nahco3aq' 1.0 ;'naso4-' 1.0 ;'kso4-' 1.0 ;'sroh+' 1.0 ;'hco3-' 1.0 ;'h2co3aq' 1.0 ;'hso4-' 1.0 ;'cscl(aq)' 1.0 ;'csi(aq)' 1.0 ;'csbr(aq)' 'end of zone' 'done'	
Codes Update	**Remarks**
1. Update in chemical matrix to optimize the connections.	
Benchmarks Update	**Remarks**
1. One benchmark was added to the current version to demonstrate the hMCD model. 2. The file location of the example is: .\Dropbox\NWMO-Projects\MIN3P-THCm\Benchmark\benchmarks_new_add\ hMCD-V1.0.203\	
Documentations Update	**Remarks**
1. Updated theory manual 2. Updated user manual	

3. Updated verification manual (Mont Terri in-situ experiments)	
Bug Fixes	**Remarks**
none	
Notes	**Remarks**
1.	
Summary of code verification (test running)	**Remarks**
Tested benchmarks are listed in the APPENDIX. Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference except the following: 1. The benchmarks under the folder \benchmarks_new_add\biomass-V1.0.56\: the number of connections considered are slightly different, which results in insignificant difference in some files *.gbs, *.gbt, *.gsc, *.gsd, *.gss, *.mas. Most difference is in the last digit, well below 0.001%. 2. The benchmark degas under the folder benchmarks_standard\reactran\degas: the number of connections considered are slightly different, 1988 for the current version, 1920 for the previous version (v1.0.191). The timesteps differ slightly (as showed in *.dt), which causes the difference in all transient output files *.gb*, *.ma*, *.mgc, *.mms, _o.m**. However, the results are identical as long as the values reaches stable and do not change with the time. Some spatial output files (e.g. *.gsc, *.gsg, *.gsm, *.gss, *.gst, *.gsgr, showed also slight difference (in the last one or two digits), well below 0.001%, indicating machine precision difference.	

Revision 191

Revision Number	191	**Version Number**	V1.0.191
Commit Date	2014-04-28	**Commit Author**	Mingliang Xie
Software Grade		signature	Date
Verified by	Danyang Su		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 12.1.5 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict			
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version.			

The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
New Features	**Remarks**
1. User specified unit for the surface complexation.	
Usage of New Features	**Remarks**
1. Add keyword 'specify output unit for SCM sorbed species concentration' in data block 2: geochemical system after the keyword 'define sorption type' and parameter 'surface-complex'. The keyword has only effect on the output unit for the SCM sorbed species concentration. 2. The default unit for the sorbed species concentration is 'mol/L H2O'. Optional unit is 'mol/L bulk', which is more suitable for unsaturated porous media. 3. Example: 'define sorption type' 'surface-complex' 'specify output unit for SCM sorbed species concentration' 'mol/L bulk' !optional unit !'mol/L H2O' ! default unit	
Codes Update	**Remarks**
1.	
Benchmarks Update	**Remarks**
1. Two benchmarks were added to the current version to demonstrate the sorption processes (e.g. ion exchange, surface complexation) modelling in unsaturated porous media. 2. One is the ion exchange in unsaturated porous media which is initially saturated. Through drainage, the soil is de-saturated. It can be found under: .\Dropbox\NWMO-Projects\MIN3P-THCm\Benchmark\benchmarks_new_add\surfx-ionx-unsat-V1.0.191\ionx-drain1\ 3. The other one daaaeals with surface complexation in unsaturated	

porous media which is initially saturated. Through drainage, the soil is de-saturated. It can be found under: .\Dropbox\NWMO-Projects\MIN3P-THCm\Benchmark\benchmarks_new_add\surfx-ionx-unsat-V1.0.191\surfx-drain1\	
Documentations Update	**Remarks**
1. Updated the user manual	
Bug Fixes	**Remarks**
1.noneasas	
Notes	**Remarks**
1.	
Summary of code verification (test running)	**Remarks**
No difference in the results obtained by the current version to those by the version V1.0.184, except the updates in file description file .fls.	

Revision 184

Revision Number	184	**Version Number**	V1.0.184
Commit Date	2014-04-17	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict			
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version.			

The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
New Features	**Remarks**
1. Remove the optional parameters that used for OpenMP version. The shared variables are indexed by rank id, so as to reduce the overhead in parallel mode.	
Usage of New Features	**Remarks**
1.	
Codes Update	**Remarks**
1.	
Benchmarks Update	**Remarks**
1.	
Documentations Update	**Remarks**
1.	
Bug Fixes	**Remarks**
1. Fix bug of duplicated boundary condition setting. If the control volume is included in two or more different boundary zones, give out error information. See the example biomass-V1.0.56. This bug maybe not activated in Release mode. 2. Fix bug in bdryflux.F90, pointer is out of upper bound in array iabvs. This bug maybe not activated in Release mode.	
Notes	**Remarks**
1.	
Summary of code verification (test running)	**Remarks**
No difference in the results obtained by the current version to those by the version V1.0.177.	

Revision 177

Revision Number	177	**Version Number**	V1.0.177
Commit Date	2014-04-10	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict			
System Requirements			**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version.			

The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
New Features	**Remarks**
1. Add gas advection item to MIN3P. The gas advection item is ported from Dust Gas Model (DGM). 2. Add Openmp parallel control parameters for jacobi assembly in reactive transport.	
Usage of New Features	**Remarks**
1. Usage of Feature 1 1.1 Add the following options to 'Section 7: control parameters - reactive transport' to invoke the following functions: 'gas advection' !- Include gas advection 'cumulative mole fractions' !- Enable cumulative gas pressure value in the output .gsg. 'enable gravity for gas phase' !- Enable gravity for the gas phase, if enabled, grad = grad - dens * gacc * (zgj - zgi), where gpi and gpj are the gas pressure at the control volume i and j. 1.2. Add the following options to 'Section 11: physical parameters - reactive transport' to invoke the following functions: 'update gas density' !- Enable the option to update the gas density. 'constant gas density' !- Use constant gas density. This commands	

should be followed by a new line with the gas density.

```
    E.g., 'constant gas density'
          1.29d+0
```

'wilke viscosity'   !- Use Wilke expression to compute gas viscosity value

```
    E.g.,  'wilke viscosity'
           2.04d-5        ;viscosity of gas component A
           1.46d-5        ;viscosity of gas component B
           1.75d-5        ;viscosity of gas component C
```

'linear viscosity'   !- Use linear expression to compute gas viscosity value

```
    E.g.,  'linear viscosity'
           2.04d-5        ;viscosity of gas component A
           1.46d-5        ;viscosity of gas component B
           1.75d-5        ;viscosity of gas component C
```

'constant viscosity'   !- Use constant gas viscosity value

```
    E.g.,  'constant viscosity'
           1.80d-5        ;viscosity of gas phase
```

2. Usage of Feature 2

Add the commands of parallel jacobi assembly for jacrt and jacbrt to "Block: Matrix assembly setting" in parallel configuration file.

E.g.,

MATRIX ASSEMBLY: TYPE IN REACTIVE TRANSPORT 1 MATRIX ASSEMBLY: TYPE IN REACTIVE TRANSPORT JACRT 0 !Set jacrt to sequential version MATRIX ASSEMBLY: TYPE IN REACTIVE TRANSPORT JACBRT 1 !Set jacbrt to parallel version	
Codes Update	**Remarks**
1. Update the default matrix export format to matrix market exchange format.	
Benchmarks Update	**Remarks**
1. Add benchmark “benchmarks_new_add\gas_advection-V1.0.175”	
Documentations Update	**Remarks**
1.	
Bug Fixes	**Remarks**
1. Fix bug in compiling error for the sequential version in debug mode. De-active the OpenMP function in this version. 2. Fix bug in update concentrations of surface sites, ulcmax = ulcmax/enat to ulcmax = ulcmax/dlog(r10) 3. Fix bug in sparse block matrices when dissolution-precipitation reactions are included. 4. Fix bug in updtbcatm.F90 when there are duplicated data in *.atm data set. 5. Fix bug in subroutine updtsvgp, new optional parameter eqg is added.	
Notes	**Remarks**
1. For the current version, the default block matrix format has been changed to 'sparse block matrices'. For some cases, if the solver does not converge well (with a lot of failing iterations), please increase the value of incomplete factorization level. By default, level 0 is used. Increasing this value to 1 or 3 will get better convergence.	
Summary of code verification (test running)	**Remarks**

The following difference are observed when comparing the current version to the previous version: *.mgc file in benchmark biomass-V1.0.56, uraninite-reoxidation-V1.0.56, amd-ex, degas, diffdry, diffvar, diffwet, pyrox and raoult; results in benchmark het_2d; The difference is caused by the use of ‘sparse block matrices’, where the solution is numerically not exactly the same as that from ‘dense block matrices’. The bug fixes as abovementioned also affect some of the benchmarks.	

Revision 167

Revision Number	167	**Version Number**	V1.0.167
Commit Date	2014-03-12	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PETSC;MPI;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict	

System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version.	

The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
New Features	**Remarks**
1. Add PETSc parallel solver. 2. Set 'sparse block matrices' as the default option. 3. Add support to export matrix in Matrix Market (MM) exchange format.	
Usage of New Features	**Remarks**
1. The following commands are used to configure PETSc solver. !> ** !> Block: Golbal solver setting !> ** !> This command indicates that the input parameters in the MIN3P input !> file have the priority over the input parameters in the parallel !> solver configuration file. !> By default, the parameters in this file will overwrite the same !> parameters in the MIN3P input file. !> For example, if there are solver convergence parameters in both !> files, you can use the following command if you want to use MIN3P !> input parameters first. !> Requirement: Optional !> GLOBAL: USE MIN3P INPUT PARAMETERS FIRST	

```
!>
!> Use number of threads from MPI calling.
!> If this command is enabled, the number of threads
!> in OpenMP calling will be replaced by the number
!> of processors in mpi calling.
GLOBAL: USE NUMBER OF THREADS FROM MPI


!> *********************************************************
!>            Block:  PETSc solver setting
!> *********************************************************
!> Set this parameter if you want to check the result of matrix
!> solver with ws209. Only valid if  i_solver_type = 0
!> Requirement: Optional if you need to compare the result of
!>            matrix solver.
PETSC: SOLVER TEST WITH WS209


!> Set PETSc Krylov method for flow problem
!> The following methods are supported in PETSc:
!> KSPGMRES      "gmres"
!> KSPBCGS       "bcgs"
!> Requirement: Optional if use sequential mode
!>            Required if use parallel mode
!> Input example:
!>             PETSC: KSP TYPE IN FLOW
!>             kspgmres or gmres
!> The default value is gmres
```

```
PETSC: KSP TYPE IN FLOW
kspgmres


!> Set PETSc preconditioner type for flow problem
!> The following methods are supported in PETSc:
!> Algorithm          Name       Parallel
!> PCNONE             "none"         N
!> PCJACOBI           "jacobi"       Y
!> PCLU               "lu"           N
!> PCBJACOBI          "bjacobi"      Y
!> PCILU              "ilu"          N
!> PCASM              "asm"          Y
!> PCKSP              "ksp"          Y
!> Requirement: Optional if use sequential mode
!>              Required if use parallel mode
!> Input example:
!>               PETSC: PRECONDITIONER TYPE IN FLOW
!>               pcbjacobi or bjacobi
!> The default value is bjacobi
PETSC: PRECONDITIONER TYPE IN FLOW
pcbjacobi


!> Set KSP solver convergence criteria
!> By default, KSP solver will reach convergence when
!> rnorm < MAX (rtol * rnorm_0, abstol) or divergence if
!> rnorm > dtol * rnorm_0. The rnorm here is preconditioned
!> residual norm. If '-ksp_norm_type unpreconditioned' is used,
!> rnorm is the true residual norm.
!> If the solver convergence criteria is set to user-defined
!> criteria, then the true residual will be calculated every
!> iteration. The solver will reach convergence when
!> rnorm < MAX (rtol * rnorm_0, abstol).
!> The following methods are supported in PETSc:
!> KSPDEFAULT        "default"
!> KSPUSERDEFINED    "userdefined"
!> Requirement: Optional if use sequential mode
!>              Required if use parallel mode
!> Input example:
!>              PETSC: KSP CONVERGENCE CRITERIA TYPE IN FLOW
!>              kspdefault or default
!> The default value is kspdefault
PETSC: KSP CONVERGENCE CRITERIA TYPE IN FLOW
kspdefault


!> Set the relative convergence tolerance for flow problem
!> The default value is 1.0E-5
PETSC: RELATIVE CONVERGENCE TOLERANCE IN FLOW
1.0E-5
!> Set the absolute convergence tolerance for flow problem
!> The default value is 1.0E-20
PETSC: ABSOLUTE CONVERGENCE TOLERANCE IN FLOW
```

```
1.0E-20
!> Set the divergence tolerance for flow problem
!> The default value is 1.0E5
PETSC: DIVERGENCE TOLERANCE IN FLOW
1.0E5
!> Set the maximum number of iterations for flow problem
!> The default value is 1000
PETSC: MAXIMUM NUMBER OF ITERATIONS IN FLOW
100

!> Set PETSc Krylov method for reactive transport problem
!> The following methods are supported in PETSc
!> KSPGMRES      "gmres"
!> KSPBCGS       "bcgs"
!> Requirement: Optional if use sequential mode
!>              Required if use parallel mode
!> Input example:
!>               PETSC: KSP TYPE IN REACTIVE TRANSPORT
!>               kspgmres or gmres
!> The default value is gmres
PETSC: KSP TYPE IN REACTIVE TRANSPORT
kspgmres

!> Set PETSc preconditioner type for reactive transport problem
!> The following methods are supported in PETSc:
!> Algorithm          Name       Parallel
!> PCNONE          "none"        N
!> PCJACOBI        "jacobi"      Y
!> PCLU            "lu"          N
!> PCBJACOBI       "bjacobi"     Y
!> PCILU           "ilu"         N
!> PCASM           "asm"         Y
!> PCKSP           "ksp"         Y
!> Requirement: Optional if use sequential mode
!>             Required if use parallel mode
!> Input example:
!>              PETSC: PRECONDITIONER TYPE IN REACTIVE TRANSPORT
!>              pcbjacobi or bjacobi
!> The default value is bjacobi
PETSC: PRECONDITIONER TYPE IN REACTIVE TRANSPORT
pcbjacobi


!> Set KSP solver convergence criteria
!> By default, KSP solver will reach convergence when
!> rnorm < MAX (rtol * rnorm_0, abstol) or divergence if
!> rnorm > dtol * rnorm_0. The rnorm here is preconditioned
!> residual norm. If '-ksp_norm_type unpreconditioned' is used,
!> rnorm is the true residual norm.
!> If the solver convergence criteria is set to user-defined
!> criteria, then the true residual will be calculated every
!> iteration. The solver will reach convergence when
!> rnorm < MAX (rtol * rnorm_0, abstol).

!> The following methods are supported in PETSc:

!> KSPDEFAULT       "default"

!> KSPUSERDEFINED    "userdefined"

!> Requirement: Optional if use sequential mode

!>             Required if use parallel mode

!> Input example:

!>              PETSC: KSP CONVERGENCE CRITERIA TYPE IN REACTIVE TRANSPORT

!>              kspdefault or default

!> The default value is kspdefault

PETSC: KSP CONVERGENCE CRITERIA TYPE IN REACTIVE TRANSPORT

kspdefault



!> Set the relative convergence tolerance for reactive transport problem

!> The default value is 1.0E-5

PETSC: RELATIVE CONVERGENCE TOLERANCE IN REACTIVE TRANSPORT

1.0E-5

!> Set the absolute convergence tolerance for reactive transport problem

!> The default value is 1.0E-20

PETSC: ABSOLUTE CONVERGENCE TOLERANCE IN REACTIVE TRANSPORT

1.0E-20

!> Set the divergence tolerance for reactive transport problem

!> The default value is 1.0E5
```

PETSC: DIVERGENCE TOLERANCE IN REACTIVE TRANSPORT 1.0E5 !> Set the maximum number of iterations for reactive transport problem !> The default value is 1000 PETSC: MAXIMUM NUMBER OF ITERATIONS IN REACTIVE TRANSPORT 100 2. If 'dense block matrices' is used in "Data Block 7: control parameters - reactive transport", then the system will use 'dense block matrices', otherwise, if 'sparse block matrices' is used or this option is not specified, the system will use 'sparse block matrices'. 3. By default, the matrix will be exported in general 1D array. Set the output matrix format to 1 in parallel solver configuration file (solver.cfg) will export the matrix into MM format. !> Set the output matrix format !> 0: export a (ia, ja), b, x as 1d array into separated files !> 1: export a (ia, ja) as matrix market exchange format, !> and b and x as 1d array into separated files !> Requirement: Optional, only for test. !> E.g. !> OUTPUT MATRIX FORMAT SELECTION !> 1 OUTPUT MATRIX FORMAT SELECTION 1	
Codes Update	**Remarks**
1.	

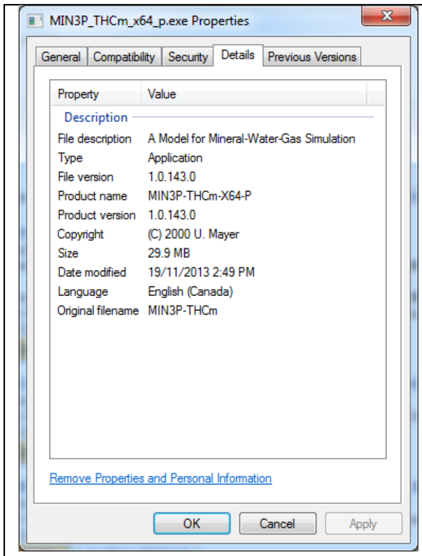
Benchmarks Update	**Remarks**
1.	
Documentations Update	**Remarks**
1.	
Bug Fixes	**Remarks**
1. Fix bug of race condition caused by dtotg and ginc in jacbrt.F90. 2. Fix bug of exporting matrix when WatSolv is called. The exported matrix is exported after incomplete factorization. Exported the matrix before incomplete factorization. 3. Remove duplicated and null declaration in the codes. Some of these declaration will results into compiling error in IVF 14.0. 4. Fix bug that causes difference in the initial concentration of sorbed species in _0.gsb and .gen has been fixed. This bug is caused by the error pointer in gamma array in outputrt.F90. The previous codes use gamma (the same as gamma(1,1)) as the pointer address while the correct pointer address should be gamma(1, ivol). This bug exists from the initial version of MIN3P-THCm. For the archived benchmarks, if the gamma(1,1) is the same as gamma(1, ivol), the results are still correct. But for concrete-clay benchmark, the bug affects the results.	
Notes	**Remarks**
The current version is hybrid MPI and OpenMP. The solver can work for both MPI and OpenMP version while the matrix assembly can only work for OpenMP.	
Summary of code verification (test running)	**Remarks**
The following differences are observed comparing the current version to the previous version: *.gsd, *.gss and *.gsv files in benchmark d51_dedolomitization; *.gbc, *.gbm, *.gbt, *.gsb, *.gsc, *.gsm and *.gst files in benchmark d52_cation_exchange; results in benchmark d6_rt_highly_saline, biomass-V1.0.56, multisite-ionx-V1.0.129, surfx-ionx-V1.0.106 and uraninite-redoxidation-V1.0.56 and other benchmarks in benchmarks_standard category;	

Revision 144

Revision Number	144	**Version Number**	V1.0.144
Commit Date	2013-11-20	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: RELEASE_X64|RELEASE_WIN32;WINDOWS 2.3 Preprocessor definition for parallel version: RELEASE_X64|RELEASE_WIN32;WINDOWS;PARDISO;OPENMP;SCHEDULE_DYNAMIC|SCHEDULE_STATIC|SCHEDULE_GUIDED 2.4 Preprocessor definition for resources: RELEASE_X64|RELEASE_WIN32|RELEASE_X64_P|RELEASE_WIN32_P 2.5 Additional Options: /fp:strict	
System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version.	

The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
New Features	**Remarks**
1. Add OpenMP parallel routine for the following codes: initicrt.F90, initbcrt.F90, initicdd.F90, initbcenergybal.F90, restart_r.F90, batreac.F90, sptldisc.F90, xyzcoord.F90, cvolume.F90, mem_vs.F90, mem_heat.F90, mem_rt.F90, mem_dd.F90, mem_evap.F90, iajavs.F90, iajavs_dp.F90, iajavs_energybal.F90, mem_njavs.F90, mem_njart.F90, datstr_1.F90, datstr_n.F90, mem_mat.F90, initppvs.F90, initpppm.F90, initppenergybal.F90, initicvs_1.F90, initsatw.F90, initicenergybal.F90, updatedd.F90, updatedd_energybalance.F90, seepfdd.F90, seepface.F90, tstepvs.F90, updatevs.F90, soilparm.F90, msysdd.F90, mbaldd.F90, msysvs.F90, energysys.F90, energy_bal.F90, velodd.F90, nexttime.F90, infcvs.F90, comp_bc_ice_sheet.F90, m_ice_sheet.F90, infheat_c.F90, infheat_d.F90, timeloop.F90, infevap.F90, ddtds.F90, ddtds_energybal.F90, mbal_mcd.F90, tsteprt.F90, infcrtdd.F90, infcrt_a.F90, infcrt_g.F90, infcrt_mcd.F90, diffcoff_mcd.F90, i2upfind.F90, mbalrt.F90, msysrt.F90 2. Add OpenMP parallel control commands for the above parallel routines. This is mainly designed for parallel test or performance test. 3. Add executable file information (File name, Version, Copyright, and etc.) to executable file properties. The version number will be automatically replaced by VisualSVN revision number when the project is built.	

 MIN3P_THCm_x64_p.exe Properties General Compatibility Security Details Previous Versions Property Value Description File description A Model for Mineral-Water-Gas Simulation Type Application File version 1.0.143.0 Product name MIN3P-THCm-X64-P Product version 1.0.143.0 Copyright (C) 2000 U. Mayer Size 29.9 MB Date modified 19/11/2013 2:49 PM Language English (Canada) Original filename MIN3P-THCm Remove Properties and Personal Information OK Cancel Apply 4. Update trunk revision 129 and 137 to this branch. 5. Add MIN3P development guidelines to docs. 6. Add MIN3PRunAndResultCompare tool to tools.	
Usage of New Features	**Remarks**
1. Usage of Feature 1 and 2: Add the following commands to enable/disable the parallel routine. !> *** !> Block: OpenMP Parallel Controls !> *** !> Format: Command for specified subroutine !> Threshold for loop amount !> If the threshold is smaller than the loop amount, OpenMP !> parallelization is enabled for this subroutine, otherwise,	

```
!> if the threshold is larger than the loop amount, OpenMP
!> parallelization is disabled for this subroutine.
!> ************************************************************
mbalrt: number of threads 1
1
mbalrt: number of threads 2
1
mbalrt: number of threads 3
1
mbalrt: number of threads 4
1
mbalrt: number of threads 5
1
mbalrt: number of threads 6
1
mbalrt: number of threads 7
1
mbalrt: number of threads 8
1
mbalrt: number of threads 9
1
mbalrt: number of threads 10
1
mbalrt: number of threads 11
1
msysrt: number of threads 1
```

1 *msysrt: number of threads 2* *1* *msysrt: number of threads 3* *1* *msysrt: number of threads 4* *1* *msysrt: number of threads 5* *1* *msysrt: number of threads 6* *1* *mbal_mcd: number of threads 1* *1* *mbal_mcd: number of threads 2* *1* *mbal_mcd: number of threads 3* *1* *infcrtdd: number of threads 1* *1* *infcrtdd: number of threads 2* *1* *infcrtdd: number of threads 3* *1* *infcrt_a: number of threads 1* *1* *infcrt_a: number of threads 2*	

1 *infcrt_a: number of threads 3* *1* *infcrt_g: number of threads 1* *1* *infcrt_g: number of threads 2* *1* *infcrt_mcd: number of threads 1* *1* *infcrt_mcd: number of threads 2* *1* *diffcoff_mcd: number of threads 1* *1* *i2upfind: number of threads 1* *1* *i2upfind_heat: number of threads 1* *1* *ddtds: number of threads 1* *1* *ddtds_energybal: number of threads 1* *1* *ddtds_energybal: number of threads 2* *1* *comp_bc_ice: number of threads 1* *1* *comp_bc_ice: number of threads 2*	

1 *timeloop: number of threads 1* *1* *infheat_c: number of threads 1* *1* *infheat_d: number of threads 1* *1* *infevap: number of threads 1* *1* *updatedd: number of threads 1* *1* *updatedd: number of threads 2* *1* *updatedd_ener: number of threads 1* *1* *updatedd_ener: number of threads 2* *1* *updatedd_ener: number of threads 3* *1* *ddvsflow: number of threads 1* *1* *seepfdd: number of threads 1* *1* *tstepvs: number of threads 1* *1* *updatevs: number of threads 1*	

1 *updatevs: number of threads 2* *2* *seepface: number of threads 1* *1* *soilparm: number of threads 1* *1* *msysdd: number of threads 1* *1* *msysdd: number of threads 2* *1* *msysdd: number of threads 3* *1* *msysvs: number of threads 1* *1* *mbalvs: number of threads 1* *1* *mbalvs: number of threads 2* *1* *mbalvs: number of threads 3* *1* *energysys: number of threads 1* *1* *energy_bal: number of threads 1* *1* *energy_bal: number of threads 2*	

1 velodd: number of threads 1 1 nexttime: number of threads 1 1 nexttime: number of threads 2 1 infcvs: number of threads 1 1 xyzcoord: number of threads 1 1 cvolume: number of threads 1 1 iajavs: number of threads 1 1 iajavs: number of threads 2 1 iajavs_dp: number of threads 1 1 iajavs_ener: number of threads 1 1 iajavs_ener: number of threads 2 1 matrix_uti: number of threads 1 1 iajart: number of threads 1	

1 *iajart: number of threads 2* *1* *initpppm: number of threads 1* *1* *initppdd: number of threads 1* *1* *initppvs: number of threads 1* *1* *initppvs: number of threads 2* *1* *initppeb: number of threads 1* *1* *initppeb: number of threads 2* *1* *initicvs: number of threads 1* *1* *initsatw: number of threads 1* *1* *initsatw: number of threads 2* *1* *initsatw: number of threads 3* *1* *initicener: number of threads 1* *1* *initprob: number of threads 1*	

1 *initprob: number of threads 2* *1* *initicrt: number of threads 1* *1* *initicdd: number of threads 1* *1* *restart_r: number of threads 1* *1* *batreac: number of threads 1* *1* *batreac: number of threads 2* *1*	
Codes Update	**Remarks**
1. Modify the default backup frequency (restart file) from 10 to 100. 2. Modify writing of "Solver Iteration Convergence Summary" from debug file to log file. 3. Modify runtime calculation, the failed timestep is not included in *_nonlinear.prt but included in *_timestep.prt. 4. Preallocate memory space (iwork, rwork, afrt, afglob, ...) for the solver to improve performance as dynamic allocate memory space will seriously degrade performance for large scale simulation. 5. Update the rnorm_max threadhold for Pardiso solver. 6. Update the output information for profiling test. The iteration parameters of Picard iteration, Newton iteration and seepage iteration are added. The previous "coupling time" is renamed as "nonlinear time".	
Benchmarks Update	**Remarks**
1.	

Documentations Update	**Remarks**
1.	
Bug Fixes	**Remarks**
1. Fix bug in calling infcvs in restart_r.F90. This subroutine is not necessary to be called for every volume. This bug is benign and exists in all the previous versions. 2. Fix bug in redundant calling tcorr in initprob. This bug is benign and exists in all the previous versions. 3. Fix bug in output iteration number.	
Notes	**Remarks**
1. Pointer is treated as a shared variable in OpenMP. To avoid race condition, some of the pointer is removed and use allocatalbe array instead. 2. Pardiso vs iterative solver with ILU preconditioning such as PETSc: PETSc is an iterative solver whereas PARDISO direct one. Iterative methods with ILU preconditioned for good conditioner number converge faster than direct on the other hand direct return near exact solution. For ill condition matrix situation change a bit - converge of iterative solver become poor and direct method return solution quicker (its speed doesn't depend on condition number) but residual can be bad. These reasoning depend on preconditioner - so that's a hard to compare this 2 absolutely different approach of solving system of linear equation.	
Summary of code verification (test running)	**Remarks**
There is no difference except *.dbg files. This files are only used to generate temporary output in debugging. We do not output debug information in the release version.	

Revision 137

Revision Number	137	**Version Number**	V1.0.137
Commit Date	2013-11-01	**Commit Author**	Mingliang Xie
Software Grade		signature	Date
Verified by	Danyang Su		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: *None* 2.3 Preprocessor definition for parallel version: WINDOWS;PARDISO;OPENMP;SCHEDULE_DYNAMIC;CONDITION_NUMBER 2.4 Additional Options: /fp:strict	
System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
New Features	**Remarks**

1. hybrid multicomponent diffusion model(hMCD) (test implementation for Mt Terri project).	
Usage of New Features	**Remarks**
1. Usage of feature 1: To use this function, two keywords are needed: 'effective porosity factor for MCD diffusion' and 'effective tortuosity factor for MCD diffusion'in data block 11: physical parameters - reactive transport. This function should be used under MCD (activated by 'multicomponent diffusion' under Datablock 1). The new input files can be found in the new-added benchmarks, see benchmarks_new_added\hMCD-V1.0.137. This example test both multisite ion exchange and hMCD model. 1.1 Define effective porosity factor *'effective porosity factor for MCD diffusion'* *1.0 ;'cs+'* *1.0 ;'na+1'* *1.0 ;'k+1'* *1.0 ;'mg+2'* *1.0 ;'ca+2'* *1.0 ;'sr+2'=0.15/0.11739* *0.667 ;'cl-1' 'free'* *1.0 ;'h+1'* *1.0 ;'o2(aq)'* *0.667 ;'so4-2'* *0.667 ;'co3-1'* *0.533 ;'i-1' =0.08/0.15*	

0.667 ;'br-1' 0.1/0.15 *1.0 ;'HTO'* *1.0 ;'co_60+2'* 1.2 Define effective tortuosity factor *'effective tortuosity factor for MCD diffusion'* *1.0 ;'cs+'* *1.0 ;'na+1'* *1.0 ;'k+1'* *1.0 ;'mg+2'* *1.0 ;'ca+2'* *3.5 ;1.277791975 ;1.0 ;'sr+2'=0.15/0.11739* *1.0 ;'cl-1' 'free'* *1.0 ;'h+1'* *1.0 ;'o2(aq)'* *1.0 ;'so4-2'* *1.0 ;'co3-1'* *0.55 ;'i-1'* *0.38 ;'br-1'* *1.0 ;'HTO'* *300.0 ;'co_60+2'* The name order should be the same as component. More details see the Mt Terri project report.	
Codes Update	**Remarks**

Benchmarks Update	**Remarks**
1. New benchmark added, see benchmarks_new_added\ surfx-ionx-V1.0.137.	
Documentations Update	**Remarks**
None	
Bug Fixes	**Remarks**
1. Fix bug of sorbed mass output for ion exchange reactions which may cause changes in the output file e.g _o.gen.	
Notes	**Remarks**
1. This approach is not fully implemented, i.e. material dependent factors are not considered.	
Summary of code verification (test running)	**Remarks**
No difference of the results simulated using the current version and the previous version V1.0.129.	

Revision 129

Revision Number	129	**Version Number**	V1.0.129
Commit Date	2013-10-24	**Commit Author**	Mingliang Xie
Software Grade		signature	Date
Verified by	Danyang Su		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: *None* 2.3 Preprocessor definition for parallel version: WINDOWS;PARDISO;OPENMP;SCHEDULE_DYNAMIC;CONDITION_NUMBER 2.4 Additional Options: /fp:strict	

System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	

New Features	**Remarks**
1. Multisite ion-exchange reaction for Gaines Thomas convention.	

Usage of New Features	**Remarks**
1. Usage of feature 1: To use this function, two keywords are needed: 'surface sites of ion-exchange' in data block 2: geochemical system and 'CEC fraction of multisite ion exchange' in Data Block 14: initial condition - reactive transport for the specification of the amount of each site. The new input files can be found in the new-added benchmarks, see benchmarks_new_added\ionx-2domains-V1.0.129. 1.1 Define number and names of surface sites and sorbed species, add the following input command/data to Data Block 2: geochemical system. *'surface sites of ion-exchange'* *2* *'-x'* *'-Y'* *'sorbed species of ion-exchange'* *12* *'mg-x(na)'* *'na-x(na)'* *'k-x(na)'* *'zn-x(na)'* *'pb-x(na)'* *'ca-x(na)'* *'mg-Y(na)'* *'na-Y(na)'*	

```
'k-Y(na)'

'zn-Y(na)'

'pb-Y(na)'

'ca-Y(na)'
```

The name of each site should include ‘-‘. The sorbed species name has to use the site name plus (na). For example mg-Y(na) stands for the sorbed species of Mg+2 on –Y and exchanges with Na+. The exact reaction will be defined in the database in the same way for single site ion excahnge reactions (see user manual).

1.2 Define the amount of each sites, add the following input command/data to Data Block 14: initial condition - reactive transport

```
'sorption parameter input of ion-exchange'

0.733          ;total cation exchange capacity (CEC) [meq/100 g solid]

1.875d0        ;dry bulk density [g/cm^3]


'CEC fraction of multisite ion exchange'

0.9            ;'-x'

0.1            ;'-Y'
```

The input format and units for the total CEC value and dry bulk density parameters remain the same as the single site ion excahnge reaction. But the keyword must be updated as the one in revision 94 as shown above.

The amount of each sites are specified by the CEC fractions. They should be given in the same order as given under the keyword 'surface sites of ion-exchange'. The total CEC fraction should be 1.0.

Codes Update	**Remarks**

Benchmarks Update	**Remarks**
1. New benchmark added, see benchmarks_new_added\multisite-ionx-V1.0129.	
Documentations Update	**Remarks**
1. Updated the theory manual 2. Updated the user manual 3. Updated the versification report	
Bug Fixes	**Remarks**
1. Fix bug of pe, pH and Eh initialization in the source code file phpe.F90, which may cause NaN value in the output files *.gbm, *.gsm.	
Notes	**Remarks**
1. Multisite ion-exchange reaction for Gapon convention is not verified yet. 2. Four benchmarks (i.e. ionx-m, ionx-m-2domains, ionx-multisite-2domain and ionx-multisite-default-dbs-sec0.73) using the multisite ion exchange model were added under .\benchmarks_new_add\surfx-ionx-V1.0.106.	
Summary of code verification (test running)	**Remarks**
No difference of the results simulated using the current version and the previous version V1.0.107.	

Revision 107

Revision Number	107	**Version Number**	V1.0.107
Commit Date	2013-8-21	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.3 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: *None* 2.3 Preprocessor definition for parallel version: WINDOWS;PARDISO;OPENMP;SCHEDULE_DYNAMIC;CONDITION_NUMBER 2.4 Additional Options: /fp:strict	

System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	

New Features	**Remarks**
1. Separation of surface-complex and ion-exchange reaction so as to support	There is

both reactions at the same time. 2. Output the matrix data and right hand side for the linear equations. This is only used for solver testing.	some difference between MIN3P and PHREEQC, further check may be needed.
Usage of New Features	**Remarks**
1. Usage of feature 1: If only surface-complex or ion-exchange is considered, you can just use the previous input file. If both surface-complex and ion-exchange is considered, you should use the following input format. You can also modify the following input file for only surface-complex or ion-exchange, just delete the unnecessary part. The new input files can be found in the new-added benchmarks, see benchmarks_new_added\surfx-ionx-V1.0.106. 1.1 Define sorption type, add the following input command/data to Data Block 2: geochemical system. *‘define sorption type of surface-complex’* *‘surface-complex’* *‘define sorption type of ion-exchange’* *‘gaines-thomas’* 1.2 Define sorbed species, add the following input command/data to Data Block 2: geochemical system. *‘sorbed species of surface-complex’* *3* *‘=feoh2+(s)’*	

```
‘=feo-(s)’

‘=feozn+(s)’


‘sorbed species of ion-exchange’

3

‘na-x(na)’

‘ca-x(na)v’

‘mg-x(na)v’
```

1.3 Define sorption parameter, add the following input command/data to Data Block 14: initial condition – reactive transport

```
‘sorption parameter input of ion-exchange’

10.0d0      ;cation exchange capacity [meq/100 g solid]; ion-exchange part, cec

1.875d0      ;dry bulk density [g/cm^3]


‘sorption parameter input of surface-complex’

‘=feoh(s)’  100.0d0  10.0d0  6.02228d0  ;surface site, mass, area, site density
```

1.4 Define equilibrate type, add the following input command/data to Data Block 14: initial condition – reactive transport

```
‘equilibrate with fixed solution composition of ion-exchange

!’equilibrate with fixed solution composition of surface-complex’
```

2. Usage of feature 2

Add the following input command/data to parallel solver configuration file (e.g,

solver.cfg) *!> Set if output the matrix data for all the linear equations.* *!> Comment out if you do not want to export it.* *!> Requirement: Optional, only for test.* *OUTPUT SPARSE MATRIX DATA SET AND RHS*	
Codes Update	**Remarks**
1. Add schedule 'guided' for parallel schedule. 2. Update the format of long names output in gen file. For the 72-character long names, export the first 12 characters if the name is followed by some other data in the same line.	
Benchmarks Update	**Remarks**
1. New benchmark added, see benchmarks_new_added\surfx-ionx-V1.0.106.	
Documentations Update	**Remarks**
Bug Fixes	**Remarks**
1. Fix bug of race condition in ddbdflux.F90 for the parallel version.	
Notes	**Remarks**
1. There is some difference between MIN3P and PHREEQC when both surface complex and ion exchange are considered, further check may be needed.	
Summary of code verification (test running)	**Remarks**
No difference of the results simulated using the current version and the previous version V1.0.94.	

Revision 94

Revision Number	94	**Version Number**	V1.0.94
Commit Date	2013-6-11	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.1.2 2. Configurations 2.1 Preprocess source file: Yes (/fpp) 2.2 Preprocessor definitions for sequential version: None 2.3 Preprocessor definition for parallel version: WINDOWS;PARDISO;OPENMP;SCHEDULE_DYNAMIC;CONDITION_NUMBER 2.4 Additional Options: /fp:strict	Configurations and options are optional if user do not want to use all the features
System Requirements	**Remarks**
Intel Visual Fortran Redistributable library is required to run the parallel version. The redistributable library can be downloaded via the URL http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.	
New Features	**Remarks**

1. Add parallel version of matrix assembly and solver for MIN3P. 2. Add condition number estimation module. 3. Update the length of name from 12 to 72. 4. Add build environment to the log file when the program runs.	
Usage of New Features	**Remarks**
Usage of features 1 and feature 2 are defined in the parallel configuration file, which is used for parallel version of MIN3P. For the formatted user guide for parallel version of MIN3P, please look into MIN3P-THCm_User_Manual_Parallel.doc. 1 PARALLEL CONFIGURATION Parallel Configuration is used to configure the system environment and input parameters. The configuration focuses on the parallel matrix solver, parallel matrix assembly, parallel performance analysis and testing. 1.1 SYSTEM REQUIREMENT Redistributable libraries are needed for the OS without developing environment. Currently, the program is compiled with Intel Visual Fortran compiler. Before running the parallel version of MIN3P, please install Intell Visual Redistributables on the system first. 1.1.1 System requirement for Windows OS For 32-bit OS, please install Intell Visual Fortran Redistributables on IA-32; For 64-bit OS, please install Intell Visual Fortran Redistributables on 64. The Redistributable libraries corresponding to the developing environment can be download via the URL	

http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.

1.1.2 System requirement for Linux OS

Compatible compiler for Fortran 90/95.

1.2 CONFIGURATION FILE ENTRY

To use parallel configuration file, please add the following command and data in the block of "Data Block 1: global control parameters" in MIN3P input file.

'parallel solver configuration file'

'solver.cfg'

Where 'solver.cfg' is path of configuration file.

1.3 CONFIGURATION FILE FORMAT

The entire configuration is saved in an ASCII text file (e.g., parallel.cfg). The line begin with exclamation mark (!) is a comment line. The character is NOT case-sensitive. The line begin with alphabet character is a command line or a data line. The line begin with number is a data line. Not all the command line or data line is required. The latter command will overwrite the former command if they have the same target.

1.4 GLOBAL SOLVER SETTING

Global solver setting includes the solver type for flow and reactive transport, as well as the number of threads for global use.

There are two solvers in the parallel version of MIN3P. The first solver is WatSolv (or WS209), which is the default solver of MIN3P. WatSolv is partially parallelized. The second solver is Pardiso, which is a shared-memory directly solver.

1.4.1 Command and data

1.4.1.1 Solver type

Set solver type for flow and reactive transport equations. Solver type is given by a command line and a data line.

Solver type

i

Set solver type for both flow equations and reactive transport equations. If I is 0, use WatSolv as the solver, if I is 1, use Pardiso as the solver. If this command is not specified, the default solver is WatSolv.

Solver type flow

i

Set solver type for flow equations only. If I is 0, use WatSolv as the solver, if I is 1, use Pardiso as the solver. If this command is not specified, the default solver is WatSolv.

Solver type reactive transport

i

Set solver type for reactive transport equations only. If I is 0, use WatSolv as the solver, if I is 1, use Pardiso as the solver. If this command is not specified, the default solver is WatSolv.

1.4.1.2 Number of threads

Set the number of threads for global use, including number of threads for matrix assembly, parallel solver as well as other parallel region.

Global: number of threads

n

Set the number of threads n for global use. The number of threads for global use includes matrix assembly, matrix solver and other parallel region. If this command is not specified, the default number of threads for global use is 1.

1.4.2 Sample configuration

```
!> **********************************************************

!>            Block:  Golbal solver setting

!> **********************************************************

!SOLVER TYPE

!1


SOLVER TYPE FLOW

0


SOLVER TYPE REACTIVE TRANSPORT

1


GLOBAL: NUMBER OF THREADS

4
```

1.5 MATRIX ASSEMBLY SETTING

Matrix assembly setting includes matrix assembly type, number of threads used for matrix assembly, and chunk size factor. Matrix assembly setting is separated for flow and reactive transport. If this part is not specified, and the number of threads for global use is 1, then the system will run in sequential mode for matrix assembly.

1.5.1 Command and data

1.5.1.1 Matrix assembly type

Set matrix assembly type for flow and reactive transport equations. Matrix assembly type is given by a command line and a data line.

Matrix assembly: type in flow

i

Set matrix assembly type for flow equations only. If I is 0, use sequential mode (non-parallel) for matrix assembly, if I is 1, use OPENMP parallel mode for matrix assembly. If this command is not specified, the default mode is sequential mode.

Matrix assembly: type in reactive transport

i

1.5.1.2 Number of threads

Set matrix assembly type for reactive transport equations only. If I is 0, use sequential mode (non-parallel) for matrix assembly, if I is 1, use OPENMP parallel mode for matrix assembly. If this command is not specified, the default mode is sequential mode.

Matrix assembly: number of threads in flow

n

Set the number of threads n in matrix assembly for flow equations only. If not specified, use ‘global: number of threads’ instead.

Matrix assembly: number of threads in reactive transport

n

Set the number of threads n in matrix assembly for reactive transport equations only. If not specified, use ‘global: number of threads’ instead.

1.5.1.3 Schedule type

matrix assembly: schedule type in flow

i

Set the schedule type in matrix assembly for flow equations only. If I is 0, use static schedule method. If I is 1, use dynamic schedule method. The default value is 0. Currently schedule type is pre-defined in compiling option. DO NOT use the setting here.

Matrix assembly: schedule type in reactive transport

i

Set the schedule type in matrix assembly for reactive transport equations only. If I is 0, use static schedule method. If I is 1, use dynamic schedule method. The default value is 0. Currently schedule type is pre-defined in compiling option. DO NOT use the setting here.

1.5.1.4 Chunk size in parallel loop

matrix assembly: chunk size factor in flow

n

Set the chunk size factor for matrix assembly for flow equations only. This value should be from 0 to ‘(loop count)/(number of processors)’.If the value is not specified or 0, use the system default value. The default value depends on the schedule type.

Matrix assembly: chunk size factor in reactive transport

n

Set the chunk size factor for matrix assembly for reactive equations only. This value should be from 0 to ‘(loop count)/(number of processors)’.If the value is not specified or 0, use the system default value. The default value depends on the schedule type.

1.5.2 Sample configuration

```
!> *********************************************************

!>              Block:  Matrix assembly setting

!> *********************************************************


MATRIX ASSEMBLY: TYPE IN FLOW

1


MATRIX ASSEMBLY: NUMBER OF THREADS IN FLOW
```

4

MATRIX ASSEMBLY: SCHEDULE TYPE IN FLOW

0

MATRIX ASSEMBLY: CHUNK SIZE FACTOR IN FLOW

0

MATRIX ASSEMBLY: TYPE IN REACTIVE TRANSPORT

1

MATRIX ASSEMBLY: NUMBER OF THREADS IN REACTIVE TRANSPORT

4

MATRIX ASSEMBLY: SCHEDULE TYPE IN REACTIVE TRANSPORT

0

MATRIX ASSEMBLY: CHUNK SIZE FACTOR IN REACTIVE TRANSPORT

0

1.6 PARDISO SOLVER SETTING

PARDISO is a thread-safe, high-performance, robust, memory efficient and easy to use software for solving large sparse symmetric and unsymmetric linear systems of equations on shared-memory. The following settings are valid only if the solver type is Pardiso.

1.6.1 Command and data

1.6.1.1 Solver test

pardiso: solver test with ws209

Set this command if you want to check the result of matrix solver with WatSolv (WS209). This command is valid only if the solver type is WatSolv. Do not include this command in the normal use. This is only used for testing.

1.6.1.2 Number of Threads

pardiso: number of threads

n

Set the number of threads n for pardiso solver. This is valid only if the solver type is Pardiso. If the number of threads is 0, use number of threads determined by pardiso, which is uaually the number of physical cores.

1.6.1.3 Solver refinement

pardiso: max iterative refinement steps in flow

n

pardiso: max iterative refinement steps in reactive transport

n

Set the maximum number of iterative refinement steps n that the solver will perform for flow equations and reactive transport equations, respectively. The solver will perform not more than the absolute value of this parameter for iterative refinement and will stop the process if a satisfactory level of accuracy of the solution in terms of backward error has been achieved.If this parameter is negative, the accumulation of the residuum is using extended precision real types. The default value is 9.

1.6.1.4 Pivoting perturbation

pardiso: pivoting perturbation in flow

n

pardiso: pivoting perturbation in reactive transport

n

This parameter controls how to handle small pivots or zero pivots for unsymmetric matrices for flow equations and reactive transport equations, respectively. It indicates the iterative refinement contraction rate. The default value is 13, which means eps = 10^(-13) is used in handling small pivots.

1.6.1.5 Preconditioning

pardiso: cgs criterion in flow

n

pardiso: cgs criterion in reactive transport

n

This parameter controls preconditioned CGS [sonn89] for nonsymmetric matrices for flow equations and reactive transport equations, respectively. The parameter n has the form 10*l+k.If k=0, the factorization is always computed as required by phase. If k=1 CGS iteration replaces the computation of LU. The preconditioner is LU that was computed at a previous step (the first step or last step with a failure) in a sequence of solutions needed for identical sparsity patterns. For example, if n is 31, the solver will use LU-preconditioned CGS iteration with a stopping criterion of 1.0e-3 for nonsymmetric matrices ; If n is 61, the solver will use LU-preconditioned CGS iteration with a stopping criterion of 1.0e-6 for nonsymmetric matrices. The default value is 0.

1.6.1.6 Maximum solver iteration

pardiso: maximum solver iteration in flow

n

pardiso: maximum solver iteration in reactive transport

n

This parameter controls the maximum solver iteration n for flow equations and reactive transport equations, to see if symbolic factorization should take before the next iteration. In pardiso solver, when the previous reordering is not quite good to get correct results, reorder the matrix (symbolic factorization) again. For nonsymmetric case, it’s better to call reorder step for each matrix, but this can waste a lot of time. When preconditioned CGS is used, this value will be compared to the number of completed iterations, otherwise, this value will be compared to the number of iterative refinement steps performed. When the iteration number is larger than the specified value, do symbolic factorization again. The default value is 9.

1.6.1.7 Maximum residual

pardiso: maximum residual in flow

e

pardiso: maximum residual in reactive transport

e

This parameter controls the maximum residual e for flow equations and reactive transport equations, to see if symbolic factorization should take before the next iteration. In pardiso solver, when the previous reordering is not quite good to get correct results, reorder the matrix (symbolic factorization) again. For nonsymmetric case, it’s better to call reorder step for each matrix, but this can waste a lot of time. When the maximum residual is larger than the specified value, do symbolic factorization again. The default value is 1.0e-5.

1.6.1.8 Symbolic factorization type

pardiso: symbolic factorization type in flow

i

pardiso: symbolic factorization type in reactive transport

i

This parameter controls the symbolic factorization type I for flow equations and reactive transport equations. If the symbolic factorization type is 0, do symbolic factorization only once before the iteration. If the symbolic factorization type is 1, do symbolic factorization at every newton iteration step. When symbolic factorization type is 0, the “matrix solver iteration” and “maximum residual” parameters are valid.

1.6.2 Tips on pardiso solver

Pardiso cannot handle ill-conditioned matrix. For some nearly ill-conditioned matrices, pardiso solver may generate inaccurate results in solving equations that cause failure in Newton iteration.   If Newton iteration fails, try to use WatSolv solver. You can also output the condition number to see if the matrix is ill-conditioned.

1.6.3 Sample configuration

```
!> ********************************************************

!>              block:  pardiso solver setting

!> ********************************************************

! PARDISO: SOLVER TEST WITH WS209


PARDISO: NUMBER OF THREADS

0
```

PARDISO: MAX ITERATIVE REFINEMENT STEPS IN FLOW *9* *PARDISO: MAX ITERATIVE REFINEMENT STEPS IN REACTIVE TRANSPORT* *9* *PARDISO: PIVOTING PERTURBATION IN FLOW* *13* *PARDISO: PIVOTING PERTURBATION IN REACTIVE TRANSPORT* *13* *PARDISO: CGS CRITERION IN FLOW* *0* *PARDISO: CGS CRITERION IN REACTIVE TRANSPORT* *0* *PARDISO: MAXIMUM SOLVER ITERATION IN FLOW* *5* *PARDISO: MAXIMUM SOLVER ITERATION IN REACTIVE TRANSPORT* *5* *PARDISO: MAXIMUM RESIDUAL IN FLOW* *1.0E-8*	

```
PARDISO: MAXIMUM RESIDUAL IN REACTIVE TRANSPORT
1.0E-8


PARDISO: SYMBOLIC FACTORIZATION TYPE IN FLOW
0


PARDISO: SYMBOLIC FACTORIZATION TYPE IN REACTIVE TRANSPORT
0
```

1.7 WATSOLV SOLVER SETTING

For the present version, WatSolv (WS209) is not fully parallelized. Only part of the loops in the solver is parallelized. The only parameters that can be controlled is the number of threads.

1.7.1 Command and data

1.7.1.1 Number of Threads

```
ws209: number of threads
n
```

Set the number of threads n for WatSolv (WS209). This is valid only if the solver type is WatSolv. If not specified, use the default value 1 as the number of threads.

1.7.2 Sample configuration

```
!> *********************************************************
!>            block:  ws209 solver setting
!> ********************************************************

WS209: NUMBER OF THREADS

4
```

1.8 OUTPUT SETTING

1.8.1 Command and data

1.8.1.1 Runtime statistics analysis

```
output runtime statistics analysis
```

This command outputs the runtime profiling, including runtime of matrix assembly, factorization and substitution in each Newton iteration step, runtime of flow and reactive transport in each time step. The output file is in tecplot data format.

1.8.1.2 Condition number

```
output condition number
```

This command output the estimated condition number of matrix before solving the equations. Enable this command will significantly increase the running time. Use this only in testing, for example, when iteration fails.

1.8.2 Sample configuration

```
!> ********************************************************

!>            Block:  Output setting

!> ********************************************************


OUTPUT RUNTIME STATISTICS ANALYSIS
```

OUTPUT CONDITION NUMBER	
Codes Update	**Remarks**
Benchmarks Update	**Remarks**
Documentations Update	**Remarks**
Bug Fixes	**Remarks**
1. For some specific cases, Newton iteration fails. This bug is caused by level of accuracy, making some simulation fail in iteration. 1) adding "/fp:strict" to the compilation option, which has higher level of accuracy and result consistency for floating-point calculation. 2) adding explicit interface instead of external statement for the subroutine/functions with optional parameters.	
Notes	**Remarks**
1. Condition number estimation is time-consuming. Do not use this in the normal use. This is mainly used for matrix check when newton iteration failed. There will be two condition number exported: classical condition number and skeel's condition number. If both condition numbers are very large (e.g., > 1.0e12), the matrix are ill-conditioned and Pardiso may fail in solving it. WatSolv solver is more accurate in handling ill-conditioned matrix. 2. The code of estimating condition number is "Academic licence", read the LICENCE file for more information. 3. Input format is not changed when update the length of name from 12 to 72. For the free format database, it can support name length longer than 12. But for the fixed format database, it only read 12 characters. This should be modified later.	

4. Installation of redistributable libraries.

Redistributable libraries are needed for the OS without developing environment. Currently, the program is compiled with Intel Visual Fortran compiler. Before running the parallel version of MIN3P, please install Intel Visual Redistributables on the system first. For 32-bit OS, please install Intel Visual Fortran Redistributables on IA-32; For 64-bit OS, please install Intel Visual Fortran Redistributables on 64. The Redistributable libraries corresponding to the developing environment can be downloaded via the URL

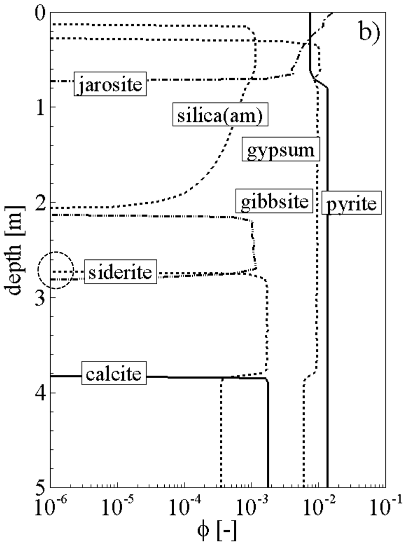
http://software.intel.com/en-us/articles/redistributable-libraries-for-intel-c-and-visual-fortran-composer-xe-2013-for-windows.

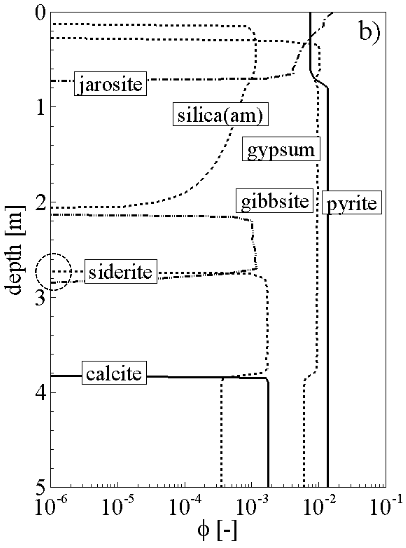
5. Selection of executable files.

There are two levels of executable file: normal level accuracy and high level accuracy. The executable files of normal level accuracy are compiled with the default setting ("/fp:fast") for floating-point calculation while the executable files of high level accuracy are compiled with "/fp:strict" for floating-point calculation. For most of the cases, normal level accuracy executable file works fine. But for some special cases, high level accuracy executable file is needed, otherwise, the solver maybe fails.

6. Result difference when "/fp:strict" is used.

Almost all the benchmarks have unrecognizable difference in results whatever "/fp:strict" is used or not. But for the benchmark amd-ex, the difference is recognizable. The figures bellow demonstrate the difference, left without "/fp:strict" and right with "/fp:strict".





7. Failure in Pardiso solver

Pardiso parallel solver cannot handle ill-conditioned matrix well that it may fail in the simulation with ill-conditioned matrices. For all the benchmarks, two benchmarks in the following folder fail: benchmarking_nwmo_report\nwmo_verification_examples\d311_hydrostatic_box_test.

Summary of code verification (test running)	**Remarks**
With the compiler option “/fp:strict” added, all the benchmarks have unrecognizable difference in results except for the benchmark amd-ex, the difference is recognizable, as discussed in Notes.6.	

Revision 67

Revision Number	67	**Version Number**	V1.0.67
Commit Date	2013-04-15	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.1 2. Configurations: Default			
System Requirements			**Remarks**
New Features			**Remarks**
1. Add maximum and minimum threshold for permeability scaling factor. For the previous version, the permeability scaling factor can range from 0 to a huge number, which may cause unrealistic result in matrix solver. 2. Add threshold “porosity threshold maximum” and “porosity threshold minimum” when “update porosity” is used. This is to avoid division by 0 in Kozeny-Carman equation and to avoid porosity 0. 3. Add ‘twothird-mix’ type in calculating surface area. ‘twothird-mix’ is based on ‘two-third’, when the phases have the initial volume fraction (phi_o) < the threshold value (phinuc), the initial surface area (scalfac) is used as the surface area, otherwise, ‘two-third’ method is used. 4. Update algorithm of residual to norm for pardiso solver. Three algorithms are considered: (a) The L^1 vector norm, (b) the L^2 (Euclidean) vector norm, and (3) the L^infinity vector norm. Current L^2 (Euclidean) vector norm is used in Pardiso solver.			
Usage of New Features			**Remarks**

1. Usage of feature 1: Add the following commands to Data Block 7 *‘permeability scaling factor threshold maximum’* *1.0d10* *‘permeability scaling factor threshold minimum’* *1.0d-10* 2. Usage of feature 2: Add the following commands to Data Block 7: *‘update porosity’* *‘porosity threshold maximum’* *0.9999d0* *‘porosity threshold minimum’* *0.0001d0* 3. Usage of feature 3: Example in section of mineral input in Data Block 13: *0.00d0 .false. ‘twothird-mix’ ;phim, minequil, update_type - ‘calcite’* *1.d-10 1.0d-1 0.00d0 1.0d-7 ;phimin, scalfac, supsatm, phinuc*	Usage of feature 4 is not open yet.
Codes Update	**Remarks**
Benchmarks Updates	**Remarks**
Documentations Update	**Remarks**

Bug Fixes	**Remarks**
Notes	**Remarks**
1. To feature 2: The hydraulic conductivity exported to the tecplot is following the formula : “condzz(nzn)*perm_fac(ivol)/sec_per_days. For the previous version, the exported hydraulic conductivity is not strict the same as that used in the internal calculation. When porosity is 0 or 1, perm_fac will not updated to avoid division by 0 in Kozeny-Carman equation. 2. To feature 3: This is an alternative method to apply ‘twothird’ and ‘constant’ for the same minerals to different zones. At present, update-type is mineral-dependent, say, the same mineral has the same update-type (e.g, ‘constant’) for different zones. For example, if we have the following zone 1 and zone 2. In zone 1, the initial volume fraction is 0 < 1.0d-7, then the initial surface 1.0d-1 is used. In zone 2, the initial volume fraction 0.25 > 1.0d-7, the surface area is calculated based on ‘two-third’ method. Zone 1 0.00d0 .false. ‘twothird-mix’ ;phim, minequil, update_type - ‘calcite’ 1.d-10 1.0d-1 0.00d0 1.0d-7 ;phimin, scalfac, supsatm, phinuc zone 2 0.25d0 .false. ‘twothird-mix’ ;phim, minequil, update_type - ‘calcite’ 1.d-10 1.0d-1 0.00d0 1.0d-7 ;phimin, scalfac, supsatm, phinuc	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference except the following benchmarks: *.gss, *.gsd and *.mve files in benchmark perm2 *.gbc, *.gbd, *.gbs, *.gsd, *.gss, *.mac, *.mmc. *.gsv and *.mve files (not all) in benchmark perm1, perm1_new, perm2_new, perm3 and perm3_new.	

Revision 63

Revision Number	63	**Version Number**	V1.0.63
Commit Date	2013-04-08	**Commit Author**	Mingliang Xie
Software Grade		signature	Date
Verified by	Danyang Su		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.1 2. Configurations: Default			
System Requirements			**Remarks**
New Features			**Remarks**
1. Total sorbed mass calculation: previous version valid only for one material zone. Improvement is the extension and valid for different zones or distributed case. 2. Added Harmonic weighting of effective diffusion coefficient for radial and 1D 2D & 3D Cartesian coordinate. 3. Added verification benchmarks.			
Usage of New Features			**Remarks**
1. Usage of features 1: no change of the input file is required. 2. Usage of features 2: A. arithmetic weighting of effective diffusion coefficient (De). De averaging is undertaken in the way:			

1. calculate por_av=[por(ivol)+por(jvol)]/2,

2. calculate De_av=tau_av*por_av*sat_av*D0

by adding the following keywords 'spatial averaging - diffusion' or *‘averaging diffusion’* and subkeywords in Data Block 7: control parameters – reactive transport:

'spatial averaging - diffusion' *; Keyword*

*‘arithmetic’ ; De_av = D0*por_av*tau_av*

B. arithmetic averaging of effective diffusion coefficient (De). De averaging is undertaken in the way:

1. calculate calculate De(ivol) and De(jvol),

2. calculate De_av=[De(ivol)+De(jvol)]/2

by adding the following keywords and subkeywords in “! Data Block 7: control parameters – reactive transport”:

'spatial averaging - diffusion' *; Keyword*

*‘arithmetic De’ ; De_av=[De(i)+De(j)]/2, in which De(i)=D0*por_i*tau_i*

C. harmonic averaging of effective diffusion coefficient (De). De averaging is undertaken in the way:

1. calculate calculate De(ivol) and De(jvol),

2. calculate De_av=[De(ivol)*De(jvol)*(delx_i+delx_j)]/[De(ivol)*delx_i+De(jvol)*delx_j]

by adding the following keywords and subkeywords in “! Data Block 7: control parameters – reactive transport”: 'spatial averaging - diffusion' *; Keyword* *‘harmonic’*	
Codes Updates	**Remarks**
Benchmarks Updates	**Remarks**
Added nine benchmarks under .\benchmark_standard\reactran\diff_harmonic\	
Documentations Update	**Remarks**
1. Updated the user manual 2. Updated the verification report	
Bug Fixes	**Remarks**
none	
Notes	**Remarks**
1. It is recommended to use the averaging methods for porosity and diffusion coefficient in a consistent way. That is to say, if harmonic averaging of De is desired, the porosity should be averaged also harmonically by just adding ‘harmonic average in porosity’ in the Data Block 7: control parameters – reactive transport. 2. If the keyword ‘averaging diffusion’ is not provided in the input file, the default method A (arithmetic averaging) is applied.	
Summary of code verification (test running)	**Remarks**
No difference of the results simulated using the current version and the previous version V1.0.56.	

Revision 56

Revision Number	56	**Version Number**	V1.0.56
Commit Date	2013-03-08	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.1 2. Configurations: Default	
System Requirements	**Remarks**
New Features	**Remarks**
1. Add biomass growth codes to MIN3P-THCm. 2. Update reactive surface area for linear relationship with mass volume fraction. The reference surface area is calculated by S(ref) = S(phi_max) * phi_nuc / phi_max, where phi_max is the given bulk volume fraction and S(phi_max) is the given surface area under the phi_max. 3. Add new input parameter ‘consider exponent n for reaction rate law’. 4. Write version informations to the output files. 5. Add porosity threshold to avoid division by 0 in Kozeny-Carman equation.	

6. Add tortuosity update. Current tortuosity is updated as Tau = tau_0 por^alpha/por_o^alpha or Tau = por^alpha 7. Assign tortuosity factor (alpha, archie) to different zones. Previous codes apply the same factor to the whole domain when tortuosity correction is set to ‘archie’.	
Usage of New Features	**Remarks**
1. Usage of feature 1: Add ‘biomass components’ to data block 2. Example input is as follows : *‘components’* *12 ;number of components* *‘h+1’ ;component names* *‘na+1’* *…* *‘biomass components’* *2 ;number of biomass components* *‘c5h7o2n’ ;biomass component names* *‘c5h7o2n(d)’* 2. Usage of feature 2: Set update type of mineral to ‘linear’ and define the value of nucleation threshold, e.g., 1.0E-7. Example input is as follows: *‘mineral input’* *0.018879 .false. ‘linear’ ;phim, minequil, update_type - c3fh6* *1.d-10 1.0d-9 0.00d0 1.0d-7 ;phimin, scalfac, supsatm,*	

```
phic_nuc

  0.127803  .false. ‘linear’                  ;phim, minequil, update_type  -
csh(1.6)

  1.d-10    1.0d-9    0.00d0    1.0d-7   ;phimin, area, supsatm,
phic_nuc
```

3. Usage of feature 3:

This parameter is read from the input file. Default setting does not consider exponent n for reaction rate law. Add the command ‘consider exponent n for reaction rate law’ to ‘Data Block 2: geochemical system’ if exponent n is required in reaction rate law. Example input is as follows:

```
 ! Data Block 2: geochemical system

 ‘geochemical system’

 ‘compute alkalinity’

 ‘consider exponent n for reaction rate law’

 ‘use new database format’
```

4. Usage of feature 4:

By default, the program only write version information to the log file and gen file. If you want to write version information to the tecplot data file, please add the command ‘write version to tecplot data file’ to “Data Block 1: global control parameters”. Example input is as follows:

```
 ! Data Block 1: global control parameters

 ‘write version to tecplot data file’
```

5. Usage of feature 5:

Add ‘porosity threshold’ to Data Block 7. Example input is as follows: *‘porosity threshold’* *0.99d0* If porosity threshold is not set in the input file, then the default value 1.0d0 will be used. 6. Usage of feature 6 and feature 7: Add ‘update tortuosity’ to block 7 and alpha value to block 9. Example input is as follows: !Data Block 7: control parameters – reactive transport *‘tortuosity correction’* *‘assigned tau’* *‘update tortuosity’* !Data Block 9: physical parameters – porous medium *‘number and name of zone’* *1* *‘concrete’* *0.1 ;porosity* *0.0383 ;tortuosity* *2.0 ;alpha, tortuosity update factor* tau=tau_0*(por/por_0)^alpha	
Codes Update	**Remarks**

Benchmarks Update	**Remarks**
Documentations Update	**Remarks**
Bug Fixes	**Remarks**

1. Fix bug in calculating ratemp in modrate.F90.
`if (dabs(ratem).lt.tinyrate) then`
`ratemp(ireac) = r0`
`end if`
`...`
`ratemp(ireac) = ratemp(ireac) * ratem_mod/ratem`
The default tinyrate value is 0 in the codes, without the command line ‘define minimum reaction rate’, the above codes will cause NaN in ratemp when ratem equals 0. Change the default value of tinyrate from 0 to 1.0D-300 to avoid division by 0.
2. Fix bug in calculating diffusion coefficient use ‘archie’ in diffcoff.F90.
previous codes:
`case(‘archie’)`
`diffcoff = diff_p * sat_p * por**m_archie`
current codes:
`case(‘archie’)`
`diffcoff = diff_p * sat_p * por**(m_archie + 1)`	

3. Fix bug in calculating satav in infcrtdd.F90 previous codes: satav=dmin1(r1, sanew(ivol)) !ivol is not corrent, should use jvol current codes: satav=dmin1(r1, sanew(jvol))	
Notes	**Remarks**
1. Modification in calculating prodrc (activity product of species involved in reaction) and prodrcinc (activity product of species involved in reaction (incremented)). For an irreversible dissolution reaction, the reaction should be excluded, if (1-IAP/K) < 0, no matter what prodrcinc is. For an irreversible precipitation reaction, the reaction should be excluded, if (1-IAP/K) > 0, no matter what prodrcinc is. 2. Add the routines for switching the (1-IAP/K) term (alternative formulation). 3. If exponent n is considered, you will find “consider exponent n for reaction rate law: true” in the log file, otherwise, you will find “consider exponent n for reaction rate law: false”. If exponent n is not considered but the supplied exponent n is not 1, then you will find warning information “WARNING: supplied exponent n will be ignored and is set to 1” in the log file.	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference except the following benchmarks: *.gbd and *.bgs files in benchmarks prb *.gsd, *.gss and *.gsv and/or *.mve files in d51_dedolomitization and perm2 More different files in d6_rt_highly_saline, amd_ex, perm1, perm1_new, perm2_new, perm3, perm3_new This difference is caused by the bug fixes.	

Revision 37

Revision Number	37	**Version Number**	V1.0.37
Commit Date	2013-01-31	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.1 2. Configurations: Default			
System Requirements			**Remarks**
New Features			**Remarks**
1. Add runtime statistics analysis of 291acobi matrix construction, factorization and substitution for different equations. 2. Add Pardiso parallel solver (factorization and substitution) and solver configuration. Further check is needed to correct the error and improve the parallel efficiency. Not recommend to use at present. 3. Modify read statement for complex and gases so as to support more than 5 components. 4. Modify the formula in calculating prodrc and prodrcinc by taking an absolute value term and switch signs. The previous formula may cause NaN problem when n is not equal 1. 5. Modify input file read, add the function getarg to read in arguments, making the program able to start from command line. Command format: min3p.exe inputfile. This feature is required in automatic			

running benchmark and result compare tool. 6. Add version stamp (revision, build time) to the start logo of the program.	
Usage of New Features	**Remarks**
1. Usage of feature 1 and feature 2: Current feature 1 and feature 2 is configured in an external file solver.cfg. This file should be put together in the same location as MIN3P input file. This file can be found in source codes. Example of solver.cfg is as follows: ! ** ! Block 1: Golbal solver setting ! ** ! Select solver ! i_solver_type = 0 use ws209 solver ! 1 use pardiso solver SOLVER TYPE 1 ! ** ! Block 2: Pardiso solver setting ! ** ! Set this parameter if you want to check the result of matrix ! solver with ws209. Only valid if i_solver_type = 0 ! PARDISO: SOLVER TEST WITH WS209	

```
! Set the number of threads for pardiso solver.
! If the solver type is not pardiso, ignore it.
! If the number of threads is less than 1, use dynamic
! number of threads determined by pardiso.
PARDISO: NUMBER OF THREADS
1

! Maximum number of iterative refinement steps that the
! solver will perform. The solver will perform not more
! than the absolute value of this parameter for iterative
! refinement and will stop the process if a satisfactory
! level of accuracy of the solution in terms of backward
! error has been achieved.
! If this parameter is negative, the accumulation of the
! residuum is using extended precision real types.
! By default, the value is 9
PARDISO: MAX ITERATIVE REFINEMENT STEPS IN FLOW
-100

PARDISO: MAX ITERATIVE REFINEMENT STEPS IN REACTIVE TRANSPORT
-100

! This parameter instructs PARDISO how to handle small
! pivots or zero pivots for unsymmetric matrices.
```

! It indicates the iterative refinement contraction ! rate. The default value is 13, which means eps = 10^(-13) ! is used in handling small pivots. PARDISO: PIVOTING PERTURBATION IN FLOW 13 PARDISO: PIVOTING PERTURBATION IN REACTIVE 13 ! ** ! Block 3: Output setting ! ** ! Set if output detail runtime statistics analysis. ! Comment out if you do not want to export it. OUTPUT RUNTIME STATISTICS ANALYSIS	
Codes Updates	**Remarks**
Benchmarks Updates	**Remarks**
Documentations Update	**Remarks**
Bug Fixes	**Remarks**
1. Modify syntax warning in fortran output format. The output format like ‘(a,i4,a,e10.3)/’ is not correct, should use ‘(a,i4,a,e10.3,/)’ instead.	
Notes	**Remarks**

1. New tool added: mtxView.exe. This tool is used for sparse matrix visualization. Input data should be matrix market exchange formats (zlib compress of data file is acceptable). 2. Add project folder visualstudio_p for parallel development environment. It shares the same source codes with serial development environment.	
Summary of code verification (test running)	**Remarks**
Comparison of the results for all examples obtained by the current and the previous MIN3P-THCm versions showed generally no difference except the following: *.gbm file in benchmarks diffdry, diffvar, and diffwet. The difference is caused by some uninitialized variables. These variables should be initialized by zero; otherwise, it will use a very small value close to zero instead. This bug will cause some balance error in the last column of the file. Example of difference Line: 5, REFERENCE: 0.0000000E+00 0.4327933-310 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.2500000E+02 0.2079485E+00 Line: 5, CURRENT: 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.2500000E+02 0.0000000E+00	

Revision 31

Revision Number	31	**Version Number**	V1.0.31
Commit Date	2013-01-29	**Commit Author**	Mingliang Xie
Software Grade		signature	Date
Verified by	Danyang Su		
Approved by	Uli Mayer		

Compiler and configurations	**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.1 2. Configurations: Default	
System Requirements	**Remarks**
New Features	**Remarks**
1. Add multicomponent diffusion (MCD) 2. Extension of radial coordinate for MCD	
Usage of New Features	**Remarks**
1. In Data Block 1: global control parameters add keyword: ‘multicomponent diffusion’ 2. Use special database including diffusion coefficient for each primary and secondary species. Example of the added D_0 value in the database comp.dbs by the end of each line in [m^2/s] ca+2 2.0 6.00 .17 40.08000 .00 0.792d-9 cd+2 2.0 .00 .00 112.39940 .00 0.719d-9 cl-1 -1.0 3.00 .01 35.45300 .00 2.032d-9 Example of the added D_0 value in the database complex.dbs by the end	The D_0 values are derived from the CRC handbook; If it is unknown, 2.0e-9 m^2/s is temporarily applied.

of each line in [m^2/s] oh- 13.3620 -13.9980 -1.00 3.50 .00 17.0074 1.00 *5.273d-9* 2 h2o 1.000 h+1 -1.000	
Codes Updates	**Remarks**
Benchmarks Updates	**Remarks**
Documentations Update	**Remarks**
1. Updated the theory manual 2. Updated the user manual 3. Updated the verification manual	
Bug Fixes	**Remarks**
1. Fix a bug in volume calculation for radial coordinate. 2. Fix bugs in reading enthalpy change for gases (readgses.F90), for sorbed species (readsorb.F90) and for minerals (readmin_new.F90) 3. Fix a bug for the saturation update after each timestep in file nexttime.F90.	
Notes	**Remarks**
1. The start logo version number is V1.1. This is not the version number generated by VisualSVN. 2. Five benchmarks were added, four of them were verified against CrunchFlow and/or PHREEQC. The one in radial-coordinate cannot be run by the CrunchFlow version that we got.	
Summary of code verification (test running)	
The difference of the results simulated using the current version and the previous version V1.0.7 is for the following benchmarks: 1. Diffdry: outputfile *.gsm, parameter Eh from 1.089e-2 (previous version) to 0.432e-310 (current version). The pe values are in the order of 1e-322, slightly change. According to the Eh=0.198*tempK*pe, the Eh value is now consistent to the pe value. The cause is the second bug fix. Similar problems are	

found for benchmarks diffvar and diffwet. 2. polyhal-pitz: The output files for the initial conditions are not changed, but the values in the output files thereafter showed small difference. For example, the maximal relative error for *_*.gst is much less than 0.016% (sulphate concentration) at 10 years. The second bug fixes resulted in the change.	

Revision 7

Revision Number	7	**Version Number**	V1.0.7
Commit Date	2012-12-10	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.1 2. Configurations: Default			
System Requirements			**Remarks**
New Features			**Remarks**
1. Add Version Information to the start logo. When MIN3P starts, it shows version and build date. 2. Add pre-build event to fortran configure for both release and debug under win32 and x64 environment.			
Usage of New Features			**Remarks**
Codes Update			**Remarks**
Benchmarks Update			**Remarks**
Documentations Update			**Remarks**
Bug Fixes			**Remarks**

Notes	**Remarks**

Revision 6

Revision Number	6	**Version Number**	V1.0.6
Commit Date	2012-12-10	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.1 2. Configurations: Default			
System Requirements			**Remarks**
New Features			**Remarks**
Usage of New Features			**Remarks**
Codes Update			**Remarks**
Benchmarks Updates			**Remarks**
Documentations Update			**Remarks**
Bug Fixes			**Remarks**
Notes			**Remarks**
Ignore the version information of code file Version.F90.			

This file is automatically generated during pre-build event based on the template VersionTemplate.F90. Every time you rebuild your project, the Version information will be changed. This would cause conflicts between different user so that this file is neglected in the VisualSVN version control. Please note: the version information in Version.F90 is generated by SubWCRev.exe, which is integrated into TortoiseSVN program.	

Revision 3

Revision Number	3	**Version Number**	V1.0.3
Commit Date	2012-12-09	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and Configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.1 2. Configurations: Default			
System Requirements			**Remarks**
New Features			**Remarks**
1. Add VisualSVN Keywords into project. 2. Add VisualSVN_Server_Client_Setting.docx to the repository. Path: docs/configure.			
Usage of New Features			**Remarks**
Codes Update			**Remarks**
Benchmarks Update			**Remarks**
Documentations Update			**Remarks**
Bug Fixes			**Remarks**

Notes	Remarks

Revision 2

Revision Number	2	**Version Number**	V1.0.2
Commit Date	2012-12-09	**Commit Author**	Danyang Su
Software Grade		signature	Date
Verified by	Mingliang Xie		
Approved by	Uli Mayer		
Compiler and Configurations			**Remarks**
1. Compiler: Intel(R) Visual Fortran Compiler XE 13.0.1 2. Configurations: Default			
System Requirements			**Remarks**
New Features			**Remarks**
1. Add cross-platform configuration into the project using preprocessor. The project can be compiled under Windows and CYGWIN. 2. Add version control information and embed version information into source codes. 3. Add Doxygen format into source codes.			
Usage of New Features			**Remarks**
Codes Update			**Remarks**
Benchmarks Update			**Remarks**
Documentations Update			**Remarks**

Bug Fixes	**Remarks**
Notes	**Remarks**
This is the initial repository of MIN3P-THCm development project. This project is based on MIN3P-THCm V1.0. PLEASE CREATE YOUR OWN BRANCH BEFORE MAKING BIG CHANGES. Repository: https://biot.eos.ubc.ca/svn/MIN3P-THCm/trunk	

Revision 1

Revision Number	1	**Version Number**	-
Commit Date	2012-11-29	**Commit Author**	VisualSVN Server
Software Grade			
Compiler and Configurations			**Remarks**
System Requirements			**Remarks**
New Features			**Remarks**
Usage of New Features			**Remarks**
Codes Update			**Remarks**
Benchmarks Update			**Remarks**
Documentations Update			**Remarks**
Bug Fixes			**Remarks**
Notes			**Remarks**
This is the initial structure of MIN3P-THCm repository.			

Revision Number		**Version Number**	
Commit Date		**Commit Author**	
Software Grade			
Compiler and Configurations			**Remarks**
System Requirements			**Remarks**
New Features			**Remarks**
Usage of New Features			**Remarks**
Codes Update			**Remarks**
Benchmarks Update			**Remarks**
Documentations Update			**Remarks**
Bug Fixes			**Remarks**
Notes			**Remarks**

APPENDIX: List of benchmarks

In the validation report, three types of verification approaches are used.

1. Verification is performed through comparison of simulated results with analytical solutions.
2. Verification is undertaken through code intercomparisons, i.e. the comparison of simulated results obtained by MIN3P-THCm and by published code(s) with equivalent functions.
3. Verification by comparison of MIN3P-THCm results to experimental data

Only type 3 simulations can be classified as validation cases.

Demonstration examples are indicated as D.

The examples included in the benchmarking are listed in the following six tables organized according to the tested capabilities of MIN3P-THCm including batch reaction (Table 1), saturated and unsaturated flow (Table 2), density dependent flow (Table 3), energy balance (Table 4), reactive transport (including diffusion, ion exchange, multisite ion exchange, multicomponent diffusion, mineral dissolution/precipitation, oxidation, retardation and biodegradation etc. (Table 5), hydromechanical coupling (Table 7), and unstructured grid cases (Table 8).

It is important to note that the parallel version using domain decomposition method cannot run the batch examples because the total grid number should be at least two for each processor.

Table 1: List of benchmarking problems for batch reactions

Name	Path	Description of Main Features	Verification Type	Version required
Examples under the folder: .\benchmarks_standard\				
As(V)_surfx	.\batch\as(V)_surfx	Surface complexation	2	V1.0.0$^+$
Surfa	.\batch\surfa	pH-dependent anion surface complexation	2	V1.0.0$^+$
Surfme	.\batch\surfme	pH-dependent cation surface complexation	2	V1.0.0$^+$
albite	.\batch\albite	weathering of calcite (fast) and albite (slow)	D	V1.0.0$^+$
appelo	.\batch\appelo	kinetically controlled dissolution of calcite	D	V1.0.0$^+$
As(III)	.\batch\As(III)	As(III)-speciation from pH = 0-14	D	V1.0.0$^+$
As(V)	.\batch\As(V)	As(V)-speciation from pH = 0-14	D	V1.0.0$^+$
Cr(III)	.\batch\cr(III)	Cr(III)-speciation from pH = 0-14	D	V1.0.0$^+$
Fe(III)	.\batch\Fe(III)	Fe(III)-speciation from pH = 0-14	D	V1.0.0$^+$
linsorb	.\batch\linsorb	dissolution and sorption of pce - linear isotherm	D	V1.0.0$^+$
nh3	.\batch\nh3\	NH_3-speciation from pH = 0-14	D	V1.0.0$^+$
Se(-II)	.\batch\Se(-II)	Se(-II)-speciation from pH = 0-14	D	V1.0.0$^+$
Se(IV)	.\batch\Se(IV)	Se(IV)-speciation from pH = 0-14	D	V1.0.0$^+$
Se(VI)	.\batch\Se(VI)	Se(VI)-speciation from pH = 0-14	D	V1.0.0$^+$
Surfa2	\batch\surfa2	pH-dependent anion surface complexation	D	V1.0.0$^+$
Surfme2	.\batch\surfme2	pH-dependent cation surface complexation	D	V1.0.0$^+$

Examples under .\nwmo_verification_examples\				
Pitzer	\d2_verification_pitzer_equations	Pitzer equations for reactions in highly saline solution	2	V1.0.0$^+$

V1.0.0$^+$ means the MIN3P-THCm version V1.0.0 or later is required to run this example

Table 2 List of benchmarking problems for saturated and unsaturated flow

Name	Path	Description of Main Features	Verification Type	Version required
Examples under the folder: .\benchmarks_standard\				
drain1	.\flow\drain1	1D gravity drainage in soil column	3	V1.0.0+
infil1	.\flow\infil1	1D infiltration in soil column	3	V1.0.0+
Haverk	.\flow\haverk	1D drainage problem in sand column	2, 3	V1.0.0+
clement	.\flow\clement	2D water table mounding	3	V1.0.0+
verif-evap	.\flow\verif-evap	1D water vapor movement	3	V1.0.0+
Clem3d	.\flow\clem3d	Infiltration and 3D transient water table mounding	D	V1.0.0+
Clement-transient	.\flow\clement-transient	2D transient water table mounding	D	V1.0.0+
shlomo	.\flow\shlomo	2D variably-saturated flow with seepage face – steady state solution	D	V1.0.0$^+$
shlomot	.\flow\shlomot	2D variably-saturated flow with seepage face - transient solution	D	V1.0.0$^+$
stedfs	.\flow\stedfs	2D steady state saturated flow	D	V1.0.0$^+$
stedvs	.\flow\stedvs	2D steady state variably-saturated flow	D	V1.0.0$^+$
tranfs	.\flow\tranfs	2D transient saturated flow	D	V1.0.0$^+$
tranvs	.\flow\tranvs	2D transient variably-saturated flow	D	V1.0.0$^+$
Examples under the folder: .\benchmarks_standard\reactran				
diff_h_cc_2D_H	.\diff_harmonic\diffh_h_cc_2D_H	Spatial weighting of hydraulic conductivity using	D	V1.0.63$^+$

		‘harmonic' method, 2D case		
diff_h_cc_3D_H	.\diff_harmonic\diffh_h_cc_3D_H	Spatial weighting of hydraulic conductivity using ‘harmonic' method, 3D case	D	V1.0.63$^+$
diffh-1D-H	.\diff_harmonic\diffh-rc-H	Spatial weighting of hydraulic conductivity using ‘harmonic' method, 1D case, radial coordinate	D	V1.0.63$^+$

V1.0.63$^+$ means the MIN3P-THCm version V1.0.63 or higher is required to run this example

Table 3 List of verification problems for density dependent flow (DDF)

Name	Path	Description of Main Features	Verification Type	Version required
Examples under the folder: .\benchmarking_nwmo_report\				
elder	.\nwmo_verification_examples\d32_elder_problem	Elder problem (2D fluid convection driven by a thermal gradient)	2	V1.0.0$^+$
henry	.\d33_modified_henry_problem	Modified Henry problem (2D lateral saltwater intrusion into aquifer)	1	V1.0.0$^+$
Diersch (coupled)	.\nwmo_verification_examples\ d311_hydrostatic_box_test\coupled	2D hydrostatic box problem, $d\rho/dc=0.7$	D	V1.0.0$^+$
Diersch (decoupled)	.\nwmo_verification_examples\ d311_hydrostatic_box_test\decoupled	2D hydrostatic box problem, $d\rho/dc=0.0$	D	V1.0.0$^+$
voss	.\nwmo_verification_examples\ d312_hydrodynamic_box_problem	Box 2 SEAWAT example	D	V1.0.0$^+$

V1.0.0$^+$ means the MIN3P-THCm version V1.0.0 or later is required to run this example

Table 4 List of verification problems for energy balance

Name	Path	Description of Main Features	Verification Type	Version required
Examples under the folder: .\benchmarking_nwmo_report\nwmo_verification_examples_D4				
radial-flow	.\d41_radial_flow_energy	flow with energy transport (1D radial)	1	V1.0.0+
verif-sutra	.\d42_density_dep_energy\centered\injection	aquifer thermal energy storage (2D radial), centered spatial weighting, injection	2	V1.0.0+
verif-sutra	.\d42_density_dep_energy\centered\pumping	aquifer thermal energy storage (2D radial), centered spatial weighting, pumping	2	V1.0.0+
verif-sutra	.\d42_density_dep_energy\upstream\injection	aquifer thermal energy storage (2D radial), upstream spatial weighting, injection	2	V1.0.0+
verif-sutra	.\d42_density_dep_energy\upstream\pumping	aquifer thermal energy storage (2D radial), upstream spatial weighting, pumping	2	V1.0.0+
henry-hilleke	.\d43_density_dep_heat_solute	Henry-Hilleke problem (2D)	1 and 2	V1.0.0+
thermohaline	.\d44_thermo_haline_convection\ negative_buoyancy	Thermal-haline convection (2D), negative initial buoyancy	2	V1.0.0+
thermohaline	.\d44_thermo_haline_convection\ positive_buoyancy	Thermal-haline convection (2D), positive initial buoyancy	2	V1.0.0+
Examples under the folder: .\benchmarking_nwmo_report\				
Salt-dome	.\nwmo_verification_examples_D45\ d45_salt_dome_problem\d80_b5	Salt dome problem, *Δ*T=80, B=5	2	V1.0.0+
Salt-dome	.\nwmo_verification_examples_D452\d200_b2	Salt dome problem, *Δ*T=200, B=2	2	V1.0.0+
Salt-dome	.\nwmo_verification_examples_D453\d133_b3	Salt dome problem, *Δ*T=133, B=3	2	V1.0.0+

V1.0.0$^+$ means the MIN3P-THCm version V1.0.0 or later is required to run this example

Table 5 List of verification problems for reactive transport

Name	Path	Description of Main Features	Verification Type	Version required
Examples under the folder: .\benchmarks_standard\reactran\				
diff_rc_s	.\diff_rc_s	Diffusion, steady-state, radial coordinates	1	V1.0.0$^+$
diff_rc_t	.\diff_rc_t	Diffusion, transient, radial coordinates	1	V1.0.0$^+$
disper_rc	.\disper_rc	Diffusion and dispersion, radial coordinates	1	V1.0.0$^+$
diffdry	.\diffdry	Gas diffusion, constant moisture content, dry conditions	1	V1.0.0$^+$
diffvar	.\diffvar	Gas diffusion, variable moisture content	1 and 2	V1.0.0$^+$
diffwet	.\diffwet	Gas diffusion, constant moisture content, wet conditions	2	V1.0.0$^+$
ex11	.\ex11	Cation exchange – comparison with PHREEQC	2	V1.0.0$^+$
surfx	.\surfx	Trace metal mobility - surface complexation	2	V1.0.0$^+$
pyrox	.\pyrox	Pyrite oxidation and gas diffusion in vadose zone	2	V1.0.0$^+$
perm1_new	.\perm1_new	Porosity-permeability changes – permeability enhancement	2	V1.0.63+
perm2_new	.\perm2_new	Porosity-permeability changes - clogging with simple chemistry	2	V1.0.63+
perm3_new	.\perm3_new	Porosity-permeability changes – clogging with complex chemistry	2	V1.0.63+
retardation	.\retardation	Linear retardation of PCE and benzene	1	V1.0.0$^+$
MCD-1	.\MCD-1	Fick's Law	2	V1.0.31+
MCD-2	.\MCD-2	Multicomponent diffusion (MCD) – diffusion only	2	V1.0.31+
MCD-2-1	.\MCD-2-1	Multicomponent diffusion (MCD) – diffusion only	2	V1.0.31+
MCD-2-adv	.\MCD-2-advection	Multicomponent diffusion (MCD) – diffusion and advection	2	V1.0.31+

MCD-2-pH	.\MCD-2-high-pH	Multicomponent diffusion (MCD) – diffusion only, high pH	2	V1.0.31+
MCD-rc	.\MCD-2-rc	MCD in radial coordinate	2	V1.0.31+
Examples under the folder: .\benchmarking_nwmo_report\nwmo_verification_examples_D5\				
caex	.\d51_dedolomitization\min3p-nwmo	Cation exchange – comparison with PHAST	2	V1.0.0$^+$
dedo	.\d52_cation_exchange\min3p	Dedolomization – mineral dissolution precipitation	2	V1.0.0$^+$
biod	.\d53_aerobic_degradation\min3p	Aerobic biodegradation of toluene	2	V1.0.0$^+$
Examples under the folder: .\benchmarks_new_add\multisite-ionx-V1.0.129\				
ionx-2m	.\ionx-m-2domains	Multisite ion exchange	2	V1.0.129$^+$
Examples under the folder: .\benchmarking_nwmo_report\nwmo_verification_examples_D6\				
polyhal-pitz	.\d6_rt_highly_saline\min3p	Reactive transport under highly saline conditions – Pitzer equations	2	V1.0.0$^+$
Examples under the folder: .\benchmarks_new_add\biomass-V1.0.56				
batch1	.\batch1	No growth batch simulation	2	V1.0.56$^+$
batch2	.\batch2	Biomass growth batch simulation	2	V1.0.56$^+$
column1	.\column1	Biomass growth column simulation	2	V1.0.56$^+$
column2	.\column2	Biomass growth and decay column simulation	2	V1.0.56$^+$
column3	.\column3	Microbially mediated chromium reduction under denitrifying conditions in a biostimulation column experiment	2	V1.0.56$^+$

Examples under the folder: .\benchmarks_new_add\mtterri-V1.0.431				
mtterri01	.\mtterri01	In-situ diffusion experiment in clay	3	V1.0.431^{+}
mtterri01Cs	.\mtterri01Cs	In-situ diffusion experiment in clay(Cs using MIE model)	3	V1.0.431^{+}
mtterri02	.\mtterri02	Effect of chemical perturbation on diffusion and retardation in clay	3	V1.0.431^{+}
mtterri02Cs	.\mtterri02Cs	Effect of chemical perturbation on diffusion and retardation in clay (Cs using MIE model)	3	V1.0.431^{+}
mttoc	.\mttoc	Back diffusion during overcoring	3	V1.0.431^{+}
mttocCs	.\mttocCs	Back diffusion during overcoring (Cs using MIE model)	3	V1.0.431^{+}
Examples under the folder: .\benchmarks_new_add\surfx-ionx-V1.0.106				
ionx	.\ionx-2-crunch-crunch-dbs-cec0.73	Ion exchange using CRUNCHFLOW database parameters	2	V1.0.106^{+}
ionx1	.\ionx-2-phreeqc-default-dbs-cec0.14	Ion exchange using PHREEQC default database parameters	2	V1.0.106^{+}
ionx2	.\ionx-2-phreeqc-default-dbs-cec0.73	Ion exchange using PHREEQC default database parameters	2	V1.0.106^{+}
ionx3	.\ionx-2-phreeqc-default-dbs-ph7-cec0.14	Ion exchange using PHREEQC default database parameters	2	V1.0.106^{+}
ionx-m	.\ionx-m	Multisite ion exchange (2 sites, single material domain)	2	V1.0.129^{+}
ionx-m2	.\ionx-m-2domains	Multisite ion exchange (2 sites, two material domains)	2	V1.0.129^{+}
ionx-m2m	.\ionx-multisite-2domains	Multisite ion exchange (3 sites, two material domains)	2	V1.0.129^{+}
ionx-md	.\ionx-multisite-2domains	Multisite ion exchange (3 sites, single material domain)	2	V1.0.129^{+}
surfx1	.\surfx-2-phreeqc-default-dbs	Surface complexation, using parameters based on the PHREEQC default database	2	V1.0.106^{+}

surfx_new	.\surfx-2-phreeqc-default-dbs-new	Surface complexation, equilibrating with fixed solution composition of ion-exchange	2	V1.0.106$^+$
surfx-ionx	.\surfx-ionx-phreeqc-default-dbs-cec0.73-eq	Surface complexation and ion exchange	2	V1.0.106$^+$
surfx-ionx-new	.\surfx-ionx-phreeqc default-dbs-cec0.73-eq-new	Surface complexation and ion exchange, equilibrating with fixed solution composition of ion-exchange	2	V1.0.106$^+$
Examples under the folder: .\benchmarks_new_add\uraninite-reoxidation-V1.0.56				
2d_un10	.\2d-run10	Uraninite reoxidation	2	V1.0.56$^+$
2d_un16	.\2d-run16	Uraninite reoxidation	2	V1.0.56$^+$
run1_new	.\run1_new	Uraninite reoxidation	2	V1.0.56$^+$

V1.0.56$^+$ means the MIN3P-THCm version V1.0.56 or later is required to run this example

Table 6 List of verification problems for reactive transport (continued)

Name	Path	Description of Main Features	Verification Type	Version required
Examples under the folder: .\benchmarks_standard\reactran\				
amd_ex	.\amd_ex	Generation and attenuation of acid mine drainage	D	V1.0.0$^+$
complex	.\complex	1D advective-dispersive transport with aqueous complexation	D	V1.0.0$^+$
comptran	.\comptran	1D advective-dispersive transport including Monod kinetics, degradation of organic contaminants	D	V1.0.0$^+$
degas	.\degas	1D advective-dispersive transport, degassing from a hydrocarbon spill	D	V1.0.0$^+$

diff-de-ave	.\diff_harmonic\diff_h_cc_de_ave	Spatial weighting of diffusion using ‘arithmetic De' method	4	V1.0.63^{+}
diffh-cc	.\diff_harmonic\diffh_cc	Spatial weighting of diffusion using ‘harmonic' method	4	V1.0.63^{+}
diff-noHarm	.\diff_harmonic\diffh_noHarm	Spatial weighting of diffusion using ‘arithmetic’ method	4	V1.0.63^{+}
diff_h_2D	.\diff_harmonic\diffh_h_cc_2D_C	Spatial weighting of diffusion using ‘harmonic' method, 2D case	D	V1.0.63^{+}
diff_h_3D	.\diff_harmonic\diffh_h_cc_3D_C	Spatial weighting of diffusion using ‘harmonic' method, 3D case	D	V1.0.63^{+}
diffh-rc-C	.\diff_harmonic\diffh-rc-C	Spatial weighting of diffusion using ‘harmonic' method, radial coordinate	D	V1.0.63^{+}
diffh-rc-DD	.\diff_harmonic\diffh-rc-DD	Spatial weighting of diffusion using ‘harmonic' method, radial coordinate, density dependent flow	D	V1.0.63^{+}
diffh-cc-DD	.\diff_harmonic\diffh-rc-DD	Spatial weighting of diffusion using ‘harmonic' method, Cartesian coordinates, density dependent flow	D	V1.0.63^{+}
dissol	.\dissol	1D reactive transport - calcite dissolution	D	V1.0.0^{+}
perm1	.\perm1	Porosity-permeability changes – permeability enhancement	D	V1.0.0^{+}
perm2	.\perm2	Porosity-permeability changes - clogging with simple chemistry	D	V1.0.0^{+}
perm3	.\perm3	Porosity-permeability changes – clogging with complex chemistry	D	V1.0.0^{+}
prb	.\prb	Groundwater remediation by a reactive barrier	D	V1.0.0^{+}
raoult	.\raoult	Dissolution and volatilization of an organic contaminant mixture in variably saturated media	D	V1.0.0^{+}
Surfa2	.\surfa2	1D advective-dispersive transport with anionic surface complexation reaction	D	V1.0.0^{+}

Surfa2	.\surfme2	1D advective-dispersive transport with cationic surface complexation reaction	D	V1.0.0^{+}
transrc	.\surfa2	1D advective-dispersive transport with transient boundary conditions	D	V1.0.0^{+}
weather	.\weather	Mineral weathering in a soil column	D	V1.0.0^{+}
Examples under the folder: .\benchmarks_new_add\isotope-V1.0.303				
waybrant	.\Gibson et al. 2011	Isotope fraction (32S and 34S)	D	V1.0.303+
ykcolumn	.\Jamieson-Hanes et al 2012\Cr_columns - 3	Cr(VI) reduction in an organic carbon column (isotopes of Cr)	D	V1.0.303+
Cr-FTCb	\Jamieson-Hanes et al 2014\CrZVI_FTC - 3b	Cr(VI) treatment by zero-valent iron (isotopes of Cr)	D	V1.0.303+
Cr-FTCc	\Jamieson-Hanes et al 2014\CrZVI_FTC – 3c	Cr(VI) treatment by zero-valent iron (isotopes of Cr)	D	V1.0.303+
Examples under the folder: .\benchmarks_new_add\hMCD-V1.0.304				
hMCD	.\	Hybrid multicomponent diffusion model	D	V1.0.304+
Examples under the folder: .\benchmarks_new_add\sulfur-V1.0.431				
sulfur	.\sulfur	Salinity dependent SRB reaction (1D)	D	V1.0.431+
sul2d	.\2D-sal	Salinity dependent SRB reaction (2D)	D	V1.0.431+
sul2dice	.\2D-sal_ice	Salinity dependent SRB reaction (2D) and one cycle of glaciation/deglaciation	D	V1.0.431+
sul2dice4	.\2D-sal_ice4	Salinity dependent SRB reaction (2D) and multiple glaciation/deglaciation cycles	D	V1.0.431+
sul2pqc	.\verification\sul2pqc	Salinity dependent SRB reaction (1D) in single domain	2	V1.0.431+
sul2pqc	.\verification\sul2pqc-cos	Salinity dependent SRB reaction (1D) in single domain	2	V1.0.524+
sul2pqc-	.\verification\sul2pqc-3domains	Salinity dependent SRB reaction (1D) in three domains	2	V1.0.431+

3domains

Examples under the folder: .\benchmarks_new_add\surfx-ionx-unsat-V1.0.191				
ionx-drain1	.\ionx-drain1	Ion exchange reaction in unsaturated porous media	D	V1.0.191+
surfx-drain1	.\surfx-drain1	Surface complexation in unsaturated porous media	D	V1.0.191+
Examples under the folder: .\benchmarks_new_add\transient-boundary-V1.0.303				
gacc	.\co2-seq-gacc	Transient boundary condition for reactive transport provided by using 'update boundary conditions'	D	V1.0.303+
sine	.\co2-seq-sine	Transient boundary condition for reactive transport provided by Polynomial and sine functions	D	V1.0.303+
Examples under the folder: .\benchmarks_new_add\co2seq-water-freezing-V1.0.377				
freezing	.\co2seq-water-freezing-V1.0.377	Water freezing and flow restriction	D	V1.0.377$^+$
Examples under the folder: .\benchmarks_new_add\dusty-gas-model-V1.0.303				
amd505	.\amd505	Dusty gas model, acid mine generation in a carbonate-rich system	D	V1.0.303$^+$
bemdisp	.\bem600disp	Dusty gas model, gas advection		
bem205	.\bem20510e-2	Dusty gas model, 2D case with methanogenesis and oxidation	D	V1.0.303$^+$
deviss139	.\deviss139	Methane attenuation and multicomponent transport in landfill cover soils, Maxwell Stefan equations	3	V1.0.303$^+$
deviss141_r	.\deviss141_r	Methane attenuation and multicomponent transport in landfill cover soils, Dusty gas model	3	V1.0.303$^+$
fen22	.\fen22	Dusty gas model, nonreactive binary transport, diffusion in highly permeable media	2	V1.0.303$^+$
fen23	.\fen23	Dusty gas model, nonreactive binary transport, advection and diffusion in highly permeable media	2	V1.0.303$^+$

fen24	.\fen24	Dusty gas model, nonreactive binary transport, Knudsen diffusion in low-permeability media	2	V1.0.303^{+}
sch5010	.\sch5010	multicomponent transport in landfill cover soils, Dusty gas model	D	V1.0.303^{+}
thor45_t	.\thor45_t	Steady state multicomponent gas transport, dusty gas model	2	V1.0.303^{+}
Examples under the folder: .\benchmarks_new_add\gas_advection-V1.0.175				
gasadv	.\amd505dgm1_gas_adv	Gas advection, constant flux boundary	D	V1.0.175^{+}
gas_wt	.\amd505dgm1_gas_adv_water_table	Gas advection, constant hydraulic head boundary	D	V1.0.175^{+}
Examples under the folder: .\benchmarks_new_add\gas-bubble-V1.0.303\Jeen et al 2012\506-4				
gb_am	.\amd505dgm1_gas_adv	Gas bubble and entrapment	D	V1.0.303^{+}
gb_jeen	.\Jeen et al 2012\506-4	Gas bubble in granular iron column	D	V1.0.303^{+}
Examples under the folder: .\benchmarks_new_add\sit-clogging-V1.0.598				
sit	.\1D_sit_verif	1D diffusion with SIT model	2	V1.0.598+
benC1a	.\clogging	1D diffusion with clogging	D	V1.0.598+
Examples under the folder: .\benchmarks_new_add\perm3-clogging-V1.0.609				
perm3	.\	1D advection and diffusion with clogging	D	V1.0.609+
Examples under the folder: .\benchmarks_new_add\ mine-waste-attenuation-V2.0.0.724				
nszd-btw	.\	2D mine waste simulation	D	V2.0.0.724
Examples under the folder: .\benchmarks_new_add\segment-output-V2.0.0.724				
tracer21	.\	Example of 2D unstructured mesh with mass output through specified boundary segments	D	V2.0.0.724
Examples under the folder: .\benchmarks_new_add\tranvs-steady-V2.0.0.724				
tranvs	.\	Example of 2D steady state flow running in transient mode	D	V2.0.0.724

Table 7 List of a verification problems for hydromechanical coupling

Name	Description of Main Features	Verification Type	Version required
basin	1D hydromechanical coupling problem for the verification of the one-dimensional vertical stress implementation	1	V1.0.0$^+$

Table 8 List of demonstration problems for unstructured grid capabilities

Name	Path	Description of Main Features	Verification Type	Version required
Examples under the folder: .\benchmarks_usg\flow\				
clem3d	.\clem3d_hexa	3D variably saturated flow simulation using hexahedral mesh	2	V1.0.609$^+$
clem3d	.\clem3d_prism	3D variably saturated flow simulation using prism mesh	2	V1.0.609$^+$
clem3d	.\clem3d_usg	3D variably saturated flow simulation using tetrahedral mesh	2	V1.0.609$^+$
clement	.\clement_usg	2D transient water table mounding using triangle mesh	2	V1.0.609$^+$
shlomot	.\shlomot_usg	2D variably saturated flow simulation with seepage face - transient solution using triangle mesh	2	V1.0.609$^+$
stedfs	.\stedfs_usg	2D steady state full saturated flow simulation using triangle mesh	2	V1.0.609$^+$
stedvs	.\stedvs_usg	2D steady state variably saturated flow simulation using triangle mesh	2	V1.0.609$^+$
tranfs	.\tranfs_usg	2D full saturated flow simulation using triangle mesh	2	V1.0.609$^+$

tranvs	.\tranvs_quad	2D variably saturated flow simulation using quadrilateral mesh	2	V1.0.609^{+}
tranvs	.\tranvs_usg	2D variably saturated flow simulation using triangle mesh	2	V1.0.609^{+}
Examples under the folder: .\benchmarks_usg\react\				
clem3d	clem3d_usg	3D variably saturated flow with calcite dissolution simulation using tetrahedral mesh	D	V1.0.609^{+}
elder	d32_elder_problem_usg	2D elder problem using triangle mesh	2	V1.0.609^{+}
henry	d33_modified_henry_usg	2D henry problem using triangle mesh	2	V1.0.609^{+}
henry-hilleke	d43_density_dep_heat_solute_usg	2D henry-hilleke problem using triangle mesh, ‘voronoi’ diagram control volume	2	V1.0.609^{+}
henry-hilleke	d43_density_dep_heat_solute_usg_cc	2D henry-hilleke problem using triangle mesh, ‘cell center’ control volume	2	V1.0.609^{+}
henry-hilleke	d43_density_dep_heat_solute_usg_ls3md	2D henry-hilleke problem using triangle mesh, ‘median dual’ control volume and ‘least square third order’ gradient reconstruction	2	V1.0.609^{+}
free-thermohaline-convection	d44_thermo_haline_convection_usg\negative_buoyancy_usg	2D thermos haline convection problem using triangle mesh	2	V1.0.609^{+}
free-thermohaline-convection	d44_thermo_haline_convection_usg\positive_buoyancy_usg	2D thermos haline convection problem using triangle mesh	2	V1.0.609^{+}
salt-dome	d45_salt_dome_problem_d80_b5_usg	2D salt dome problem d80_b5 using triangle mesh	2	V1.0.609^{+}
salt-dome	d45_salt_dome_problem_d133_b3_usg	2D salt dome problem d133_b3 using triangle mesh	2	V1.0.609^{+}
salt-dome	d45_salt_dome_problem_d200_b2_usg	2D salt dome problem d200_b2 using triangle mesh	2	V1.0.609^{+}
diersch	d311_hydrostatic_box_test_usg\diersch_coupled_usg	2D hydrostatic box problem using triangle mesh	2	V1.0.609^{+}
diersch	d311_hydrostatic_box_test_usg\diersch_decoupled_usg	2D hydrostatic box problem using triangle mesh	2	V1.0.609^{+}

voss	d312_hydrodynamic_box_problem_usg	2D hydrodynamic box problem using triangle mesh	2	V1.0.609^{+}
diffh-2D-C	diff_h_cc_2D_C_usg	2D diffusion problem using triangle mesh	2	V1.0.609^{+}
het-2d	het_2d_complex_usg	2D calcite dissolution with complex geometry using triangle mesh	D	V1.0.609^{+}
het-2d	het_2d_usg	2D calcite dissolution using triangle mesh	D	V1.0.609^{+}
hetero-frac-slope	hetero_frac_slope_usg	2D calcite dissolution with complex geometry and anisotropic heterogeneous material properties using triangle mesh	D	V1.0.609^{+}
patchf	patchf_usg	3D solute transport using prism mesh	3	V1.0.609^{+}
pile	pile_hexa	3D calcite dissolution using hexahedral mesh for pile alike domain	D	V1.0.609^{+}
sand	ssbench4_sand_usg	2D ssbench problem using triangle mesh	2	V1.0.609^{+}
stripf	stripf_usg	2D slotue transport using triangle mesh	3	V1.0.609^{+}
tranrt	tranrt_quad	2D calcite dissolution using quadrilateral mesh	D	V1.0.609^{+}