



REFLECT DELIVERABLE D4.1

User's guide of a coupled hydro-thermal-chemical code for fluids



Summary:

This document presents the developed code, porousMedia4Foam, an open-source hydro-geochemical package that can decipher fluid flow and chemical reactive processes occurring within multi- {scale, dimensional, phase, compositional} environments. This report gives details regarding the installation of porousMedia4Foam, running the first simulations and describing many application cases in single-phase and multiphase systems.

Authors:

Saideep Pavuluri, BRGM, French Geological Survey
Laurent André, BRGM, French Geological Survey
Christophe Tournassat, ISTO
Francis Claret, BRGM, French Geological Survey
Cyprien Soullaine, ISTO



Title:	User's Guide of a Coupled Hydro-Thermal-Chemical Code for Fluids		
Lead beneficiary:	BRGM		
Due date:	31 July 2021		
Nature:	Public		
Diffusion:	All project partners, European Commission, general public		
Status:	Final		
Document code:	REFLECT_D4.1		
DOI:	http://doi.org/10.48440/gfz.4.8.2021.002		
License information:	CC-BY-4.0		
Recommended citation:	Pavuluri S., André L., Tournassat T., Claret F., Souleine C.: <i>The H2020 REFLECT project: Deliverable 4.1 - User's Guide of a Coupled Hydro-Thermal-Chemical Code for Fluids</i> , GFZ German Research Centre for Geosciences, http://doi.org/10.48440/gfz.4.8.2021.002		
ORCID:	Saideep Pavuluri: 0000-0002-6454-781X Laurent Andre: 0000-0002-8844-1585 Christophe Tournassat: 0000-0003-2379-431X Francis Claret: 0000-0002-6203-7795 Cyprien Souleine: 0000-0002-6151-3286		
Revision history	Author	Delivery date	Summary of changes and comments
Version 01	Pavuluri et al.	28/06/2021	First draft
Final version	Pavuluri et al.	23/07/2021	Implementing remarks and comments from P. Shoeibi Omrani

	Name	Function	Date
Deliverable responsible	Laurent ANDRE	Hydrogeochemist (BRGM)	23/07/2021
WP leader	Laurent ANDRE	Hydrogeochemist (BRGM)	23/07/2021
Reviewer	Pejman Shoeibi Omrani	Senior Scientist (TNO)	14/07/2021
Project Coordinator	Simona Regenspurg	Project coordinator (GFZ)	26/07/2021

This document reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

porousMedia4Foam

USER GUIDE

VERSION 1

AUTHORS:

Pavuluri Saideep (saideep025@gmail.com)

André Laurent (l.andre@brgm.fr)

Tournassat Christophe (christophe.tournassat@univ-orleans.fr)

Claret Francis (f.claret@brgm.fr)

Soulaine Cyprien (cyprien.soulaine@cnrs-orleans.fr)

Table of Contents

1	Brief Introduction to <i>porousMedia4Foam</i>	3
1.1	Reactive Transport Modelling	4
1.2	Description about <i>porousMedia4Foam</i>	4
1.3	Structure of the user manual	5
2	Installing and getting started with <i>porousMedia4Foam</i>	7
2.1	Softwares and versions	7
2.2	Installing <i>porousMedia4Foam</i>	7
2.3	Getting started with <i>porousMedia4Foam</i>	9
3	Setting up and running cases in <i>porousMedia4Foam</i>	11
3.1	Requisites to run a case	11
3.2	Example cases	13
3.2.1	Calcite dissolution - dolomite precipitation/ dissolution . . .	13
3.2.2	Celestite dissolution - Barite precipitation in a 2D flow cell .	25
4	Multiphase benchmark using <i>porousMedia4Foam</i>	33
4.1	Case setup	33
4.2	Running the case	34
4.3	Post-processing the results	34
5	Mathematical models implemented in <i>porousMedia4Foam</i>	37
5.1	Porous media models	37
5.1.1	absolutePermeabilityModels	38
5.1.2	dispersionModels:	38
5.1.3	surfaceAreaModels:	38
5.1.4	reducedSaturationModels:	39
5.1.5	capillarityModels:	39
5.1.6	relativePermeabilityModels:	40
5.2	Geochemical models	40
5.2.1	phreeqcRM	41
5.2.2	simpleFirstOrderKineticMole	42
5.2.3	transportOnly	42
5.2.4	flowOnly	42
5.2.5	uphreeqcRM	43
5.2.6	unsaturatedComponentTransport	43
5.2.7	unsaturatedFlowOnly	43
5.3	Solvers	43
5.3.1	dbfFoam: Multi-scale micro-continuum flow model	44

5.3.2	darcyFoam: Darcy's law	45
5.3.3	constantVelocityFoam: Constant flow rate	46
5.3.4	impesFoamMMFRT: two-phase flow solver	46
6	Creating a new library, solver in <i>porousMedia4Foam</i>	47
6.1	Creating a new permeability model	47
6.1.1	Coding the mathematical expression:	48
6.1.2	Compiling the implemented model:	49
6.1.3	Using the implemented model:	49
6.2	Creating a new solver	49
7	References	51

Brief Introduction to *porousMedia4Foam*

porousMedia4Foam is an open-source hydro-geochemical package that can decipher fluid flow and chemical reactive processes occurring within multi-{scale, dimensional, phase, compositional} environments. Topics of research interest include but are not only limited to — investigating the use of sub-surface settings for the storage of carbon-dioxide, hydrogen gas, nuclear wastes, contaminant transport within vadose zones, recovery of geothermal fluids. *porousMedia4Foam* provides a versatile platform highlighting the following:

- robust solvers — allow users to model single-phase hydro-geochemistry at different scales i.e. at the pore-, continuum-, and, hybrid- scales. Additionally, two phase compositional flows can also be modelled at the continuum scale,
- generic code architecture — allow users to extend/ manipulate existing libraries and solvers to create new models with relative ease,
- unstructured grids — allow users to run hydro-geochemical analysis on any complicated porous geometries,
- plethora of choices available for users — switch (on/ off) following criteria as per the specifics of the case being modelled. For example, user may consider flow with/ without solving for the chemistry, the geochemical interactions can be taken into account considering thermodynamic local equilibrium/ kinetic reactions/ combination of local thermodynamic equilibrium and kinetics, activate/ deactivate evolution of transport properties such as porosity, permeability, reactive surface area, dispersivity.

1.1 Reactive Transport Modelling

The above mentioned sub-surface applications involve complex interactions between Thermo-Hydro-Mechanical-Chemical (THMC) parameters that include temperature, pressure, contact angle, surface tension, reaction rate and reactive surface area of minerals, diffusion coefficients of chemical species etc [3, 54, 2, 43, 42, 49, 32]. Minor changes in THMC parameters at the pore-scale can potentially trigger significant changes in the fluid displacement patterns [26, 29, 27]. The link between these THMC parameters are non-linear and usually difficult to comprehend [22, 36, 39].

A typical sub-surface setup spatially expands for several kilometers along all directions i.e. along the depth, length and width, and the time horizon of interest to analyse reactive flow physics range from few years to centuries of years. At present, Reactive Transport Modelling (RTM) has created a niche to comment on the efficient design, operational strategies and analyse the safety of complex sub-surface applications [21, 46, 11, 9]. Unlike in hydro-geochemical experiments that are limited to core plugs (cylindrical plugs usually have a diameter approximately around 0.01 m and height approximately around 0.02 m [28]) and analysed for the ongoing hydro-geochemical interactions for a period of few days, RTM offers flexibility to analyse cases with spatio-temporal scales on par with real-life sub-surface applications. The main advantage of RTM is that it integrates both the essential ingredients required to assess the ongoing dynamic processes in sub-surface settings which include 1. fluid flow, and, 2. geochemistry. There already exist several reactive transport packages such as TOUGHREACT [53], CrunchFlow [47], MIN3P [20], PFLOTRAN [18] amongst others. The review by [44] provided an extended description of various reactive transport packages and compared the packages on different grounds such as code availability (i.e. open-source or not), possibility to run simulations in parallel, and numerical schemes, along with implemented geochemistry and physical modelling features.

The development of reactive transport packages is a continuous and dynamic task primarily benefiting from, and triggering new research directions on the coupling of basic processes responsible for complex non-linear THMC behaviors [45].

1.2 Description about *porousMedia4Foam*

In this section we provide a brief introduction to *porousMedia4Foam*. A more elaborate discussion about the package, models available and numerical benchmarks performed using *porousMedia4Foam* can be found in [40, 35].

porousMedia4Foam is built on the skeletal framework of OpenFOAM[®]. OpenFOAM is a well established open-source finite-volume based Computational Fluid Dynamics (CFD) package with capability to run simulations in parallel [13]. OpenFOAM has extensively been used to model and analyse diverse set of applications ranging from single [25] to multiphase phase flows [34], laminar [10] and turbulence flows [19], capillary

driven flows [33], phase-change [23] etc. OpenFOAM uses C++ as the primary coding language. The rich set of existing libraries and the use of Object Oriented Programming (OOP) in OpenFOAM assisted us in developing new classes and inheriting existing classes as per requirements.

porousMedia4Foam was initially developed to model fluid flows (single and multi-phase) in porous media [14]. In this current release (version #1), we extended the fluid flow modelling capabilities of *porousMedia4Foam* to additionally account for geochemical interactions. Main highlights of the current release include – 1. a unique multi-scale framework is now available that allows the user to run hybrid-scale hydro-geochemical cases for single phase systems, and, 2. couple two well-established open-source packages (OpenFOAM and PHREEQC) to model hydro-geochemical systems. While developing *porousMedia4Foam*, care has been taken by the authors to follow a generic coding structure so that a novice user can adapt the code as per requirements with relative ease. *porousMedia4Foam* already comes loaded with a number of commonly used models to describe the evolution of porous media and transport properties. These source files will serve as a reference if the user wants to modify the empirical relations to describe the evolution of various properties.

Figure 1 provides an overview of the main elements of the *porousMedia4Foam* package. These include: 1. porous media models, 2. geochemical models, and, 3. solvers. Within the coding architecture followed to develop *porousMedia4Foam*, suitable **class** are created for the porous media models and geochemical models. These classes specifically detail the essential traits of the porous medium and geochemistry respectively. An **object** is created for the geochemical models class in the solvers. Using this created object in the solvers, suitable functions defined in the classes are called during run time to update the porous media and transport properties based on chemical equilibrium of the system.

1.3 Structure of the user manual

In Chapter 2 we give details regarding the installation of *porousMedia4Foam* and running the first simulation. In Chapter 3 we discuss a single phase hydro-geochemical test case mentioning the files that must be checked/ edited before running any simulation using *porousMedia4Foam*. We follow the discussion considering a simple multi-phase benchmark in Chapter 4. In Chapter 5 we elaborate on the mathematical models implemented in *porousMedia4Foam* and redirect the users to the corresponding source codes for reference.

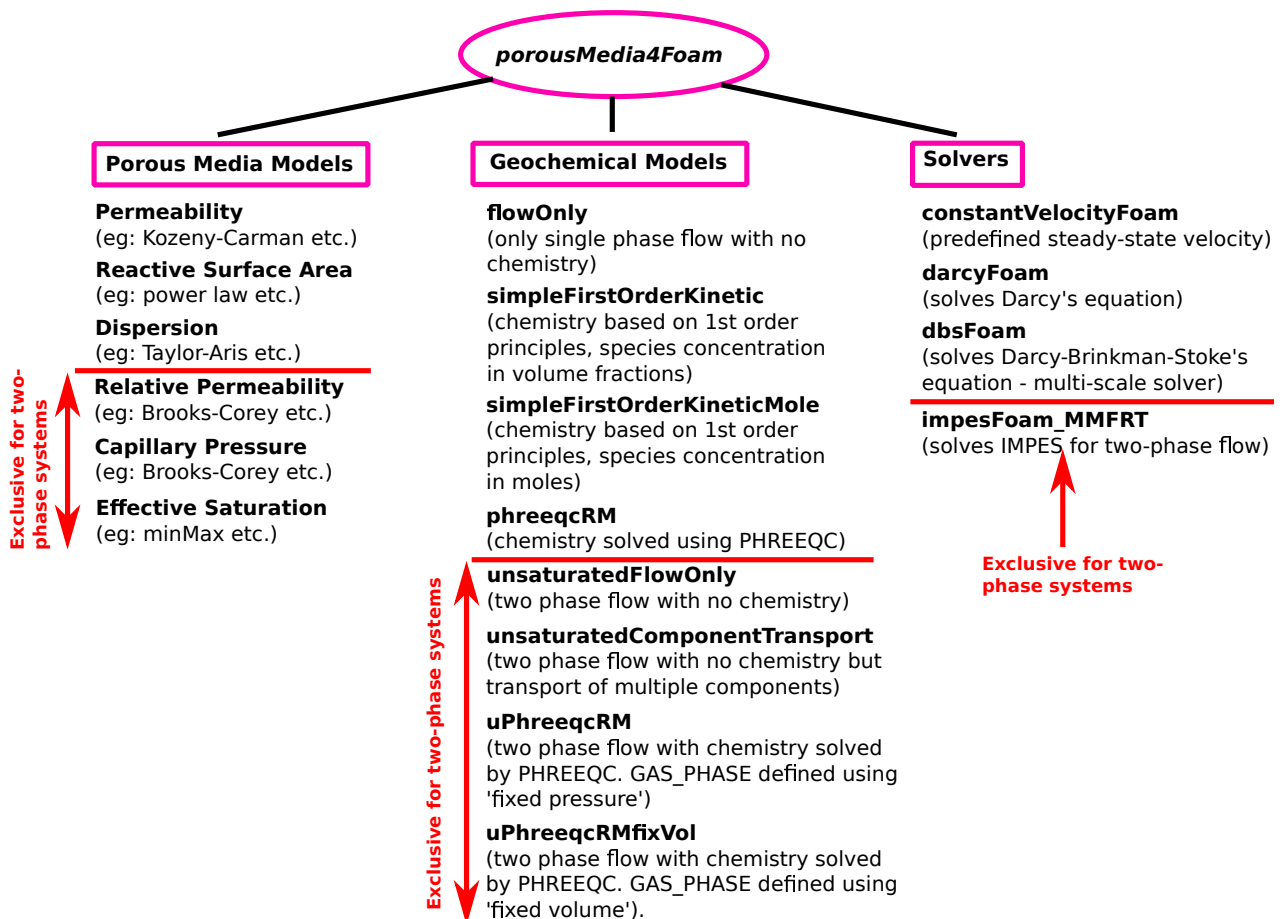


Figure 1: Main elements of the *porousMedia4Foam* package. The **Porous Media Models** comprise of commonly used models that describe the evolution of porous media properties (example: permeability, reactive surface area) and transport properties (relative permeability, dispersivity). Users are provided with a plethora of choices to run — for example, only fluid flows, fluid flows with transport of species or solve for hydro-geochemistry. Chemistry can be solved using — simple first order kinetics or by using an external geochemical module in the form of PHREEQC [30]. Details regarding the list of choices available are mentioned under **Geochemical Models** in the figure. List of **Solvers** available in *porousMedia4Foam* are also mentioned in the figure.

Installing and getting started with *porousMedia4Foam*

In this section we describe the installation procedure for *porousMedia4Foam*. We also run a sample test case for getting started with the hydro-geochemical package.

2.1 Softwares and versions

The current version of *porousMedia4Foam* is compatible with OpenFOAM version 7 (<https://github.com/OpenFOAM/OpenFOAM-7>) and PhreeqCRM version 3.6.2 (<https://water.usgs.gov/water-resources/software/PHREEQC/documentation/phreeqcrm/index.html>). *porousMedia4Foam* can be run on Ubuntu as-well as on Windows (using Windows Subsystem for Linux, WSL) operating systems.

Firstly, the user is advised to install OpenFOAM on their machine. For Ubuntu operating systems: detailed installation procedure is available at - <https://openfoam.org/download/7-ubuntu/>. For Windows operating systems: detailed installation procedure is available at - <https://openfoam.org/download/windows/>.

2.2 Installing *porousMedia4Foam*

- **Step 1: Enter repository where the package is to be installed.**

The user is advised to enter the repository where the package is to be installed. For example, following the conventional installation procedure of OpenFOAM, one may have the following file structure `$HOME/OpenFOAM/spavuluri-7`. To install the package in this repository one must run the following command in

the terminal to enter the necessary working directory (NOTE - spavuluri7 in the command below must be edited by the user as per their respective repository name),

```
cd $HOME/OpenFOAM/spavuluri-7
```

- **Step 2: Cloning the source code from Github.**

Users can download/ clone *porousMedia4Foam* from Github by copying the following command in terminal (both the following lines as one command),

```
git clone -b porousMedia4Foam -single-branch  
https://github.com/csoulain/porousMedia4Foam
```

After running the above command a repository named *porousMedia4Foam* will be available in the present working directory that contains the source code and tutorials. Within this *porousMedia4Foam* repository, 5 sub-folders (libraries, phreeqrm, solvers, tutorials and utilities) and 2 shell script files (Allwclean, Allwmake) are available.

- **Step 3: Compiling libraries and creating solvers from source code.**

While making a first installation of *porousMedia4Foam*, the *Allwmake* file is to be run by copying the following command in the terminal,

```
./Allwmake
```

The above command will take a few minute to process where the source code is compiled. Upon finishing the compilation, essential libraries for the solvers are initially created. Thereafter the solver executables are created. The created libraries (shared object libraries with *.so* extension) and solvers will be available in the following repository -

```
$WM_PROJECT_USER_DIR/platforms/linux64GccDPInt32Opt
```

- **Step 4: Checking the installation.**

A simple check to know if the above three steps were successful, the user may run any solver executable in the present working directory. For example, the command *dbSFoam* is used to begin a hydrib-scale simulation. At this moment there exist no test case for the solver to be used. Hence, the following error message is expected which complains about a missing *controlDict* file as shown below,

```
1 /*  
-----*\  
2 ===== |  
3  \\      /  F ield      | OpenFOAM: The Open Source CFD Toolbox  
4  \\      /  O peration   | Website: https://openfoam.org  
5   \\    /   A nd         | Version: 7  
6   \\  /    M anipulation |  
7  \*-----*\  
   */  
8 Build   : 7-3bcbaf946ae9  
9 Exec    : constantVelocityFoam  
10 Date    : Jun 10 2021
```

```

11 Time      : 14:37:08
12 Host      : "hgxxn93"
13 PID       : 420696
14 I/O       : uncollated
15 Case      : /home/local/ISTO/spavuluri/OpenFOAM/spavuluri-7
16 nProcs    : 1
17 sigFpe    : Enabling floating point exception trapping (FOAM_SIGFPE).
18 fileModificationChecking : Monitoring run-time modified files using
    timeStampMaster (fileModificationSkew 10)
19 allowSystemOperations : Allowing user-supplied system call
    operations
20
21 // * * * * *
    * * * * * //
22 Create time
23
24
25
26 --> FOAM FATAL ERROR:
27 cannot find file "/home/local/ISTO/spavuluri/OpenFOAM/spavuluri-7/
    system/controlDict"
28
29 From function virtual Foam::autoPtr<Foam::ISstream> Foam::
    fileOperations::uncollatedFileOperation::readStream(Foam::
    regIOobject&, const Foam::fileName&, const Foam::word&, bool)
    const
30 in file global/fileOperations/uncollatedFileOperation/
    uncollatedFileOperation.C at line 538.
31
32 FOAM exiting

```

2.3 Getting started with *porousMedia4Foam*

In this section, we run our first hydro-geochemical case. This case models calcite dissolution - dolomite precipitation/ dissolution in a 1D column of length 0.5 m. An acidic solution (pH = 2) is continuously injection into the system. Details regarding the setup for the same case is described later in Chapter 3. Within the package, this case is available in the following repository -

`porousMedia4Foam/tutorials/noPoroFeedback/calcDissDolPrecip` and the user must enter this repository to run the test case. The `calcDissDolPrecip` repository comprises of 3 mandatory sub-folders (0, constant, system) and a shell script file (`Allrun`). Description regarding the files present within the mandatory sub-folders are elaborated in Chapter 3. To simulate the test case, the user must run `Allrun` file as follows,

```
./Allrun
```

The `Allrun` file executes three commands sequentially

1. *blockMesh* - to generate the mesh,

2. *constantVelocityFoam* - solver being used to run the hydro-geochemical case, and
3. *paraFoam* - to view the simulation results in Paraview.

3

Setting up and running cases in *porousMedia4Foam*

In this chapter we elucidate how to setup cases for *porousMedia4Foam* simulations. The three important stages to be followed in order are 1. pre-processing, 2. running simulation, and, 3. post-processing the results. To explain these three stages in detail, we refer to a few test cases in this chapter.

The examples considered in this chapter are monophasic. In addition to the cases discussed in this chapter, there exist few more examples in the `porousMedia4Foam/tutorials` repository. Cases available in the tutorials repository are numerical benchmarks discussed in [52, 37, 22]. Users can refer to [40, 35] for an elaborate scientific discussion of all the benchmark studies performed using *porousMedia4Foam*.

3.1 Requisites to run a case

The setup for a case in *porousMedia4Foam* follows the same skeletal structure as that of OpenFOAM. The following folders (**0**, **constant**, **system**) and files (blue refer to optional files - ex: *pH*, *setFieldsDict*, ..., rest being mandatory files) are required to run any case in *porousMedia4Foam*.

- **0**
 - inertMineral
 - mineral volume fraction, *Ys.mineralName* — eg: *Ys.Calcite*, *Ys.Barite*
 - permeability, *K*
 - pressure, *p*

- pH
- species concentration, $Y.specieName$ — eg: $Y.Na$, $Y.Cl$
- velocity, \mathbf{U}
- **constant**
 - gravity, \mathbf{g}
 - phreeqcInputFile
 - phreeqcDataBase
 - transportProperties
 - turbulenceProperties
- **system**
 - blockMeshDict
 - controlDict
 - fvSchemes
 - fvSolution
 - setFieldsDict
 - singleGraph

- **The 0 repository:**

0 repository specifies the initial and boundary conditions for required variables. The required variables include primary variables (for example: pressure, velocity etc.) and other additional variables (for example: permeability). While using the geochemical packages *phreeqcRM* and *uphreeqcRM*, the species concentration $Y.specieName$ and pH files are not required to be provided by the user. These files are generated upon starting the run. In-case the user provides initial and boundary data for the species concentration and pH , the provided data will be overwritten when the simulation starts. To set the concentration of species and pH as per requirement, the user must edit the `constant/phreeqcInput` file.

- **The constant repository:**

constant repository is used to specify all the 'static information' essential for the simulation. This information include details regarding the activation of gravity (file name: **g**), transport properties such as fluid density, viscosity, models to be used for describing the evolution of porous media and transport properties (file name: **transportProperties**), turbulence models (file name: **turbulenceProperties**). As flow in the sub-surface is usually slow, turbulence is often neglected and only laminar flows are accounted. While using the *phreeqcRM* and *uphreeqcRM*

geochemical modules, users have to provide two additional files within the constant repository: 1. `phreeqcInputFile` and, 2. `phreeqcDatabase`. The `phreeqcInputFile` describes the initial chemical composition of species and solution being injected into the system whereas, `phreeqcDatabase` describes the chemical database which include specifics of all species, possible chemical reactions, kinetic rate laws etc.

- **The system repository:**

system repository is used to specify the simulation run parameters. These include information regarding the schemes to be used for solving equations (file name: **fvSchemes**), the tolerances to be achieved while solving linear matrices (file name: **fvSolutions**). Information regarding the time step size, data output controls are specified in **controlDict**. **blockMeshDict** is a mandatory file that describes the geometry of the system, identify the boundaries, name the boundaries, and indicate the mesh size. Few optional files such as **setFieldsDict**, **singleGraph** are also available in this repository and their use will be explained while discussing the example case below.

3.2 Example cases

In this section, we discuss an example in detail to give a comprehensive overview for the users about the necessary files that must be edited/ checked before running any case with *porousMedia4Foam*.

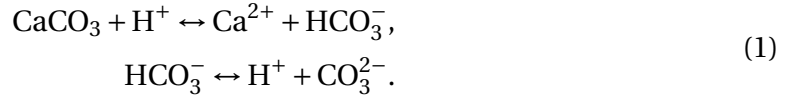
3.2.1 Calcite dissolution - dolomite precipitation/ dissolution

A 1D column of length $\ell = 0.5$ m is initially, uniformly occupied with calcite having volume fraction 0.03 and inert minerals occupying volume fraction 0.57. The inert minerals do not participate in any chemical reaction. The porosity of the column is $\phi = 0.4$. For this case, we do not account for porosity feedback dependent on the ongoing geochemical reactions. Hence the porosity stays constant throughout the simulation. An acidic solution (pH = 2) is continuously injected along with chloride (Cl^-) and magnesium (Mg^{2+}) ions with a velocity $\mathbf{V}_f = 0.01$ m/s. Initial conditions and description of the primary components continuously injected into the system are provided in Table 1.

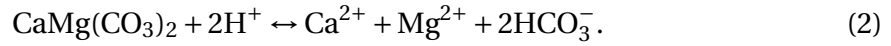
Primary components	Units	Initial conditions	Injected solution
pH	-	8.2	2
Ca^{2+}	mol/L	5.113×10^{-4}	0
Mg^{2+}	mol/L	0	9.967×10^{-3}
CO_3^{2-}	mol/L	1.018×10^{-3}	0
Cl^-	mol/L	9.97×10^{-16}	31.51×10^{-3}

Table 1: Example 1 – Initial conditions and description of the primary components injected for the case of calcite dissolution - dolomite precipitation/ dissolution.

The spatial domain is discretized using 50 cells (each cell $dx = 10\text{mm}$ long). We followed the evolution of calcite, dolomite volume fractions and primary ion concentrations for one hour of reaction time. We account for the dissolution of calcite based on kinetic equilibrium whereas the precipitation and dissolution of dolomite is taken into account based on thermodynamic equilibrium. Following equations describe the dissolution of calcite,



Ca^{2+} and HCO_3^- released during dissolution of calcite react with Mg^{2+} that is continuously injected into the system to precipitate dolomite $\text{CaMg}(\text{CO}_3)_2$. Due to the continuous injection of an acidic solution, the precipitated dolomite dissolves according to the following equation,



We now explain the three essential steps to be followed 1. pre-processing, 2. simulation, and 3. post-processing to run and analyze the above described hydro-geochemical system.

Pre-processing: in this step, all essential information such as - the geometry, initial and boundary conditions, spatio-temporal discretisation, numerical schemes, choice of linear solvers and models to be used are specified by the user. This information will then be used by the numerical solver for running the case.

Pre-processing - step 1 (PP1): Building the geometry and meshing the system

For the geometry illustrated in Fig. 2 we commence by constructing the geometry, labeling the boundaries and gridding the system. This is done by editing the following

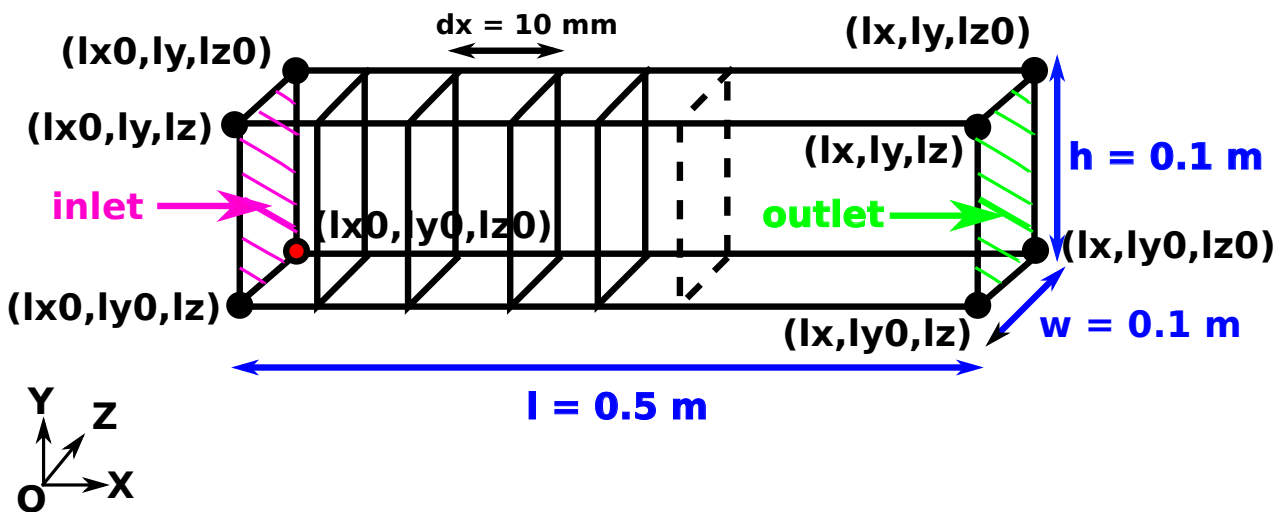


Figure 2: Case setup for investigating calcite dissolution - dolomite precipitation/ dissolution in a 1D column.

file: `system/blockMeshDict`. The `blockMeshDict` file for the case is shown below

for reference with comments (text in green). It is important to note that OpenFOAM is a 3D CFD package. As the case to be modelled is a 1D case, the height h and width w are set to $h = w = 0.1$ m (it can be set to any other scalar value other than 0.1 m also). Along Y and Z axis, there exist only one cell and it is tagged with an 'empty' boundary condition.

```

1 /*-----*-- C++
   *-----*\
2 | ===== |
   |         |
3 | \\      / F i e l d       | OpenFOAM: The Open Source CFD Toolbox
   |         |
4 | \\      / O p e r a t i o n   | Version:  2.3.0
   |         |
5 |  \\    /  A n d             | Web:      www.OpenFOAM.org
   |         |
6 |   \\  /    M a n i p u l a t i o n  |
   |         |
7 /*-----*-----
   */
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     object       blockMeshDict;
14 }
15 // * * * * *
   * * //
16
17 convertToMeters 1; //unit in meters;
18
19 lx0 0;
20 ly0 0;
21 lz0 0;
22
23 lx 0.5;
24 ly 0.1;
25 lz 0.1;
26
27 vertices //coordinates of the system
28 (
29     ($lx0 $ly0 $lz0)
30     ($lx  $ly0 $lz0)
31     ($lx  $ly  $lz0)
32     ($lx0 $ly  $lz0)
33     ($lx0 $ly0 $lz)
34     ($lx  $ly0 $lz)
35     ($lx  $ly  $lz)
36     ($lx0 $ly  $lz)
37 );
38
39 blocks

```

```

40 (
41     hex (0 1 2 3 4 5 6 7) (50 1 1) simpleGrading (1 1 1) //uniform
    gridding - 50 cells
42 );
43
44 edges
45 (
46 );
47
48 boundary
49 (
50
51     outlet //outlet boundary face vertices
52     {
53         type patch;
54         faces
55         (
56             (2 6 5 1)
57         );
58     }
59
60     inlet //inlet boundary face vertices
61     {
62         type patch;
63         faces
64         (
65             (0 4 7 3)
66         );
67     }
68
69     frontAndBack //frontAndBack boundary face vertices
70     {
71         type empty;
72         faces
73         (
74             (3 7 6 2)
75             (1 5 4 0)
76             (0 3 2 1)
77             (4 5 6 7)
78         );
79     }
80 );
81
82 mergePatchPairs
83 (
84 );
85
86
87 //
    *****
    //

```

We briefly explain the important sections of the shown `blockMeshDict` file. For an elaborate discussion on how to write `blockMeshDict` and to know additional capabilities of `blockMesh` the users can refer to [13].

Lines 1 - 16: a common format used by OpenFOAM to read input and write output files. In general, this content stays same for all input and output files. We further neglect these set of lines at the start of files that are shown later.

Line 17: the default unit for spatial dimensions are 'meters (m)'. `convertToMeters` keyword is used to scale the vertices to attain suitable geometrical dimensions.

Lines 19 - 25: scalar variables (`lx0`, `ly0`, `lz0`, `lx`, `ly`, `lz`) are defined so that these variables can be used to define the co-ordinates of the geometry.

Lines 27 - 37: define 8 vertices of the column geometry using the 6 variables (`lx0`, `ly0`, `lz0`, ...) defined earlier. For example, the vertex #0 is defined by (`lx0`, `ly0`, `lz0`), and the vertex #1 is identified by (`lx`, `ly0`, `lz0`) and so on. Refer to Fig. 2.

Lines 39 - 42: define a hexagonal block using the vertex numbers `hex (0 1 ... 7)`. We spatially discretize the geometry with 50 cells along X-axis, 1 cell along Y-axis and Z-axis. The spacing is uniform `simpleGrading (1 1 1)` and the mesh is Cartesian.

Lines 48 - 80: name all boundaries of the geometry. For example: the boundary face formed by vertices 2, 6, 5 and 1 is labelled as an `outlet`. The face formed by vertices 0, 4, 7 and 3 form the `inlet` boundary. The remaining 4 faces belong to the `frontAndBack` boundary.

PP2: Setting the initial and boundary conditions

In this step, we specify the initial and boundary conditions for the following variables: 1. velocity U , 2. inert mineral volume fraction `inertMineral`, and, 3. mineral volume fractions `Ys.Calcite` and `Ys.Dolomite`.

As mentioned earlier in the case setup, an acidic solution along with chloride Cl^- and Mg^{2+} ions are injected at a constant velocity $V_f = 0.01$ m/s. The corresponding file for velocity is shown below for reference and can be found in `0/U`.

```

1 dimensions      [0 1 -1 0 0 0 0];    //units m/s
2
3 internalField   uniform (0.01 0 0);    //50 cells having Ux=0.01m/s
4
5 boundaryField
6 {
7     outlet
8     {
9         type          zeroGradient;

```

```

10     }
11     inlet
12     {
13         type            fixedValue;
14         value            uniform (0.01 0 0); //constant injection
15     }
16     frontAndBack
17     {
18         type            empty;
19     }
20 }

```

Line 1: specifies the units of velocity (m/s). OpenFOAM strictly performs dimensional checks in order to avoid any apparent errors during the mathematical calculations. The dimension set used in OpenFOAM follows the following order: [mass *kg* — length *m* — time *s* — Temperature *K* — Quantity *mol* — Current *A* — Luminous intensity *cd*].

Line 3: specifies the internal field for velocity. This line indicates that the velocity is $V_{f,x} = 0.01$ m/s in all 50 cells of the column.

Line 5 - 20: specifies the boundary conditions for the case. The inlet has a constant injection $V_{f,x} = 0.01$ m/s.

After setting the velocity file, we now set the initial and boundary conditions for the solid minerals. Starting with `0/inertMinerals` which occupy 57% volume fraction in a cell. The file looks as shown below,

```

1
2 dimensions            [0 0 0 0 0 0 0];    //dimensionless
3
4 internalField          uniform 0.57;      //inert mineral occupies 0.57 volume
5     fraction
6 boundaryField
7 {
8     outlet
9     {
10        type            zeroGradient;
11    }
12    inlet
13    {
14        type            zeroGradient;
15    }
16    frontAndBack
17    {
18        type            empty;
19    }
20 }
21

```

```

22
23 //
    *****
    //

```

Similarly, corresponding files for calcite having an initial volume fraction of 3% (file: 0/Ys.Calcite)

```

1 dimensions      [0 0 0 0 0 0 0];    //dimensionless
2
3 internalField   uniform 0.03;      //initial volume fraction in 50 cells
4
5 boundaryField
6 {
7     outlet
8     {
9         type          zeroGradient;    //Neumann b.c
10    }
11    inlet
12    {
13        type          zeroGradient;    //Neumann b.C
14    }
15    frontAndBack
16    {
17        type          empty;
18    }
19 }

```

and dolomite having no presence initially within the column are created (file: 0/Ys.Dolomite),

```

1 dimensions      [0 0 0 0 0 0 0];    //dimensionless
2
3 internalField   uniform 0;          //dolomite not present initially
4
5 boundaryField
6 {
7     outlet
8     {
9         type          zeroGradient;    //Neumann b.c
10    }
11    inlet
12    {
13        type          zeroGradient;    //Neumann b.c
14    }
15    frontAndBack
16    {
17        type          empty;
18    }
19 }

```

PP3: Setting transportProperties

In this step, we specify the transport properties that have to be used for the case. These include description about the fluid density, fluid viscosity, models that have to be used

to describe the evolution of porous media properties, geochemical models that are to be used etc. The constant/transportProperties file is shown below for reference,

```

1 transportModel Newtonian;
2 mu mu [1 -1 -1 0 0 0 0] 1e-3; //fluid viscosity
3 rho rho [1 -3 0 0 0 0 0] 1000; //fluid density
4
5 //
6 *****
7 //
8
9 geochemicalProperties
10 {
11     rhol rhol [1 -3 0 0 0 0 0] 1000; //fluid density
12     geochemicalModel phreeqcRM; //use phreeqcRM geochemical module
13
14     mineral (Calcite Dolomite); //we have calcite, dolomite in system
15
16     absolutePermeabilityModel heterogeneousScalarConstant; //permeability
17     is constant
18     heterogeneousScalarConstantCoeffs
19     {
20         K0 K0 [0 2 0 0 0 0 0] 1.e-11; //initial permeability value
21     }
22
23     dispersionModel linearDispersion; //dispersion model
24     linearDispersionCoeffs
25     {
26         Di Di [0 2 -1 0 0 0 0] 1e-9; //molecular diff.
27         alphaL alphaL [0 1 0 0 0 0 0] 1e-4; //hydrodyn. disp.
28     }
29
30     CalciteProperties //Properties of calcite min.
31     {
32         activatePhaseEquilibrium false; //calcite treated by kinetics
33         rhos rhos [1 -3 0 0 0 0 0] 2707; //density
34         Vm Vm [0 3 0 0 -1 0 0] 36.93336e-6; //molar volume
35         surfaceAreaModel constantSurfaceArea; //model to describe
36         reactive surf. ar.
37         constantSurfaceAreaCoeffs
38         {
39             A0 A0 [0 -1 0 0 0 0 0] 1; //initial reactive surf. ar.
40         }
41     }
42
43     DolomiteProperties //Properties of dolomite
44     {
45         activatePhaseEquilibrium true; //consider thermodynamic eq.
46         rhos rhos [1 -3 0 0 0 0 0] 2707;
47         Vm Vm [0 3 0 0 -1 0 0] 64.5e-6;
48         surfaceAreaModel none;

```

```

46 }
47
48 phreeqcRM          //essential info for phreeqc run
49 {
50     PhreeqcInputFile "phreeqcInput"; //input initialization file
51     PhreeqcDataBase  "phreeqc.dat"; //database file
52     use_SNIA false; //use Strang alg.
53     activateUpdatePorosity false; //porosity is constant
54     StrangSteps 6; //Strang alg. coeff.
55 }
56 }

```

Line 1 - 3: indicate that the solution is a Newtonian fluid having density $\rho = 1000$ kg/m³ and viscosity $\mu = 0.001$ kg/ms.

Line 7 - 56: we create a mandatory dictionary labelled `geochemicalProperties`. Within this dictionary, the models to be used for the simulation are specified.

Line 10: we indicate the solver that we would like to use PHREEQCs reaction module `phreeqcRM` to solve the geochemistry. Other possible options are mentioned in Table 8.

Line 12: indicate the minerals present within the system. For this case, there exist **Calcite, Dolomite**. If there exist no minerals in the system, the arguments are left empty like `mineral ()`;

Note that, the name of the minerals mentioned here must match the name of minerals specified in the 0 repository as `Ys.Calcite`, `Ys.Dolomite`.

Line 14 - 18: indicate the permeability model (`heterogeneousScalarConstant`) and initial permeability value `K0`. Other alternative permeability models available in *porousMedia4Foam* can be found in Table 2.

Line 20 - 25: indicate the dispersion model (`linearDispersion`) along with the molecular diffusion `Di` and hydrodynamic dispersion `alphaL` coefficients. Other alternative dispersion models available in *porousMedia4Foam* can be found in Table 3.

Line 28 - 38: indicate the properties of calcite mineral such as density `rhos`, molar volume `Vm`, model to be used for describing the evolution of reactive surface area. In line 30, the `activatePhaseEquilibrium` is set to false as the dissolution of calcite is taken into account using kinetics.

Line 40 - 46: indicate the properties of dolomite mineral. In line 42, the `activatePhaseEquilibrium` is set to true as dolomite precipitation and dissolution are taken into account based on thermodynamic equilibrium.

Line 48 - 56: indicate the parameters for `phreeqcRM` module. The module name here must correspond to the `geochemicalModel` mentioned on line #10 of this file.

Line 50: the name of the PHREEQC input file is specified here. This file is present in `constant/phreeqcInput`. This file is described below separately.

Line 51: the PHREEQC database file present in `constant/phreeqc.dat`. This file consists of the chemical database which include all essential species, reaction between species, kinetic rate laws. For this case, the database file is a copy of standard PHREEQCs database file.

Line 52: users choice to run the simulation using Sequential Non Iterative Algorithm (SNIA) or Strang algorithm. If `use_SNIA` is set to false, Strang algorithm is activated. We recommend users to start the run using Strang algorithm.

Line 53: an option available for the user to account for porosity variation caused due to the ongoing chemical reactions. If `activateUpdatePorosity` is set to false, porosity is considered to be a constant all through the run.

Line 54: while solving a time step using Strang algorithm, we split the time step into sub-time steps to reduce apparent numerical dispersion errors. This number indicates that the time step is split into 6 sub-time steps and the chemistry is run using PHREEQC at the third sub-time step.

We will now look into a mandatory file required to run the test cases: `phreeqcInput`. This file is available in the `constant` repository.

```

1 SOLUTION 0
2   pH 2
3   Cl 30 charge #mmol/L
4   Mg 10 #mmol/L
5 END
6 # CELLS WITH CALCITE
7 SOLUTION 1-50
8   pH 8.2 charge
9   Ca 1 Calcite 0
10  C(4) 1 CO2(g) -3.45
11  Cl 1e-12
12 END

```

Line 1 - 5: SOLUTION 0 refers to the solution that is being injected into the system. Based on the data provided in Table 1, the solution being injected into the system is described in this section. The concentration of Cl^- and Mg^{2+} are specified in milli moles per litre of water (mmol/L - default units used by PHREEQC).

Line 7 - 12: SOLUTION 1 - 50 describe the initial chemical composition of the system. This data includes the pH of the internal control volumes and the equilibrium concentrations of Ca, C ions.

The other mandatory file required to run cases in *porousMedia4Foam* is the chemical database file. This database file consist of the chemical description of various species, possible interaction with other species and description of kinetic reaction rate laws. `phreeqc.dat` is PHREEQCs database that is used for this case study. The corresponding file can be found in the `constant` repository. We show a snippet of the BASIC interpreter used to compute the kinetic reaction rate of calcite for this case. The kinetic rate of dissolution of calcite is given by,

$$r_{calcite} = k_{calcite} A_{calcite} (1 - \Omega), \quad (3)$$

where $r_{calcite}$ is the reaction rate of calcite in $\text{mol}/\text{m}^3\text{s}$, $k_{calcite} = 1 \text{ mol}/\text{m}^2\text{s}$ is the reaction rate constant of calcite, $A_{calcite} = 1 \text{ m}^2/\text{m}^3$ is the reaction surface area of calcite, Ω is the saturation index of calcite.

```

1 Calcite
2   -start
3 1  REM   PARM(1) = specific surface area of Calcite (large grains), [m
   ^2/m^3]
4 10  si_cc = SR("Calcite")
5 20  rate = 1*PARM(1)*(1 - si_cc) #mol/m3/s, 1 is the effective rate
   constant mol/m2/s
6 30  mole = rate*time #mol/m3
7 35  mole = mole*0.001 #mol/l
8 40  save mole
9   -end

```

PARM(1) in line #20 of the BASIC interpreter refers to the reactive surface area of calcite. This information is recovered from the `porousMediaModels/specificSurfaceArea` module (Section 5.1.3).

Finally, the files in the `system` repository have to be tuned as per the simulation requirements. More details about this section can be found in the OpenFOAM user guide. The numerical schemes to be used for solving the mathematical equations are specified in `system/fvSchemes`. Care must be taken to use the same schemes in the benchmark cases else discrepancies are known to occur. The algebraic matrix solvers are defined in the `system/fvSolution` file. Necessary tolerance values are to be provided by the users. To change the output time, simulation end time and other output parameters users have to edit the `system/controlDict` file. At times, there might be a need to set specific variables at specific zones of the setup. For example, the user might want to have different permeabilities at different sections of the system. For these setups, the users may use the `setFieldsDict` file in the `system` repository. An example can be found in `porousMedia4Foam/tutorials/case8/structuredMesh/system/setFieldsDict`. In this file, the permeability, the mineral volume fractions are different at different sections of the flow cell and are initialized using the `setFieldsDict` file.

Running the simulation: in this step, we run the simulation based on the parameters set in the pre-processing step. The list of solvers available in the package are described in Section 5.3. For this case, we use the `constantVelocityFoam` solver. Once the name of the solver is entered in the terminal, the simulation starts. During the run, output related to the convergence of transport equation of species is continuously written to the screen.

Post-processing: after the simulation run finishes, several output files are written. For example in this case the output is written even 20 minutes. These files can be used to post-process to extract the essential data. For example, the following is a plot showing the mineral concentration over the length of the channel at 20 minutes, 40 minutes and 60 minutes. The results of *porousMedia4Foam* are compared with that of PHREEQCs transport module for the mineral volume fractions of both calcite and dolomite and

also the concentration of primary ions within the system.

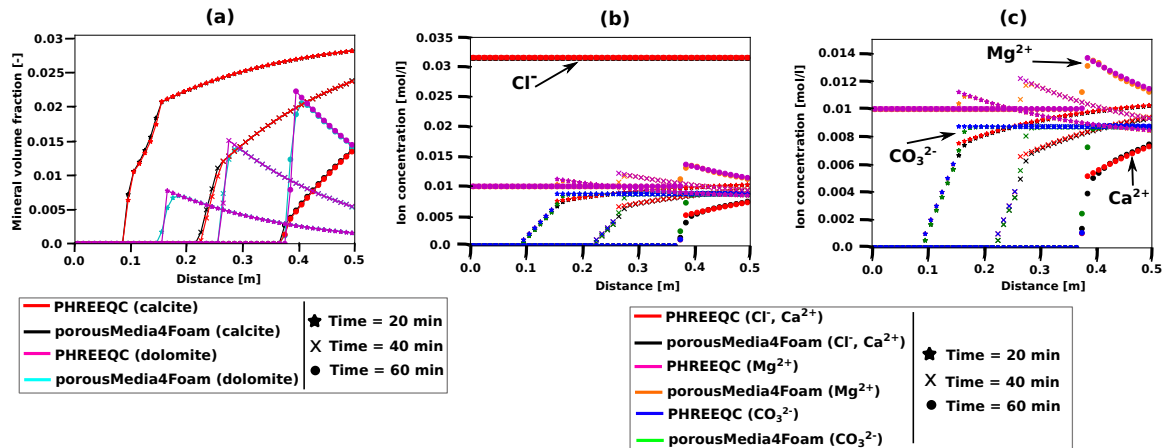


Figure 3: Calcite dissolution and dolomite precipitation/ dissolution in a 1D column. Primary ions are Cl^- , Ca^{2+} , Mg^{2+} and CO_3^{2-} . (a) Evolution of calcite and dolomite volume fractions. (b) Evolution of primary ion concentrations. (c) A zoom of (b) highlighting the evolution of primary ion concentrations.

To achieve such sort of a plot, the `system/singleGraph` dictionary is used. The code snippet is shown for reference.

```

1 start (0.005 0.05 0.05);
2 end (0.495 0.05 0.05);
3 fields (Ys.Calcite Y.Cl Y.C Y.Ca Y.Mg Ys.Dolomite);
4
5 // Sampling and I/O settings
6 #includeEtc "caseDicts/postProcessing/graphs/sampleDict.cfg"
7
8 // Override settings here, e.g.
9
10 setConfig
11 {
12     axis x; // y, z, xyz
13     nPoints 50;
14 }
15
16 // Must be last entry
17 #includeEtc "caseDicts/postProcessing/graphs/graph.cfg"

```

In the above snippet,

Line 1 - 2: indicate the lines starting and ending co-ordinates where the user would like to extract the data.

Line 3: indicates the fields that are required to be extracted from the output files written.

Lines 10 - 14: indicate the number of sampling points. Here we consider 50 to match the cells considered for this case.

Analysing results of the discussed benchmark: As an acidic solution is continuously being injected into the column, the calcite within the system dissolves according

to Eq.1. The released ions of Ca^{2+} and HCO_3^- react with the Mg^{2+} forming dolomite according to Eq. 2. As we notice in Fig.3a, at $t = 20$ min, calcite has completely been dissolved at the inlet of the column upto a length of $\ell \approx 0.09$ m based on kinetic rates of the reaction. Over the remaining portion of the column $\ell = 0.09 - 0.5$ m calcite has partially been dissolved as well as we notice a decrease in the calcite volume fraction from initial values of 0.03. Over time, the process of calcite dissolution carries on further as we notice at times $t = 40$ and 60 min in Fig.3a.

According to the thermodynamic equilibrium of the system, dolomite commences to precipitate and later dissolve at the inlet of the system. At $t = 20$ min, as seen in Fig. 3a, dolomite exist all over the column length except between $\ell \approx 0 - 0.165$ m. Over time, we notice that the dolomite continues to precipitate and dissolve as seen in Fig. 3a at times $t = 40$ and 60 min.

3.2.2 Celestite dissolution - Barite precipitation in a 2D flow cell

In this case, we investigated reactive transport occurring in a 2D flow cell containing celestite (SrSO_4) having a bimodal grain size distribution. The setup was discussed in detail in [37] and is illustrated in Fig. 4a. Users can refer to this test case in the repository `porousMedia4Foam/tutorials/celestiteBariteFlowcell`.

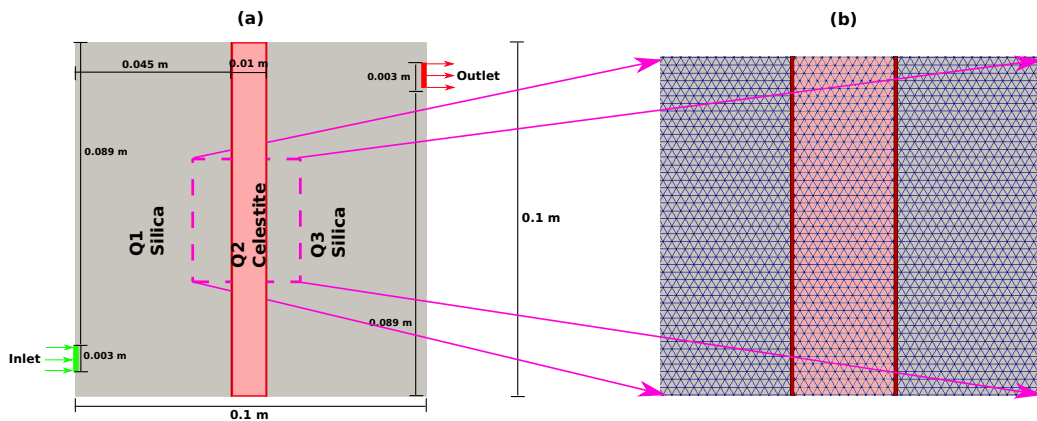
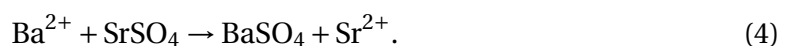


Figure 4: Description of the case setup – (a) Schematic of the flow cell. (b) A zoom of a section of the flow cell meshed with unstructured triangulated mesh.

The flow cell comprised of three compartments - Q1, Q2 and Q3. Q1 and Q3 were composed of inert mineral (quartz). Q2 comprised of a reactive mineral, celestite. An acidic solution comprising of barium (Ba^{2+}) and chloride (Cl^-), was injected continuously from the inlet at a constant rate. Once the barium ions reached the reactive zone Q2 in the flow cell, celestite dissociated into strontium (Sr^{2+}) and sulphate (SO_4^{2-}) ions. The barium ions reacted with sulphate ions resulting in the precipitation of barite (BaSO_4) according to the following reaction:



As barite has a greater molar mass ($V_{m,\text{barite}} = 52.09 \times 10^{-6} \text{ m}^3/\text{mol}$) compared to celestite ($V_{m,\text{celestite}} = 46.38 \times 10^{-6} \text{ m}^3/\text{mol}$), the geochemical reactions would result in reduction of porosity. For this case, we account for porosity feedback. Changes in mineral volume fractions due to precipitation/ dissolution will subsequently vary the porosity of the medium. The dissolution of celestite was taken into account based on kinetics while the formation of secondary minerals (barite, witherite, strontianite etc) relied on thermodynamic equilibrium. In the modelling framework for this case, celestite grains of different size were treated as two separate minerals. Initially, on the one hand, smaller celestite grains volume fraction and reactive surface area were 0.223 and $A_{\text{celestite,small},0} = 20000 \text{ m}^2/\text{m}_{\text{mineral}}^3$, respectively. On the other hand, the volume fraction of larger celestite grains was 0.447 and their reactive surface area was $A_{\text{celestite,large},0} = 100 \text{ m}^2/\text{m}_{\text{mineral}}^3$. The reactive surface area of celestite (both small and large grains) evolved as a linear function of celestite's mineral content (power-law with $n = 1$, Table 4). The permeability-porosity relationship was described using Kozeny-Carman law (Table 2). The initial permeability of Q1 and Q3 zones was $k = 1.82 \times 10^{-11} \text{ m}^2$ and for the Q2 zone it was $k = 1.8 \times 10^{-14} \text{ m}^2$. The species transport was modelled using a linear dispersion law (Table 3) with $D = 10^{-9} \text{ m}^2/\text{s}$, $\alpha_L = 10^{-5} \text{ m}$ and $n = 1$.

Pre-processing: in this step we discuss the geometry used for the case along with description of initial and boundary conditions for various data fields. The spatial domain was discretized using a Cartesian mesh with $\Delta x = \Delta y = 1 \text{ mm}$ resulting in a total of 10000 cells. To construct the Cartesian mesh, we use *blockMesh* utility. Users can refer to `system/blockMeshDict` in the tutorial repository to build the mesh and label the boundaries of the domain. Users can refer to earlier tutorial cases `blockMeshDict` description and follow the same procedure to build the mesh and label the boundaries of the flow cell.

Further to show the modelling capabilities of *porousMedia4Foam*, we also ran this benchmark study considering unstructured (triangulated) mesh. Fig. 4b shows a zoom of a section of the flow domain with the triangular meshes. For this scenario, the domain comprised of 31854 cells. We developed this triangulated mesh using a third party GUI (Graphic User Interface) software — **Salome** (<https://www.salome-platform.org/user-section/about/mesh>). The Salome platform additionally provides capabilities to label the boundaries of the domain while building the mesh. After meshing the geometry and labelling the boundaries, the using a GUI (Graphic User Interface). The file is saved in **Salome** in a *fileName.unv* format and built in OpenFOAM using the following command –

```
ideasUnvToFoam fileName.unv
```

The above command will create essential `constant/polyMesh` files.

Similar to the earlier discussed test case, now we describe the initial and boundary conditions for the case. All necessary fields that are required to have boundary conditions and initial values are defined in the 0 repository. Essential files include:

- volume fraction of inert minerals — `inertMineral`
- permeability of the medium — `K`
- pressure — `p`
- velocity — `U`
- mineral volume fractions — `Ys.mineralName`

Due to the heterogeneous distribution of permeability and mineral volume fractions we make use of the *setFields* utility to initialise the data. To run *setFields* it is essential to provide the `systems/setFieldsDict` file. For example, we show a snippet of the `setFieldsDict` code below to initialize permeability of the medium.

```

1 defaultFieldValues
2 (
3     volScalarFieldValue K 0.
4 );
5
6 regions
7 (
8     boxToCell
9     {
10        box (0. 0 -1) (0.045 0.1 1);
11        fieldValues
12        (
13            volScalarFieldValue K 1.82e-11
14        );
15    }
16
17    boxToCell
18    {
19        box (0.045 0 -1) (0.055 0.1 1);
20        fieldValues
21        (
22            volScalarFieldValue K 1.8e-14
23        );
24    }
25
26    boxToCell
27    {
28        box (0.055 0 -1) (0.1 0.1 1);
29        fieldValues
30        (
31            volScalarFieldValue K 1.82e-11
32        );
33    }
34 );

```

In the above snippet:

Lines 1 - 4: define a default value for the permeability (referred to by K in *porous-Media4Foam*). Here, we set the default value of permeability to 0 on Line #3.

Lines 8 - 15: replace the default permeability value with actual values. These specific lines set permeability of the Q1 zone.

Lines 17 - 24: initialize permeability of Q2 zone, and,

Lines 26 - 33: initialize permeability of Q3 zone.

Similar procedure is followed to initialize the *inertMineral*, *Ys.mineralName* fields.

Finally, we define the initial chemical composition of the system, solution being injected into the system and the models that are used to describe the evolution of porous media properties in the constant repository. Within `constant/transportProperties` essential details regarding the models being used to update porous media properties are defined. A snippet of the `transportProperties` file is shown below.

```
1 transportModel Newtonian;
2 mu mu [ 1 -1 -1 0 0 0 0 ] 1e-3;
3 rho rho [ 1 -3 0 0 0 0 0 ] 1000;
4
5 geochemicalProperties
6 {
7   geochemicalModel phreeqcRM;
8
9   mineral (Witherite BaCl2 BaCl22H2O BaCl2H2O Barite Strontianite SrCl2
10           SrCl22H2O SrCl26H2O Celestite_SmlGrns Celestite_LrgGrns);
11
12 absolutePermeabilityModel KozenyCarman;
13 KozenyCarmanCoeffs
14 {
15   updateFromInitialPoroPerm true;
16   K0 K0 [0 2 0 0 0 0 0] 1.;
17 }
18 dispersionModel archiesLaw;
19 archiesLawCoeffs
20 {
21   Di Di [0 2 -1 0 0 0 0] 1e-9;
22   alphaL alphaL [0 1 0 0 0 0 0] 1e-5;
23   n 0;
24   addDispersion true;
25 }
26
27 BariteProperties
28 {
29   activatePhaseEquilibrium true;
30   rhos rhos [1 -3 0 0 0 0 0] 4480;
31   //https://en.wikipedia.org/wiki/Baryte
```

```

32 Vm Vm [0 3 0 0 -1 0 0] 52.09e-6;
33 surfaceAreaModel none;
34 }
35
36 //define other mineral (mentioned on line #9) properties as well
37
38 Celestite_SmlGrnsProperties
39 {
40 activatePhaseEquilibrium false;
41 rhos rhos [1 -3 0 0 0 0 0] 1930;
42 //https://en.wikipedia.org/wiki/Strontium_sulfate
43 Vm Vm [0 3 0 0 -1 0 0] 46.38e-6;
44 surfaceAreaModel mingliangLaw;
45 mingliangLawCoeffs
46 {
47 A0 A0 [0 -1 0 0 0 0 0] 20000.;
48 n 1.;
49 initVF 0.223;
50 }
51 }
52
53 Celestite_LrgGrnsProperties
54 {
55 activatePhaseEquilibrium false;
56 rhos rhos [1 -3 0 0 0 0 0] 1930;
57 //https://en.wikipedia.org/wiki/Strontium_sulfate
58 Vm Vm [0 3 0 0 -1 0 0] 46.38e-6;
59 surfaceAreaModel mingliangLaw;
60 mingliangLawCoeffs
61 {
62 A0 A0 [0 -1 0 0 0 0 0] 100.;
63 n 1.;
64 initVF 0.447;
65 }
66 }
67
68 phreeqcRM
69 {
70 PhreeqcInputFile "test.phr"; //define chemical solution
71 PhreeqcDataBase "JenaFin.DAT"; //phreeqc database
72 activateUpdatePorosity true; //update porosity
73 numThreads 4; //use 4 threads for runs
74 use_cvODE true; //use cvODE to solve chem.
75 cvODETol 1e-7; //set solver tol.
76 }
77
78 }

```

Line 7: indicates we want to use PHREEQCs reaction module to solve the chemistry.

Line 9: indicate all mineral names that are required to be tracked for this run.

Lines 11 - 16: indicate the permeability model (KozenyCarman) being used for this benchmark and define an initial value (K_0). This value is only for reference, and the data provided by the user through `setFieldsDict` will be used for the calculation.

Lines 18 - 25: indicate the dispersion model being used for the case. Line #21 specifies the molecular diffusion coefficient and Line #22 specifies the hydrodynamic dispersion coefficient.

Lines 27 - 66: define properties of minerals present within the system which include – density (ρ_{hos}) and molar volume (V_m). Additionally, the models used to define the evolution of mineral surface area are indicated along with the initial reactive surface area values (A_0) and initial mineral volume fraction (initVF).

Lines 68 - 76: define essential information for PHREEQC solver. Line #70 indicates the name of the PHREEQC input file comprising of details regarding the initial solution and solution being injected into the system. The file `test.phr` can be found in the `constant` repository. Line #71, indicates the chemistry database file which indicates the species present within the system, potential chemical reactions taking place within the system and kinetic reaction rate for celestite grains. The file `JenaFin.DAT` can be found in the `constant` repository.

Running the simulation: we run the simulation using `darcyFoam` solver. The flow within the system is established based on single phase Darcys law. We followed the reactive transport processes within the flow cell for 300 hours.

Post-processing: Fig. 5 shows the porosity and permeability data at 150 hours and 300 hours of simulation.

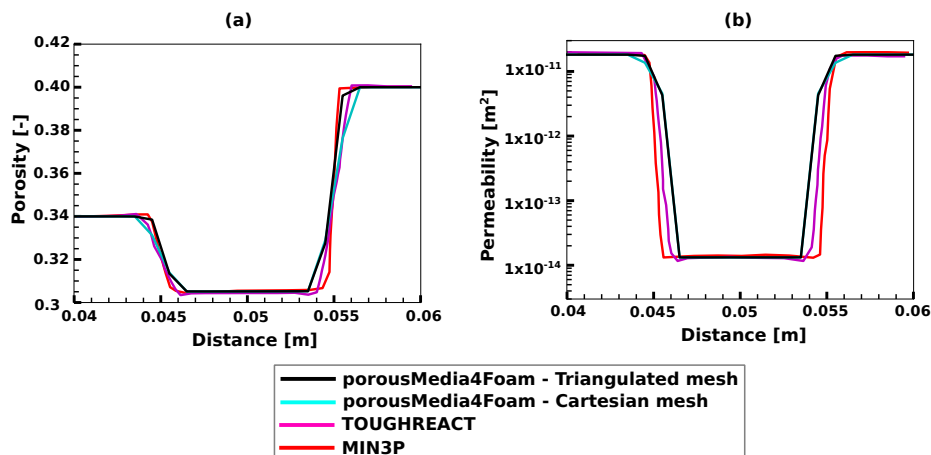


Figure 5: Evolution of (a) porosity and (b) permeability at 300 hours. The results of TOUGHREACT and MIN3P are taken from [37]. The data was collected at a height of $y = 0.01$ m (from bottom of the flow cell) with the Q2 zone lying between $x = 0.045$ m and $x = 0.055$ m.

Fig. 6 shows the primary ion species concentration at 150 hours and 300 hours of simulation.

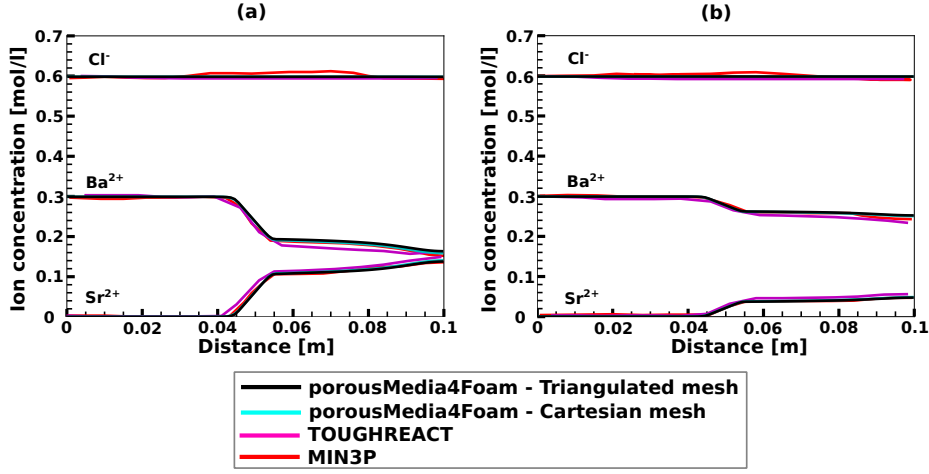


Figure 6: Evolution of primary ion species concentration (a) 150 hours and (b) 300 hours. Results of TOUGHREACT and MIN3P are taken from [37]. The data was collected at a height of $y = 0.01$ m (from bottom of the flow cell) along the length of the flow cell.

To extract the field data of porosity, permeability, primary ion concentrations shown in Figs. 5, 6 we make use of system/singleGraph file. Users can refer to the earlier discussed example to understand the description of the singleGraph file.

Fig. 7 shows the evolution of mineral concentrations (total celestite, small celestite, large celestite and barite) over time. To obtain the mineral concentration in moles (mol), the total volume of the mineral within the system is computed initially using Paraview's *integrate variables* function. The mineral volume is then divided by the mineral molar volume to obtain the concentration of mineral.

For Figs. 5, 6, 7, we compare the results of porousMedia4Foam with TOUGHREACT and MIN3P data taken from [37].

Analysing results of the discussed benchmark: As we notice from Figs. 5, 6, 7 – the results of *porousMedia4Foam* (considering Cartesian, triangulated meshes) are consistent with the results of MIN3P and TOUGHREACT. As Ba²⁺ ions reach the Q2 zone comprising of celestite minerals, the smaller celestite grains having larger reactive surface area are prone to faster reaction compared to larger celestite grains having smaller reactive surface area. Hence, we notice in Fig. 7 the smaller celestite grains concentration depletes over time however, the concentration of larger celestite grains only marginally depletes. The reaction of Ba²⁺ with celestite releases Sr²⁺ and SO₄²⁻. The incoming Ba²⁺ ions react with SO₄²⁻ resulting in the precipitation of barite. As we notice in Fig. 7, as the concentration of celestite keeps decreasing, the concentration of barite keeps rising.

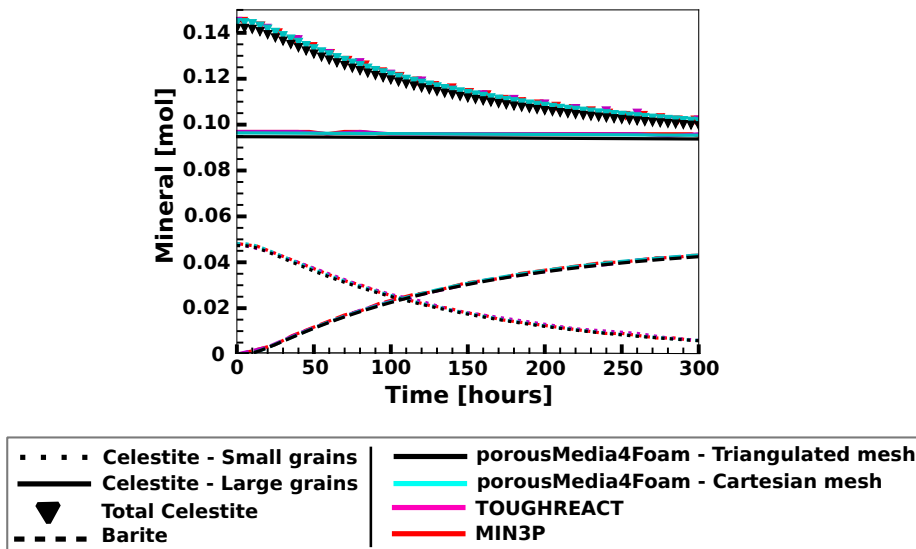


Figure 7: Mineral concentration of celestite and barite within the flow cell. Results of TOUGHREACT and MIN3P are taken from [37].

Due to difference in the molar volume of celestite and barite, the dissolution of celestite and precipitation of barite results in a decrease in the porosity of the Q2 zone. Subsequently, according to Kozeny-Carman relation the permeability of the Q2 zone decreases as seen in Fig. 5.

Following the above described three steps – **pre-processing, running the simulation, post-processing**, the user can run any hydro-geochemical cases by tuning the above mentioned files accordingly.

Multiphase benchmark using *porousMedia4Foam*

After witnessing the modelling capabilities of *porousMedia4Foam* in a monophasic system, we proceed to extend the package for a diphasic system. Similar sort of diphasic system environments are encountered in geothermal systems and conventional ground water flows such as in vadoze zones. Currently, the two-phase hydro-geochemical modelling section is under development by the authors. In this chapter, we show a simple benchmark to compare the results of PHREEQC with that of the coupled solver. We use the `uphreeqcRM` geochemical module (see Section 5.2.5) to run the following test case. We use a fixed pressure - varying volume setup for the run. Hence, the pressure remains constant throughout the run but the saturation may vary.

4.1 Case setup

For this benchmark case we consider a 1D vertical column of length $\ell = 1$ m. The column comprise of no solids i.e. the porosity is unity. The column is spatially discretized into 400 cells (each cell $dy = 2.5$ mm). The gravity has been turned off, and the column is initially occupied by the liquid phase ($S_b = 0.999$). The temperature of the column is set to 240°C . There is no injection of fluids into the system. After initialising the system, and running the case, we immediately attain an equilibrium solution according to the water-steam thermodynamic parameters. The water-steam saturation curve is shown in Fig. 8 for reference.

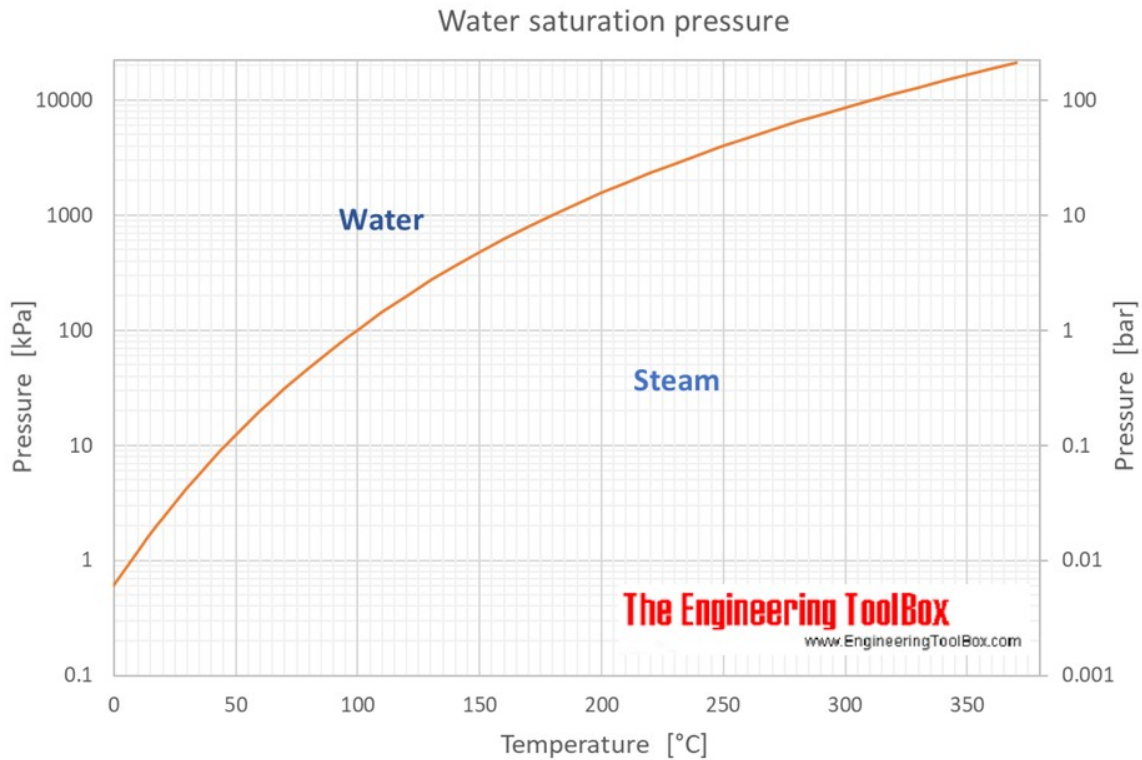


Figure 8: Water - steam saturation curve. The red line indicates the phase transition zone of water. Image source: https://www.engineeringtoolbox.com/water-vapor-saturation-pressure-d_599.html.

4.2 Running the case

After setting up the system as mentioned above, the case is run with the two-phase Darcy solver - `impesFoamMMFRT`.

4.3 Post-processing the results

We compare the results of the coupled solver against the solution of PHREEQC. Keeping the temperature constant at $T = 240^\circ$, the pressure is changed to 20 atm, 30 atm and 40 atm. As it can be seen in Fig. 8, upon increasing the pressure, the phase transition occurs. When the $P = 20$ atm and $T = 240^\circ$ (zone below the two-phase equilibrium), the water completely vaporises to form steam and form a single phase system. When the pressure is increased to $P = 30$ atm, the system is exactly on the phase equilibrium. That is, the control volumes would partially be saturated by the liquid phase and the rest occupied by the gaseous phase (steam). This results in a two-phase system. If the pressure is further increased $P = 40$ atm, the system is above the two-phase equilibrium and the system continues to remain in the liquid state.

We compare the results of saturation (S_b - which is the liquid saturation) and the number of moles of vapour in a cell volume ($Y_{H_2O}^g$). As noticed in Fig. 9, we observe a consistent match of the results between the coupled solver and PHREEQC. We restrict the saturation of the liquid phase to lie inbetween $0.001 \leq S_b \leq 0.999$. These results

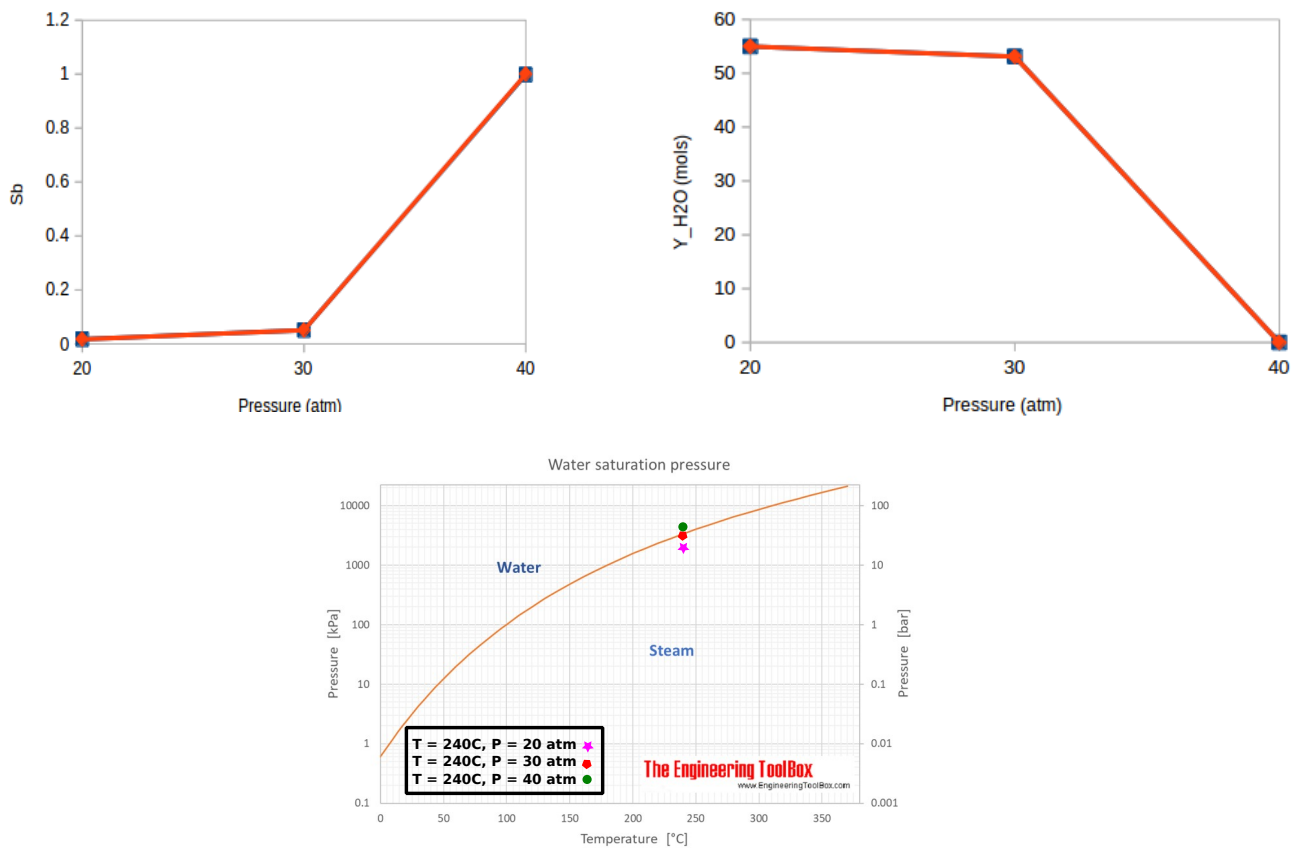


Figure 9: Comparison of saturation and vapor moles in a system when the temperature is fixed at $T = 240^\circ\text{C}$ and the pressure is varied between $P = 20, 30, 40$ atms. The coupled solver results are benchmarked against the results of PHREEQC.

indicate a successful coupling of the OpenFOAM package with PHREEQC over which we would like to build the code further.

Mathematical models implemented in *porousMedia4Foam*

As we have shown the modelling capabilities of *porousMedia4Foam* in Chapters 3, 4 we now go underneath the skin of *porousMedia4Foam* to see what equations are being solved during the simulations and what modelling options are currently on offer to the users. As we had briefly discussed the main elements of *porousMedia4Foam* in Chapter 1 by referring to Fig. 1, we will commence this chapter by elaborating a bit more about these key elements.

5.1 Porous media models

Due to the ongoing hydro-geochemical interactions, mineral precipitation/ dissolution processes occur ubiquitously within a reactive environment. These processes subsequently alter the porosity and nature of fluid flow within the porous medium. A common approach to quantify changes in fluid flows is by linking the evolution of porosity ϕ , mineral volume fractions Y_s to parameters such as permeability k , dispersivity D^* , and, reactive surface area of minerals A_c .

The source files for this section can be found in the following repository:

`porousMedia4Foam/libraries/porousModels`.

This repository comprises of two base modules namely `porousModels` and `unsaturatedPorousModels`. The source code for base module `porousModels` is available in:

`porousMedia4Foam/libraries/porousModels/porousModels`.

Within this module there exist few other sub-modules namely `absolutePermeabil-`

ityModels, dispersionModels, surfaceAreaModels. These sub-modules comprise of the mathematical models that describe the evolution of porous media and transport properties. Frequently used models that describe the evolution of porous media and transport properties are already available for use in these sub-modules. We discuss these models briefly by indicating their mathematical equations.

5.1.1 absolutePermeabilityModels

Permeability is a property of the porous medium that measures the ability of the medium to transmit fluids. The units are m^2 . For reference, permeability occurs in Eqs. 12 (Darcy-Brinkman-Stokes equation), 14 (Darcys equation), 20, 21 (Darcys two phase flow equations). Usually the permeability evolves as a function of porosity ϕ and the models implemented in *porousMedia4Foam* are:

Name	Expression	Comments
none	$k = 0$	
Constant	$k = k_0$	k_0 = initial permeability
Power-law	$k = k_0 \left(\frac{\phi}{\phi_0}\right)^n$	n = user defined variable
Kozeny-Carman	$k = k_0 \left(\frac{\phi}{\phi_0}\right)^3 \left(\frac{1-\phi_0}{1-\phi}\right)^2$	[17][7]
Hele-Shaw	$k = \frac{h^2}{12}$	

Table 2: Summary of the permeability-porosity models implemented in *porousMedia4Foam*. In the table, ϕ_0 refers to porosity data at initial time $t = 0$ s.

5.1.2 dispersionModels:

Transport of species in a medium are dependent on two factors: 1. the advective fluxes, and, 2. the dispersivity. The dispersivity of species j represented by D_j^* is a sum of molecular diffusion D_j and hydrodynamic dispersion α_L . This term occurs in the transport equation of species Eq. 6. Table 3 indicate all the dispersivity models available in the present release.

Name	Expression	Comments
none	$D_j^* = 0$	for modelling transport by advection only
diffusionOnly	$D_j^* = D_j$	
archiesLaw	$D_j^* = \phi^n D_j$	tortuosity represented by ϕ^n . By default, $n = 0$
linearDispersion	$D_j^* = \phi^n D_j \left(1 + \frac{\alpha_L}{D_j} U \right)$	tortuosity represented by ϕ^n

Table 3: Summary of the dispersion models implemented in *porousMedia4Foam*.

5.1.3 surfaceAreaModels:

Reactive surface area of minerals A_e controls the reaction rate of minerals subjected to hydro-geochemical interactions which consequently impact the mass transfer rate in Eq. 5. The units are m^2/m^3 . Typically, this parameter is linked to specific minerals volume fraction $Y_{s,i}$. The implemented models in *porousMedia4Foam* that describe the

evolution of reactive surface area of minerals are given in the table below.

Name	Expression	Comments
None	$A_e = 0$	used if A_e is not necessary
Constant	$A_e = A_0$	
Volume of solid	$A_e = \nabla Y_s Y_s$	pore-scale [41]
Power-law	$A_{e,i} = A_{0,i} (Y_{s,i})^n$	n = user defined variable

Table 4: Summary of the specific surface area models implemented in *porousMedia4Foam*. Units of specific surface area are m^{-1} .

In addition to the above mentioned parameters, additional parameters are required to describe multiphase systems. These include defining the, 1. relative permeability $k_{r,\alpha}$, 2. capillary pressure p_c , and, 3. effective saturation S_e . Popular choices of empirical relations that describe the evolution of relative permeability and capillary pressure (eg: Brooks-Corey [6], Van Genuchten [51]) are available as part of this package. The effective saturation is a parameter that is used while computing the relative permeability and capillary pressure. The effective saturation is calculated using a relation that requires specifying the irreducible saturation of both phases [14]. These properties are defined in the module `unsaturatedPorousModels`. The source code for this module can be found in:

`porousMedia4Foam/libraries/porousModels/unsaturatedPorousModels`. Within this module there exist the following sub-modules namely `capillarityModels`, `relativePermeabilityModels`, `reducedSaturationModels`. In the following discussion 'b' refers to the wetting phase and 'a' refers to the non-wetting phase.

5.1.4 reducedSaturationModels:

Reduced/ effective saturation S_e is a dimensionless parameter used to compute the relative permeability $k_{r,\alpha}$ and capillary pressure p_c (discussed below). There are two basic models implemented in *porousMedia4Foam* to compute the reduced saturation and are indicated in Table 5.

Name	Expression	Comments
minMax	$S_e = \frac{S_b - S_{min}}{S_{max} - S_{min}}$	S_{min}, S_{max} are user defined variables
saturation	$S_e = S_b$	value equal to saturation

Table 5: Summary of the models implemented in *porousMedia4Foam* to compute the reduced/ effective saturation.

5.1.5 capillarityModels:

The capillary pressure indicates the pressure difference between the two phases (wetting and non-wetting phases) and has the unit of Pascal Pa . At the Darcy scale, the

capillarity is taken as a function of the effective saturation S_e (discussed above). The capillary pressure is used in Eq. 21 while solving for two-phase Darcy flow. Frequently used models to compute capillary pressure at Darcy scale are shown in Table 6.

Name	Expression	Comments
none	$p_c = 0$	no capillary pressure.
BrooksAndCorey	$p_c(S_e) = p_{c,0} S_e^{-\alpha}$	$p_{c,0}$ = entry capillary pressure, $\frac{1}{\alpha}$ = pore-size distribution index. $p_{c,0}, \alpha$ provided by user.
VanGenuchten	$p_c(S_e) = p_{c,0} \left(S_e^{\frac{-1}{m}} - 1 \right)^{\frac{1}{n}}$	m = Van Genuchten coefficient, $\frac{1}{n} = 1 - m$.
Linear	$p_c(S_e) = p_{c,0} + (1 - S_e)(p_{c,max} - p_{c,0})$	$p_{c,0}$ = min. capillary pressure, $p_{c,max}$ = max. capillary pressure. $p_{c,0}, p_{c,max}$ provided by the user.

Table 6: Summary of the models implemented in *porousMedia4Foam* to compute the capillary pressure.

5.1.6 relativePermeabilityModels:

Relative permeability $k_{r,\alpha}$ is a flow property that indicates the resistance of flow of a fluid phase in a system due to the co-existence of another fluid phase in a system. The relative permeability is a dimensionless parameter and the two common empirical equations used to determine the relative permeability of fluids are given in Table 7.

Name	Expression	Comments
BrooksAndCorey	$k_{ra}(S_e) = k_{ra,max}(1 - S_e)^m$ $k_{rb}(S_e) = k_{rb,max}(S_e)^m$	$k_{ra,rb,max} = 1$ (default), m = exponent defined by user
VanGenuchten	$k_{ra}(S_e) = k_{ra,max}(1 - S_e)^{\frac{1}{2}} \left(1 - (S_e)^{\frac{1}{m}} \right)^{2m}$ $k_{rb}(S_e) = k_{rb,max}(S_e)^{\frac{1}{2}} \left(1 - \left(1 - (S_e)^{\frac{1}{m}} \right)^m \right)^2$	$k_{ra,rb,max} = 1$ (default), m = exponent defined by user,

Table 7: Summary of the models implemented in *porousMedia4Foam* to compute the relative permeability of fluids. Sub-script 'a' refers to the non-wetting phase and 'b' refers to the wetting phase.

5.2 Geochemical models

Dealing with geochemical interactions is an important aspect of reactive transport modelling. The design of the code in *porousMedia4Foam* is generic such that any 'third party' geochemical solvers can be plugged in to the existing package. One main highlight of the current release is the coupling of the OpenFOAM flow solver with PHREEQC. PHREEQC is a widely used open-source geochemical simulation platform for batch reactions.

As shown in Fig. 1 under the list of 'Geochemical Models', *porousMedia4Foam* provides a plethora of choices for its users to choose a suitable geochemical modelling platform as per requirement during the start of the simulation. Within the repository:

`porousMedia4Foam/library/porousModels`

there exist two modules named `geoChemicalModels` and `unsaturatedGeochemicalModels`.

The `geochemicalModels` module comprise of the sub-modules presented in Table 8 that describe the geochemical interactions for single phase systems. The geochemical sub-modules update the species composition C_j in the solution, the solid mineral volume fractions $Y_{s,i}$ and also return the rate of change of the solid mineral content as,

$$\dot{m}_{s,i} = -\frac{\partial \rho_{s,i} Y_{s,i}}{\partial t}, \quad (5)$$

where $\rho_{s,i}$ is the density of solid mineral i .

Name	Model	Comments
<code>phreeqcRM</code>	PHREEQC	[30]
<code>simpleFirstOrderKinetic</code>	first order kinetic, C_i is volume fraction	[41]
<code>simpleFirstOrderKineticMole</code>	first order kinetic, C_i in mol	[22]
<code>transportOnly</code>	no geochemistry	
<code>flowOnly</code>	no transport, no geochemistry	

Table 8: Summary of the geochemical packages implemented in *porousMedia4Foam* for simulating hydro-geochemical processes.

5.2.1 phreeqcRM

The `phreeqcRM` class calls the general-purpose geochemical reaction model PHREEQC through the `PhreecRM` module. It carries out the transport of the aqueous solution composition, C_j (in moles of elements), along with equilibrium and kinetic reactions with the solid minerals described by $Y_{s,i}$. The aqueous solution is defined either in terms of components or in terms of species. By default, `phreeqcRM` is set with components.

The geochemistry setup is carried out using an input file that follows PHREEQC format. Hence, the aqueous composition are defined in the block SOLUTION (0 for the composition of the injected fluid at the inlet boundary, 1 for the initial aqueous composition in the bulk). The EQUILIBRIUM_PHASES and KINETICS blocks are generated automatically within the code and the user only have to assign before the calculation which mode of reactions is used for each mineral. Moreover, *porousMedia4Foam* can load any customized database using PHREEQC format.

The coupling between transport and reactions relies on an operator-splitting approach based on the Strang's algorithm [48]. First, all species concentration fields, C_j , are transported sequentially using the advection-dispersion equations,

$$\frac{\partial \phi C_j}{\partial t} + \nabla \cdot (\mathbf{v}_f C_j) - \nabla \cdot (\phi D_j^* \nabla C_j) = 0, \quad (6)$$

where \mathbf{v}_f is the fluid velocity computed with the flow solver (see Section 5.3) and D_j^* is an effective diffusion coefficient that accounts for tortuosity and hydrodynamic dispersion effects (see Section 5.1.2). The transport equation is discretized on the computational domain using the finite-volume method and solved implicitly using OpenFOAM's engines.

Then, the volume fractions of solid minerals, $Y_{s,i}$, are updated according to phase equilibrium and/or kinetic reaction calculations provided by PHREEQC. Reaction kinetics use the surface area computed at every time steps using the surface area models in section 5.1.3. It corresponds to the surface area per volume and its units are m^2/m^3 (or m^{-1}). Hence, the RATE block provided in PHREEQC database to describe reaction rates has to be defined accordingly.

5.2.2 simpleFirstOrderKineticMole

simpleFirstOrderKineticMole is a simple geochemical engine for solving the transport of a single species labelled A that reacts with solid minerals using first order kinetic reactions. It is an extension to multiple minerals of the model used in the benchmark presented in [22] in which pore-scale simulators were used to model the dissolution of a calcite crystal by hydrochloric acid.

The chemical reaction reads,



The mass balance equation for species A reads,

$$\frac{\partial \phi C_A}{\partial t} + \nabla \cdot (\mathbf{v}_f C_A) - \nabla \cdot (\phi D_j^* \nabla C_A) = - \left(\sum_{j=1}^{N_s} A_{s,j} k_{A,j} \gamma_A \right) C_A, \quad (8)$$

where \mathbf{v}_f is the fluid velocity, D_j^* is an effective diffusion, A_e is the reactive surface area (in m^{-1}), $k_{j,s}$ is the constant of reaction of the species A with the mineral s (in $mol.m^{-2}.s^{-1}$) and γ_A is the activity of A in water (in $m^3.mol^{-1}$). In simpleFirstOrderKineticMole, the concentration field, C_j , is defined in mol/m^3 . The equation is discretized on the computational grid using the finite-volume method and solved implicitly.

The distribution of solid minerals evolves according to,

$$\frac{\partial Y_{s,i}}{\partial t} = -A_{s,A} k_{i,A} \gamma_A V_{m_{s,i}} C_A. \quad (9)$$

5.2.3 transportOnly

transportOnly solves the advection-dispersion equation,

$$\frac{\partial \phi C_j}{\partial t} + \nabla \cdot (\mathbf{v}_f C_j) - \nabla \cdot (\phi D_j^* \nabla C_j) = 0, \quad (10)$$

without considering geochemistry. It allows the transport of species using the dispersion models implemented in *porousMedia4Foam* (see Table 3).

5.2.4 flowOnly

flowOnly is an empty class for computing velocity profiles without species transport nor geochemistry. For example, [36] used the multi-scale flow solver of *porousMedia4Foam* to compute the steady-state velocity profile in absence of geochemical reactions within

a two-scale domain, i.e. a domain that contains both solid-free regions and porous regions. This option is particularly interesting in cases for which geochemistry and flow can be treated independently from each other.

Moving on to multiphase systems, the corresponding sub-modules are created within the base module labelled `unsaturatedGeochemicalModels`. Table 9 summarizes the available sub-modules to deal with two-phase geochemical interactions.

Name	Model	Comments
<code>uphreeqcRM</code>	PHREEQC	[30]
<code>unsaturatedComponentTransport</code>	no geochemistry	
<code>unsaturatedFlowOnly</code>	no transport, no geochemistry	

Table 9: Summary of the geochemical packages implemented in *porousMedia4Foam* for simulating hydro-geochemical processes in a two-phase system.

5.2.5 uphreeqcRM

The `uphreeqcRM` sub-module is built on the `phreeqcRM` sub-module to account for two phase chemical interactions. In PHREEQC, the second phase is considered to be the `GAS_PHASE`. The gas phase in PHREEQC can be treated by either fixing the volume of the gas phase in a control volume and computing the pressure or by fixing the pressure to compute the volume at given temperatures. For example, if the pressure is fixed and the volume of the gas is computed using PHREEQC, the saturation is updated after solving for the chemistry but the pressure remains constant. The updated saturation is then used in the two-phase Darcy equation to solve for the flow.

5.2.6 unsaturatedComponentTransport

This sub-module solves only for the transport of species not taking into account chemical interactions. The two-phase transport equation of species is given by,

$$\sum_{\alpha} \frac{\partial \phi S_{\alpha} C_{j\alpha}}{\partial t} + \nabla \cdot (\mathbf{v}_{f\alpha} C_{j\alpha}) - \nabla \cdot (\phi D_j^* \nabla C_{j\alpha}) = 0, \quad (11)$$

where $C_{j\alpha}$ refers to the concentration of component j in phase α , $\mathbf{v}_{f\alpha}$ refers to the velocity of phase α .

5.2.7 unsaturatedFlowOnly

This sub-module solves only two-phase Darcy flow considering neither geochemical interactions nor transport of species.

5.3 Solvers

As shown in Fig. 1, *porousMedia4Foam* comes with a rich set of solvers capable to model and analyse single and two-phase hydro-geochemical processes. *porousMedia4Foam*

for hydro-geochemical simulations includes three flow models: a multi-scale flow solver based on the micro-continuum approach, a continuum-scale Darcy solver and a constant velocity solver. Additionally, an other solver is available to model two-phase flows. (see Table 10).

Name	Model	Comments
dbsFoam	Micro-continuum Darcy-Brinkman-Stokes	pore-scale, hybrid-scale, continuum-scale
darcyFoam	Darcy's law	continuum-scale only
constantVelocityFoam	constant velocity profile	uniform/ non-uniform velocity distribution profiles
impesFoamMMFRT	two-phase flow solver	continuum-scale

Table 10: Summary of the flow solvers implemented in *porousMedia4Foam* for simulating hydro-geochemical processes.

Amongst the available solvers, there are *constantVelocityFoam* — considers a steady-state velocity all through the simulation run, *darcyFoam*, *impesFoam_MMFRT* — solve the Darcys equation for single and multiphase flows respectively. A unique capability of *porousMedia4Foam* is the ability to model single phase hydro-geochemistry at multiple scales i.e. at the pore-, continuum-, and, the hybrid- scales using the *dbsFoam* solver. This unique feat can be achieved using the Darcy-Brinkman-Stokes multi-continuum approach that will be discussed in the next section.

5.3.1 dbsFoam: Multi-scale micro-continuum flow model

dbsFoam is a multi-scale flow solver based on the micro-continuum modelling approach developed in [43]. Micro-continuum approaches are intermediate between a pure Navier-Stokes description of the transport for which all the porosity is fully resolved , and a pure continuum-scale modelling for which the flow is governed by Darcy's law This hybrid-scale approach relies on the Darcy-Brinkman-Stokes (DBS) equation [5] that allows for the modelling of flow and transport in regions free of solid and porous regions in a single framework [24, 43]. DBS equation arises from the integration of Navier-Stokes equations over a control volume [50, 15, 4, 12]. The momentum equation reads,

$$\frac{1}{\phi} \left(\frac{\partial \rho_f \mathbf{v}_f}{\partial t} + \nabla \cdot \left(\frac{\rho_f}{\phi} \mathbf{v}_f \mathbf{v}_f \right) \right) = -\nabla p_f + \rho_f \mathbf{g} + \nabla \cdot \left(\frac{\mu_f}{\phi} (\nabla \mathbf{v}_f + {}^t \nabla \mathbf{v}_f) \right) - \mu_f k^{-1} \mathbf{v}_f, \quad (12)$$

where ϕ is the porosity, \mathbf{v}_f is the seepage velocity, p_f is the fluid pressure, \mathbf{g} is the gravity, ρ_f is the fluid density, μ_f is the fluid viscosity and k is the matrix permeability. The porous media properties including porosity and permeability change dynamically with geochemical processes and are updated at every time steps.

Eq. (12) is valid throughout the computational domain regardless the content of a cell. In regions that contain fluid only, $\phi = 1$, and the drag force $\mu_f k^{-1} \mathbf{v}_f$ vanishes so that the momentum equation tends to the Navier-Stokes equation. In regions that contain an aggregate of fluid and solid, $0 < \phi < 1$, and the drag force is dominant over the inertial and viscous forces so that Eq. (12) tends asymptotically to Darcy's law.

The momentum equation, Eq. (12), can be used to model pore-scale flows. Indeed, if a solid region is approximated by a low-permeability low-porosity matrix, the velocity in this region goes to near zero which forces a no-slip boundary condition at the fluid/-solid interface. This feature of the DBS equation is particularly relevant to solve Navier-Stokes problems using Cartesian grids only (also called penalized approach) [1, 43] and to move the fluid/solid interface with respect to geochemical processes such as precipitation/dissolution [41, 22] or swelling by using the local porosity field, ϕ , as a phase indicator function [8].

The pressure-velocity coupling is achieved by solving the momentum equation along with the micro-continuum continuity equation for multiple minerals. For an incompressible Newtonian aqueous fluid, the latter reads,

$$\nabla \cdot \mathbf{v}_f = \sum_{i=1}^{N_s} \dot{m}_{s,i} \left(\frac{1}{\rho_f} - \frac{1}{\rho_{s,i}} \right), \quad (13)$$

where $\rho_{s,i}$ is the solid density of mineral i and $\dot{m}_{s,i}$ represents the rate of phase change of the solid into fluid, or of a fluid into solid. For example, it can represent the rate of solid minerals that is dissolved into aqueous solution. Inversely, it can describe an amount of fluid that is removed of a control volume because it has precipitated. The right-hand side of Eq. (13) is provided by the geochemistry calculation (Section 5.2). Although this term is often neglected in continuum-scale models, it may be necessary to keep it to ensure the mass balance at the fluid/solid interface in pore-scale simulations [41], as well as in continuum scale simulations [38].

The flow model formed by Eqs (12)-(13) is discretized using the finite volume method and solved sequentially. The pressure-velocity coupling is handled by a predictor-corrector strategy based on the PIMPLE algorithm implemented in OpenFOAM. It consists in a combination of PISO (Pressure Implicit with Splitting of Operator, [16]) and SIMPLE (Semi-Implicit Method for Pressure Linked Equations, [31]). PIMPLE algorithm allows both transient and steady-state simulations. Moreover, PIMPLE enables larger time steps than PISO. Further information regarding the numerics is found in [43].

5.3.2 darcyFoam: Darcy's law

darcyFoam is a standard continuum-scale solver that is based on the Darcy's law,

$$\mathbf{v}_f = -\frac{k}{\mu_f} (\nabla p_f - \rho_f \mathbf{g}), \quad (14)$$

for describing flow in porous media. Numerically, Eq. (14) is combined along with Eq. (13) to form a Laplace equation solving implicitly for the pressure field, p_f . Then, the velocity field is calculated point-wise using Eq. (14) and p_f . If activatePorosityFeedback is switched on, Darcy's law is recalculated at every time steps to update the velocity and pressure fields according to the new permeability value.

Boundary conditions can be described by imposing fixed pressure or fixed velocity values on the domain edges. However, as darcyFoam solves implicitly for the pressure

field, the boundary conditions on the velocity are transformed into pressure gradient conditions using Darcy's law:

$$\mathbf{n} \cdot \nabla p_f = -\mathbf{n} \cdot (\mu_f k^{-1} \mathbf{v}_f - \rho_f \mathbf{g}), \quad (15)$$

where \mathbf{n} is the vector normal to the domain boundary. In the code, Eq. (15) is achieved using the boundary condition `darcyGradPressure` [14].

5.3.3 constantVelocityFoam: Constant flow rate

The flow solver `constantVelocityFoam` is used to model cases in which the chemical species are transported using a steady-state velocity field, \mathbf{v}_f –uniform or non-uniform– provided as an input data that can come from a separate flow simulation. `constantVelocityFoam` is particularly useful if the feedback between geochemical reactions and the flow is negligible. Indeed, in such a case, the characteristic timescale of flow changes is much longer than the characteristic time of species transport and the calculation of the velocity profile can be uncoupled from the species transport.

5.3.4 impesFoamMMFRT: two-phase flow solver

The mass balance equations for a two-phase flow systems are given by,

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \mathbf{v}_{f\alpha} = q_\alpha, \quad (16)$$

where $\mathbf{v}_{f\alpha}$ is the fluid velocity. The fluid velocity is determined using Darcy's equation,

$$\mathbf{v}_{f\alpha} = -\frac{k k_{r\alpha}}{\mu_\alpha} (\nabla p_\alpha - \rho_\alpha \mathbf{g}). \quad (17)$$

On substituting Eq. 17 in Eq. 16 and using the following supplementary relations,

$$\sum_\alpha S_\alpha = 1, \quad (18)$$

$$p_c(S_e) = p_a - p_b, \quad (19)$$

two following two mass balance equations can be derived,

$$-\phi \frac{\partial S_b}{\partial t} + \nabla \cdot \left(\frac{k k_{ra}(S_e)}{\mu_a} (\nabla p_a - \rho_a \mathbf{g}) \right) = q_a \quad (20)$$

$$\phi \frac{\partial S_b}{\partial t} + \nabla \cdot \left(\frac{k k_{rb}(S_e)}{\mu_b} (\nabla p_a - \rho_b \mathbf{g} - \nabla p_c(S_e)) \right) = q_b \quad (21)$$

The above system of equations are solved for p_a and S_b using the IMPES (Implicit Pressure Explicit Saturation) algorithm. An elaborate discussion regarding the procedure followed to solved the formed equations is discussed in [14].

6

Creating a new library, solver in *porousMedia4Foam*

porousMedia4Foam comes loaded with a vast variety of commonly used models that describe the evolution of porous media and transport properties as elaborated in Chapter 5. As we had mentioned in Chapter 1 that reactive transport processes involve complicated non-linear coupling of Thermo-Hydro-Mechanical-Chemical (THMC) parameters, the users might end up requiring to change the code to create new libraries and/or solvers as per requirements based on experimental evidences. In this chapter, we provide an example on how to create a new permeability model and a compressible single phase Darcy solver in *porousMedia4Foam*.

6.1 Creating a new permeability model

The list of permeability models implemented in *porousMedia4Foam* are discussed in Section 5.1.1. In this section we explain how to add a new permeability model in the package. For example, if the user wants to implement the following model to describe the evolution of permeability based on porosity ϕ during the run,

$$k = k_0 \phi_{n-1}^m, \quad (22)$$

we detail the procedure below. In the above equation, k is the permeability, k_0 refers to the initial permeability of the medium, ϕ_{n-1}^m refers to the porosity at previous time step and m is a user defined exponent. We implement this model in the `absolutePermeabilityModels/example` for reference. Three sequential steps to be followed to incorporate this model into *porousMedia4Foam* are – 1. code the mathematical expression, 2. compile the code, and, 3. use the created model.

6.1.1 Coding the mathematical expression:

Firstly, we move to the directory where all the permeability models are implemented,

```
cd porousMedia4Foam/libraries/porousModels/porousModels/absolutePermeabilityModels.
```

We then make a copy (named: `example`) of an existing model (`heterogeneousScalarConstant`) as follows,

```
cp -r heterogeneousScalarConstant example.
```

Within the created `example` repository, we rename the `heterogeneousScalarConstant.C`, `heterogeneousScalarConstant.H` files as `example.C`, `example.H` respectively. The user need to enter both these files (`example.H`, `example.C`) and replace the word 'heterogeneousScalarConstant' with 'example'.

In `example.H` file, we add the following three lines (Lines #72 - #74),

```
word epsName_;
const volScalarField & eps_;
scalar m;
```

In `example.C` file, we add the following three lines to initialize the variables introduced above (Lines #91 - #93),

```
epsName_(exampleDict_.lookupOrDefault<word>("eps", "eps")),
eps_(mesh.lookupObject<volScalarField>(epsName_)),
m(exampleDict_.lookupOrDefault("m", 2))
```

The first two lines above tells the solver to search for the porosity `eps` in the `transportProperties` file or in the output files written during the run. The third line above initializes a default scalar value for the exponent $m = 2$ or gives an opportunity for the user to select a value in the `constant/transportProperties` file.

During the simulation, the `updatePermeability()` function is called where the cell centre value of permeability (`K_`), cell surface value of permeability (`Kf_`) and inverse of permeability (`invK_`) are computed as shown below.

```
void Foam::absolutePermeabilityModels::example::updatePermeability()
{
K_ = K0_*Foam::pow(eps_.oldTime(), m);
//K = K0*(eps)^m, eps @ old time step
Kf_ = fvc::interpolate(K_);
invK_ = 1./K_;
}
```

6.1.2 Compiling the implemented model:

Now, we tell the compiler to also include the created model while compiling the package. In order to do so, we add the following line (Line #6) in file `porousMedia4Foam/libraries/porousModels/Make/files`:

```
porousModels/absolutePermeabilityModels/example/example.C.
```

Upon running `.\Allwmake` from the `porousMedia4Foam` repository, the file is compiled and all the essential data to use this model will be available in the created corresponding shared object library named `libporousModels.so`.

6.1.3 Using the implemented model:

Finally, to use the presented model, the user need to mention the model name (example) while defining the `absolutePermeabilityModel` in `constant/transportProperties` file.

A sample of a snippet to use the 'example' model to evaluate the permeability would look like,

```
absolutePermeabilityModel example;  
exampleCoeffs  
{  
m 1; //exponent for the porosity  
}
```

6.2 Creating a new solver

The list of available solvers in *porousMedia4Foam* is discussed in Section 5.3. In this section we discuss the procedure one has to follow to create a new solver. As an example, we create a new solver named **compressibleDarcyFoam**. The compressible single phase Darcy equation is given by,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = q, \quad (23)$$

where ρ is the fluid density, t is the time, U is the Darcy velocity (Eq. 14) and q is the source/ sink term. For reference the source code for this solver can be found in:

```
porousMedia4Foam/solvers/compressibleDarcyFoam.
```

We commence by making a copy of the `darcyFoam` solver named `compressibleDarcyFoam`. Within `compressibleDarcyFoam`, we edit the name of the file `darcyFoam.C` to `compressibleDarcyFoam.C`. Within the `compressibleDarcyFoam.C` file, the compressible Darcy equation (Eq. 23) is solved as:

```
fvScalarMatrix pEqn  
(  
eps*fvc::ddt(Rho) - fvm::laplacian(rhorAUf,p) + fvc::div(phig)  
);
```

```
pEqn.solve();
```

Solving the above `pEqn.solve()` will establish a flux caused due to pressure gradient. To obtain the total flux, one must add the flux caused due to pressure gradient and gravitational forces. Later, the velocity of the fluid is back calculated from the computed flux (units in kg/s) field. For running this solver, few additional variables are required apart from those used for the incompressible Darcy flow solver. These are defined in the `createFields.H` file.

Finally, in the `Make/files`, name of the file to be compiled shall be updated from `darcyFoam.C` to `compressibleDarcyFoam.C`. For convenience and to avoid confusion the name of the executable is also changed as *compressibleDarcyFoam*. Finally, in `porousMedia4Foam/./Allwmake` file the following line is added: `wmake solvers/compressibleDarcyFoam` to consider the created solver for compilation.

The same procedure is to be followed by the user to create any new models or solvers in *porousMedia4Foam*.

References

- [1] Angot, P., Bruneau, C.H., Fabrie, P.: A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik* **81**(4), 497–520 (1999)
- [2] Bea, S.A., Mayer, U., MacQuarrie, K.: Reactive transport and thermo-hydro-mechanical coupling in deep sedimentary basins affected by glaciation cycles: model development, verification, and illustrative example. *Geofluids* **16**(2), 279–300 (2016)
- [3] Bildstein, O., Claret, F.: Stability of clay barriers under chemical perturbations. *Developments in Clay Science* **6**, 155–188 (2015)
- [4] Bousquet-Melou, P., Goyeau, B., Quintard, M., Fichot, F., Gobin, D.: Average momentum equation for interdendritic flow in a solidifying columnar mushy zone. *International Journal of Heat and Mass Transfer* **45**(17), 3651 – 3665 (2002). DOI [http://dx.doi.org/10.1016/S0017-9310\(02\)00077-7](http://dx.doi.org/10.1016/S0017-9310(02)00077-7). URL <http://www.sciencedirect.com/science/article/pii/S0017931002000777>
- [5] Brinkman, H.C.: A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles. *Appl. Sci. Res.* **A1**, 27–34 (1947)
- [6] Brooks, R.H., Corey, A.T.: Hydraulic properties of porous media. Ph.D. thesis, Colorado State University. Libraries (1964)

- [7] Carman, P.C.: Fluid flow through granular beds. *Trans. Inst. Chem. Eng.* **15**, 150–166 (1937)
- [8] Carrillo, F.J., Bourg, I.C.: A darcy-brinkman-biot approach to modeling the hydrology and mechanics of porous media containing macropores and deformable microporous regions. *Water Resources Research* **55**(10), 8096–8121 (2019). DOI 10.1029/2019WR024712. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019WR024712>
- [9] Druhan, J., Tournassat, C., et al.: *Reactive Transport in Natural and Engineered Systems*, vol. 85. Walter de Gruyter GmbH & Co KG (2020)
- [10] Espinosa-Gayosso, A., Ghisalberti, M., Ivey, G.N., Jones, N.L.: Particle capture and low-reynolds-number flow around a circular cylinder. *Journal of Fluid mechanics* **710**, 362–378 (2012)
- [11] Gaus, I., Audigane, P., André, L., Lions, J., Jacquemet, N., Durst, P., Czernichowski-Lauriol, I., Azaroual, M.: Geochemical and solute transport modelling for co2 storage, what to expect from it? *International journal of greenhouse gas control* **2**(4), 605–625 (2008)
- [12] Goyeau, B., Lhuillier, D., Gobin, D., Velarde, M.: Momentum transport at a fluid-porous interface. *International Journal of Heat and Mass Transfer* **46**(21), 4071–4081 (2003)
- [13] Greenshields, C.J., et al.: *Openfoam user guide*. OpenFOAM Foundation Ltd, version **3**(1), 47 (2015)
- [14] Horgue, P., Soullain, C., Franc, J., Guibert, R., Debenest, G.: An open-source toolbox for multiphase flow in porous media. *Computer Physics Communications* **187**, 217–226 (2015)
- [15] Hsu, C., Cheng, P.: Thermal dispersion in a porous medium. *International Journal of Heat and Mass Transfer* **33**(8), 1587–1597 (1990)
- [16] Issa, R.I.: Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics* **62**, 40–65 (1985)
- [17] Kozeny, J.: *Über kapillare leitung der wasser in boden*. Royal Academy of Science, Vienna, Proc. Class I **136**, 271–306 (1927)
- [18] Lichtner, P.C., Hammond, G.E., Lu, C., Karra, S., Bisht, G., Andre, B., Mills, R., Kumar, J.: *Pflotran user manual: A massively parallel reactive flow and transport model for describing surface and subsurface processes*. Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States); Sandia ... (2015)
- [19] Lysenko, D.A., Ertesvåg, I.S., Rian, K.E.: Modeling of turbulent separated flows using openfoam. *Computers & Fluids* **80**, 408–422 (2013)

- [20] Mayer, K., Amos, R., Molins, S., Gerard, F.: Reactive transport modeling in variably saturated media with MIN3P: Basic model formulation and model enhancements, vol. 26. Bentham Science Publishers Sharjah, UAE (2012)
- [21] Mayer, K.U., Frind, E.O., Blowes, D.W.: Multicomponent reactive transport modeling in variably saturated porous media using a generalized formulation for kinetically controlled reactions. *Water Resources Research* **38**(9), 13–1 (2002)
- [22] Molins, S., Soullaine, C., Prasianakis, N.I., Abbasi, A., Poncet, P., Ladd, A.J., Starchenko, V., Roman, S., Trebotich, D., Tchelepi, H.A., et al.: Simulation of mineral dissolution at the pore scale with evolving fluid-solid interfaces: Review of approaches and benchmark problem set. *Computational Geosciences* pp. 1–34 (2020)
- [23] Nabil, M., Rattner, A.S.: interthermalphasechange foam—a framework for two-phase flow simulations with thermally driven phase change. *SoftwareX* **5**, 216–226 (2016)
- [24] Neale, G., Nader, W.: Practical significance of brinkman’s extension of darcy’s law: coupled parallel flows within a channel and a bounding porous medium. *The Canadian Journal of Chemical Engineering* **52**(4), 475–478 (1974)
- [25] Nieves-Remacha, M.J., Kulkarni, A.A., Jensen, K.F.: Openfoam computational fluid dynamic simulations of single-phase flows in an advanced-flow reactor. *Industrial & Engineering Chemistry Research* **54**(30), 7543–7553 (2015)
- [26] Noiriél, C.: Resolving time-dependent evolution of pore-scale structure, permeability and reactivity using x-ray microtomography. *Reviews in Mineralogy and Geochemistry* **80**(1), 247–285 (2015)
- [27] Noiriél, C., Daval, D.: Pore-scale geochemical reactivity associated with co₂ storage: new frontiers at the fluid–solid interface. *Accounts of chemical research* **50**(4), 759–768 (2017)
- [28] Noiriél, C., Gouze, P., Bernard, D.: Investigation of porosity and permeability effects from microstructure changes during limestone dissolution. *Geophysical research letters* **31**(24) (2004)
- [29] Noiriél, C., Steefel, C.I., Yang, L., Bernard, D.: Effects of pore-scale precipitation on permeability and flow. *Advances in water resources* **95**, 125–137 (2016)
- [30] Parkhurst, D.L., Wissmeier, L.: Phreeqcrm: A reaction module for transport simulators based on the geochemical model phreeqc. *Advances in Water Resources* **83**, 176–189 (2015)
- [31] Patankar, S.V.: *Numerical Heat Transfer And Fluid Flow*. Taylor & Francis, Washington, DC (1980)

- [32] Pavuluri, S.: Direct numerical simulations of spontaneous imbibition at the pore-scale: impact of parasitic currents and dynamic capillary barriers. Ph.D. thesis, Heriot-Watt University (2019)
- [33] Pavuluri, S., Maes, J., Doster, F.: Spontaneous imbibition in a microchannel: Analytical solution and assessment of volume of fluid formulations. *Microfluidics and Nanofluidics* **22**(8), 1–18 (2018)
- [34] Pavuluri, S., Maes, J., Yang, J., Regaieg, M., Moncorgé, A., Doster, F.: Towards pore network modelling of spontaneous imbibition: contact angle dependent invasion patterns and the occurrence of dynamic capillary barriers. *Computational Geosciences* pp. 1–19 (2019)
- [35] Pavuluri, S., Tournassat, C., Claret, F., Soullaine, C.: Reactive transport modelling with a coupled openfoam[®]-phreeqc platform (2021)
- [36] Poonoosamy, J., Soullaine, C., Burmeister, A., Deissmann, G., Bosbach, D., Roman, S.: Microfluidic flow-through reactor and 3d raman imaging for in situ assessment of mineral reactivity in porous and fractured porous media. *Lab on a Chip* **20**(14), 2562–2571 (2020)
- [37] Poonoosamy, J., Wanner, C., Epping, P.A., Águila, J., Samper, J., Montenegro, L., Xie, M., Su, D., Mayer, K., Mäder, U., et al.: Benchmarking of reactive transport codes for 2d simulations with mineral dissolution–precipitation reactions and feedback on transport parameters. *Computational geosciences* pp. 1–22 (2018)
- [38] Seigneur, N., Mayer, K.U., Steefel, C.I.: Reactive transport in evolving porous media. *Reviews in Mineralogy and Geochemistry* **85**(1), 197–238 (2019). DOI 10.2138/rmg.2019.85.7
- [39] Soullaine, C., Maes, J., Roman, S.: Computational microfluidics for geosciences. *Frontiers in Water* **3**, 11 (2021)
- [40] Soullaine, C., Pavuluri, S., Claret, F., Tournassat, C.: porousmedia4foam: Multi-scale open-source platform for hydro-geochemical simulations with openfoam. *Earth and Space Science Open Archive ESSOAr* (2021)
- [41] Soullaine, C., Roman, S., Kavscek, A., Tchelepi, H.A.: Mineral dissolution and wormholing from a pore-scale perspective. *Journal of Fluid Mechanics* **827**, 457–483 (2017)
- [42] Soullaine, C., Roman, S., Kavscek, A., Tchelepi, H.A.: Pore-scale modelling of multiphase reactive flow: application to mineral dissolution with production of CO_2 . *Journal of Fluid Mechanics* **855**, 616–645 (2018)
- [43] Soullaine, C., Tchelepi, H.A.: Micro-continuum approach for pore-scale simulation of subsurface processes. *Transport in Porous Media* **113**(3), 431–456 (2016)

- [44] Steefel, C., Appelo, C., Arora, B., Jacques, D., Kalbacher, T., Kolditz, O., Lagneau, V., Lichtner, P., Mayer, K.U., Meeussen, J., et al.: Reactive transport codes for subsurface environmental simulation. *Computational Geosciences* **19**(3), 445–478 (2015)
- [45] Steefel, C.I.: Reactive transport at the crossroads. *Reviews in Mineralogy and Geochemistry* **85**(1), 1–26 (2019)
- [46] Steefel, C.I., DePaolo, D.J., Lichtner, P.C.: Reactive transport modeling: An essential tool and a new research approach for the earth sciences. *Earth and Planetary Science Letters* **240**(3-4), 539–558 (2005)
- [47] Steefel, C.I., Molins, S.: Crunchflow. Software for modeling multicomponent reactive flow and transport. User's manual. Lawrence Berkeley National Laboratory, Berkeley (2009)
- [48] Strang, G.: On the construction and comparison of difference schemes. *SIAM journal on numerical analysis* **5**(3), 506–517 (1968)
- [49] Tournassat, C., Steefel, C.I.: Reactive transport modeling of coupled processes in nanoporous media. *Reviews in Mineralogy and Geochemistry* **85**(1), 75–109 (2019)
- [50] Vafai, K., Tien, C.: Boundary and inertia effects on flow and heat transfer in porous media. *International Journal of Heat and Mass Transfer* **24**(2), 195–203 (1981)
- [51] Van Genuchten, M.T.: A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil science society of America journal* **44**(5), 892–898 (1980)
- [52] Xie, M., Mayer, K.U., Claret, F., Alt-Epping, P., Jacques, D., Steefel, C., Chiaberge, C., Simunek, J.: Implementation and evaluation of permeability-porosity and tortuosity-porosity relationships linked to mineral dissolution-precipitation. *Computational Geosciences* **19**(3), 655–671 (2015)
- [53] Xu, T., Spycher, N., Sonnenthal, E., Zhang, G., Zheng, L., Pruess, K.: Toughreact version 2.0: A simulator for subsurface reactive transport under non-isothermal multiphase flow conditions. *Computers & Geosciences* **37**(6), 763–774 (2011)
- [54] Zhang, R., Yin, X., Winterfeld, P.H., Wu, Y.S.: A fully coupled thermal-hydrological-mechanical-chemical model for co₂ geological sequestration. *Journal of Natural Gas Science and Engineering* **28**, 280–304 (2016)