**GwaterFreakFoam a coupling of OpenFoam and Phreeqc for groundwater flow and reactive transport.**

NB: part in red are to be done

**This manual is for version 19/06/22**

# Theory

## Flow

### General equation

The model includes flow, multispecies transport and reaction. The reaction part is solved with PhreeqcRM, so please refer to the Phreeqc manual for more information for chemistry. Flow and transport are solved by OpenFoam. The advantages of OpenFoam are multiple : (i) a finite volume approach allowing complex geometry, (ii) a modular approach that allows to include rapidly new equations or complex behavior and (iii) several transport solver that perform fast and accurate solutions in the finite volume domain.

For flow, the model solves the Darcy equation

(1)

Where *Stor* is the storage term, *Mf* the mobility, the vector of gravity flow linked to density variation and *Q* the flow source term. Here:

where *k* is the permeability (m2), the density and the *µ* viscosity of the fluid.

The storage term is equal to 10-4 (or given as stor0) for confined flow and described below for unconfined flow.

where here ***g*** is the gravity vector.

If required the equation can be solved without the temporal term in order to reach steady state (it is done to initiate the model) using the SIMPLE solver included in OpenFoam.

In the openFoam formulation for the **equations**, if fvm:: is used before the concerned term, it means that it will be solved implicitly, while if fvc:: is used it means an explicit term. The implicit approach shall be preferred but it is not always possible to use it. The above equation is written:

hEqn (

stor.fvm::ddt(h) + fvm::laplacian(-Mf,h) + fvc::div(phiG) == fvOptions(h)

);

### Unconfined flow

If the model is unconfined, the model calculates the relative saturated thickness as

(2)

where *h* is the head, *zbot* is the bottom of the cell, and *thk* its thickness, all in meters (or other length units consistent with the rest of the model), so *sw* is dimensionless. The value of *sw* is bounded by 10-4 (dry cells) and 1 (fully saturated). The *sw* term is used as finally it can be considered as a relative saturation of the cell, except that this relative saturation is not distributed over the whole cell. In equation (1) *sw* decreases the section for flow and thus we set in case of unconfined flow, on contrary *sw* is simply set to one over the whole domain. It can be seen here that this part of the equation cannot be solved implicitly, and thus *sw* is recalculated at each time step. However one must be careful as in OpenFoam all the calculations are for a volume and here the unconfined properties apply only to the horizontal flow, this is why we use the values of hydraulic conductivity at faces and during calculation for unconfined flow only the vertical faces hydraulic conductivity are modified.

Under unconfined conditions the water flux phiw is where S is the surface of the concerned face, this is correct as Mf includes the effect of sw.

In OpenFoam the tensor operation to set the equation for a given geometry are transparent: OpenFoam calculates the product of the tensor parameters with the surface vector of each cell face to reach the expression of *Mf* at the cell faces. However, since its creation Modflow (and also Modflow USG) behave differently: it is assumed that the hydraulic conductivity anisotropy is not oriented according to the horizontal direction but is parallel to the sediment layer, even if it is tilted. A specific option for the permeability field has therefore been written to reproduce this behavior. For the inter-layer faces the hydraulic conductivity is set to the value of Kv (vertical hydraulic conductivity) whether the face is horizontal or tilted. At the same time the harmonic averaging is done.

### Unsaturated medium

For unsaturated medium the major variable is *hp* the pressure head, and the equation is similar as Eq. 1 (Bear 1988):

(3)

Here the storage coefficient is replaced by *Ch* the capillary capacity, with (Nielsen, Th. Van Genuchten, and Biggar 1986):

(4)

Where *sw* has not the same meaning as for the saturated medium as it is here the water saturation, its maximum value and its minimum, with *Se* the effective saturation expressed as . and *m* are the Van Genuchten coefficient of the capillary curve (Genuchten 1980).

### Dual phase flow

In its present formulation the gwaterFreakFoam library includes multiphase but for two phases only. The mathematical formulation is the same as the one provided by Horgue et al. (2015), as this part of the library comes from their work, which can be expressed as follows:

(5)

And

(6)

Where is the gas pressure, the porosity, and the capillary and water fluxes across cell surface respectively. Here the fluid mobility *Mf* is the total mobility which includes water and gas.

The two equations are solved iteratively with the *P* term being implicit and the flux term being explicit in eq.5. For eq. 6, the *sw* term is implicit and the other terms are explicit.

The classical formulation of relative permeability is used with the Mualem approach (Mualem 1986).

## Transport

The transport of any component is based on the following equation (Zheng and Bennett 2002):

(7)

Where is the concentration in the fluid phase i (aqueous or gas), the fluid content ( for unconfined or for unsaturated medium), the fluid phase flux and the dispersion tensor. includes both dispersion and diffusion. In the gas phase, the effective diffusion is calculated using the Millington and Quirck formulation (Millington 1959). *Q* is the source term. Here all terms are implicit, except which is explicit, coming from the resolution of the flow equation. The transport steps size are, in the present formulation, equal to the flow step size, while this could be adapted.

If *n* components are present like in reactive transport, *n* equations similar to eq.7 are written and solved separately.

For unit consistency Q is here in kg.m-3.s-1. This equation is up to now (version 19/0622) valid only for constant porosity in the domain. In case of unconfined situation is replaced by .

Automated Time stepping for transport is linked to the assigned to dCmax and dCresiduals assigned in system/controlDict file (This has not really been tested)

The source term can be associated to water flow and thus linked to the flow equation source term.

## Reactions

As already stated the reactions are solved by PhreeqcRM, which is included in the OpenFoam library. This approach implies that only the operator splitting can be used, i.e. a transport step is done for all species and then Preeqc is run and the concentrations of all species are sent back to OpenFoam. This means that **the user has to set a correct time step** for reactions to occur, there is no other option for operator splitting.

PhreeqcRM uses total component amount (i.e. the sum of all complexes of one element) as input and output. In order to solve for ionic strength and redox species, the transport also includes the amount of water, oxygen, hydrogen and charge imbalance. Therefore the total number of components is equal to the sum of all components + 4, but for instance N(5) and N(-3) account for only one component. It also means that the total number of components will not be the same as the one that can be seen in a phreeqc file. It also means that the initial concentrations of species are automatically calculated from phreeqc run.

The dissolved concentrations Cw are in mol/.

Gas phase

The gas phase transfer is calculated by phreeqc and then transferred to OpenFoam. For transport calculation, the mole number of one substance relative the total number of moles can be considered as the ratio of the partial pressure divided by the total pressure and thus the concentrations in the library has been set to a proportion:

This formulation is used as it allows the transport step to be independent from the pressure equation resolution.

The number of moles in gas in a representative volume (RV) is

Where is gas volume (L) in general per/L of medium and the molar volume. This formulation is independent of the perfect gas law as the molar volume can be obtained through the Peng Robinson equation for instance. The molar volume is calculated by Phreeqc. The moles transfer between the liquid and gas phase is also calculated by Phreeqc.

Be aware that phreeqcVm uses L as units of RV, while in opeFoam we use m3.

During reaction the partial pressure of each compound can change during the reaction and thus the total pressure can also change. At this stage we need to consider two approaches:

* For the case of diffusion in soil with no gas movement, the total pressure
* For dual phase modeling, the partial pressures resulting from phreeqc calculation are summed up to reach the total pressure and this value, transformed to Pa, is then used as the initial pressure for the next flow step calculation.

## Implementation in OpenFoam

### fvOptions

fvOptions is used to set internal conditions both for flow and transport. Three types of conditions can be set:

* A fixed value (Dirichlet type) like fixed head pressure or concentration (in dissolved or gas phase)
* A fixed rate (Neumann type): discharge or recharge Q or mass flux.
* A rate dependent of the equation variable (Cauchy type), Q = Sp.x +Su where x is the equation variable.

For rates, the fvOptions term has to be given in s-1 in order to be consistent with the equation and the internal way of dealing with surface of volume of openFoam. For a well the rate is thus given by Q/cell volume. For the surface recharge, the value (m/s) then only needs to be divided by the cell thickness. As volumes can change with cell geometry, in the GFF library the division by the cell volume is done internally, so the recharge is provided in volume/time units.

The GFF library uses a formulation of the internal conditions similar to the Modflow one, i.e. rivers, general head boundary (GHB) and drains. The three conditions are similar, with where Cond is the conductance (m2/s) and H0 is the head value fixed at the river, drain or GHB. The drain conditions is specific in the sense that the flow exists only for h>H0. The river condition is specific in the sense that the river also provides a bed value, so when h is below a given value (<H0) the exchange stops. At present (19/06/22 version) in GFF GHB and drain condition are similar to Modflow ones while river is treated as GHB.

In fvOption the corresponding formulation is and thus and . In fact, as written previously for unit consistency the Su and Sp terms are divided by the cell volume.

# Structure of the library

For the flow, according to the options the following equations can be solved:

* Darcy confined flow for column experiments for instance
* Groundwater flow which includes confined and unconfined flow
* Unsaturated flow which solves Richards equation
* Dual phase flow

According to the chosen options the solvers can use different options for the transport :

* No transport (gwaterFoam0)
* one component in water (gwaterFoam1)
* one component in several phases. Included phases are water (w), air (a), solid (s) and possibly napl (n). The napl is immobile (it is not multiphase). There is a possible exchange between each phase (gwaterFoam2)
* several components in water with reaction thanks to phreeqc (gwaterFoam3)
* several components in water and gas phase (gwaterfoam4)

The library also provides options for the effect of temperature and density:

* Tmp : temperature is transported without any effect on other properties
* Tcoupl : temperature is transported and coupled to flow : temperature modifies the density and viscosity of the fluid

# Installation

Up to now GwaterFreakFoam works on linux (windows version may come later), so you need to have a linux computer or install linux on you computer. If you have a windows computer, one of the fastest and easiest solution is to install WSL on your computer (<https://docs.microsoft.com/en-us/windows/wsl/install-win10>). Choose Ubuntu for the linux version.

You need to install OpenFoam, see <https://openfoam.org/download/>. You will install openFoam8, we use OpenFoam 7, normally it shall work also on version 8.

Then you need to install phreeqcRM, first read <https://water.usgs.gov/water-resources/software/PHREEQC/PhreeqcRM_ReadMe.txt>. All documentation on functions is at <https://water.usgs.gov/water-resources/software/PHREEQC/documentation/phreeqcrm/index.html> but you may not need it.

When PhreeqcRM is installed you need to copy libphreqcrm-….so (+ the symbolic link) to /usr/lib because the c++ compiler will search at this place the libraries.

Then write source/opt/openfoam7/etc/bashrc  in your ~/.bashrc and you’ll have access to OpenFoam environnement variables (try by typing echo $WM\_PROJECT\_DIR).

Then you’ll need to compile the gwaterFreakFoam library by unzipping it in your home folder, then go into the folder and type first the two following commands “wmake fvOptions” and “wmake libraries/phaseModels” (sorry the allwmake does not work, will see that later). For “wmake solvers”, it is a little more complex because in our solver folder there are four C files gwaterFoam0 gwaterFoam1, gwaterFoam2 and gwaterFoam3, so we will need to run wmake for each of these files. Open the solvers/Make/files file and modify the name of the .C file and the corresponding file in EXE for each 0,1,2,3,4 files and run wmake (in gwaterFoam/solvers).

In order to run one case, unzip it and go into the folder and just type “gwaterFoam0” (or 1,2,3 or 4).

If you work with WSL on windows, it is very easy to have the files in the windows folders so you can manage them with the tools you know. To see them in linux make a symbolic link “ln –s /mnt/c (this is your c drive)/”your window folder” “the same name”. Then you can cd to this folder and run your case, the files will be produced in your windows folder.

# Input files

The description below is to understand how it works, normally all these files can be produced from a orti usg file, with the opfoam.py and opfoamWriter.py files.

## Folders

For a classical OpenFoam test case, the main folder of each case contains three major folders

* **system** : it shall contains fvSchemes (the advection schemes and interpolation choices) and fvSolutions (contains the solver choice). If the domain is a simple rectangular box, the user can add a blockMeshDict (see example) for the model geometry.
* **constant** : contains all the variables that do not change during the computation. It contains eps (porosity), K (permeability in m2), g (gravity vector). The TransportProperties file contains the main parameters for flow and transport (see more details below). If options are required the *fvOptions* file is here, it is detailed below. For 3D cases a *zbot* file is required here : it contains the bottom of each cell in the domain, this is required for the unconfined calculations (for 2D zbot is considered to be 0 if not provided)
* **0** : this is the initial folder to put the initial state of each variable. It may contain at the minimum h (head) and Uw (the water velocity) and Ug when gas is present.

**Units**. In openFoam the units are set in the international system, with the convention kg, m, s for the first three unit that we use, *i.e.* [1 -1 -2 0 0 0 0] will represent *kg.m-1.s-2*. This also means the results will be given for a number of second since the start.

**Geometry**. The geometry files for openfoam are quite complex, they include faces, points, owner, neighbor and boundary. For a simple domain, it can be constructed by the blockMeshDict file (in system) and typing blockMesh in the main folder. There are a lot of options and other utilities for building mesh. When using Orti3d the geometry files are produced by the file openfoam and openfoamWriter from a rectangular geometry or a USG model.

For radial model, it is possible to build it from blockmeshDict (see radialCO2gas example)

## System

### fvSchemes

In openFoam each part of an equation has a name and thus a scheme can be specified for each part of any equation. This is valid for the temporal scheme, the divergence and laplacian terms. The temporal scheme allows for instance to choose between Euler or CrankNicolson temporal schemes. For instance below a vanLeer scheme is declared for dissolved concentration. The file also allows to set the way the interpolation of spatial variable is done at the cell faces (below upwind interpolation for kr).

ddtSchemes

{

default Euler;

ddt(sw) CrankNicolson 0.5;

}

gradSchemes

{

default Gauss linear;

}

divSchemes

{

default none;

div(phiw,Cw) Gauss vanLeer;

}

laplacianSchemes

{

default Gauss linear corrected;

}

interpolationSchemes

{

default linear;

krg upwind phig;

krw upwind phiw;

}

There are no default values, so in general set default to none.

### fvSolutions

this file allows to select the matrix solver for each equation, for instance

h{solver PBiCGStab;preconditioner DIC;tolerance 1e-12;relTol 0;}

where we can see the choice of the solver, preconditioner tolerance…

If picard iterations are used the parameters are set here.

### ControlDict

The controlDict allows to specify the time steps, writing steps, start and end time and the way to write the output files. The time steps constraints can be specified as a simple max time steps or is also used to set the constraints in terms of courant number (or residual concentrations variation between two time steps).

Interestingly the controlDict parameters can be changed during the simulation (usefull for a restart or increase time steps).

Using variable size time steps is tricky, it requires several files

### blockMeshDict

this file is used to build the geometry for simple geometries, see the openfoam documentation or examples to see how it works. blockMeshDict is used by the program blockMesh to produce the files that are in constant/polyMesh (see below).

## 0 folder files

Here are contained the initial values and boundary conditions of the field considered in the problem. The boundary values have to be set at boundaries provided in blockMeshDict or in boundary (in the constant/PolyMesh folder). For flow the files can be h (m), hp (m) or p (Pa). For transport of one component the Cw file must be provided (concentrations in kg/m3). When phreeqc is used the initial files are not provided, they are calculated from the initChemp.pqi file.

## Constant folder Files

***Fixed fields***

The hydraulic conductivity fields (Kh and Kv) are here, they are scalarField, they will be transformed to a K tensor (as explained in theory part). The porosity field, a scalar, is given in the eps file. The gravity vector is in the g file, with an orientation along the z axis in GFF library. Two other fields are required for calculations (and because all cells are vertically aligned), this is thk which is the thickness of each cell (at the center of the cell) and zbot, the bottom of the cell all in meters. These fields are used in particular to calculate the wet thickness in case of an unconfined aquifer.

***Constant/Transportproperties***

This file contains the main properties of the model, like :

* phase.w{rho rho [1 -3 0 0 0 0 0] 1e3;mu mu [1 -1 -1 0 0 0 0] 1e-3;} this is the density and viscosity of the water. Similarly phase.g is required in dual phase model.
* flowStartSteady : if the user wants the flow start at steady state (usefull for transport)
* activateUnconfined 1; the flow is unconfined
* activateUnsat 1; for unsaturated flow
* activate2phase 1; to activate two phase flow
* activateCapillarity 1; to activate capillarity in dual phase (default)
* activateReaction 1; to activate reaction when phreeqc is used (default)
* reactionSteps 25; the number of flow and transport step between each phreeqc run
* sw\_min 0.2; needed for unsaturated or dualphase flow the minimum saturation
* alpha\_vg alpha\_vg [0 -1 0 0 0 0 0] 0.0182; the value of the α parameter (Van Genuchten equations) for unsaturated medium in m-1.
* n\_vg 2.5; the value of the n parameter (Van Genuchten equations) for unsaturated medium
* alphaL alphaL [0 1 0 0 0 0 0] 1.0; longitudinal dispersivity for species transport
* alphaT alphaT [0 1 0 0 0 0 0] 0.1; transverse dispersivity for species transport
* nlay 1; ncell\_lay 370; the number of layers and number of cells/layers for chemistry calculations
* phreeqcVm 6.84e-2; molar volume (L/mol) of the gas for the simulation (called phreeqcVm as it shall be consistent with phreeqc)

## Constant options files

**Constant/fvOptions file**

An example part of the file is

hWell

{

type scalarmySemiImplicitSource;

active true;

selectionMode cellSet;

cellSet hwell;

volumeMode absolute;

injectionRateSuSp {sw (1 0); h (1 0); }

}

*Type*: is the program that reads the data and transform the openfoam equations (for groundwaterFrekFoam it can be scalarmySemiImplicitSource or scalarmyFixedValueConstraint)

*selectionMode*: can be cellset or all or point (we use only cellSet)

*cellSet*: this is the list of cells where the process is effective, it is composed of one letter for the general equation (*h* and *p* for flow and *c* for transport of dissolved components and *g* for gaseous) and then then three letter for the type of condition (wel for wells, rch for recharge, ghb for general head, drn for drains)

*volumeMode*: not used

*injectionRateSuSp* : a keyword typical of the used program in fvOptions, injectionRateSuSp is used for scalarmySemiImplicitSource. injectionRateSuSp includes two variables: the first one is used as a complementary variable to provide information, while the second one is the variable on which the equation applies. For instance here *h* is the variable so the modification will apply to the hEqn (this is the equation where the variable *h* is). fieldValues applies for scalarmyFixedValueConstraint with only the variable that is fixed. For all variables there are two numbers: the first one is the multiplier for the Su term which is the addition term, while the second is for Sp which is the multiplier term (i.e. it will apply as a multiplier of the current variable in the equation). In GFF the parameters Su and Sp are fixed (0 or 1) because the values at this place cannot vary in space and time, while it is necessary. The files in options will enclose these varying values.

**Constant/polymesh/sets files**

This folder contains several cellSets that are associated with one process, as described above. Here hwell is the cellSet that contains the list of all cells that contain wells.

**Constant/options files**

The values of the parameters necessary for the internal conditions for each cell and time are contained in options files that have the same name as the cellSet. For flow the names are hwel, hrch, hriv, hdrn and hghb for the five types of conditions (Wells, recharge, rivers, drain and general head respectively). For transport the names are the same but starting with c. The files are ordered as follows

In the first line there is the number of layers and the number of cells per layer, then for the next lines the format is “Time (d) cell\_nb Su Sp “ # one line for each cell in each time.

***Time units here are in days***. This is done because seconds are a little bit difficult to handle. All data are transformed to seconds internally.

Example:

5 453

0 0 0 0

1 43 -3e-4 0 // this is for one day and first cell of the cellSet (cell nb 43)

1 57 -4e-4 0 // this is for one day and 2nd cell of the cellSet (cell nb 57)

….

17820 -2e-4 0 // this is for time=two days and first cell of the cellSet

…

Last time+0.01 0 0 0

Where Su (here) is the injection rate in m3/d and Sp is the term that allows to vary the discharge/recharge in proportion to the head value (Cauchy type of condition). For recharge the rate is the daily recharge thickness multiplied by the cell area

## Chemistry specific file

Ractiv: this file contains a list of the cells number where the chemistry will be calculated

The main folder includes the files required for phreeqc:

phreeqc.dat the phreeqc database

initChem.pqi : this is the phreeqc file of your problem

phqfoam.txt : it structured as follows

1st line

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nb of cells (or only the active ones, if a sub domain for chemistry is defined) | Nb of dissolved components | Nb of gas components | Nb of different solutions | Units for phase exchange.. |

Then they are 7 lines that contain the number for:

|  |  |  |
| --- | --- | --- |
| Group (not written) | 1st number | List of numbers |
| Solution | 0 or -1 | One number or a list of ncells values |
| Equilibrium phases | “ | “ |
| Exchange | “ | “ |
| Surface | “ | “ |
| Gas phase | “ | “ |
| Solid solutions | “ | “ |
| Kinetics | “ | “ |

For each line, whether the first number in the line is 0 and then all cells have the same solution number (use -1 if the group is not used), or the first number is -1 and it must be followed by the solution number for each cell.

Phqinit.txt: very similar to phqfoam except that it contains a cell number=number of solutions: it is used to initialize the solutions (and potentially the gases)

If a solution is injected, it needs to be done through fvoptions file, the sets and the file cwel (or cfix..) in constant/option folder, as presented above, except that instead of the rate value for wells, just specify the solution number. This will specify the fvOptions for all Cwi and Cgi equations.

Solutions: this file is written automatically by gwaterFoam3/4: it contains the chemistry of the solutions present in the problem. (needs to be written by gwaterFoam because we need 2O, H, O and e- amounts)

Gases: same as above but for gas components concentrations

# For developers

There is a curious but useful use of .H files in a lot of OpenFoam libraries. I fact the .H contain most f the calculations while normally these are in the .C. This allows to simply include them when a calculation is required. We mostly use this approach, when a .C file to link to other parts of openFoam is required, it is in a specific folder (like fvOptions, phases..)

## Fields

There are scalar, vector and tensor fields, they can be defined by cells : e.g. volScalaraField but also on the faces surfaceScalarField. During the definition the field can be defined as fieldName.timeConstant, they will remain constant and be in the constant folder, or fieldName.timeName () they can change over time and be stored in the time folders. During the creation of a field one can set its dimension an its initial value. If no initial value is specified openfoam will ask for the file.

In the library, K is a tensor (to account for anisotropy), so the mobility is also a tensor. Then in order to have for instance the mobility multiplied by gravity, the operation is “Mf & g” which is a tensor and not “Mf \* g” which would be valid if Mf would be a simple scalar.

## Phreeqc coupling

The phreeqc coupling is done through the initPhreeqc.H file but it is a little bit tricky, as the initChem.pqi file needs to be read in order to set the number of components in the simulation.

The operations are done in the following order:

* gwaterFoam3/4 reads the files constant/options/ractiv (active cells), the phqfoam.txt file (distribution of initial solutions exchange…) that is internally called ph\_data. The first line of ph\_data is used to specify the nxyz, number of cells in the model and n\_comp, the total number of components
* then ph\_data are sent to freak class (setData), the database if set (setDB) using the address of phreeqc.dat file and the pqi file (setChemistry) using the initChem.pqi file.
* Freak.init() is run once for a number of cells=number of solutions to calculate the composition of the initial solutions
* gwaterFoam3 uses the initial solutions to write constant/options/solutions file that contains the components of each solution (written sequentially by solution number), this will be used by fvOptions. In gwaterFoam4 an additional file gases is written for initial gaseous components.
* freak.init() is run again but for the whole domain cells. Phreeqc stores the solid phases components (minerals, exchangers, surfaces)
* createFvOptions.H is run that reads all option data (see fvOptions below)
* createCwiFields.H is run to build the n\_comp variables Cwi in OpenFoam (the Cwi variables have the size of the total domain, not ractiv length)
* run the flow equation
* run the transport of n\_comp species (CwiEqn)
* finds the place where chemistry has changed (any species relative changing of more 5e-5) stored in rchange array
* transfer the Cwi concentrations to a one dimensional array c\_ph (n components one after the other with ractiv size for each component) and provides if to freak class through the setC function
* specify where to calculate through freak.setCalc which uses setSaturation: 1 for calculating the chemistry and 0 for not calculating (it seems that there is no other way to inactivate calculation in phreeqcRM)
* set time step and runs phreeqc
* reset all Cwi concentrations to 0, and then get the freak.C concentrations and copy them to Cwi. The reset to 0 is required because transport will calculate concentrations around the ractiv zone and we don’t want to lose time with these concentrations

PhreeqcRM does not want null conc at the beginning, put 1e-12

CreateMapping function : it creates a correspondance between a cell in the model and the ones that phreeqc will run; it should provide for all active cell a vector, this one has to start from 0 and have an increment of 1 for each new cell to be run.

## fvOptions

we use two types of fvOptions : semiImplicitSource and fixedValue. In order to have data varying for these source a specific file is added (in constant/options) which has the same name as the cellSet, internally it is called cellsData. This file is read in the file in cellsetOptions.C. Then the values given in the cellsData are multiplied by the values provided in fvOptions (1 but formatted as tensor or other according to the equation) in the file semiImplicitSource.C or fixedValuconstraint.C.

fvOptions for the semiImplicitSource is added to the equation “== fvOptions() “ while for the fixedValue it is included by adding “fv.Options.constrain(eqn)” before solving the eqn.

(OpenFoam seems to reorder the cells so it is necessary to provide the cell number in order to set the cell. The index of cellSetData is calculated in cellSetOptions.C and used in semiImplicit and fixed. Not sure this comment is still valid)

!! it seems that fvOptions of type fixedValueConstraint does not accept two input (while it is compiled correctly it is never used)

Vdash has been removed from the formula => absolute = specific

In order to have access to *sw* field for the recharge, we use the fvOtions with two variables (sw and h).

!! It seems that with Su<0 and Sp>0, for stabilization for the first time step the equation shall be written laplacian … = fvOptions, while for time running this is “ddt + laplacian = -fvOptions “ (minus sign before fvOptions) if not the time steps are very small whether on one side or the other. This is due to the fact that when Sp is positive fvOptions is treated as implicit but if Sp is negative fvOptions is treated as explicit (<https://www.cfd-online.com/Forums/openfoam-programming-development/182107-fvmatrix-fvoptions-susp-automatic-implicit-explicit-source-term-treatment.html>, there is a solution there, but it did not provide good results)

For semiImplicitSource drains the Q=Su+Sp\*h formulation is used and should normally limit the results to the values of Q<0 (draining). In order to limit the oscillations close to the equilibrium we use a simple condition: if Q>0 => Su=Su/2 and Sp= Sp/2. (The same formulation should be done for river<bottom)

Flow

fvOptions is fixedValue for fixed head, semiImplicit for wells, recharge, drains, GHB and rivers. Wells and recharge use only the Su part= the injected discharge, while drains, ghb an rivers use both Su and Sp, this is the discharge is linked to h value in the cell. The discharge in the cellsData file is the total rate for the cell (for wells it is the true pumping or injecting rate, while for recharge it is the recharge height multiplied by the cell area). This value is divided by the cell volume in the semiImplicitSource.C file. The time units are in days in cellsData files and transformed to seconds in the semiImplicitSource.C file.

Transport

For concentrations injected in a well, the cellsData file contains the concentration and the cellOptions.C file also retrieve the hwell data (that contain the well flow rate) to calculate q\*conc the mass injection rate. For the pumping wells, no concentration is specified but the local concentration is used to calculate the mass retrieval rate. It is necessary in order to remove the mass pumped by the well.

For recharge a similar approach is used: the recharge concentration is given by crch cellData file and the hrch provide the flow rate injected.

For drains, ghb and rivers, the concentrations in these cells are provided in cdrn, cgb, and criv cellsData files. Even if normally concentrations in drains or ghb cells with heads below surrounding cells are not necessary as water is discharged towards these cells, they must be provided as 0 values, if not, sometimes it leads to instability in calculations leading to very high concentrations.

# Annex

Phreeqc

If you ever happen to want to link against installed libraries in a given directory, LIBDIR, you must either use libtool, and specify the full pathname of the library, or use the '-LLIBDIR' flag during linking and do at least one of the following:

- add LIBDIR to the 'LD\_LIBRARY\_PATH' environment variable during execution

- add LIBDIR to the 'LD\_RUN\_PATH' environment variable during linking

- use the '-Wl,-rpath -Wl,LIBDIR' linker flag

- have your system administrator add LIBDIR to '/etc/ld.so.conf'

For fluxes calculation

phiw = hEqn.flux() provides correct fluxes (null balance in a cell),

while - ((Mf & fvc::interpolate(fvc::grad(h)) ) & mesh.Sf() ) does not (why, I don’t know)

Interpolation schemes possibles

CoBlended

Gamma

MUSCL

Minmod

OSPRE

QUICK

SFCD

SuperBee

UMIST

biLinearFit

blended

cellCoBlended

clippedLinear

cubic

cubicUpwindFit

downwind

filteredLinear

filteredLinear2

filteredLinear3

fixedBlended

limitWith

limitedCubic

limitedLinear

limiterBlended

linear

linearFit

linearPureUpwindFit

linearUpwind

localBlended

localMax

localMin

midPoint

outletStabilised

pointLinear

quadraticFit

quadraticLinearFit

quadraticLinearUpwindFit

quadraticUpwindFit

reverseLinear

skewCorrected

upwind

vanAlbada

vanLeer

weighted

)

PhreeqcRM surface

PHREEQC version 2 and 3 have the same units convention for SURFACE

The units in a SURFACE definition result in a number of moles of surface sites. Moles are either defined directly (default) as a product of sites per nanometer^2 times surface area (with an internal factor to convert sites to moles and nanometers to meter^2), or as a proportion of a mineral or kinetic reactant. Regardless of the different input options, the result is moles of sites, not moles per liter water or moles per representative volume, just moles.

The following is excerpted from the documentation for the PhreeqcRM method SetUnitsSurface:  
  
**In PHREEQC input, surfaces are defined by moles of surface sites (Mp). SetUnitsSurface specifies how the number of moles of surface sites in a reaction cell (Mc) is calculated from the input value (Mp).  
Options are 0, Mp is mol/L of RV (default), Mc = Mp\*RV, where RV is the representative volume (SetRepresentativeVolume); 1, Mp is mol/L of water in the RV, Mc = Mp\*P\*RV, where P is porosity (SetPorosity); or 2, Mp is mol/L of rock in the RV, Mc = Mp\*(1-P)\*RV.**

The SetUnitsSurface option is used when the PhreeqcRM method InitialPhreeqc2Module or InitialPhreeqcCell2Module is invoked. The moles in a PhreeqcRM cell are calculated from the PHREEQC SURFACE definition using the specified units option.

As Pht3d is in mol/L bulk volume, the only solution is to set option to 1 in phreqcRM (initChem.pqi) and transform the amounts from Pht3d to mol/L (considering porosity)