porousMultiphaseFoam v2107: An open-source tool for modeling saturated/unsaturated water flows and solute transfers at watershed scale

Pierre Horgue, F. Renard, G.S. Gerlero, Romain Guibert, Gerald Debenest

porousMultiphaseFoam v2107: An open-source tool for modeling saturated/unsaturated water flows and solute transfers at watershed scale

P. Horgue[a,*], F. Renard[b], G. S. Gerlero[c], R. Guibert[a], G. Debenest[a]

[a] *Institut de Mécanique des Fluides de Toulouse (IMFT) - Université de Toulouse, CNRS-INPT-UPS, Toulouse FRANCE*
[b] *CEA, DAM, DIF, F-91297 Arpajon*
[c] *Centro de Investigación en Métodos Computacionales (CIMEC, UNL–CONICET), S3000GLN Santa Fe, Argentina*

Abstract

The first releases of porousMultiphaseFoam proposed an open-source software suite to solve the equations for multiphase flow (generalized Darcy's law) in porous media or groundwater flows (Richards' equation) by taking advantage of OpenFOAM, a finite volume platform with automatic discretization on three-dimensional unstructured grids and good parallel efficiency. Recently, the porousMultiphaseFoam toolbox has been confronted with complex cases of fast water flows and solute transfers in realistic hydrological configurations with variable forcing conditions (heterogeneous infiltration and local tracer injection). Several developments have been carried out to make it possible to simulate those cases, which extend the toolbox with: (i) a set of solvers dedicated to groundwater flows, including coupled water flow and solute transport and simplified 2D approaches, (ii) improved numerical techniques for problems with strong non-linearities, (iii) libraries/executables for pre-processing of input data (geographical information and time-variable forcing terms) and (iv) passive or coupled scalar transport (tracer) with groundwater solvers that support any number of species. New solvers are validated on several (un-)saturated configurations by a direct comparison with a well validated finite element code.

Keywords: Porous media, Unsaturated flow, Richards' equation, Newton's algorithm, OpenFOAM

1. Introduction

The modeling of saturated/unsaturated flow in soils is of importance in a wide range of scientific fields such as environmental monitoring, waste management or hydrogeology. Multiphase flow through porous media can be modeled by solving mass conservation for each phase expressing velocities using the Generalized Darcy's law [21]. However, a classical approach for the air-water flow modeling (for hydrology for example) consists in neglecting the air pressure gradients to reduce the problem to the resolution of the water mass conservation, via the so-called Richardson [25] or Richards' equation [24].

Despite its apparent simplicity, the Richards' equation remains challenging to be accurately solved due to its highly non-linear terms which require iterative procedures such as Picard's or Newton's methods. Moreover, the modeling of hydro-geological events may involve a wide range of time and space scales, inducing strong requirements on space grids or time steps. To reduce the computational cost, it is then necessary to use advanced numerical techniques for time step management and/or adaptive mesh refinement. Main difficulties and recent works related to the numerical solution of the Richards' equation, as well as an inventory of the techniques that have been developed, can be found in two recent reviews [9] and [31]. Over the last two decades, several open-source tools dedicated to the 3D modeling of Richards' equation have been developed, such as ParFlow [1], PFLOTRAN [15], Dumux [10] and RichardsFOAM [22], with special attention brought to the parallel efficiency of the different solvers since the three-dimensional modeling of groundwater at the

*Corresponding author
Email address: phorgue@imft.fr (P. Horgue)

watershed scale can be particularly expensive. The open-source tools listed previously exhibit parallel linear behavior up to at least one thousand processors, such as the porousMultiphaseFoam (PMF) used in this work [16, 17].

The groundwater modeling is often simplified by neglecting the unsaturated upper part and using the Dupuit–Forchheimer assumption to consider that the vertical component of the velocity is zero [2]. This groundwater free surface approximation allows then to simulate groundwater flow as a two-dimensional flow by solving the water height without capillary effects. The strong assumptions required by the 2D approach are counterbalanced by the drastic reduction of the computational costs which is particularly effective under specific conditions. A more accurate simplified approach consists in coupling 1D (for the unsaturated part) and 2D (for the saturated part) models to simulate unsaturated and saturated groundwater flow at the watershed scale while keeping low computational costs [28, 19, 23].

Simultaneously to the fluid flow, the solute transport through porous media has also been widely studied with some specific effects such as the dispersion [14, 29] (velocity fluctuations at the micro-scale which increase the spreading of the tracer) or the retardation factor [7] (linked to the sorption coefficients). Where the precise characterization of the parameters remains difficult to establish [12], some usual formulations are commonly accepted in the community and generally implemented in the available numerical tools.

A general open-source toolbox based on OpenFOAM was initially developed for generic two-phase flow [16], including the commonly used soil water retention models (for relative permeability and capillary pressure). The PMF toolbox was first extended to Richards' equation [17], using existing libraries and functions, and providing an efficient tool dedicated to air-water flows. The toolbox has also served as the basis for other codes such as the hybridPorousInterFoam [5] or for two-phase flow with adaptative mesh refinement [27]. However, it has become apparent that multiple features were missing in order to offer a complete modeling suite such as:

(i) A fast two-dimensional free-surface modeling tool,
(ii) Passive/coupled tracer solvers for 2D-saturated/unsaturated flows,
(iii) Pre-processing tools for realistic configuration with Geographical Information System (GIS),
(iv) Improved numerical techniques to allow the modeling of unsaturated complex cases.

The work presented in this paper intends to validate the multiple developments which have been made during the last years. Due to the good parallel efficiency observed for OpenFOAM executables, one will be able to simulate flows and transfers in three-dimensions at the watershed scale.

2. Mathematical model and governing equations

2.1. Mathematical model for saturated flow (2D modeling)

The groundwater flow modeling of the saturated zone can be simplified to a two-dimensional problem by considering a constant head-over-depth assumption (called Dupuit-Forchheimer's assumption). In this configuration, the momentum equation reduces to the Darcy's law and the continuity equation reads :

$$\varepsilon \frac{\partial \psi}{\partial t} - \nabla \cdot \left(H_{water} \frac{K \rho \|\boldsymbol{g}\|_2}{\mu} \nabla \psi \right) = V_{source}, \tag{1}$$

where $\psi\,[m]$ is the groundwater head, $\varepsilon\,[-]$ the porosity, $K\,[m^2]$ the permeability, $\rho\,[kg.m^{-3}]$ the density of the fluid, $\mu\,[Pa.s]$ the dynamic viscosity of the fluid and $\boldsymbol{g}\,[m.s^{-2}]$ the gravity. The water depth $H_{water}\,[m]$ is defined as:

$$H_{water} = \psi - z_0, \tag{2}$$

where $z_0\,[m]$ is the height of the impermeable bottom of the aquifer. The flow velocity $V_{source}\,[m.s^{-1}]$ is used for rainwater infiltrations or other inflow sources.

The boundary conditions available for 2D-saturated flows are:

– Dirichlet condition to impose potential : $\psi_{boundary} = \psi_{fixed}$,

– Neumann condition on the potential gradient to impose flow velocity u_{fixed}. Fixed gradient value is computed as :

$$\nabla\psi_{boundary} \cdot \overrightarrow{n} = \frac{\mu}{H_{water} K \rho \|\boldsymbol{g}\|_2} u_{fixed} \cdot \overrightarrow{n} \tag{3}$$

Seepage condition

During medium and high intensity infiltrations, it happens that the local capacity of the aquifer is reached and that any additional infiltration results in a surface runoff which does not further affect the groundwater flow. This physical feature needs to be handled to correctly model the behavior of a watershed over a long period. Each computation cell i gets a maximal value $z_{max,i}$ relative to the local thickness of the aquifer. This additional geometric information is obtained using bilinear interpolation from Digital Elevation Model (DEM) files and represents the local maximum value for the potential ψ_i. At any linear iteration k, the potential ψ_i^k and the flow balance ϕ_i^k for each cell i can be used to determine if the cell is overflowing and therefore if a fixed potential value should be applied:

$$\text{if } \psi_i^k \quad \geqslant \quad z_{max,i} \text{ and } \phi_i^k \geqslant 0 \text{ then } \psi_i^{k+1} = z_{max,i} \tag{4}$$

The dynamic forcing is applied after each time-step (transient simulation) or solver iteration (steady simulation).

2.2. Mathematical model for unsaturated flow (vadose zone)

Modeling variably saturated flow in the vadose zone generally requires solving Richards' equation, a physics-based equation where air pressure gradient has been neglected to keep only the liquid phase conservation equation. Several forms of the Richards' equation can be found in the literature such as the pressure-head form, the saturation-based form and the mixed form proposed by Celia et al. [6], the latter of which has been implemented in the software. This modified version consists of using the water content θ for the accumulation term and pressure head h for the other terms, which reads:

$$\frac{\partial\theta}{\partial t} + \frac{\theta S_s}{\varepsilon}\frac{\partial h}{\partial t} - \nabla \cdot \left(\frac{K\rho\|\boldsymbol{g}\|_2}{\mu} k_{r,h} \nabla h \right) + \nabla \cdot \left(\frac{K\rho}{\mu} k_{r,h} \boldsymbol{g} \right) = Q_{source}, \tag{5}$$

where $\frac{\theta S_s}{\varepsilon}\frac{\partial h}{\partial t}$ is the aquifer specific storage term ($S_s\,[-]$ is the specific storage coefficient) and $k_r(h)\,[-]$ is the relative permeability and $Q_{source}\,[m^3.s^{-1}]$ is a volumic source term. The mixed-formulation has the advantages of being mass conservative (like the $\theta-$based form) as well as being able to handle locally saturated areas (as the $h-$based form).

This two dependent variables (θ, h) equation cannot be directly linearized, and water content θ has to be expressed as a function of head pressure h by considering the capillary capacity C_h defined as:

$$C_h = \frac{\partial\theta}{\partial h} \tag{6}$$

The capillary capacity $C_h\,[m^{-1}]$ and the relative permeability $k_{r,h}\,[-]$ can be computed using usual relationships such as Brooks and Corey [4], Van Genuchten [30] or Ippisch et al. [18].

Boundary conditions available for unsaturated flows are :

– Dirichlet condition to impose pressure head value: $h_{boundary} = h_{fixed}$

– Neumann condition on the pressure-head gradient to impose flow velocity u_{fixed}. Fixed value is computed as :

$$\nabla h_{boundary} \cdot \overrightarrow{n} = \frac{\mu}{K\rho\|\boldsymbol{g}\|_2 k_{r,h}} u_{fixed} \cdot \overrightarrow{n} \tag{7}$$

Note that the Richards' equation (5) must be solved using an iterative procedure to handle the (possibly strong) non-linearities of the relative permeability $k_{r,h}$ and $\theta - h$ relationships (Picard or Newton's algorithm for example, see Section 3.2).

2.3. Mathematical model for tracer transport

The general conservation equation for the transport of a tracer with concentration $C\,[-]$ reads [3, 2] :

$$R\frac{\partial \Theta C}{\partial t} + \nabla \cdot U_w C - \Theta \nabla \cdot (D_{eff} \nabla C) = -Q_C - \Theta R \lambda C, \tag{8}$$

where $\lambda\,[s^{-1}]$ is the decay coefficient, U_w the water flux equals to:

$$U_w \quad = \quad \begin{cases} U\,[m.s^{-1}] & \text{for unsaturated solvers} \\ U \times H_{water}\,[m^2.s^{-1}] & \text{for 2D-saturated solvers} \end{cases}, \tag{9}$$

Θ equals to:

$$\Theta \quad = \quad \begin{cases} \theta\,[-] & \text{for unsaturated solvers} \\ \varepsilon\, H_{water}\,[m] & \text{for 2D-saturated solvers} \end{cases}, \tag{10}$$

and $R[-]$ the retardation factor :

$$R = \begin{cases} 1 + \frac{(1-\varepsilon_{total})\rho_s K_d}{\theta}\,[-] & \text{for unsaturated solvers} \\ 1 + \frac{(1-\varepsilon_{total})\rho_s K_d}{\varepsilon}\,[-] & \text{for 2D-saturated solvers} \end{cases}, \tag{11}$$

where $\varepsilon_{total}\,[-]$ is the total porosity (larger than kinematic porosity ε), $\rho_s\,[kg.m^{-3}]$ is the solid matrix density and $K_d\,[m^3.kg^{-1}]$ the volumetric partitioning coefficient. The effective dispersion coefficient $D_{eff}\,[m^2.s^{-1}]$ is computed using local values of velocity $\boldsymbol{U}\,[m.s^{-1}]$ field as [11, 2]:

$$D_{eff} = \frac{D_m}{\tau} + \frac{\underline{\underline{\alpha}}|\boldsymbol{U}|}{\varepsilon} \tag{12}$$

where $D_m\,[m^2.s^{-1}]$ is the molecular diffusion coefficient, $\tau\,[-]$ the tortuosity and $\underline{\underline{\alpha}}\,[m]$ the dispersion tensor constructed using velocity direction and $\alpha_L/\alpha_T\,[m]$ longitudinal and transversal dispersion coefficients. An equation taking the form of (8) has to be solved for each tracer species transported by the fluid. Q_c stands for source term whose dimension depends on the solver considered: $[s^{-1}]$ for unsaturated and $[m.s^{-1}]$ for 2D-saturated solvers.

3. Numerical methods

3.1. Equation discretization

The toolbox has been developed using the OpenFOAM environment which is a development framework for solving equations using the finite volume method. The OpenFOAM native functions allow one to automatically discretize classical operators ($\frac{\partial}{\partial t}$, ∇, Δ) on unstructured three-dimensional meshes using a wide range of numerical schemes. In the validation cases studied in this work, a classical second order centered scheme is used for spatial discretization of convective and diffusive terms. A second order backward scheme is used for time discretization of tracer transport and saturated flow equations. The time discretization of Richards' equation (unsaturated flow) is carried out using a backward Euler scheme to simplify the implementation of non-linear Picard and Newton's methods (cf Section 3.2).

3.2. Non-linear methods for Richards' equation

To linearize the Richards' equation (5), it is necessary to express the implicit water content θ at iteration $k+1$ as a function of pressure head h using capillary capacity C_h (cf eq. 6) as:

$$\theta^{k+1} = \theta^k + C_h^k \left(h^{k+1} - h^k\right), \tag{13}$$

where θ^k and h^k are the current iteration values (C_h^k computed using h^k). The derivative $\frac{\partial\theta}{\partial t}$ can then be discretized using a backward Euler scheme:

$$\frac{\partial\theta}{\partial t} \approx \frac{\theta^{k+1} - \overline{\theta}}{\Delta t} = \frac{1}{\Delta t}\left(C_h^k\left(h^{k+1} - h^k\right) + \theta^k - \overline{\theta}\right) \tag{14}$$

where $\overline{\theta}$ is the water content at the previous time step. This makes possible to express the linearized form of conservative equation 5 as a function of solved variable h^{k+1} and current iteration values h^k and θ^k:

$$\frac{1}{\Delta t}\left(C_h^k\left(h^{k+1} - h^k\right) + \theta^k - \overline{\theta}\right) - \nabla\cdot\frac{K\rho\|\boldsymbol{g}\|_2}{\mu}k_{r,h^k}\nabla h^{k+1} + \nabla\cdot\frac{K\rho}{\mu}k_{r,h^k}\boldsymbol{g} = Q_{source} \tag{15}$$

This linearized mixed form of the Richards' equation (15) can be rewritten as:

$$\mathbf{C}^k\frac{h^{k+1} - \overline{h}}{\Delta t} - \mathbf{M}^k h^{k+1} + \mathbf{L}^k = S^k \tag{16}$$

with the diagonal matrix $\mathbf{C}^k$ containing the accumulative terms, $\overline{h}$ the old-time pressure head vector, $\mathbf{M}^k$ head-pressure gradient–related terms, $\mathbf{L}^k$ gravity-related terms and S^k the source-term vector. Picard's algorithm consists of solving the equation 16 iteratively while updating the matrix coefficients using the pressure-head vector h^k at iteration k. Letting R^k be the equation residual at iteration k:

$$R^k = \mathbf{C}^k\frac{h^k - \overline{h}}{\Delta t} - \mathbf{M}^k h^k - \mathbf{L}^k - S^k \tag{17}$$

This method converges linearly to the solution when:

$$|R^k| < r_{tol} \tag{18}$$

with r_{tol} the user-defined residual tolerance. This method was the one initially implemented in PMF, yet it may converge slowly for highly non-linear configurations and/or low tolerance values. It is particularly unsuitable for the modeling of high flow rates in the vadose zone (see test case in section 5.1.4) where the time-steps required for the convergence of this algorithm imply a prohibitive computational cost. However, the method is robust and has a wide range of convergence.

To improve the convergence rate and reduce the computational cost, one can use Newton's method, which consists of solving the equation:

$$\mathbf{J}^k\left(h^{k+1} - h^k\right) = -R^k \tag{19}$$

where $\mathbf{J}^k$ is the Jacobian matrix of the residual function R which is equal (row i, column j):

$$J_{ij} = \frac{\partial R_i}{\partial h_j} \tag{20}$$

This method converges to the solution quadratically and thus allows solving highly non-linear problems, as shown in section 5.1.4.

Both methods are used sequentially in unsaturated solvers (groundwaterFoam and groundwaterTransportFoam), by solving the equation first using Picard's method until $|R^k| < r_{tol,Picard}$ and then refining the solution with Newton's method until $|R^k| < r_{tol,Newton}$. This approach used by several numerical codes, including METIS, takes the advantage of the wide convergence range of Picard's algorithm to approach the solution roughly and then converges quadratically with Newton's method.

3.3. Time step management

An efficient time step management is required for modeling transient saturated/unsaturated water flow because the characteristic time scales can vary by several orders of magnitude. A short-time and high amplitude water injection rate into an unsaturated column for example would require small time-steps at the beginning ($\Delta t \approx 10$ seconds for several hours of physical time) while the following capillary drainage should be simulated over a long period of several months ($\Delta t \approx 1$ day or more). Because of the non-linearity of Richards' equation, a general CFL condition cannot be directly derived. Two methods for time step management are generally used:

(i) the *heuristic* one is based on the number of linear iterations (Picard or Newton) required at each time step,

(ii) the *adaptive* one is based on the time truncation error estimates (related to the numerical scheme used for time discretization).

D'Haese et al. [8] have proposed a comparison study of these two methods and their variants by analyzing the advantages and limitations of both approaches which are case-dependent. In this work, time stepping is based on a rigorous mathematical analysis performed on the numerical scheme used for time discretization which can be applied to all implemented solvers (2D-saturated, unsaturated, passive transport or coupled). The time step is computed to maintain a relative time truncation error (estimated using Taylor series) below a user-defined value ϵ_{error}. The time-step computations for the available time schemes in the presented code are :

1. backward Euler: $\Delta t = \sqrt{\frac{2\epsilon_{error} x_{max}}{\left(\partial^2 x/\partial t^2\right)_{max}}}$
2. 2$^{\text{nd}}$ order backward scheme:$\Delta t = \sqrt[3]{\frac{4\epsilon_{error} x_{max}}{\left(\partial^3 x/\partial t^3\right)_{max}}}$
3. Crank-Nicolson: $\Delta t = \sqrt[3]{\frac{12\epsilon_{error} x_{max}}{\left(\partial^3 x/\partial t^3\right)_{max}}}$

where x_{max} is the value of the variable at the cell where the maximal derivative has been computed. A third order time scheme had to be implemented in the toolbox to allow adaptive time step for available time schemes.

4. Accessory numerical tools

4.1. Toolbox file structure

The file structure of the toolbox is depicted in Figure 1. One can see old libraries or solvers developed for PMF in 2015 and the features added in the new version.

4.2. `eventFile` *structure*

To help the set up of realistic cases with time-varying conditions, a generic OpenFOAM class `eventFile` has been developed with three major concrete subclasses available :

- `eventFileInfiltration` for homogeneous or heterogeneous infiltration (2D-saturated solvers),
- `patchEventFile` for solute/water injection on boundaries (unsaturated solvers),
- `sourceEventFile` for local water or solute injection (all solvers).

For each time iteration, forcing terms are automatically computed from linear interpolation between the two closest dates. An example of `sourceEventFile` is given in Figure 2.

An additional concrete class `outputEventFile` allows specifying output writing times while avoiding any constraints on the time step, with the solutions being constructed by time interpolation. Extension .evt has been chosen to identify the files.

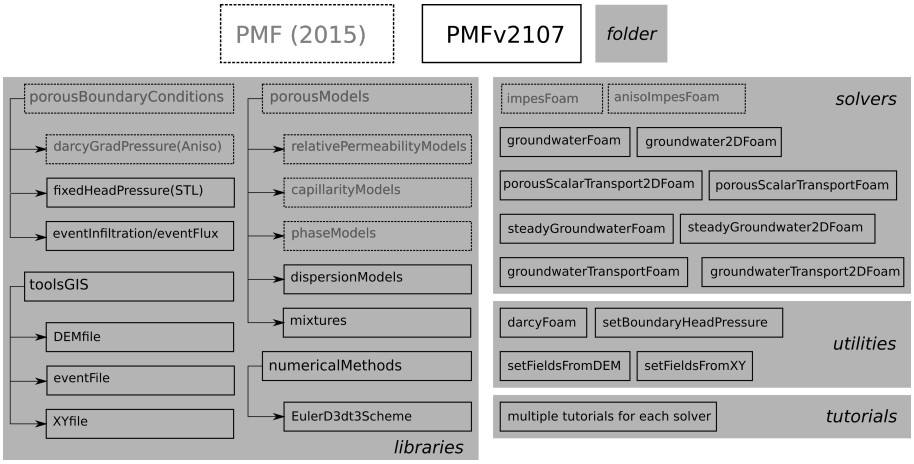


Figure 1: File structure

```
date Date1
X1 Y1 Z1 Value11
X2 Y2 Z2 Value12
...
XM YM ZM Value1M
date Date2
X1 Y1 Z1 Value21
...
date DateN
...
XM YM ZM ValueNM
```

Figure 2: `sourceEventFile` example with N dates and M injections points

4.3. Pre-processing tools

OpenFOAM executables have been developed to convert GIS data into informations for OpenFOAM simulations:

- `setFieldsFromDEM`: a variant of the official OpenFOAM `setFields` tool which performs bilinear interpolation from DEM file in ASCII format (.xyz). This tool is generally used to set up the local height of the aquifer (z_{min} and z_{max}).
- `setFieldsFromXY`: similar to `setFieldsFromDEM` with an interpolation performed using the three closest points of any list of points. This tool is used in this work to set up a permeability field with non-uniform values.
- `setBoundaryHeadPressure`: utility to set up boundary conditions for unsaturated (h) or 2D-saturated (ψ) solvers using a uniform water height or using a DEM or STL input file.

A simple Darcy utility has also been added (`darcyFoam`) to obtain an initial single-phase velocity field which can be used for initialization and/or passive scalar transport.

4.4. Multi-species transport

The transport properties of species (dispersion coefficients, retardation, decay and other parameters) have been developed inside a new class `multiScalarMixture` derived from the OpenFOAM class `basicMixture`. This makes easy to manage the transport of a N-component mixture with homogeneous or heterogeneous properties.

5. Validation of new developments

To validate the development presented in the previous sections, each solver is tested independently in a configuration adapted to the studied mathematical model. The validation of the code is performed by comparing results with those obtained using METIS [13], a finite-element solver for saturated/unsaturated flow that can handle multiple features such as solute transport or reactions. This code, developed during the last 30 years and still in development, has been used in several studies [23, 26, 20] and is considered here as our reference solution. Numerical schemes for the finite-element validation code METIS are similar to those used by the PMF toolbox (*i.e.*, second order in time and space).

The validation cases presented in the following are inspired by real cases and the physical values (time, flux, concentration) are representative of realistic configurations. The differences in orders of magnitude between output variables can thus be different and to improve the readability of this section, the choice is made (i) to present the real quantities in the configuration description of each test and (ii) to plot numerical comparisons with dimensionless variables. Quantities are systematically made dimensionless using the largest value from the reference solution, *i.e.* the METIS simulation.

5.1. Unsaturated flow solvers

The development of new solvers in OpenFOAM is independent of dimensionality of the domain (1D/2D/3D) thanks to the automatic discretization of the equations. Nevertheless, this is not the case for the reference code (METIS) and therefore validation is performed on a one-dimensional case even though the solvers in PMF can be used in two or three dimensions.

The physical configuration for the validation of unsaturated solvers consists of a 1D column 34.50 m in height discretized with 5009 computational cells (each with height $\Delta z = 0.0249$ m). We consider the bottom of the column to be in contact with the saturated area by imposing a Dirichlet boundary condition ($H_{bottom} = 0$ m) while an infiltration velocity (steady or transient) is imposed on the top. The model and physical properties are summarized in Table 1.

viscosity	μ	1.10^{-3}	[Pa.s]
density	ρ	100	[kg.m^{-3}]
Van Genuchten coefficients	m	0.3007	[-]
	α	13.0	[m^{-1}]
Permeability	K	7.10^{-12}	[m^{2}]
Kinematic porosity	ε	0.27	[-]
Specific storage	S_S	0.001	[-]

(a) Flow properties

Total porosity	ε_{total}	0.30	[-]
Tortuosity	τ	1	[-]
Molecular diffusivity	D_m	1.10^{-9}	[m^{2}.s^{-1}]
Dispersivity coefficient	α_L	1.0	[m]
	α_T	0.2	[m]
Volume partitioning coefficient	K_d	5.10^{-5}	[m^{3}.kg^{-1}]
Radioactive decay coefficient	λ	1.10^{-9}	[s^{-1}]

(b) Transport properties

Table 1: Properties for unsaturated validation cases

5.1.1. Steady flow (`steadyGroundwaterFoam`)

The first case consists in solving a steady flow with a constant infiltration velocity on the top of the column ($V_{top} = 4.753 \cdot 10^{-9}$m/s). The convergence residual for the pressure head equation is set to 10^{-10}

and the numerical results are presented in Figure 3 (H is scaled using the height of the column). Although the agreement is excellent between the two codes, a slight difference can be seen close to the bottom where the height tends to 0. Due to the relatively large α value in the Van Genuchten model, the saturation variation occurs over a small height ($\frac{H}{H_{column}} < 0.01$) with the presence of a strong pressure head gradient on few cells. Even if the size of the elements is similar between the two codes, METIS solves mass conservation at the nodes while OpenFOAM (with its finite volume method) solves conservation at the cell centers, so that boundary conditions are imposed slightly differently. This fact explains the small differences in the gradient computation near the boundaries.

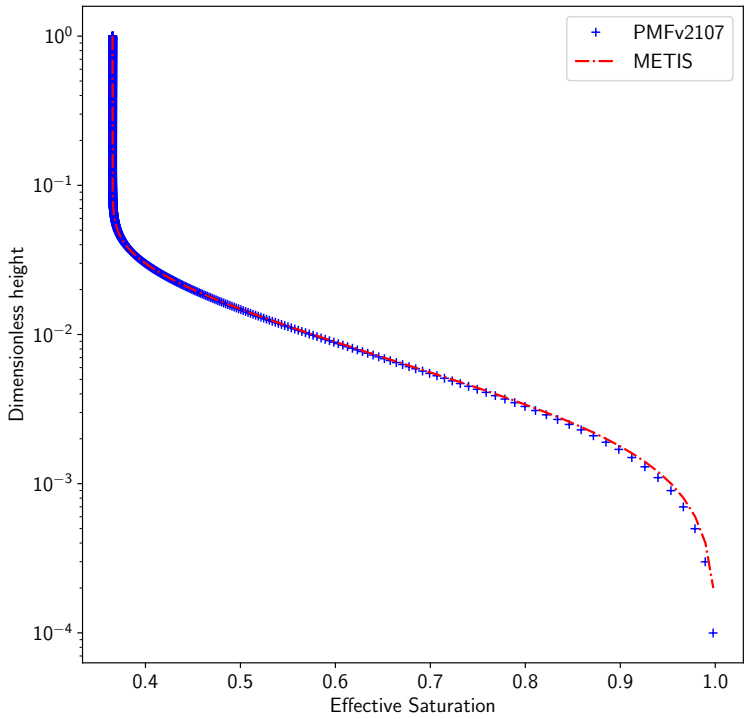


Figure 3: Saturation distribution with PMF and METIS in the steady case.

5.1.2. Transient flow (`groundwaterFoam`*)*

Starting from the steady solution of the previous case, the infiltration velocity at the top takes the form of a piecewise linear function, with values varying between 0 and 1000 times the reference injection value ($V_{top} = 4.753 \cdot 10^{-9}$m/s) over 2.10^8 seconds ($\sim$ 2315 days). This variation of infiltration was chosen to verify robustness and stability in the case of a high amplitude and sudden injection in an unsaturated zone. The results over time are compared in terms of water flow rate out of the column (cf Fig 4) and show good agreement. Some small temporary differences are observed during the drainage phase (second half of the simulation), when flow dynamic is driven by the local pressure head gradients which may slightly differ between the discretization methods (as for the steady configuration).

5.1.3. Passive transport (`porousScalarTransportFoam`*)*

Using the pre-computed saturation and velocity field of the steady case, a passive tracer is injected from the top ($Q = 1.374.10^{-8}$ kg.m^{-2}.s^{-1}) for a period of 6622 seconds. The physical time of the simulation is $1.262.10^9$ seconds ($\sim$ 40 years) and solute flux going out the column for both codes are presented in Figure 5. The good agreement observed allows to validate the good implementation of multiple elements influencing

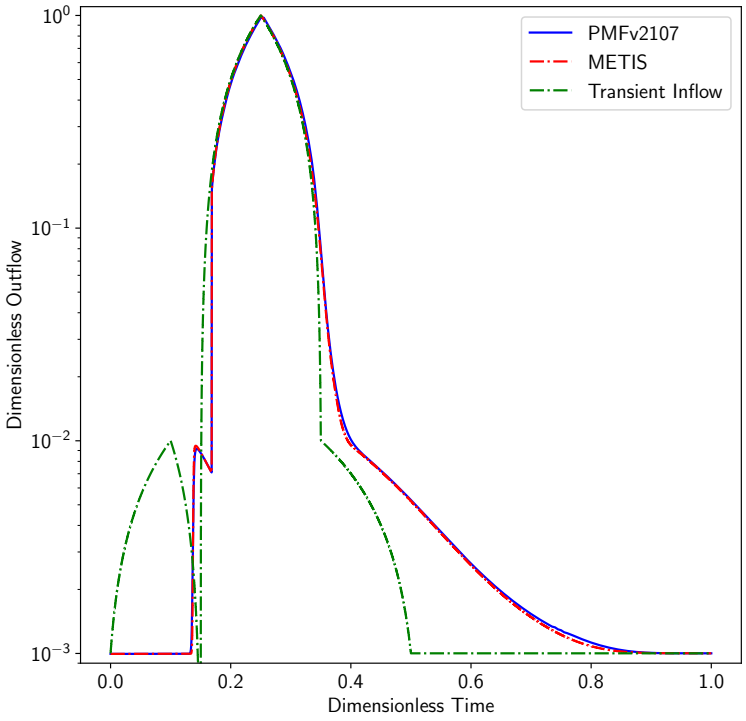


Figure 4: Comparison of water outflow with PMF and METIS, transient flow case

the solute transport, namely: the heterogeneous saturation, the dispersion, the decay and the retardation factor.

5.1.4. Coupled water/tracer simulation (`groundwaterTransportFoam`*)*

Starting from the steady configuration, water is injected inside the column (at $z = 24$ m) at fixed flow rate $Q_{in} = 3.82 \cdot 10^{-5}$ m^3/s (*i.e.* 0.135 m^3/h) with an imposed solute concentration $C_{in} = 6.12 \cdot 10^{-11}$ kg.m^{-3} from $t_{start} = 0$ to $t_{stop} = 91$ days; the injection is then stopped and the simulation continues until $t_{end} = 730$ days. This configuration is challenging because it implies successively: (i) an abrupt variation of local water content, (ii) the displacement of a water front close to the saturated point ($h \approx -0.0033$ and $\frac{\theta}{\theta_{max}} \approx 0.997$) followed by a quasi-steady case, (iii) an abrupt change related to the end of injection and (iv) a capillary drainage over a long time period. Water and solute fluxes going out the column are presented in Figure 6

This configuration highlights the efficiency of the time step management method based on the time-step truncation error. This case presents various time scales which can be seen by observing the simulation time-steps reported in Figure 7. At the early beginning ($t < 1$ h), small time steps ($0.1 < \Delta t < 20$ s) are required to correctly capture the local and rapid increase of saturation at the injection point. Then, the front displacement inside the column implies sharp local saturation and pressure head changes from ($t \approx 1$ to $t \approx 21$ h) with $\Delta t \approx 9$ s. Finally, the quasi-steady flow is established when the front reaches the bottom, which allows time step management to greatly increase the time step size to reach the maximal value $\Delta t_{max} = 90$ hours (user-defined value corresponding to the solver outputs). The abrupt changes of saturation can be visualized at $t = 2160$ h (90 days) when the injection is stopped. However, the capillary drainage that follows this event is slow and allows the solver to quickly restore the large time step.

Moreover, this configuration highlights the convergence issues of the Picard's algorithm compared to Newton's algorithm. Due to the strong non-linearities of the Van Genuchten function, particularly close to the saturation point (the derivative of the capillary function tends to infinity when saturation tends to 1), Picard's algorithm does not converge easily and requires both many linear iterations and small time steps. This results

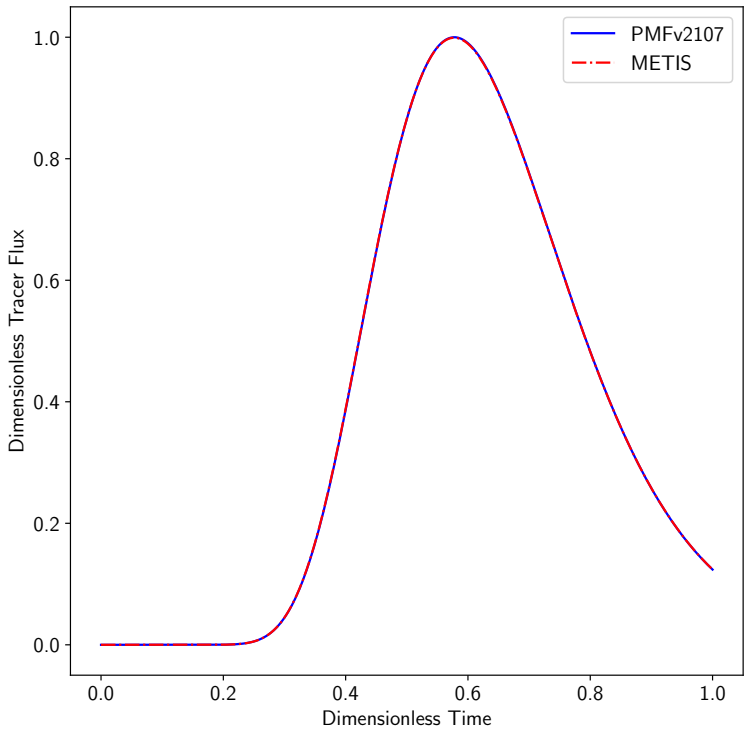


Figure 5: Tracer flux when solving for passive transport with PMF and METIS

in a prohibitive computational cost which makes it impossible to simulate this configuration in a reasonable time (actual execution times range from fewer than 4 minutes when using the newly implemented Newton's algorithm to several hours with Picard's algorithm on a standard CPU). The direct comparison between the two codes on a single-core simulation highlights computation times of the same order of magnitudes with an increased efficiency of PMF, more or less important depending on the case.

5.2. 2D-saturated flow solver (2D watershed modeling)

In the same way as for the validation of unsaturated solvers, the 2D-saturated solvers are tested on a realistic watershed. The same unstructured mesh presented in Figure 8a is used for both codes. However, it must be noted that due to the different numerical methods used, METIS (Finite Element) computes the solution on the mesh nodes while OpenFOAM (Finite Volume) computes the solution on mesh cells. Figure 8 shows the permeability field (figure 8b) and Digital Elevation Models (DEM) for the upper (figure 8c) and the lower (figure 8d) limits of the aquifer. As for the unsaturated simulations, METIS is considered our reference solution for the validation of PMF and comparisons are carried out on probes placed in the domain and visible in Figure 8d. All probes are used for the validation of the steady case but only the marked probes (probe A, shown in green , and B, marked in orange) are used for transient comparison.

5.2.1. Steady flow (`steadyGroundwater2DFoam`)

The first case consists of solving a steady flow with a constant and homogeneous infiltration velocity on the whole watershed ($V_{infiltration} = 5.59 \cdot 10^{-9}$m/s). The convergence residual for the potential equation is set to 10^{-10}. Figure 9a shows the potentials and velocity fields in the steady configuration with the watershed outlet located in the lower right corner. Figure 9b shows the good agreement in terms of potentials between METIS and PMF at each probe.

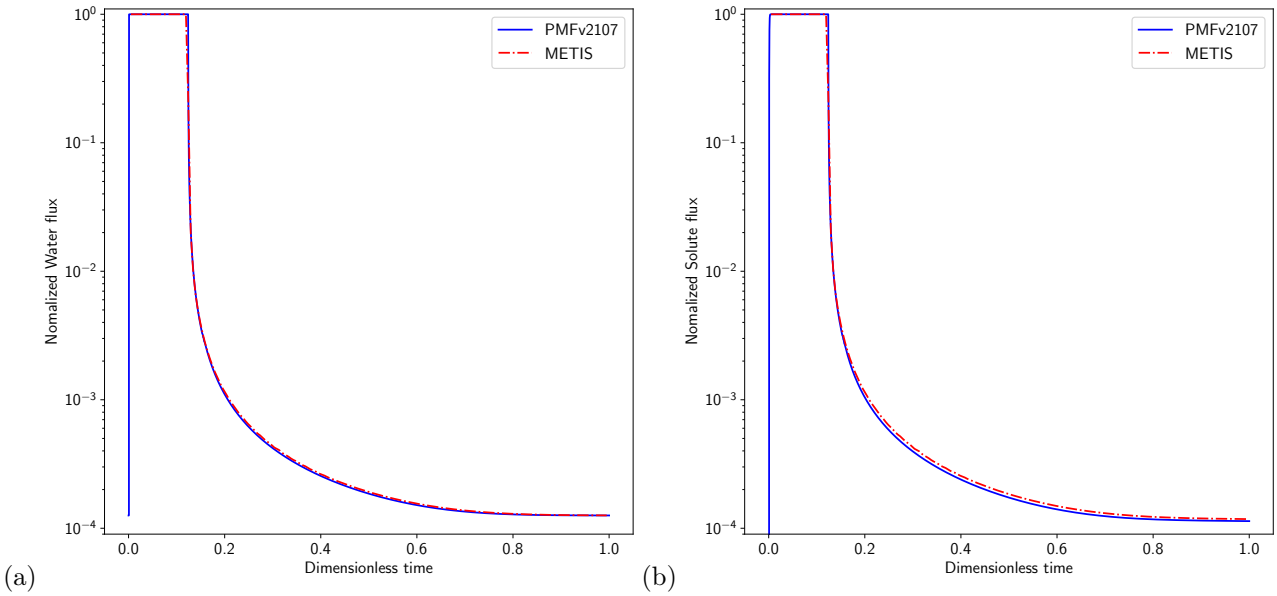


Figure 6: Fluxes on the bottom of the column as a function of time: (a) water flux, (b) solute flux

5.2.2. Transient flow (`groundwater2DFoam`*)*

Starting from the steady solution of the previous case, the infiltration velocity on the top varies in time linearly and disorderly between 0 and 4 times the reference injection value ($V_{inf,ref} = 5.59 \cdot 10^{-9}$ m/s, the amplitude of variation corresponds to realistic values at the level of the saturated zone) over $4.2.10^8$ seconds ($\sim$ 4861 days). Temporal variation of dimensionless infiltration value ($\frac{V_{inf}}{V_{inf,ref}}$) are visible in Figure 10 as well as the potential variations of probes A and B. Comparisons show good agreement with a slight difference between codes that occur during the event. This difference may be due to several factors related to numerical methods used such as the seepage condition which occurs in cells for PMF and in nodes for METIS or the post-processing operations (PMF results in mesh cells are interpolated to get probes value placed on mesh nodes).

5.2.3. Passive transport (`porousScalarTransport2DFoam`*)*

Using the pre-computed saturation and velocity field, a passive tracer is injected ($Q = 5.10^{-6}$ kg.m^{-2}.s^{-1}) for a period of 4.10^8 seconds while the physical final time of the simulation is $t_{end} = 4.73.10^9$ seconds ($\sim$ 40 years). Concentrations fields at $t = 0.2 \times t_{end}$ and $t = t_{end}$ are presented in Figure 11. To validate both transversal and longitudinal dispersion, probes A and B have been placed respectively in the direction of and perpendicular to the main flow. The good agreement between the observed probes (c.f. Figure 12) allows to validate the correct implementation of the multiple elements influencing the solute transport, namely: the heterogeneous potential and water height fields, the dispersion related to the velocity field, the decay and the retention coefficient.

5.2.4. Coupled simulation (`groundwaterTransport2DFoam`*)*

Starting from the steady 2D simulation, a passive tracer is injected in a similar manner to the previous case ($Q = 5.10^{-6}$ kg.m^{-2}.s^{-1}) for a period of 4.10^8 seconds with the final physical time of the simulation equal to $t_{end} = 4.73.10^9$ seconds ($\sim$ 40 years). A transient and variable infiltration is applied during the simulation inducing slight concentration variations related to water height changes as can be seen in Figure 13. As for previous 2D validation cases, the specificities of numerical codes induce slight differences with smaller transient variations in the case of METIS but with a globally similar dynamic behavior.

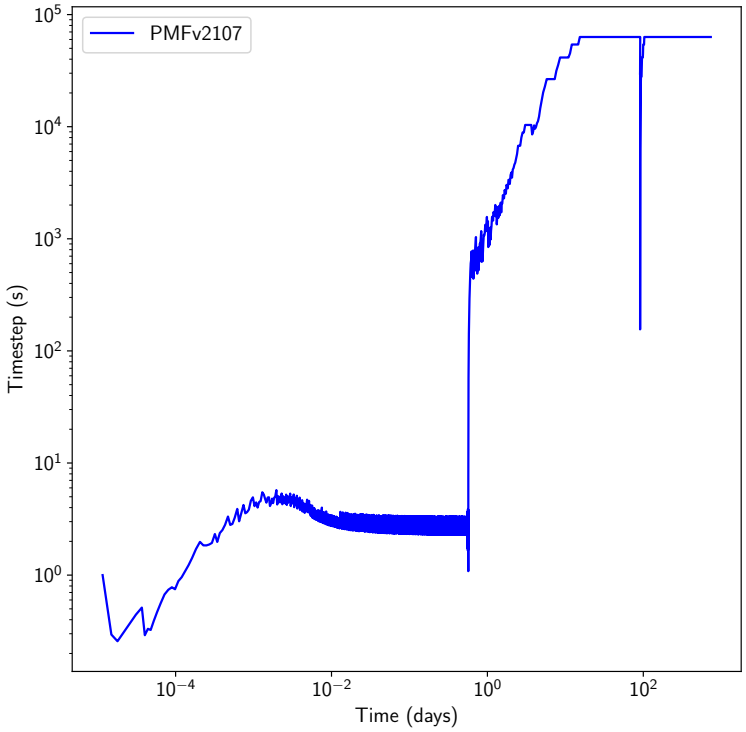


Figure 7: Evolution of (automatically adjusted) time steps during the simulation

5.3. Parallel Efficiency

To ensure that the good parallel efficiency of PMF is preserved within the new developments, we performed a weak scaling test consisting in solving the same problem with the discretized domain splitted among an increasing number of processors. Test is performed on the two-dimensional steady case (cf 5.2.1) using a much finer mesh with a mesh resolution of $\Delta x = \Delta y = 3\ m$ and a total of 2.05 M cells. Simulations are performed using between 18 and 576 cores on the CALMIP Olympe cluster (Intel® Skylake 6140). Results are reported in Table 2 with CPU time t_{CPU} computed as:

$$t_{CPU} = t_{SIM} \times n_{cores}, \tag{21}$$

where t_{SIM} is the total physical time of the simulation and n_{cores} the number of cores used for computation. The CPU time per linear iteration $t_{CPU,iter}$ is obtained as:

$$t_{CPU,iter} = \frac{t_{CPU}}{n_{iter}} \tag{22}$$

with n_{iter} the number of linear iterations which may vary according to the domain decomposition. The results are plotted in Figure 14 and compared to linear behavior expected from the reference, i.e. one computation node composed of 36 cores (the 18-cores simulation occurs on a half node and can be shared with other tasks). The results highlight a super-linear behavior until 144 cores, as was previously observed with the OpenFOAM platform [16, 17]. The parallel efficiency measured as CPU time per linear iteration remains lower than the reference ($t_{CPU,iter} = 86\,ms$) up to 288 cores before showing a deterioration. This deterioration has been observed previously with an efficiency decrease occurring when the number of cells per processor becomes lower than 5000 (around 3500 in this case). Poor results observed for 576 cores (with a large number of linear iterations inducing large physical time), are mainly due to the increase of the under-relaxation (0.1 vs. 0.5 for other configurations). This modification is mandated by the decomposition in small subdomains that makes the solving process unstable in the standard configuration.

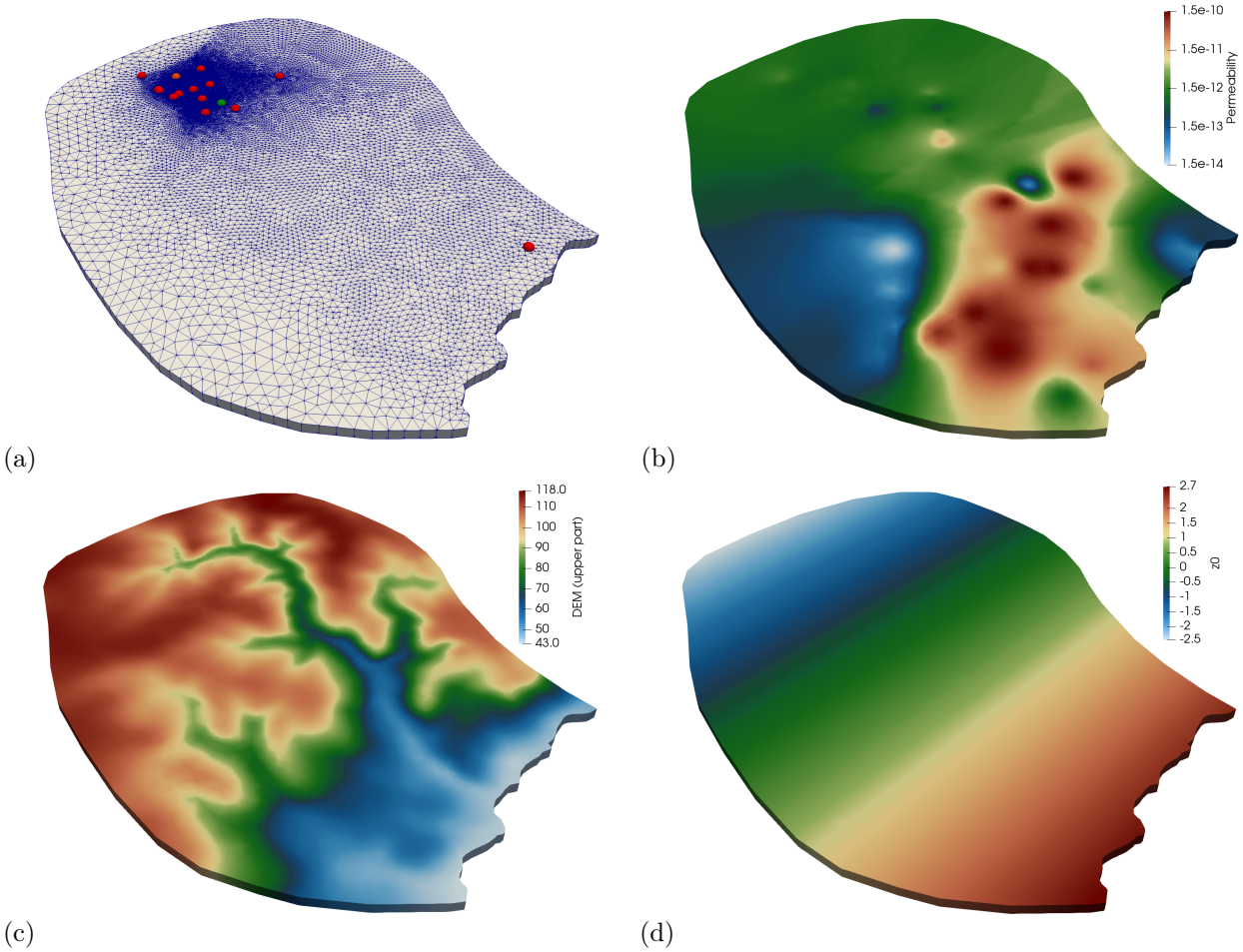
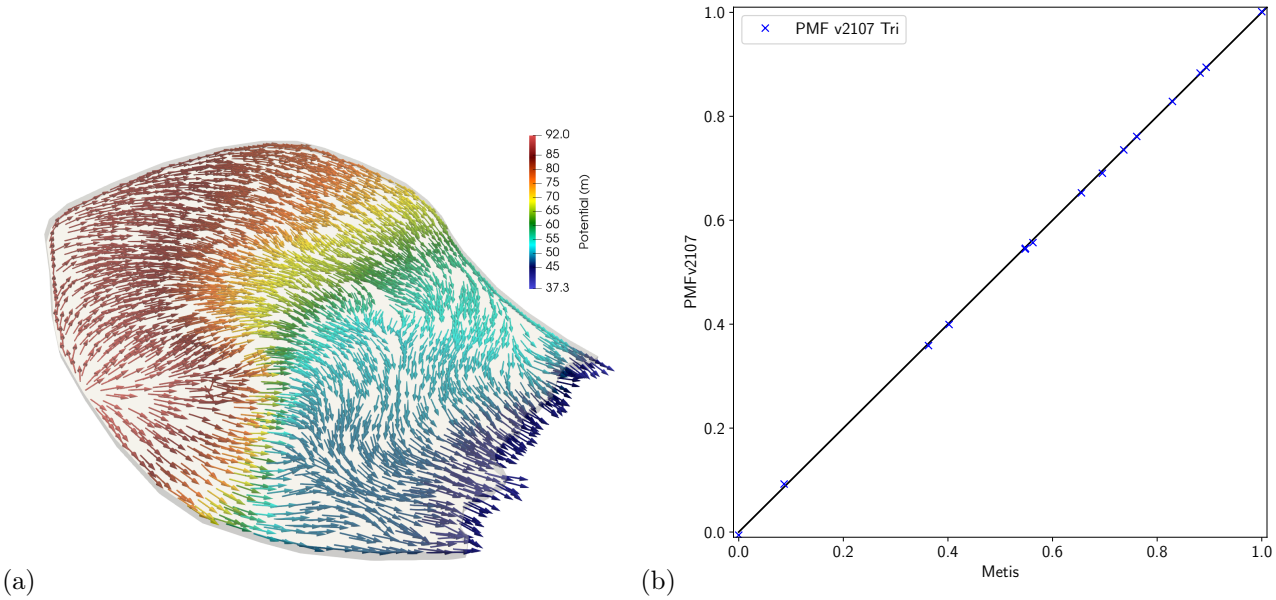


Figure 8: Watershed for the 2D validation cases: (a) mesh (b) permeability field (c) DEM for upper limit (d) DEM for lower limit. Probes used for METIS/PMF comparison are visible on (a)

6. Conclusion

The porousMultiphaseFoam toolbox, initially designed as a generic tool for the efficient parallel solution of multiphase flow through porous media, had shortcomings in hydrological modeling, with some missing preprocessing and solver features. The tool has now been improved and offers a very complete hydrogeological modeling suite with advanced numerical methods (such as automatic time step management and nonlinear algorithms), fast modeling methods (including simplified 2D approaches), passive/coupled transport modeling, and preprocessing tools essential to the setup of realistic cases. Every part of the new developments has been validated by a dedicated test case and a comparison with an existing reference code.

Main perspective of this work concerns the 3D modeling of water and solute transfers of a real configuration. The cases studied in this work are 1D or 2D to reduce computational cost and allow direct comparison with an well 1D/2D validated code. However, the developments made in the OpenFOAM environment can be directly used for 3D simulations and its good parallel efficiency make possible to compare simplified approach (1D+2D) with full modeling (3D) on real cases. Moreover, some improvements are planned in term of features and ergonomics. The development of a simple reactions module is in the implementation stage, as well as a coupling with a chemical modeling code as has been done by analogous packages based on OpenFOAM. In terms of ergonomics, an interface between the simulation code and a GIS management tool will be set up to allow use of the toolbox by a larger community.



(a) (b)

Figure 9: Steady flow: (a) visualization of velocity and potential fields, (b) comparison of potential values between METIS and PMF at each probe.

n_{cores}	$n_{cells/core}$	n_{iter}	t_{SIM} [s]	t_{CPU} [s]	$t_{CPU,iter}$ [ms]
18	113.6×10^3	45.2×10^3	279.9	5037	111
36	56.8×10^3	46.5×10^3	111.6	4018	86
72	28.4×10^3	43.1×10^3	39.2	2818	65
144	14.2×10^3	43.3×10^3	19.7	2837	66
288	7.1×10^3	46.4×10^3	12.9	3712	80
576	3.5×10^3	118.6×10^3	20.1	11600	98

Table 2: Simulation costs for parallel efficiency evaluation

Funding information

This work was performed using HPC resources from CALMIP (Grant P21009).

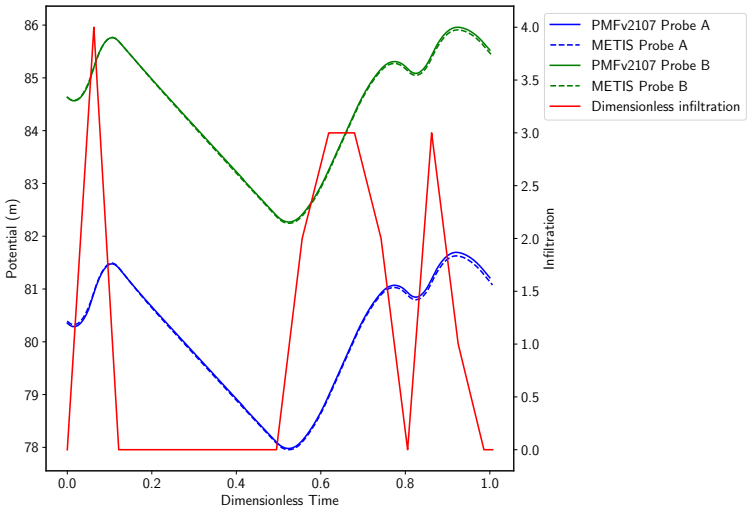
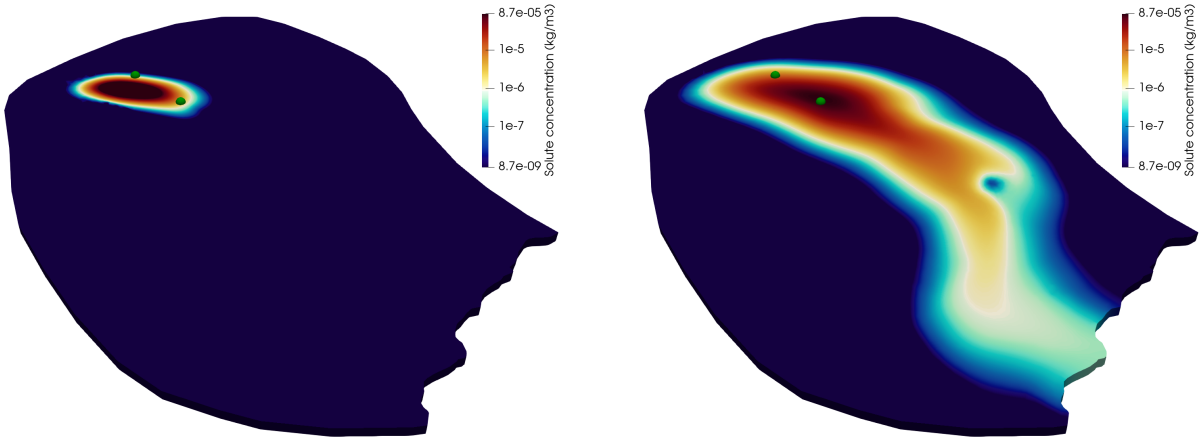


Figure 10: Comparison of transient potential variations for probes A and B



(a) (b)

Figure 11: Concentration visualization at (a) $t = 0.2 \times t_{end}$ (b) $t = t_{end}$

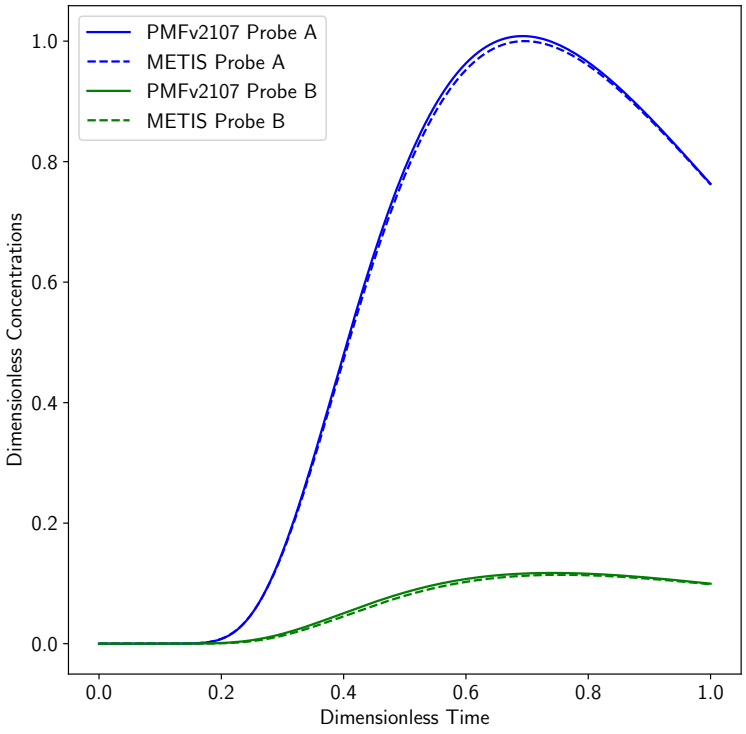


Figure 12: Comparison of tracer concentrations at probes A and B

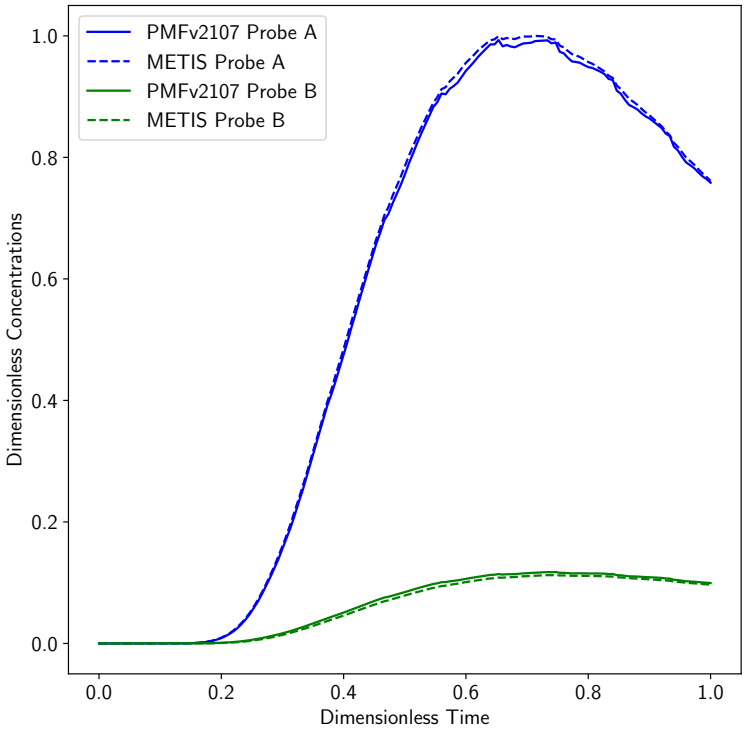


Figure 13: Comparison of tracer concentrations at probes A and B

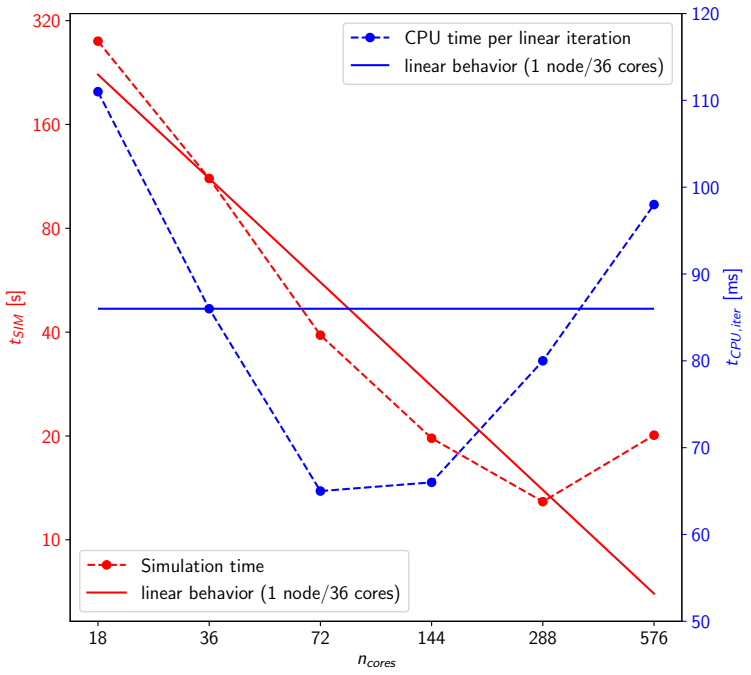


Figure 14: Log-log representation of simulation time and CPU time per linear iteration versus number of cores. Linear behaviour is extrapolated from the reference (1 node/36 cores).

References

[1] Steven F. Ashby and Robert D. Falgout. A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. *Nuclear Science and Engineering*, 124(1):145–159, 1996.

[2] J. Bear. *Dynamics of fluids in porous media*. New York American Elsevier, 1972.

[3] J. Bear and A. H.-D. Cheng. Modeling groundwater flow and contaminant transport. page 434, 2010.

[4] R. H. Brooks and A. T. Corey. Hydraulic Properties of Porous Media and Their Relation to Drainage Design. In *Transactions of the ASAE*, volume 7, pages 0026–0028, 1964.

[5] F. Carrillo, I. C. Bourg, and C. Soulaine. Multiphase flow modeling in multiscale porous media: An open-source micro-continuum approach. *Journal of Computational Physics: X*, 8:100073, sep 2020.

[6] Michael A. Celia, Efthimios T. Bouloutas, and Rebecca L. Zarba. A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research*, 26(7):1483–1496, jul 1990.

[7] H. Deng, Z. Dai, A. V. Wolfsberg, M. Ye, P. H. Stauffer, Z. Lu, and E. Kwicklis. Upscaling retardation factor in hierarchical porous media with multimodal reactive mineral facies. *Chemosphere*, 91(3):248–257, 2013. doi: 10.1016/j.chemosphere.2012.10.105.

[8] C. M. F. D'Haese, M. Putti, C. Paniconi, and N. E. C. Verhoest. Assessment of adaptive and heuristic time stepping for variably saturated flow. *International Journal for Numerical Methods in Fluids*, 53: 1173–1193, 2007.

[9] Matthew W. Farthing and Fred L. Ogden. Numerical Solution of Richards' Equation: A Review of Advances and Challenges. *Soil Science Society of America Journal*, 81(6):1257–1269, 2017.

[10] B. Flemisch, M. Darcis, K. Erbertseder, B. Faigle, A. Lauser, K. Mosthaf, S. Müthing, P. Nuske, A. Tatomir, M. Wolff, and R. Helmig. DuMux: DUNE for multi-{phase,component,scale,physics,...} flow and transport in porous media. *Advances in Water Resources*, 34(9):1102–1112, sep 2011.

[11] J.J. Fried and M.A. Combarnous. Dispersion in porous media. 7:169–282, 1971.

[12] Yanjun Gan, Qingyun Duan, Wei Gong, Charles Tong, Yunwei Sun, Wei Chu, Aizhong Ye, Chiyuan Miao, and Zhenhua Di. A comprehensive evaluation of various sensitivity analysis methods: A case study with a hydrological model. *Environmental Modelling and Software*, 51:269–285, 2014.

[13] P. Goblet. Programme METIS. Simulation d'écoulement et de transport miscible en milieu poreux et fracturé. Notice de conception. Technical report, LHM/RD/89/23., Technical Report, 1989.

[14] Fabrice Golfier, Michel Quintard, Fabien Cherblanc, Brendan A. Zinn, and Brian D. Wood. Comparison of theory and experiment for solute transport in highly heterogeneous porous medium. *Advances in Water Resources*, 30(11):2235–2261, 2007.

[15] G. E. Hammond, P. C. Lichtner, and R. T. Mills. Evaluating the performance of parallel subsurface simulators: An illustrative example with PFLOTRAN. *Water Resources Research*, 50(1):208–228, jan 2014.

[16] P. Horgue, C. Soulaine, J. Franc, R. Guibert, and G. Debenest. An open-source toolbox for multiphase flow in porous media. *Computer Physics Communications*, 187:217–226, oct 2015.

[17] Pierre Horgue, Jacques Franc, Romain Guibert, and Gérald Debenest. An extension of the open-source porousMultiphaseFoam toolbox dedicated to groundwater flows solving the Richards' equation. 2015.

[18] O. Ippisch, H. J. Vogel, and P. Bastian. Validity limits for the van Genuchten-Mualem model and implications for parameter estimation and numerical simulation. *Advances in Water Resources*, 29(12): 1780–1789, 2006.

[19] M. Kuznetsov, A. Yakirevich, Y.A. Pachepsky, S. Sorek, and N. Weisbrod. Quasi 3d modeling of water flow in vadose zone and groundwater. *Journal of Hydrology*, 450-451:140–149, 2012.

[20] Alexandra Mattei, Patrick Goblet, Florent Barbecot, Sophie Guillon, Yves Coquet, and Shuaitao Wang. Can soil hydraulic parameters be estimated from the stable isotope composition of pore water from a single soil profile? *Water (Switzerland)*, 12(2), 2020.

[21] M. Muskat. *Physical principles of oil production*. 1949.

[22] L. Orgogozo, N. Renon, C. Soulaine, F. Hénon, S. K. Tomer, D. Labat, O. S. Pokrovsky, M. Sekhar, R. Ababou, and M. Quintard. An open source massively parallel solver for Richards equation: Mechanistic modelling of water fluxes at the watershed scale. *Computer Physics Communications*, 185(12): 3358–3371, dec 2014.

[23] F. Renard and A. Tognelli. A new quasi-3D unsaturated-saturated hydrogeologic model of the Plateau de Saclay (France). *Journal of Hydrology*, 535:495–508, 2016. doi: 10.1016/j.jhydrol.2016.02.014.

[24] L.A. Richards. Capillary conduction of liquids through porous mediums. *Journal of Applied Physics*, 1931.

[25] L.F. Richardson. *Weather prediction by numerical process*. 1922.

[26] T. Robineau, A. Tognelli, F. Renard, L. Schaper, T. Robineau, and P. Goblet. A double medium approach to simulate groundwater level variations in a fissured karst aquifer. *Journal of Hydrology*, 565 (September):861–875, 2018.

[27] L. Sugumar, A. Kumar, and S. K. Govindarajan. Grid Adaptation of Multiphase Fluid Flow Solver in Porous Medium by OpenFOAM. *Petroleum and Coal*, 62(412):1301–1316, 2020.

[28] N. K. C. Twarakavi, J. Simunek, and S. Seo. Evaluating interactions between groundwater and vadose zone using the hydrus-based flow package for modflow. *Vadose Zone Journal*, 7(2):757–768, May 2008.

[29] R. Urteaga, E. Elizaldeta, and C. L. A. Berli. Transverse solute dispersion in microfluidic paper-based analytical devices (μPADs). *Analyst*, (10), 2018.

[30] M. Th. van Genuchten. A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. *Soil Science Society of America Journal*, 44(5):892–898, 1980.

[31] Y. Zha, J. Yang, J. Zeng, C. H. M. Tso, W. Zeng, and Liangsheng Shi. Review of numerical solution of Richardson-Richards equation for variably saturated flow in soils. *Wiley Interdisciplinary Reviews: Water*, 6(5):1–23, 2019.