ARTICLE IN PRESS

Computer Physics Communications ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

An open-source toolbox for multiphase flow in porous media☆

P. Horgue[a,*], C. Soulaine[a,b], J. Franc[a], R. Guibert[a], G. Debenest[a]

[a] Université de Toulouse, INPT, UPS, IMFT (Institut de Mécanique des Fluides de Toulouse), Allée Camille Soula, 31400 Toulouse, France

[b] Department of Energy Resources Engineering, School of Earth Sciences, Stanford University, 367 Panama Street, Stanford, CA 94305, United States

ARTICLE INFO

Article history:
Received 31 January 2014
Received in revised form
26 September 2014
Accepted 3 October 2014
Available online xxxx

Keywords:
Porous media
Multiphase flow
IMPES method
OpenFOAM®

ABSTRACT

Multiphase flow in porous media provides a wide range of applications: from the environmental understanding (aquifer, site-pollution) to industrial process improvements (oil production, waste management). Modeling of such flows involves specific volume-averaged equations and therefore specific computational fluid dynamics (CFD) tools. In this work, we develop a toolbox for modeling multiphase flow in porous media with OpenFOAM®, an open-source platform for CFD. The underlying idea of this approach is to provide an easily adaptable tool that can be used in further studies to test new mathematical models or numerical methods. The package provides the most common effective properties models of the literature (relative permeability, capillary pressure) and specific boundary conditions related to porous media flows. To validate this package, solvers based on the IMplicit Pressure Explicit Saturation (IMPES) method are developed in the toolbox. The numerical validation is performed by comparison with analytical solutions on academic cases. Then, a satisfactory parallel efficiency of the solver is shown on a more complex configuration.

Program summary

Program title: porousMultiphaseFoam

Catalogue identifier: AEUR_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEUR_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: GNU General Public License

No. of lines in distributed program, including test data, etc.: 36 355

No. of bytes in distributed program, including test data, etc.: 1106 569

Distribution format: tar.gz

Programming language: C++.

Computer: Any x86.

Operating system: Generic Linux.

Classification: 12.

External routines: OpenFOAM® (version 2.0 and superior)

Nature of problem:
This software solves multiphase flow in porous media.

Solution method:
The numerical approach is based on the finite-volume method (FVM). Mass conservation equations for each fluid phase are reformulated into a pressure–saturation system. The generalized Darcy's law is used to compute phase velocities in the porous medium. The pressure–saturation is solved using a segregated

☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail address: phorgue@imft.fr (P. Horgue).

http://dx.doi.org/10.1016/j.cpc.2014.10.005

method based on the IMPES method. Parallel computing can be performed using the standard domain decomposition method of OpenFOAM®.

Running time:
The Buckley–Leverett examples provided only take a few seconds to run.

1. Introduction

Simulation of multiphase fluid flow in heterogeneous porous media is of great importance in many areas of science and engineering including:

- hydrology and groundwater flow,
- oil and gas reservoirs,
- gas–liquid contactors,
- waste management, biodegradation, and so on.

In this work, only the common features of these different flows are considered, *i.e.* an isothermal and incompressible two-phase flow with capillary effects. Others physical features such as phase changes or compressibility of phases are not in the scope of this paper but are possible further developments of the presented toolbox. Due to the high complexity of the solid structure and possible large dimensions of the computational domain, the common strategy consists of defining volume averaged balance equations with effective properties such as permeabilities, porosity, etc., which take into account the microscopic flow morphology of the studied problem. With such an approach, a cell of the grid contains both fluid and solid. As usually done for the multiphase flow in porous media, the concept of "saturation" is defined as the volumic filling rate of a fluid phase (gas or liquid) with the void space of this cell and all properties, phase velocities, phase pressures, etc. are considered homogeneous within the computational cell. Readers interested in the averaging process can be referred to Das and Hassanizadeh [1] where a state-of-the-art in modeling and experimental techniques to study multiphase flow phenomena in porous media has been done with a focus on upscaling.

In the last decade, several open-source simulators dedicated to porous media flows have been developed such as, for example, Dumu^x [2], MRST [3], OpenGeoSys [4] and PFlotran [5]. The open-source platform used in this work, OpenFOAM® [6,7], does not belong to this list since it has not been conceived as a specialized simulation tool but as a general toolbox for solving partial differential equations. However, with growing community and popularity, the use of OpenFOAM® to simulate flow through porous materials becomes more and more prevalent. In the usual OpenFOAM® solvers, porous medium flows are modeled by adding viscous and inertial resistance terms in the Navier–Stokes momentum equation to obtain, in the porous domain, the commonly called Darcy–Forchheimer law [8]. A mask function allows to define both "porous" areas with Darcy and Forchheimer coefficients, and "free" areas where the classical momentum equation is solved. The porous medium model is generic and can therefore easily be used to develop new OpenFOAM® solvers. It has been used, for example, to study compressible reacting flows [9], mass transfer in solid oxide fuel cells [10] or interaction of waves and coastal porous structures [11,12]. However, the current porous medium handling in OpenFOAM® does not allow to simulate the common features of multiphase flow in porous media, mainly because it lacks some essential elements to this modeling, such as, phase saturations, relative permeability models, capillarity models, and specific boundary conditions. With an efficiency demonstrated in many fields of fluid mechanics, it seems therefore an appealing possibility to develop, in the OpenFOAM® standards, a dedicated toolbox that could serve as a basis for the study of multiphase flow in porous medium.

In this paper, we present a toolbox to simulate multiphase flow in porous media. Instead of solving a modified Navier–Stokes system, we solve the mass conservation equations for each fluid where the phase velocities are expressed using a generalization of Darcy's law [13]. Comprehensive reviews of the numerical methods available to solve this kind of problem can be found in the literature (see for example Aziz and Settari [14], Gerritsen and Durlofsky [15] or Chen et al. [16]). Two main methods can be retained to treat multiphase flow in porous media: (i) a sequential approach, IMplicit Pressure Explicit Saturation (IMPES) and (ii) a coupled approach, i.e. the "fully-implicit". The IMPES methodology treats all terms that depend on saturation, except the transient terms, as explicit functions of saturation. This allows saturation to be decoupled from the pressure, resulting in a smaller system of equations to be solved implicitly. This reduces significantly the computational effort. However, because IMPES involves some explicit terms, integration may be numerically unstable. As a result, the computational time saved by reducing the size of the system of nonlinear equations can be lost in small time stepping to solve saturations and could lead to numerical instabilities, or in some cases, to non-convergence. The "fully-implicit" approach solves the same equations as the IMPES method, except that it treats pressure and saturation variables implicitly. Thus, the "fully-implicit" method is unconditionally stable. One could refer to Cao [17] to have a large overview of the different formulations.

Given the sequential nature of OpenFOAM®, we have adopted the IMPES method to develop a dedicated toolbox for multiphase flow in porous media. This package, called *porousMultiphaseFoam*, includes two solvers *impesFoam* and *anisoImpesFoam* (for iso- and anisotropic porous medium, see Section 2.1), the most widely used porous multiphase models for relative permeabilities and capillarities and a new boundary condition to impose phase velocities.

The paper is organized as follows. In Section 2, we present the mathematical model and its implementation in OpenFOAM®. Then in Section 3, we describe the content of the *porousMultiphaseFoam* package. Finally, in Section 4 the toolbox is validated over several tests and the parallel performance is evaluated on a cluster.

2. Mathematical model

2.1. Mass–momentum conservation equations

When considering porous medium at the macro-scale, the flow is governed by volume averaged equations. Each computational cell contains both solid and void space (or pore-space) which is represented at the macro-scale as the porosity

$$\varepsilon = \frac{V_{void}}{V_{cell}}, \tag{1}$$

where V_{void} is the volume occupied by the void space and V_{cell} the volume of the cell. To deal with multiphase flow, we have to introduce the notion of saturation S_i defining the filling rate of the phase

Please cite this article in press as: P. Horgue, et al., An open-source toolbox for multiphase flow in porous media, Computer Physics Communications (2014), http://dx.doi.org/10.1016/j.cpc.2014.10.005

i within the pore-space of a computational cell

$$S_i = \frac{V_i}{V_{void}}, \tag{2}$$

where V_i represents the volume occupied by the i-phase within the computational cell. From their definitions, saturations vary in the range [0; 1]. In this work, we study the flow of a non-wetting phase a and wetting phase b through the porous medium. Saturations satisfy the following relationship for fluid saturated media

$$S_a + S_b = 1. \tag{3}$$

Considering an incompressible multiphase flow in a porous medium, the macro-scale mass balance equation for each phase i reads:

$$\varepsilon \frac{\partial S_i}{\partial t} + \nabla \cdot \mathbf{U}_i = q_i, \tag{4}$$

where $\mathbf{U}_i$ stands for the superficial velocity and q_i is a source term, used for injection or extraction wells.

In the generalized Darcy's model [13], the superficial velocity of each phase i is computed as

$$\mathbf{U}_i = -\frac{\mathbf{K}_i}{\mu_i} \cdot (\nabla p_i - \rho_i \mathbf{g}), \tag{5}$$

where the apparent permeability $\mathbf{K}_i$ is expressed as follows

$$\mathbf{K}_i = \mathbf{K} k_{ri}(S_b). \tag{6}$$

$\mathbf{K}$ is the permeability tensor of the porous medium and $k_{ri}(S_b)$ is the relative permeability of the phase i, whose value between 0 and 1 depends on the local saturation of the wetting phase S_b. This modeling suggests that the presence of another fluid reduces the pore space available, and therefore, reduces the permeability. The two most widely used relative permeability correlations (Brooks and Corey [18], Van Genuchten [19]) are detailed in the models' presentation (see Section 2.4) and implemented in the library. Two solvers are developed in the toolbox depending on the porous medium considered, isotropic or anisotropic. In the numerical validation, the porous medium is considered as isotropic which means that the tensor $\mathbf{K}$ can be replaced by a scalar K. Note that in both cases the permeability field can be heterogeneous, i.e. whose value vary in space (the permeability is defined as a tensor field $\mathbf{K}$ or a scalar field K). Both solvers are useful as the isotropic solver requires less memory and computation time.

Due to capillary effects inside the porous medium, we do not have equality between averaged pressure fields of each phase. In classical multiphase porous medium approach, we generally define a macro-scale capillary pressure p_c depending on the saturation S_b [20]

$$p_c(S_b) = p_a - p_b.$$

The p_c values depending on the considered flow and porous medium properties are usually obtained experimentally and then correlated on a capillary pressure model. The three most widely used capillary pressure correlations (Brooks and Corey [18], Van Genuchten [19] and linear model) are detailed in Section 2.5 and implemented in the library. The capillary pressure correlation eliminates an unknown of the system and the mass conservation equations read:

$$-\varepsilon \frac{\partial S_b}{\partial t} + \nabla \cdot \left(-\frac{K k_{ra}(S_b)}{\mu_a} (\nabla p_a - \rho_a \mathbf{g}) \right) = q_a, \tag{7}$$

$$\varepsilon \frac{\partial S_b}{\partial t} + \nabla \cdot \left(-\frac{K k_{rb}(S_b)}{\mu_b} (\nabla p_a - \rho_b \mathbf{g} - \nabla p_c(S_b)) \right) = q_b, \tag{8}$$

with p_a and S_b the system variables.

2.2. The IMPES method

The system described by the mass conservation equations (7) and (8) has strong non-linearities due to relative permeabilities and capillary pressure correlations. Then, the resolution of the coupled system requires the use of non-linear solver (Newton–Raphson method for example) and consequently involves substantial computation time. As explained in the introduction, the IMPES algorithm used in this work and proposed first by Sheldon et al. [21] is an alternative method consisting in a segregated resolution of the coupled equations. This method needs a new model formulation detailed below.

2.2.1. Model formulation

The mass conservation equations are reformulated into a pressure–saturation system by summing Eqs. (7) and (8). The system then reads:

$$\nabla \cdot \left(-\frac{K k_{ra}(S_b)}{\mu_a} (\nabla p_a - \rho_a \mathbf{g}) \right) + \nabla \cdot \left(-\frac{K k_{rb}(S_b)}{\mu_b} (\nabla p_a - \rho_b \mathbf{g} - \nabla p_c(S_b)) \right) = q_a + q_b, \tag{9}$$

$$\varepsilon \frac{\partial S_b}{\partial t} + \nabla \cdot \mathbf{U}_b = q_b. \tag{10}$$

The principle of this approach is to solve implicitly the global mass conservation, i.e. the pressure equation (9), while the saturation equation (10) is explicitly solved. The detailed algorithm as implemented in the toolbox is presented in Section 2.2.4.

2.2.2. Implemented formulation

To simplify the formulation, we define phase mobility M_i and gravitational contribution L_i as follows:

$$M_i = \frac{K k_{ri}(S_w)}{\mu_i}, \tag{11}$$

$$L_i = \frac{K k_{ri}(S_w)}{\mu_i} \rho_i. \tag{12}$$

Even if in the generalized Darcy's law the relation $L_i = \rho_i M_i$ is satisfied, we have found it convenient to separate each contribution. Actually, it must be noted that more complex models involving viscous resistance terms between phases for instance [22,23] could be written using this generic formulation. Therefore, the same solver basis could be used for further investigations with more sophisticated multiphase flow models.

Moreover, assuming that the capillary pressure only depends on saturation, the capillary term ∇p_c can be reformulated as

$$\nabla p_c = \frac{\partial p_c}{\partial S_b} \nabla S_b, \tag{13}$$

which allows to express the pressure equation (9) as a Poisson-type equation

$$\nabla \cdot ((M_a + M_b) \nabla p_a) = -\nabla \cdot \left((L_a + L_b) \mathbf{g} - M_b \frac{\partial p_c}{\partial S_b} \nabla S_b \right) + q_a + q_b, \tag{14}$$

and the saturation equation as

$$\varepsilon \frac{\partial S_b}{\partial t} + \nabla \cdot \left(-M_b \nabla p_a + L_b \mathbf{g} + M_b \frac{\partial p_c}{\partial S_b} \nabla S_b \right) = q_b. \tag{15}$$

To improve code readability, three fluxes depending on different contributions (pressure gradient, gravity and capillary pressure)

Please cite this article in press as: P. Horgue, et al., An open-source toolbox for multiphase flow in porous media, Computer Physics Communications (2014), http://dx.doi.org/10.1016/j.cpc.2014.10.005

are defined on each face of the computational grid:

$$\phi_p = \left(M_{a,c\to f} + M_{b,c\to f}\right) \nabla p_a \cdot \mathbf{S}_f, \tag{16}$$

$$\phi_g = \left(L_{a,c\to f} + L_{b,c\to f}\right) \mathbf{g} \cdot \mathbf{S}_f, \tag{17}$$

$$\phi_{p_c} = M_{b,c\to f} \left(\frac{\partial p_c}{\partial S_b} \nabla S_b\right)_{c\to f} \cdot \mathbf{S}_f, \tag{18}$$

where the operator $c \to f$ indicates that face-centered values are interpolated from cell-centered value using a numerical scheme detailed in Section 2.2.5. The global flux is computed as follows

$$\phi = \phi_p + \phi_g + \phi_{p_c}, \tag{19}$$

and the flux of phase b can be expressed

$$\phi_b = \frac{M_{b,c\to f}}{M_{a,c\to f} + M_{b,c\to f}} \phi_p + \frac{L_{b,c\to f}}{L_{a,c\to f} + L_{b,c\to f}} \phi_g + \phi_{p_c}. \tag{20}$$

2.2.3. Time-step limitations

To determine the time step for pressure equation, two conditions can be used in the provided solvers. The first limitation is directly inherited from the classical OpenFOAM® multiphase solvers [6,24] and is related to the Courant number Co, defined for the incompressible phase$i$

$$Co_i = \max_{\forall cell} \left(0.5 \frac{\sum_{f=0}^{m} |\phi_i|}{V_{cell}} \right) \Delta t, \tag{21}$$

with m the number of neighbor faces f to the considered cell. The coefficient for time-step change is then expressed

$$c_{\Delta t} = \frac{Co_{fixed}}{max\left(Co_a, Co_b\right)}. \tag{22}$$

To avoid sudden and too large increases of the time-step which could lead to numerical instabilities, we define the time-step of pressure equation (9) as follows

$$\Delta t_p^* = min\left(min\left(c_{\Delta t}, 1 + 0.1 c_{\Delta t}\right), 1.2\right) \Delta t_{last}, \tag{23}$$

that limits to a maximum increase of 20%. Several tests show that it is necessary to impose $Co_{fixed} \leqq 0.1$ to ensure stability to the numerical simulations.

The second possible limitations for pressure equation the most commonly used in the IMPES method is the CFL condition discussed by various authors [25–27]. It has been implemented in the provided toolbox and is defined as follows

$$CFL = \max_{\forall cell} \times \left[\frac{\Delta t}{\varepsilon V_{cell}} \left(2 \frac{\partial p_c}{\partial S_b} \frac{M_a M_b}{K\left(M_a + M_b\right)} \sum_{f=0}^{m} T_f + \frac{\partial F_b}{\partial S_b} \sum_{f=0}^{m} \phi \right) \right], \tag{24}$$

where F_b is the fractional flow

$$F_b = \frac{\frac{k_{rb}}{\mu_b}}{\frac{k_{ra}}{\mu_a} + \frac{k_{rb}}{\mu_b}}, \tag{25}$$

and T_f the transmissivity of the face f

$$T_f = \frac{K_f \|\mathbf{S}_f\|}{\Delta x_f}, \tag{26}$$

where Δx_f is the distance between the centers of two neighboring cells. The coefficient for time-step change, used in Eq. (23), is then expressed

$$c_{\Delta t} = \frac{C_{max}}{CFL}, \tag{27}$$

with $C_{max} \leqslant 1$. Note that to ensure stability for the various test cases provided in the toolbox, C_{max} is set to 0.75.

The stability for both conditions, *CFL* or *Co*, is not ensured if source/sink term are present since they are not take into account in these formulations. Moreover, if we consider further code developments in the conservation equations, stability will probably not be ensured. In anticipation of these potential changes, we added a limitation related to an user-defined maximal variation of saturation $\Delta S_{b,max}$. Then, the variation of S_b between two time steps should satisfy

$$\Delta S_{b,n\to n+1} \leq \Delta S_{b,max}, \tag{28}$$

which can be reformulated as

$$\Delta t_{S_b}^* = min \left(\frac{V_c \Delta S_{b,max}}{\varepsilon \left(-\sum_{f=0}^{m} \mathbf{U}_b \cdot \mathbf{S}_f + V_c q_b \right)} \right). \tag{29}$$

Then, the global time-step for the next iteration is given by

$$\Delta t_n = min\left(\Delta t_{S_b}^*, \Delta t_p^*\right). \tag{30}$$

2.2.4. Algorithm

1. Δt_{n+1} is computed from the two conditions (C_o or *CFL* and $\Delta S_{b,max}$).
2. Saturation S_b^{n+1} is explicitly computed using the last known flux field ϕ_b^n

$$\varepsilon \frac{S_b^{n+1} - S_b^n}{\Delta t_{n+1}} + \nabla \cdot \phi_b^n = q_b. \tag{31}$$

3. Properties depending on the saturation (M_a^{n+1}, M_b^{n+1}, L_a^{n+1}, L_b^{n+1} and $(\frac{\partial p_c}{\partial S_b} \nabla S_b)^{n+1}$) and related fluxes ($\phi_g^{n+1}$ and $\phi_{p_c}^{n+1}$) are updated.
4. Pressure field p^{n+1} is implicitly computed solving the pressure equation

$$-\nabla \cdot \left(M_{a,c\to f}^{n+1} + M_{b,c\to f}^{n+1}\right) \nabla p^{n+1} + \nabla \cdot \phi_g^{n+1} + \nabla \cdot \phi_{p_c}^{n+1} = q_a + q_b. \tag{32}$$

5. Then ϕ_p^{n+1} and, therefore, ϕ^{n+1} and ϕ_b^{n+1} can be updated for the next time step.

2.2.5. Numerical schemes

As the saturation has great influence on relative permeabilities and capillary pressure, it is necessary to use numerical schemes suitable to ensure robustness and stability to the segregated solver. In the provided solver *impesFoam*, each field (k_{ri}, K and $\frac{\partial p_c}{\partial S_b}$) has a user-defined interpolation scheme that can be modified in the simulation configuration files. In the following numerical validation of the solver (Section 4), we use the classical numerical schemes of the IMPES method which are:

- Relative permeability k_{ri}: first order upwind scheme for stability in the presence of a saturation front.
- Intrinsic permeability K: harmonic average for high heterogeneities.
- Derivative of the capillary pressure $\frac{\partial p_c}{\partial S_b}$: linear interpolation.

We should note that the generic implementation of interpolated field in the *impesFoam* solver allows the use of all the numerical schemes proposed by OpenFOAM® (high order, TVD, NVD, etc. [6,7,24]).

2.3. Wellbore models

In this work, we do not focus on the wellbore modeling in porous media which is not trivial and has been discussed by several

Please cite this article in press as: P. Horgue, et al., An open-source toolbox for multiphase flow in porous media, Computer Physics Communications (2014), http://dx.doi.org/10.1016/j.cpc.2014.10.005

authors [14,28]. We set up a simple structure in the software to allow the subsequent development of more complex models. For that, we consider constant injection and extraction flow rates of wellbores in the developed solver. Two mask functions, defined in the domain, are used to set up the positions of injection and extraction points (1 indicates the presence of a wellbore, 0 the absence). The user-defined global flow rate is equally divided between all the computational cells occupied by the wellbores, depending on their volume. We consider that wellbores inject wetting phase and extract the two phases, depending on the mobility. Under these assumptions, the source–sink terms for each phase can be expressed as:

$$q_a = -\frac{M_a}{M_a + M_b} Q_{extraction}, \tag{33}$$

$$q_b = Q_{injection} - \frac{M_b}{M_a + M_b} Q_{extraction}. \tag{34}$$

2.4. Relative permeability models

Two relative permeability models are provided in the developed library. Both models involve the notion of effective saturation, which is a normalized saturation of the wetting phase. The effective saturation reads

$$S_{b,eff} = \frac{S_b - S_{b,irr}}{1 - S_{a,irr} - S_{b,irr}}, \tag{35}$$

where $S_{a,irr}$ and $S_{b,irr}$ are the user-defined irreducible (minimal) saturations of the phases a and b respectively.

Both correlations depend on the effective saturation $S_{b,eff}$, the power coefficient m, and the maximal relative permeabilities, $k_{ra,max}$ and $k_{rb,max}$ (set to 1 if not specified by the user).

Brooks and Corey Model [18].

$$k_{ra}\left(S_{b,eff}\right) = k_{ra,max}\left(1 - S_{b,eff}\right)^m \tag{36}$$

$$k_{rb}\left(S_{b,eff}\right) = k_{rb,max} S_{b,eff}^m. \tag{37}$$

Van Genuchten Model [19].

$$k_{ra}\left(S_{b,eff}\right) = k_{ra,max}\left(1 - S_{b,eff}\right)^{\frac{1}{2}}\left(1 - \left(S_{b,eff}\right)^{\frac{1}{m}}\right)^{2m} \tag{38}$$

$$k_{rb}\left(S_{b,eff}\right) = k_{rb,max} S_{b,eff}^{\frac{1}{2}}\left(1 - \left(1 - S_{b,eff}^{\frac{1}{m}}\right)^m\right)^2. \tag{39}$$

2.5. The capillary pressure models

As for the relative permeability models, the capillary pressure correlations are based on the notion of effective saturation. However, the macro-scale capillary pressure tends to infinity when saturation S_b tends to $S_{b,irr}$ (and its derivative when S_b tends to $S_{b,max}$ in the Van Genuchten model). To accept irreducible and maximal saturations in numerical simulations, we define the effective saturation for capillary pressure S_{b,p_c} as follows

$$S_{b,p_c} = \frac{S_b - S_{p_c,irr}}{S_{p_c,max} - S_{p_c,irr}}, \tag{40}$$

where the minimal $S_{p_c,irr}$ is a user-defined parameter that should satisfy

$$S_{p_c,irr} < S_{b,irr}. \tag{41}$$

For the Van Genuchten model, the maximal saturation $S_{p_c,max}$ should satisfy

$$S_{p_c,max} > S_{b,max}. \tag{42}$$

Brooks and Corey Model [18]. The correlation for capillary pressure reads

$$p_c\left(S_{b,p_c}\right) = p_{c,0} S_{b,p_c}^{-\alpha}, \tag{43}$$

where $p_{c,0}$ is the entry capillary pressure and $\frac{1}{\alpha}$ the pore-size distribution index. Deriving the Eq. (43), the capillary term in the pressure (14) and saturation (15) equations can be expressed

$$\frac{\partial p_c}{\partial S_{b,p_c}}\left(S_{b,p_c}\right) = -\frac{\alpha p_{c,0}}{S_{p_c,max} - S_{p_c,irr}} S_{b,p_c}^{-\alpha}. \tag{44}$$

Van Genuchten model [19]. The correlation for capillary pressure reads

$$p_c\left(S_{b,p_c}\right) = p_{c,0}\left(\left(S_{b,p_c}\right)^{-\frac{1}{m}} - 1\right)^{\frac{1}{n}}, \tag{45}$$

where $p_{c,0}$ is the entry capillary pressure and m the Van Genuchten coefficient. The coefficient n is generally correlated with m with the following relationship:

$$\frac{1}{n} = 1 - m. \tag{46}$$

In the provided toolbox, this relationship is used to compute the n coefficient when it is not explicitly defined. Deriving the Eq. (45), the capillary term in the pressure (14) and saturation (15) equations can be expressed as

$$\frac{\partial p_c}{\partial S_{b,p_c}}\left(S_{b,p_c}\right) = -\frac{1-m}{m}\frac{p_{c,0}}{S_{p_c,max} - S_{p_c,irr}} \times \left(\left(S_{b,p_c}\right)^{-\frac{1}{m}} - 1\right)^{-m}\left(S_{b,p_c}\right)^{-\frac{1+m}{m}}. \tag{47}$$

Linear model. The linear model is also available with

$$p_c\left(S_{b,p_c}\right) = p_{c,0} + \left(1 - S_{b,p_c}\right)\left(p_{c,max} - p_{c,0}\right), \tag{48}$$

where $p_{c,0}$ and $p_{c,max}$ respectively the minimal and maximal capillary pressure. The capillary term in pressure and saturation equations is then given as follows

$$\frac{\partial p_c}{\partial S_{b,p_c}}\left(S_{b,p_c}\right) = -\left(p_{c,max} - p_{c,0}\right). \tag{49}$$

2.6. "Darcy velocity" boundary condition

In the IMPES method, solving the pressure equation implies some limitations in terms of boundary conditions. For example, it is not straightforward to impose phase velocities on boundaries. To make it possible in the *impesFoam* solver, we developed a suitable Neumann boundary condition (called `darcyGradPressure` in the toolbox) for the pressure field. Assuming fixed phase velocities on the considered boundary, the total velocity can be expressed

$$\begin{aligned}\mathbf{U}_{fixed} &= \mathbf{U}_{a,fixed} + \mathbf{U}_{b,fixed} \\ &= -\left(M_a + M_b\right)\nabla p_a + \left(L_a + L_b\right)\mathbf{g} - M_b\frac{\partial p_c}{\partial S_b}\nabla S_b.\end{aligned} \tag{50}$$

Then, we can expressed the Neumann boundary condition on the pressure field

$$\mathbf{n}.\nabla p_a = \mathbf{n}.\left[\left(M_a + M_b\right)^{-1}\left(\mathbf{U}_{fixed} - \left(L_a + L_b\right)\mathbf{g} + M_b\frac{\partial p_c}{\partial S_b}\nabla S_b\right)\right], \tag{51}$$

where $\mathbf{n}$ denotes the normal to the face boundary. A similar boundary condition called `darcyGradPressureAniso` is defined for the *anisoImpesFoam* solver. Note that in that case, the tensor $\mathbf{K}$ needs to be invertible.

Please cite this article in press as: P. Horgue, et al., An open-source toolbox for multiphase flow in porous media, Computer Physics Communications (2014), http://dx.doi.org/10.1016/j.cpc.2014.10.005

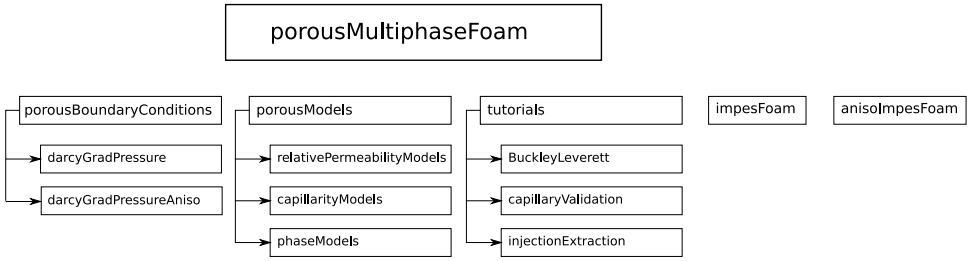

Fig. 1. Structure of the OpenFOAM® porous multiphase toolbox.

```
phasea{
rho rho [1 -3 0 0 0 0 0] 1e0;
mu mu [1 -1 -1 0 0 0 0] 1.76e-5;
}

phaseb{
rho rho [1 -3 0 0 0 0 0] 1e3;
mu mu [1 -1 -1 0 0 0 0] 1e-3;
}

relativePermeabilityModel BrooksAndCorey;

capillarityModel BrooksAndCorey;

BrooksAndCoreyCoeffs{
m 3;
Sminpc Sminpc [0 0 0 0 0 0 0] 0;
Smaxpc Smaxpc [0 0 0 0 0 0 0] 1;
pc0 pc0 [1 -1 -2 0 0 0 0] 5;
alpha 0.2;
}
```

Fig. 2. Example of a `transportProperties` file.

```
dimensions [1 -1 -2 0 0 0 0];

internalField uniform 0;

boundaryField {

boundaryExample{
type        darcyGradPressure;
}

}
```

```
dimensions [1 -1 0 0 0 0 0];

internalField uniform 0;

boundaryField {

boundaryExample{
type        fixedValue;
value       uniform (1e-5 0 0);
}

}
```

Fig. 3. Example of pressure `p` file (left) and velocity `Ub` file (right) for `darcyGradPressure` boundary condition.

3. Description of software components

The global organization of the `porousMultiphaseFoam` toolbox is depicted in Fig. 1.

The toolbox is divided in 4 parts: `porousModels`, `porousBoundaryConditions`, `impesFoam` and `tutorials`.

3.1. *porousModels*

This block compiles the `libporousModels.so` library containing the relative permeability, capillary pressure and phase models (see Sections 2.4 and 2.5). Model parameters needed by the classes should be defined in the usual configuration file `transportProperties`. An example of a configuration file for a Brooks and Corey correlation is presented in Fig. 2.

3.2. *porousBoundaryConditions*

This block compiles `libporousBoundaryConditions.so` library containing two new boundary conditions as detailed in Section 2.6 and derived from the OpenFOAM® basic boundary condition `fixedGradientFvPatchField`. The boundary condition is called in the pressure file `p` as depicted in Fig. 3(left) while the velocities have usual Dirichlet boundary conditions (see an example of `Ub` file in Fig. 3(right)).

3.3. *impesFoam*

The solver *impesFoam* solves equations described in the Section 2.2 considering isotropic porous medium (permeability K is a scalar field). This solver is used in the numerical validation of the developed library. It can be used as a development basis to integrate other features of multiphase flow in isotropic porous media.

3.4. *anisoImpesFoam*

The solver *anisoImpesFoam* solves same equations described as the *impesFoam* except that the intrinsic permeability `K` is a tensor field. Two injection cases are available in the provided toolbox. It

Please cite this article in press as: P. Horgue, et al., An open-source toolbox for multiphase flow in porous media, Computer Physics Communications (2014), http://dx.doi.org/10.1016/j.cpc.2014.10.005

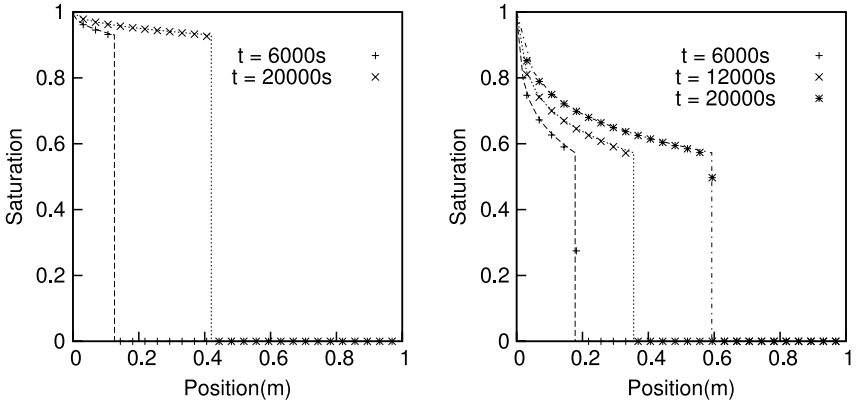

Fig. 4. Saturation profiles for the Brooks and Corey model (left) and the Van Genuchten model (right). Dash lines are theoretical saturation profiles.

can be used as a development basis to integrate other features of multiphase flows in anisotropic porous media.

3.5. `tutorials`

This block contains the validation tests presented in the Section 4: Buckley–Leverett and capillary validation. An injection/extraction test case is also provided to ensure the proper implementation of the source/sink terms.

4. Numerical validations

The toolbox is validated using the solver *impesFoam*, i.e. the isotropic version of the IMPES method. However, numerical methods are the same for anisotropic solver and two injection test cases are provided in the `tutorials` to show an example of the use of the anisotropic solver *anisoImpesFoam*.

4.1. Buckley–Leverett

We consider the simplified case of Buckley–Leverett, i.e. a two-phase flow in a 1D domain (length $L = 1$ m, porosity $\varepsilon = 0.5$, intrinsic permeability $K = 1 \times 10^{-11}$ m^2, 400 computation cells) without capillary effects. This simplified case allows to develop a semi-analytical solution to get the shock saturation (saturation at the front), the front velocity and the saturation profile behind the front [29].

In the following tests, three fluids are considered (air, water and oil) whose properties are summarized in Table 1. The domain is initially fully saturated with the non-wetting fluid (air or oil depending in this case, $S_b = S_{b,irr}$ and $S_a = S_{a,max}$,) and we inject the wetting fluid (water) with a fixed constant velocity $U_b = 1 \times 10^{-5}$ m s^{-1}.

The numerical validation is performed for the two relative permeabilities models, considering a water–air system for the Brooks and Corey model and a water–oil system for the Van Genuchten model. Model and algorithm parameters are summarized in Table 1. The comparison between numerical and semi-analytical results is shown in Fig. 4. A good agreement is found with some minor numerical diffusion mainly due to the "upwind" scheme used for the relative permeability computation.

The case of the gravity regime is also studied, considering vertical injection of water in a air-saturated system for both models. Except the gravity contribution, the simulation parameters remain unchanged. In the gravity regime, i.e. when the flow rate injection is low, the front saturation is given by solving

$$U_b - \frac{K k_{rb}(S_{b,front})}{\mu_b}\rho_b g_y = 0, \tag{52}$$

Table 1
Parameters for: (a) fluid, (b) model and (c) algorithm.

(a)		
Fluid	ρ (kg m^{-3})	μ (Pa s)
Air	1	1.76×10^{-5}
Water	1000	1×10^{-3}
Oil	800	1×10^{-1}
(b)		
Model	m	
Brooks and Corey [18]	3	
Van Genuchten [19]	0.5	
(c)		
Variable	Value	
p_a tolerance	10^{-12}	
CFL	0.75	
$\Delta S_{b,max}$	0.01	

which gives $S_{b,front} = 0.467$ for Brooks and Corey model and $S_{b,front} = 0.754$ for the Van Genuchten model. The computed front velocities are respectively 4.28×10^{-5} and 2.65×10^{-5} m s^{-1}. The good agreement between numerical and analytical results is shown in Fig. 5.

4.2. Capillary–gravity equilibrium

We now consider a two-phase flow (air/water), with capillary pressure effects in a vertical 1D domain, similar to the previous section ($H = 1$ m). The bottom boundary condition is now a fixed wall ("Darcy velocity" boundary condition with $\mathbf{U}_a = \mathbf{U}_b = \mathbf{0}$ m/s) and the top boundary condition is a Dirichlet condition with fixed reference pressure $p = 0$ Pa and irreducible saturation $S_b = S_{b,irr}$. We initialize the lower half of the domain with $S_b = 0.5$. Then we simulate the flow over a long period (2×10^6 s) to allow the establishment of a saturation profile along the vertical axis. When the stationary state is reached, we have

$$U_a = U_b = 0, \tag{53}$$

and then we can write:

$$\frac{\partial p_a}{\partial y} = \rho_a g_y, \tag{54}$$

$$\frac{\partial p_c}{\partial y} = \rho_b g_y - \frac{\partial p_a}{\partial y}. \tag{55}$$

The capillary pressure gradient can be therefore simply expressed in term of gravity contribution

$$\frac{\partial p_c}{\partial y} = (\rho_b - \rho_a) g_y. \tag{56}$$

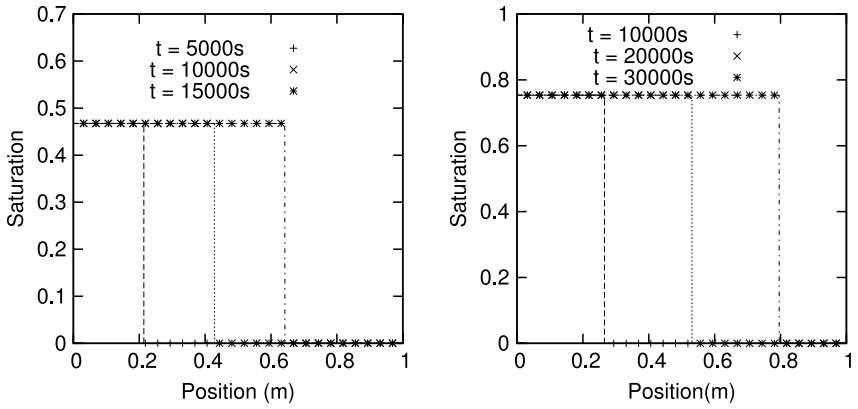

Fig. 5. Saturation profiles in the gravity regime for the Brooks and Corey model (left) and the Van Genuchten model (right). Dash lines are theoretical saturation profiles.

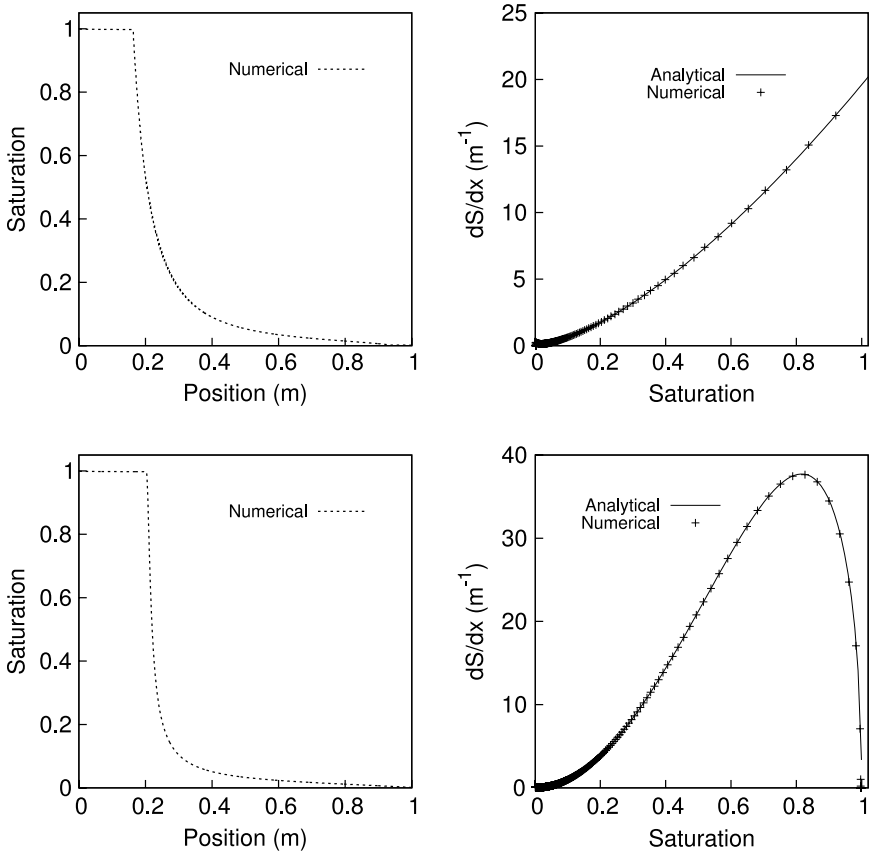

Fig. 6. Saturation profiles (left) and gradients (right) depending on the capillary pressure model (top: Brooks and Corey, bottom: Van Genuchten).

Therefore, the final saturation field should satisfy

$$\frac{\partial S_b}{\partial y} = \frac{(\rho_b - \rho_a)\, g_y}{\frac{\partial p_c}{\partial S_b}(S_b)}, \tag{57}$$

where $\frac{\partial p_c}{\partial S_b}(S_b)$ follows a given correlation described in Section 2.5. Algorithm parameters are identical to the previous test case (cf. Table 1) and the model parameters are summarized in Table 2.

Saturation profiles at the capillary–gravity equilibrium are presented in the Fig. 6 (left). The comparison between theoretical and numerical evaluations of saturation gradients (depending on the saturation) validates the numerical implementation of the presented models (see Fig. 6).

Table 2
Model parameters for capillary validation.

Model	$p_{c,0}$	m	α
Brooks and Corey [18]	1000		0.5
Van Genuchten [19]	100	0.5	

4.3. Performance test: viscous fingering in a heavy oil reservoir

We now consider a water injection ($U_b = 1 \times 10^{-4}$ m.s^{-1}) in a horizontal oil saturated system (see fluid properties on Table 1). The size of the reservoir is 1×0.4 m^2 (1.6×10^6 computation cells with a 2000×800 mesh). The permeability of the two-dimensional domain is heterogeneous by blocks with a value between 1×10^{-13}

Please cite this article in press as: P. Horgue, et al., An open-source toolbox for multiphase flow in porous media, Computer Physics Communications (2014), http://dx.doi.org/10.1016/j.cpc.2014.10.005

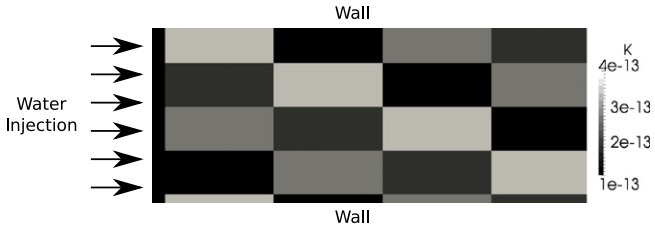

Fig. 7. Heavy oil reservoir permeability field and boundary conditions.

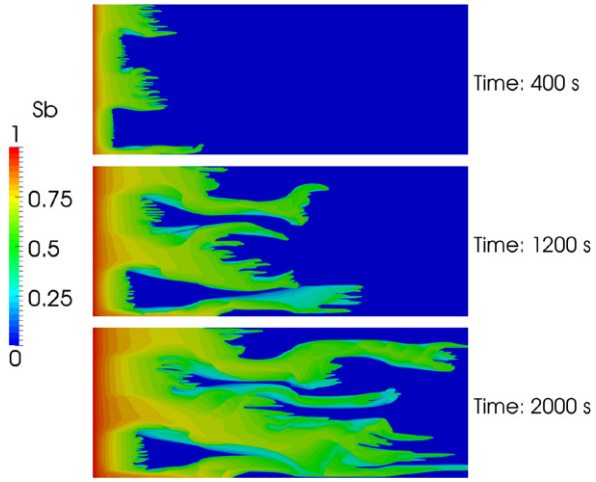

Fig. 8. Viscous fingering in a heavy oil reservoir (Water saturation field S_b).

and 4×10^{-13} m^2 (see Fig. 7). The Van Genuchten model is used for relative permeability and capillary pressure with $m = 0$, 5 and $p_{c,0} = 5$ Pa.

In this condition, we observe the emergence of multiple instabilities in the saturation front area (see Fig. 8(a)). The development of these instabilities leads to a so called "viscous fingering" (see Fig. 8(b)), a phenomenon due to the important viscosity ratio between the two fluids [30].

The viscous fingering presents an important challenge to oil industry and needs to be accurately understood and modeled because these instabilities decrease the efficiency of oil recovery processes. We do not focus in this study on these phenomena characterization and the reader interested could refer to previous experimental [30–34] and numerical studies [35–38]. Actually, the number of "viscous fingers" depends on the reservoir properties but also on numerical parameters such as grid refinement and algorithm tolerance. Therefore, the complete characterization would require a thorough study that is not in the scope of this paper. However, it is an interesting illustration of the possibilities of the solver by simulating a complex multiphase flow where the saturation front occupies almost the whole domain. A unique configuration is tested and used to evaluate the parallel performance of our solver.

In OpenFOAM®, the parallel algorithm is based on the domain decomposition method. Before the simulation, the whole domain is decomposed in n small domains, n being the number of computational cores used for the simulation. Then, each core solves a smaller linear system and information exchanges at boundaries between cores are done using the Message Passing Interface (MPI) communications protocol. Several decomposition methods are available in OpenFOAM® (*simple*, *scotch*, *manual*, etc.) and can be used independently of the considered solver. We use in our simulations the *simple* method which decomposes the domain in $n_x \times n_y \times n_z$ equal parts, where n_i are user-defined values. The

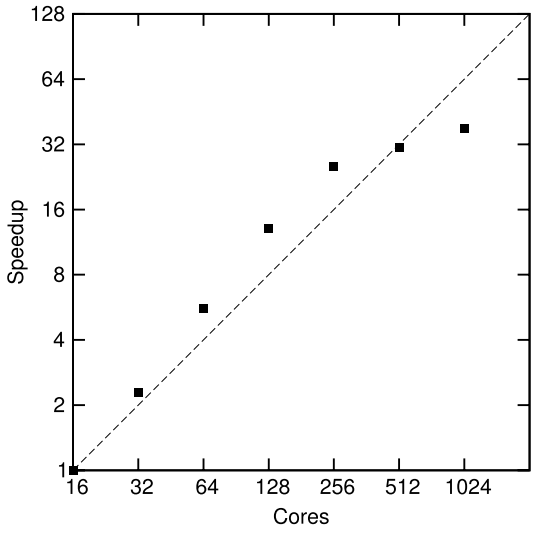

Fig. 9. Log–log representation of the speedup with the *impesFoam* solver (reference is 16 cores and linear solver is PCG).

pressure equation (14) is solved with the standard preconditioned conjugate gradient (PCG) solver with a fixed tolerance of 10^{-6}.

The "viscous fingering" phenomena is simulated on the Hyperion cluster which consists of 368 computation nodes of 2 quad-core Nehalem EX processors at 2.8 GHz with 8 MB of cache per processor. The MPI version installed on the cluster is the MPT-2.04, a specific version of MPI optimized for SGI clusters. Simulations were performed from 16 (the reference) to 1024 cores. The cluster was charged during the simulations and allocates randomly computation nodes, whose 8 cores are fully dedicated to the requested task. The total simulation, i.e. 4000 s of physical time, takes around 700 h of CPU time. The speedup σ for a simulation with n cores is computed as follows:

$$\sigma_n = \frac{T_{16}}{T_n} \tag{58}$$

where T_n is the computation time for n cores. The parallel efficiency ϵ_n is defined:

$$\epsilon_n = \frac{16}{n}\sigma_n. \tag{59}$$

The results in term of speedup are presented in Fig. 9. The numerical results show a super linear speedup until 256 cores ($\epsilon_{256} \approx 1.58$). This is not an unusual behavior since it has already been observed on standard OpenFOAM applications, see for example the 3D cavity flow simulations performed by the IT Center For Science [39]. In that study, Navier–Stokes equations are solved on 10 millions computation cells and the parallel efficiency can reach $\epsilon \approx 1.6$. In our case, we assume that the explicit part of the resolution (saturation equation and the flow properties computation at each time step) can partly explain this observation. Indeed, increasing the number of processors decreases the number of computation cells and then RAM memory necessary for each processor. As the explicit treatment need low computation but high access memory (RAM and cache), this could lead to a high parallel efficiency superior to 1. This effect decreases above 256 cores, because partial linear systems become smaller and the information exchange between cores takes relatively more computation time. A linear scaling is reached for 512 cores ($\epsilon_{512} \approx 0.97$). Then, the parallel efficiency for 1024 cores is low ($\epsilon_{1024} \approx 0.59$) because the linear system for each core become very small (only 1560 cells per core).

Parallel efficiency has also been tested using the OpenFOAM® multi-grid solver (GAMG) on shorter simulations ($t_{start} = 1000$ to

$t_{end} = 1010$ s) from 16 to 512 cores. For the reference case (16 cores), the GAMG solver reduces the global computational time of the simulation from 679 s (PCG) to 436 s (GAMG). The GAMG solver exhibits also a super linear speed-up until 256 cores but lower than the PCG solver (respectively $\epsilon_{256} \approx 1.09$ and $\epsilon_{256} \approx 1.58$). This results in a similar computational time for both solvers ($T_{256} \approx 26$ s) when using 256 processors and slower simulations above ($T_{512} \approx 19$ s for GAMG and $T_{512} \approx 14$ s for PCG). Note that the optimization of the GAMG solver parametrization may improve global efficiency and should require a thorough study.

5. Conclusion

A toolbox for the simulation of multiphase flow in porous media has been developed using the standards of OpenFOAM®. This toolbox includes libraries for porous models (relative permeability, capillary pressure and phase model) and a specific porous boundary condition. A classical IMPES solver has been developed to validate the provided models by comparison with analytical solutions. A study on the parallel efficiency (up to 1024 cores) has also been performed on a complex multiphase flow. The presented solver shows a satisfactory speedup, provided to solve a sufficiently large problem. The provided solver can serve as a basis to develop other features, such as new multiphase or improved wellbore models. Moreover, the easily modifiable nature of the OpenFOAM® platform can be useful to test, for example, new numerical schemes or solution methods.

Acknowledgments

This work was granted access to the HPC resources of CALMIP under the allocation 2013-p13147. We are grateful to Dr. Michel Quintard for interesting discussions.

References

[1] D. Das, S. Hassanizadeh, Upscaling multiphase flow in porous media: from pore to core and beyond, 2005.
[2] B. Flemisch, M. Darcis, K. Erbertseder, B. Faigle, A. Lauser, K. Mosthaf, S. Mthing, P. Nuske, A. Tatomir, M. Wolff, et al., Dumu^x: Dune for multi-{phase, component, scale, physics, …} flow and transport in porous media, Adv. Water Resour. 34 (9) (2011) 1102–1112.
[3] K. Lie, S. Krogstad, I. Ligaarden, J. Natvig, H. Nilsen, B. Skaflestad, Open-source matlab implementation of consistent discretisations on complex grids, Comput. Geosci. 16 (2) (2012) 297–322.
[4] O. Kolditz, S. Bauer, L. Bilke, N. Böttcher, J. Delfs, T. Fischer, U. Görke, T. Kalbacher, G. Kosakowski, C. McDermott, C. Park, F. Radu, K. Rink, H. Shao, H. Shao, F. Sun, Y. Sun, A. Singh, J. Taron, M. Walther, W. Wang, N. Watanabe, Y. Wu, M. Xie, W. Xu, B. Zehner, Opengeosys: an open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (thm/c) processes in porous media, Environ. Earth Sci. 67 (2) (2012) 589–599.
[5] P.C. Lichtner, G.E. Hammond, C. Lu, S. Karra, G. Bisht, B. Andre, R.T. Mills, J. Kumar, 2013, PFLOTRAN Web page, http://www.pflotran.org.
[6] H. Jasak, Error analysis and estimation for the finite volume method with applications to fluid flows, 1996.
[7] H.G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys. 12 (1998) 620.
[8] P. Forchheimer, Wasserbewegung durch boden, Z. Ver. Deutsch. Ing.
[9] F. Piscaglia, A. Montorfano, A. Onorati, Multi-dimensional computation of compressible reacting flows through porous media to apply to internal combustion engine simulation, Math. Comput. Model. 52 (7–8) (2010) 1133–1142.
[10] V. Novaresio, M. García-Camprubí, S. Izquierdo, P. Asinari, N. Fueyo, An open-source library for the numerical modeling of mass-transfer in solid oxide fuel cells, Comput. Phys. Commun. 183 (1) (2012) 125–146.
[11] P. Higuera, J.L. Lara, I.J. Losada, Three-dimensional interaction of waves and porous coastal structures using OpenFOAM®. Part I: Formulation and validation, Coast. Eng. 83 (2014) 243–258.
[12] P. Higuera, J. Lara, I. Losada, Three-dimensional interaction of waves and porous coastal structures using OpenFOAM®. Part II: Application, Coast. Eng. 83 (2014) 259–270.
[13] M. Muskat, Physical Principles of Oil Production, McGraw-Hil Edition, New York, 1949.
[14] K. Aziz, A. Settari, Petroleum reservoir simulation, 1979.
[15] M. Gerritsen, L. Durlofsky, Modeling fluid flow in oil reservoirs, Annu. Rev. Fluid Mech. 37 (2005) 211–238.
[16] Z. Chen, G. Huan, Y. Ma, Computational Methods for Multiphase Flows in Porous Media, 2006.
[17] H. Cao, Development of techniques for general purpose simulators, (Ph.D. thesis), Stanford University, 2002.
[18] R. Brooks, A. Corey, Hydraulic Properties of Porous Media, in: Hydrol. Pap., 1964.
[19] M.T. Van Genuchten, A closed-form equation for predicting the hydraulic conductivity of unsaturated soils1, Soil Sci. Soc. Am. J. 44 (5) (1980) 892.
[20] M.C. Leverett, Capillary behavior in porous solids, Trans. AIME 142 (1) (1940) 152–169.
[21] J.W. Sheldon, B. Zondek, W.T. Cardwell, One-dimensional, incompressible, non-capillary, two-phase fluid flow in a porous medium, T. SPE. AIME 216 (1959) 290–296.
[22] P.A.C. Raats, A. Klute, Transport in soils: the balance of momentum, Soil Sci. Soc. Am. J. 32 (1968) 452–456.
[23] P. Baveye, G. Sposito, The operational significance of the continuum hypothesis in the theory of water movement through soils and aquifers, Water Resour. Res. 20 (5) (1984) 521–530.
[24] H. Rusche, Computational fluid dynamics of dispersed two-phase flows at high phase fractions (Ph.D. thesis), 2002.
[25] K. Coats, IMPES stability: selection of stable timesteps, SPE J. 8 (2) (2003) 181–187.
[26] K. Coats, IMPES stability: The CFL limit, SPE J. 8 (3) (2003) 291–297.
[27] C. Preux, F. McKee, Study and approximation of IMPES stability: the CFL criteria, Finite Vol. Complex Appl. VI Probl. Perspect. Springer Proc. Math. 4 (2011) 713–721.
[28] D. Peaceman, Representation of a horizontal well in numerical reservoir simulation, SPE Adv. Tech. Ser. 1 (1) (1993) 7–16.
[29] S. Buckley, M. Leverett, Mechanism of fluid displacement in sands, Trans. AIME 146 (1942) 107–116.
[30] G.M. Homsy, Viscous fingering in porous media, Annu. Rev. Fluid Mech. 19 (1987) 271–311.
[31] J. Chen, D. Wilkinson, Pore-scale viscous fingering in porous-media, Phys. Rev. Lett. 55 (1985) 1892–1895.
[32] R. Lenormand, E. Touboul, C. Zarcone, Numerical models and experiments on immiscible displacements in porous media, J. Fluid Mech. 189 (1988) 165–187.
[33] D. Brock, F. Orr Jr., Flow visualization of viscous fingering in heterogeneous porous media, SPE J. 22614 (1991) 211–217.
[34] S. Doorwar, K. Mohanty, Viscous fingering during non-thermal heavy oil recovery, in: SPE Annu. Tech. Conf. Exhib, Denver, Colorado, USA, 2011.
[35] C. Chen, E. Meiburg, Miscible porous media displacements in the quarter five-spot configuration. Part 1. The homogeneous case, J. Fluid Mech. 371 (1998) 233–268.
[36] C. Chen, E. Meiburg, Miscible porous media displacements in the quarter five-spot configuration. Part 2. Effects of heterogeneities, J. Fluid Mech. 371 (1998) 269–299.
[37] A. Riaz, H.a. Tchelepi, Numerical simulation of immiscible two-phase flow in porous media, Phys. Fluids 18 (1) (2006) 014104.
[38] A. Ferrari, I. Lunati, Direct numerical simulations of interface dynamics to link capillary pressure and total surface energy, Adv. Water Resour. 57 (2013) 19–31.
[39] CSC IT Center for Science, http://www.csc.fi/english/research/sciences/CFD/CFDsoftware/openfoam/ofpage.

Please cite this article in press as: P. Horgue, et al., An open-source toolbox for multiphase flow in porous media, Computer Physics Communications (2014), http://dx.doi.org/10.1016/j.cpc.2014.10.005