PORE-SCALE SIMULATION OF MASS TRANSFER ACROSS IMMISCIBLE INTERFACES

A report submitted to the department of
Energy Resources Engineering
of Stanford University
in partial fulfillment of the requirements
for the degree of Master of Science

Marguerite Graveleau
June 2016

I certify that I have read this report and that in my opinion it is fully adequate, in scope and in quality, as partial fulfillment of the degree of Master of Science in Energy Resources Engineering.

(Prof. Hamdi Tchelepi) Principal Adviser

Abstract

Interphase mass transfer in porous media involving multiple fluid phases is a fundamental process that appears in a large number of situations of applied science and engineering including the injection and sequestration of CO2 into the sub-surfaces, the aquifer contamination by non-aqueous phase liquids (NAPL) and the primary migration of bitumen in petroleum reservoirs. In those situations, two immiscible phases share the pore-space, oil and water, gas and liquid, etc. One component is miscible in both and able to cross the interface, a tracer, a polluting chemical component or a gas dissolving into the liquid phase for example. Quantifying this mass transfer allows not only to predict the mass loss of one phase towards the other but also to understand and model its possible effects on the flow, such as a change in wettability, physical properties of the fluids, dissolution of a gas.

However, mass exchange has proven difficult to predict. Indeed, it is highly dependent on the physical properties of both fluids and solid, but also on the interfacial area between the phases. It is, therefore, essential to be able to predict the two-phase flow, depending on wettability, solid topology, fluid injection, and previous conditions. In particular, in the processes mentioned earlier such that CO_2 sequestration the flow in the subsurface does not necessarily obey classical laws at the reservoir scale but gravity, viscous, and capillary instability leading to fingering has been observed. That is why in this work we go back to the pore-scale to understands the underlying phenomena affecting the mass transfer and the flow.

A solver was implemented to simulate two-phase flow at the pore scale, with a miscible component crossing the interface, with or without phase change. It was based on an existing computational fluid dynamic (CFD) software: OpenFoam. The two immiscible phases are modeled under the Volume-of-fluid (VOF) formulation. The concentration of the miscible component is treated consistently with the VOF approach following Haroun's guidelines [1], and extended to handle contact line fluid/fluid/solid. The phase change is also implemented.

The species transport solver gives results very consistent with analytical solutions, and allows calculation of mass transfer coefficient in complex porous media. The phase change solver only gives preliminary results but shows good behavior in the case of the dissolution of a droplet in a surrounding fluid.

Acknowledgements

These two years as a master student have a been a wonderful experience. I feel indebted to many people who have helped and encouraged me throughout, and I would like to take this opportunity to thank them.

I would like to start by thanking my advisor, Hamdi Tchelepi, for giving me the opportunity to dive into the world of research. He has been a source of support and inspiration these past two years. His insight has set a new standard for me in the way science is to be taught, and how research should be approached.

A very special thank goes to Cyprien Soulaine. He has been a constant source of support, encouragement, knowledge and insight, and has made me discover the world of computational fluids dynamics. His patient stewardship has kept me on track for the past two years and this work would not have been possible without him.

This work has been financially supported by TOTAL and the STEMS project, which contributions are gratefully acknowledged.

The SUPRI-B team, and in particular the members of the pore-scale community within Stanford have been a great source of help, interesting and helpful discussion, seeds to new ideas for my work.

Finally, I would like to thank the OpenFOAM community over the world, which is very dedicated to making this software and open tool to everyone, through an opensource code, but also support and help to any user.

Contents

List of Figures

Chapter 1

Introduction

Interphase mass transfer in porous media involving multiple fluid phases is a fundamental process that appears in a large number of situations of applied science and engineering including the injection and sequestration of CO2 into the sub-surfaces, the aquifer contamination by non-aqueous phase liquids (NAPL) and the primary migration of bitumen in petroleum reservoirs. In all these processes, two immiscible phases share the pore space. Their distribution in the domain is process dependant involving complex configurations such as the entrapment of one of the phases by the flowing phase or the formation of thin films. Molecules may cross the interface that separates the different fluid phases. For example, the migration of bitumen produces important changes in composition and crude oils become progressively more paraffinic with increasing distance of migration [3]. According to the situation, this mass transfer may have different degrees of implication, ranging from a simple change in the fluids composition to stronger impacts on the flow properties both locally and at very large scales. For example, once the supercritical CO2 is injected into the Earth's subsurface, it flows as a separate phase forming an immiscible interface with the brine already in place. Complex capillary mechanisms, mostly governed by the wettability of the mineral surface, lead to the trapping of CO2 ganglia in the pore space. The CO2 from the supercritical phase dissolves in the aqueous phase to form carbonic acid which lowers the pH of the brine as the carbonic acid dissociates to the bicarbonate ion [2, 4]. The acid ions are then transported by advection-diffusion to the mineral surface where the dissolution and precipitation of the minerals might occur [5]. These pore scale processes associated with the injection and sequestration of CO2 into deep saline aquifer can completely reorganize the pore space which means that the rock permeability and porosity evolve and consequently impact the flow properties at larger scales [6]. Figure 1.1 presents a summary of the thermodynamic processes involved in the CO_2 sequestration. In our work, we will focus on the drainage/imbibition processes and the mass transfer occurring during the dissolution of one phase into the other.

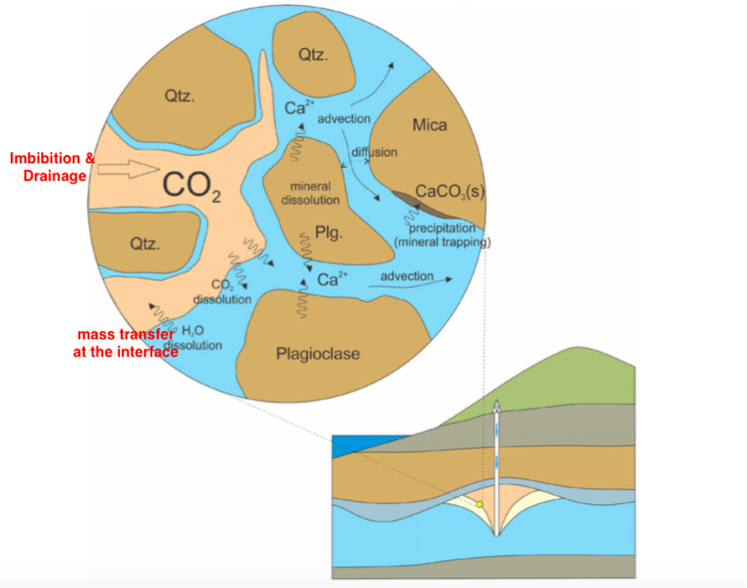


Figure 1.1: Processes involved in the sequestration of CO_2 (in red processes discussed in this thesis) [2]

1.1 State of the art

Significant efforts have been made to model the interphase mass transfer for subsurface processes, in particular, the most recent progress were related to the contamination of the water-tables from the NAPL dissolution [7, 8]. As it is commonly practiced for flow and transport in porous media, the interphase mass transfer can be investigated with different approaches, with a pore scale modeling approach where the solid skeleton of the porous structure and all the interfaces between the fluids are explicitly described, or with a physics based on macro-scale equations averaged over a Representative Elementary Volume (REV) of the porous medium. Although physical mechanisms remain the same for the two approaches, the mathematical tools used to represent the physics at these different scales may differ significantly. On the one hand, at the pore scale, the thermodynamics equilibrium of the system is defined as the equality of the chemical potential for each species at the interface between the phases. When this condition is violated, there is mass flux from one phase to one another to reach another thermodynamics equilibrium state. With REV-based approaches, on the other hand, non-local equilibrium models are often used to describe the multicomponent multiphase mass transfer and an exchange coefficient is introduced to quantify the mass exchange in the REV. These two modeling approaches are intrinsically related to each other and upscaling techniques such as volume averaging have been used to derive the macro-scale equations from the pore-scale physics [9, 10, 11, 12]. For example, Darcy's law has been derived from the Navier-Stokes equations via homogenization [13]. Those upscaling techniques, however, are only valid in a small range of physical and chemical properties. Global behavior of the flow in the porous medium highly depends on gravity effects, viscosity ratio and capillary number, flow regimes that are not necessarily modeled by the generalized Darcy's law at the larger scale. In particular Cinar et al [14] showed that CO_2 sequestration occurs under settings where one might encounter capillary percolation, fingering, of gravity instability (Figure 1.2).

Quantifying the mass transfer can be very complex. Most models define a mass exchange coefficient linking the flux of mass at the interface to the difference of concentration between the phases at the interface. Clearly, the mass exchange coefficient is a function of the interfacial area since it must estimate the rate of mass transfer of the compounds across the interface. This is a difficult data to assess since it depends on many factors including solid topology, mineral wettability and boundary conditions of the REV [15]. Indeed, the phase distribution may form very different patterns like fingering instabilities, ganglia of the non-wetting phase trapped by capillarity, or thin films formed by the wetting phase. The macro-scale expression of the interphase

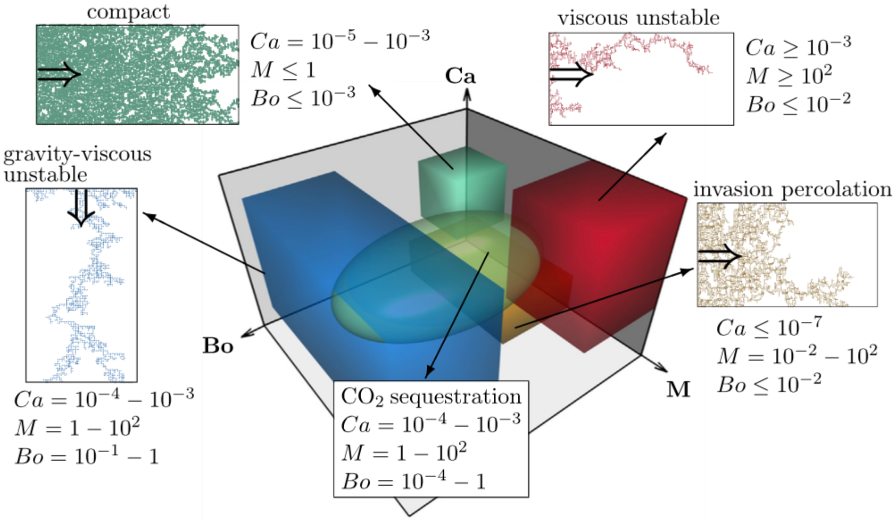


Figure 1.2: Flow regime depending on the capillary number Ca, the Bond number Bo and the viscosity ratio M

mass transfer may vary a lot according to these different situations. Hence, under thin films conditions, most of the mass transfer occurs from the film area. Various correlations developed based on ideal situations like the double film theory that postulates that the local mass transfers occur in a thin layer on each side of the interface [16], like the penetration model [17] or like the surface renewal theory [18] usually failed to predict the mass transfer across interface in sub-surface processes, mostly because of the complexity of flow in natural porous media. Hence, process-dependant correlations have been proposed from one-dimensional column experiments [19]. If the interfacial area is probably one of the most influential parameters on interphase mass transfer, many other parameters influence the process. It has been shown that the interphase mass transfer coefficient increases significantly with velocity since it affects the rate of renewal of the compounds at the interface. However, no universal law has been proposed yet to quantify this dependency.

During the last decades, there have been important improvements in the experimental and numerical techniques to get more insights, directly from the pore scale, about the multicomponent multiphase mass transfer mechanisms. The use of glass bead or micromodel experiments combined with image analysis allows a direct visualization of the different processes involved [20, 21, 22, 23, 24]. Dye is often used to map the concentration evolution of a component in the system. Powers et al. [25], however, reported that this could modify significantly the interfacial tension by 10 to 30% and may affect mass transfer rates as well. Moreover, particle image velocimetry technique allows high-resolution measurement of the velocity profile in micromodels and offer new possibilities to investigate the pore scale processes [26]. Magnetic resonance imaging is also used to acquire three-dimensional images of NAPL blobs during dissolution from columns packed with angular silica gel grains or spherical glass beads [27, 28]. The team of Dr. Tetsu Tokunaga and Dr. Jiamin Wan in Berkeley has conducted experiments where supercritical CO_2 is injected in a 2D porous medium made of cylindrical grains, initially filled with brine. After drainage and imbibition, the CO_2 phase is trapped. In a second time, the residual CO_2 dissolves into the brine (Figure 1.3).

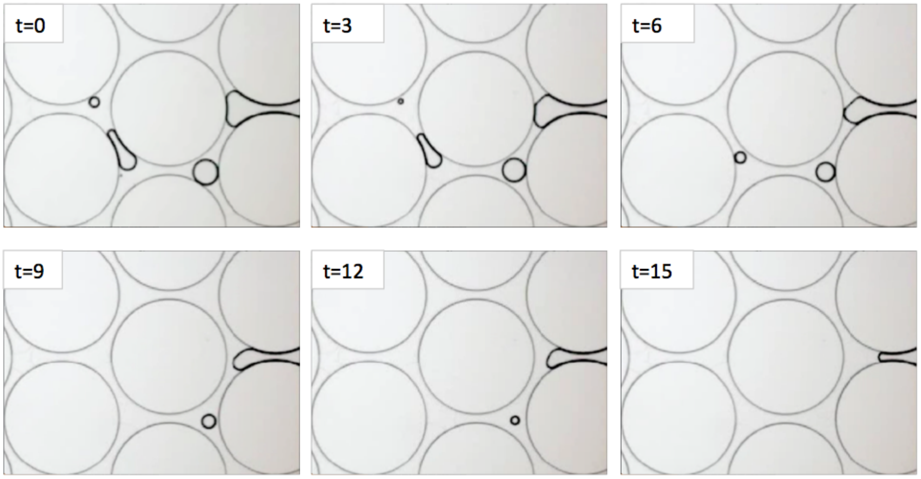


Figure 1.3: Dissolution of super-critical CO2 in brine in an artificial porous medium in unit time (Tetsu K. Tokunaga, Jiamin Wan)

The use of numerical models to simulate interphase mass transfer at the pore scale is not new. The advances in pore-scale simulators have closely followed the progress of small-scale experimental techniques and image analysis. The numerical models can be divided into the direct modeling approaches where Navier-Stokes or Boltzmann equations are directly solved in the void of the porous structure and into the pore network models (PNM) in which the porous medium is represented as a network of pore bodies and pore throats where the flow is ruled by Poiseuille's law. Most of the early progress for the pore-scale simulation of interphase mass transfer were related to NAPL dissolution using PNM [22, 29, 30, 31, 32, 33]. In these works, all the displacement mechanisms were not covered since it was assumed that the dissolving phase was immobile, trapped in the pore throats or bodies. Since in the case of NAPL dissolution most of the mass transfer occurs from the film area, the cylindrical bonds of the PNM were replaced by a throat of rectangular cross section to account for the stagnant film residing in angular pores. Mass transfer models for pore-scale corner flow was developed from analytical solution [29, 34, 23] or finite elements simulation [35] and input to the PNM with various success. The direct modeling approaches, on the other hand, can relax the restrictions of the PNM, in particular, the approximation of the pore space geometry and the hypothesis that the dissolving phase is immobile.

Two challenges arise when simulating mass transfer. First, it is necessary to be able to simulate two-phase flow, which is a computational fluid dynamics problem. Second, the miscible component evolves in both phases, and concentration obeys particular boundary conditions at the interface. Several methods have been developed to simulate multiphase flow in the pore space. Lattice-Boltzmann simulations [36], Lagrangian mesh-free methods such as smoothed particle hydrodynamics (SPH) [37] do not require explicit and complicated interface tracking algorithms and are easily parallelized. Regarding transfer across interfaces, Lattice-Boltzmann methods (LBM) have been proposed for the simulation of multiphase mass transfer and reaction of dilute species [38, 39, 40, 41]. Tartakovsky et al. have developed a smoothed particle hydrodynamics (SPH) model to simulate the flow of mixtures of aqueous and non-aqueous phase liquids in porous media and the dissolution of the non-aqueous phase in the aqueous phase [37].

The use of interface capturing methods on Eulerian grid is, however, preferable for simulating multiphase flow, for their efficiency and ability to handle large density and viscosity ratios [42]. The direct solution of Navier-Stokes equations is performed using interface capturing methods such as Level Set (LS) or Volume of Fluid (VOF)

[43, 44]. Those methods have proven very efficient to simulate multiphase processes such as drainage or imbibition in complex pore-space [45, 46, 47, 48, 49, 50]. However, despite recent improvements in computational fluid dynamics (CFD) techniques, challenges remain as to model accurately the dynamics of the interface between two immiscible fluids in the pore space at low capillary numbers. One of them is the apparition of parasitic velocities at the vicinity of the interface due to an inaccurate computation of the curvature. In capillary driven flows, those spurious currents can become significantly stronger than the average velocity in the phases, and lead to complete unphysical displacement of the interface. Smoothing techniques have been applied to reduce those parasitic velocities [51, 52], but may by themselves affect the behavior of the interface. This issue is still a topic of active research for two-phase flow at low capillary number [53]. Another challenge if the two-phase flow simulation is the implementation of the contact line fluid/fluid/solid. The visible contact angle does not correspond to the nano-scale contact angle but is scaled up [52], since a refinement to the nanoscale is not computationally feasible for the pore-space.

Another difficulty is to transport a concentration field in the system. Until recently, numerical studies were restricted to the case where concentration field was continuous at the interface. Thermodynamics equilibrium states that chemical potentials for the miscible component are equals on both sides of the interface are equal. It is classically translated into a partitioning coefficient through Henry's law. Numerical methods have been developed for the discontinuous case (coefficient different from one) by artificially ensuring the equality of fluxes [54, 55] . Haroun et al. [1] proposed a robust formulation recently referred to as Continuum Species Transfer (CST) formulation [56, 57] to treat the jump discontinuity consistently with the VOF approach while satisfying the continuity of the mass flux across the interface. This technique has been applied with success to simulate the mass transfer in liquid film flowing along corrugated surfaces [1, 58, 59]. The approach, however, excluded the presence of triple lines at the solid walls.

1.2 Thesis outline

The objective of this work is to implement a solver able to simulate mass transfer across immiscible interfaces in the pore-space. The model is an extension of the CST technique [1] to simulate sub-surface processes with moving contact line and phase change. The thesis is organized as follow. In Chapter 2 we define the physics associated with two immiscible phases sharing the pore-space, and the thermodynamics of a miscible component able to cross the interface. Chapter 3 describes how, from this physical model, a numerical model is derived, consistent with a Volume-of-fluid formulation. Its implementation using the software OpenFoam and the algorithm used are presented in Chapter 4. Finally, various cases are simulated in Chapter 5, including drainage in a porous medium, mass transfer coefficient estimation, and growth of a droplet due to phase change induced by diffusion.

Chapter 2

Physical Model

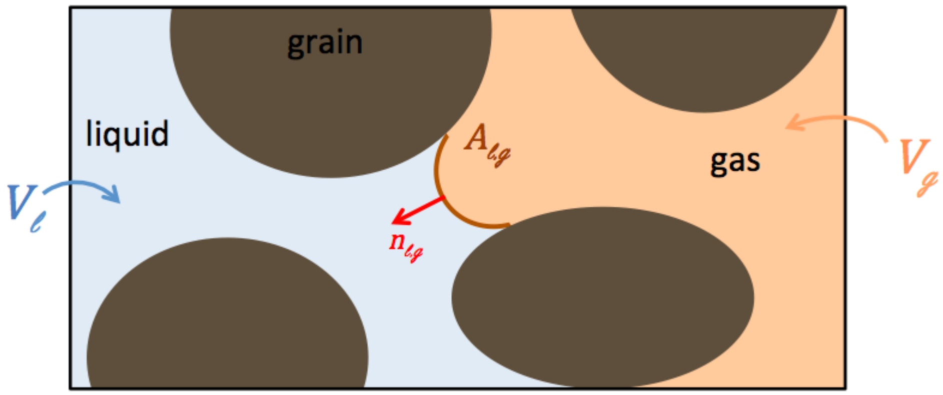


Figure 2.1: Illustration of the porous medium with two immiscible phases

At the pore-scale, solid and fluids are clearly separated, we consider the solid to be completely still and inert (no reaction with the flow) and we are interested in the evolution of the fluid phases in the pore space (Figure 2.1).There are two aspects, on the first hand two immiscible phases share the pore-space and flow, for example, gas and liquid. Their motion is governed by mass balance equations with the assumption of incompressibility of the two phases, and Navier-Stokes momentum balance equations, which are presented in the first section. On the other hand, we consider a chemical component miscible in both phases, for example, a tracer, or a chemical pollution affecting the subsurface flow.

The objective of this chapter is to describe the physical phenomena of interest and the thermodynamic hypotheses made, leading to the governing equations and boundary conditions of the investigated physics.

2.1 Two-phase flow

We consider two immiscible phases, referred to as phases 1 and 2, or liquid l and gas g. We also assume that they are incompressible so that their densities are constant. We define V_i the volume occupied by the fluid i, and A_{12} the interface between the fluids (see Figure 2.2). Two principles lead us to the governing equations and boundary conditions of our system: mass conservation, and momentum equilibrium.

2.1.1 Phase tracking

Phase i can be tracked with the indicator function $\mathbb{1}_i$. For phase 1 this gives:

$$\mathbb{1}_1 = \begin{cases} 1 & \text{in phase 1 (usually the wetting phase, typically liquid)} \\ 0 & \text{in phase 2 (usually gas)} \end{cases} . \tag{2.1}$$

With this definition, this function satisfies the following properties, that are useful when deriving further relations

$$\begin{aligned} \mathbb{1}_1(1 - \mathbb{1}_1) &= 0, \\ \mathbb{1}_1^2 &= \mathbb{1}_1. \end{aligned} \tag{2.2}$$

In each phase, the fluid obeys the classical thermodynamics laws described in the following section, and the boundary conditions at the interface between the fluids and with the solid.

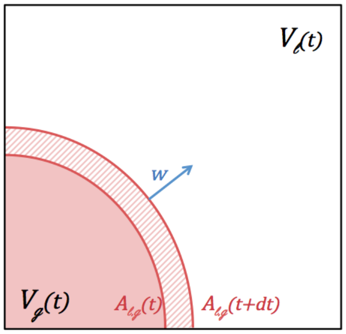


Figure 2.2: Evolution of the volume of liquid $V_l(t)$ in a cell

2.1.2 Mass conservation

Equation

We work with isothermal condition at the scale of our domain, and consider the fluids to be incompressible. Consequently, their densities ρ and viscosities μ are constant. Mass conservation of each phase writes:

$$\frac{\partial \rho_i}{\partial t} + \nabla.\left(\rho_i \mathsf{v}_i\right) = 0 \quad \text{in } V_i \text{ with } i = 1, 2, \tag{2.3}$$

where v_i is the velocity of the fluid i. Under the incompressibility assumption ($\rho_i =$ cst), this becomes:

$$\nabla.\mathsf{v}_i = 0 \quad \text{in } V_i \text{ with } i = 1, 2. \tag{2.4}$$

Interface condition

At the interface, mass conservation writes:

$$\rho_1 \left(\mathsf{v}_1 - \mathsf{w}\right).\mathbf{n}_{12} = \rho_2 \left(\mathsf{v}_2 - \mathsf{w}\right).\mathbf{n}_{12} \quad \text{on } A_{12}, \tag{2.5}$$

where w is the speed of the interface (Figure 2.2), and $\mathbf{n}_{12}$ is the normal to the interface, pointing from phase 1 to phase 2.

Boundary condition with the solid

The solid begin inert, there is no dissolution or deposition, and mass conservation at the boundary with the solid is:

$$\mathbf{n}_s.\mathsf{v}_i = 0, \tag{2.6}$$

where $\mathbf{n}_s$ is the normal to the solid.

2.1.3 Momentum equilibrium

Equation

In each phase i the momentum equilibrium holds, under the classical Navier-Stokes formulation:

$$\rho_i \frac{\partial \mathsf{v}_i}{\partial t} + \rho_i \mathsf{v}_i.\nabla \mathsf{v}_i = -\nabla p_i + \rho_i \mathbf{g} + \nabla.\left(\mu_i \left(\nabla \mathsf{v}_i +^t \nabla \mathsf{v}_i\right)\right) \quad \text{in } V_i \text{ with } i = 1, 2, \tag{2.7}$$

where p is the pressure, μ the viscosity, and $\mathbf{g}$ the gravity. The left hand side corresponds to the inertia. In our setting in porous media, it is usually be negligible, however we keep it in order to derive a general formulation. The right hand side includes the pressure forces, the effects of gravity, and the viscous forces. We consider the hydrostatic pressure $p_i' = p_i - \rho_i z g$ to simplify the equation, the gravity is therefore automatically taken into account in the pressure, even though in our domain size and settings, gravity effects are negligible.

Interface condition

At the interface, the equilibrium of forces writes:

$$\begin{aligned} &\mathbf{n}_{12}.\left(-\rho_1 \left(\mathsf{v}_1 - \mathsf{w}\right) \mathsf{v}_1 - \mathsf{I} p_1 + \mu_1 \left(\nabla \mathsf{v}_1 +^t \nabla \mathsf{v}_1\right)\right) \\ &\qquad = \mathbf{n}_{12}.\left(-\rho_2 \left(\mathsf{v}_2 - \mathsf{w}\right) \mathsf{v}_2 - \mathsf{I} p_2 + \mu_2 \left(\nabla \mathsf{v}_2 +^t \nabla \mathsf{v}_2\right)\right) + H_{12} \sigma \mathbf{n}_{12} \quad \text{on } A_{12}, \end{aligned} \tag{2.8}$$

where $\mathbf{n}_{12}$ is the normal to the interface fluid/fluid, I is the unity tensor, $H_{12} = -\nabla.\mathsf{n}_{12}$ [60] is the curvature of the interface, and σ is the surface tension. In all the following, multiplication of vector correspond to the dyadic product $\mathsf{uu} = \mathsf{u} \otimes \mathsf{u} = \mathsf{uu}^t$. The forces at stake are dissipation, pressure, and viscous forces on both side, plus the interfacial tension term.

Boundary condition with the solid

The solid is not moving, its velocity is zero, and by continuity $\mathbf{t}_s.\mathsf{v}_i = 0$ where $\mathbf{t}_s$ is the tangent to the solid. Combined with the boundary condition due to mass conservation (Eq. 2.6), we obtain the no slip condition at the boundary with the solid:

$$\mathsf{v}_i = 0 \quad \text{for } i = 1, 2, \text{ on } \Gamma_s, \tag{2.9}$$

where Γ_s is the boundary with the solid.

Contact angle

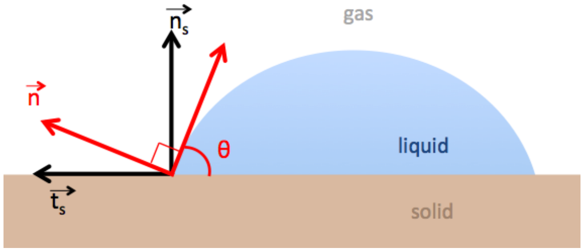


Figure 2.3: Schemetic of a triple point solid/fluid/fluid. The contact angle θ is the one formed by the tangent to the interface with the solid.

At the triple point liquid/gas/solid, the contact angle is defined as in Figure 2.3. Its value will define the wetting fluid of the setting ($\theta < 90°$ in the wetting fluid). In the example of a droplet on a table surrounded by gas, if the solid is hydrophilic the droplet spreads and θ is very small, whereas is the solid is highly hydrophobic, the droplet remains as spherical as possible and θ is very close to 180°. The contact angle depends on the solid composition and surface but also on both fluids properties. Imposing a contact angle θ_0 is equivalent to imposing the normal to the interface at the solid boundary:

$$\mathbf{n}_{12} = \cos\theta_0 \mathbf{n}_s + \sin\theta_0 \mathbf{t}_s \quad \text{on } \Gamma_s \cap A_{12} \tag{2.10}$$

where $\mathbf{n}_{12}$ the normal to the interface, $\mathbf{n}_s$ is the normal to the solid, and $\mathbf{t}_s$ the tangent to the solid, therefore both independent of the flow and constant.

However, a simple contact angle is a very simplistic representation. Contact line dynamics go back to the nanoscale, where molecules of fluids are represented [61]. Such refinement is not computationally manageable in CFD simulation, and the contact angle is scaled-up to the numerical scale, to obtain the visible contact angle. Even at that scale, there can be effects of the molecular dynamic, like hysteresis, or velocity-dependent contact angle referred as dynamic contact angle [62]. Although it can be modeled, we make the hypothesis of a constant contact angle $\theta_0 =$cst.

2.2 Transport of concentration

2.2.1 Concentration

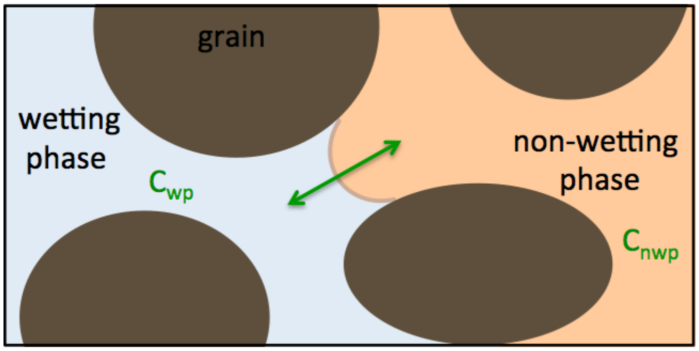


Figure 2.4: Illustration of the porous medium with two immiscible phases and a miscible component

The last element of the model is a miscible component in each phase. This can account for example for a colored tracer in multiphase flow: the two phases are immiscible, however, the tracer is present in both phase. It is also the case when CO_2 dissolves into brine, although the gas and liquid phases are immiscible, CO_2 exists both in the gas or supercritical phase and in the brine as a dissolved species. To account for this component A, we introduce as a new variable its concentration $C_{i,A}$

in phase i. In each phase i the concentration is governed by the classical advection-convection equation:

$$\frac{\partial C_{i,A}}{\partial t} + \nabla.\left(\mathsf{v}_i C_{i,A}\right) = \nabla.\left(D_{i,A} \nabla C_{i,A}\right) \quad \text{in } V_i \text{ with } i = 1, 2, \tag{2.11}$$

where $D_{i,A}$ is the molecular diffusion coefficient of A in phase i. The diffusion term in the right-hand side is a direct result of Fick's first law. As described by Taylor and Krishna [63] for multicomponent mass transfer, the generalized Fick's law states that the diffusion of a component depends on the gradients of concentration of the other components present in the mixture. Here, all these gradients are lumped into a single term, where D_A corresponds to the diffusivity of the species A into the mixture. It varies with temperature and the inverse of the pressure [63]. We work under isothermal conditions, and we assume that in the range of pressure we consider, the diffusion coefficient remains constant.

2.2.2 Interface conditions

At the interface, mass conservation imposes continuity of mass fluxes on each side. Mass fluxes is either due to the diffusion of the component, or its velocity, and the equality of fluxes at the interface gives:

$$\left(C_{1,A}\left(\mathsf{v}_1 - \mathsf{w}\right) - D_{1,A} \nabla C_{1,A}\right).\mathbf{n}_{12} = \left(C_{2,A}\left(\mathsf{v}_2 - \mathsf{w}\right) - D_{2,A} \nabla C_{2,A}\right).\mathbf{n}_{12} \quad \text{on } A_{12}. \tag{2.12}$$

Additionally thermodynamics equilibrium imposes equality of chemical potential at the interface:

$$\mu_1 = \mu_2 \quad \text{on } A_{12}. \tag{2.13}$$

This latter condition is usually described by a partitioning relation such as Henry or Raoult laws (Figure 2.5), stating that concentration in the liquid phase is proportional to the partial pressure of the species in the gas phase. Boundary condition from Eq. 2.13 becomes:

$$C_{1,A} = H_A C_{2,A} \quad \text{on } A_{12}, \tag{2.14}$$

where H_A is the partitioning coefficient or Henry's constant. Under the isothermal hypothesis, however, it considered constant in this work. When this condition is not fulfilled, there is a mass transfer from one phase to one another in order to reach the thermodynamics equilibrium.

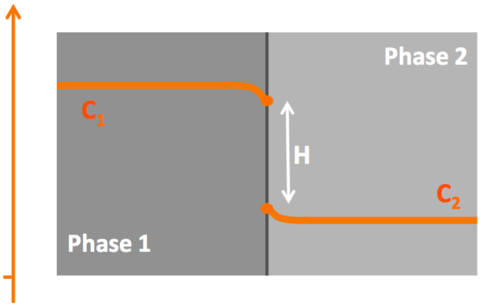


Figure 2.5: Schematic of the concentration profile at the interface, illustration of Henry's law with the jump of concentration

2.2.3 Boundary condition with the solid

The solid being inert (no chemical reaction like deposition of dissolution), the boundary condition with the solid for the concentration is:

$$\mathbf{n}_s.\nabla C_{i,A} = 0 \text{ on } A_{is} \text{ with } i = 1, 2, \tag{2.15}$$

where $\mathbf{n}_s$ is the normal to the solid, and A_{is} is the solid surface in contact with phase i.

Chapter 3

Numerical Model

In the previous chapter, the governing equation of the physical model we want to simulate has been described. The numerical model differs from the physical model in the sense that it is an approximation. In the physical model, variables like pressure and velocity are continuously defined over the entire pore-space domain. In a numerical model, the pore-space is meshed, and variables are discrete, defined only at the center of the cell, or on faces or corners. They are averages of the physical variables in the cell.

There are several existing techniques to model two-phase flow at the pore scale. We chose to use the Volume of fluid method (VOF), for it is a mass-conservative formulation. This chapter presents the derivation of the VOF equations governing the numerical variables, from the conservation laws presented in the previous chapter.

3.1 Variables of the model

The mathematical description introduced in Chapter 2 supposes to know at every spatial coordinates the quantities of interest, which we numerically can not afford since work on a meshed space. To derive the conservation laws valid on the discretized space, we need first to define cell average operators. For a variable β_i with value in phase i, we distinguish two average operator: a superficial average operator $\bar{*}$ and an intrinsic phase operator, $\overline{*}^i$. The first one is defined as,

$$\overline{\beta_i} = \frac{1}{V}\int_{V_i} \beta_i dV, \tag{3.1}$$

and the second one as,

$$\overline{\beta_i}^i = \frac{1}{V_i}\int_{V_i} \beta_i dV, \tag{3.2}$$

where V is the cell volume and V_i is the volume occupied by the phase i in the cell. Both are related by the simple relation, $\overline{\beta_i} = \frac{V_i}{V}\overline{\beta_i}^i$.

The operator $\bar{\beta}$ defines the average of quantity β_i in a cell of the grid, it takes a physical quantity continuously defined in the pore-space and creates a discrete numerical variable defined at the center of the cell (Figure 3.1). Pressure, velocities, and other variables are kept track of at the center of each cell.

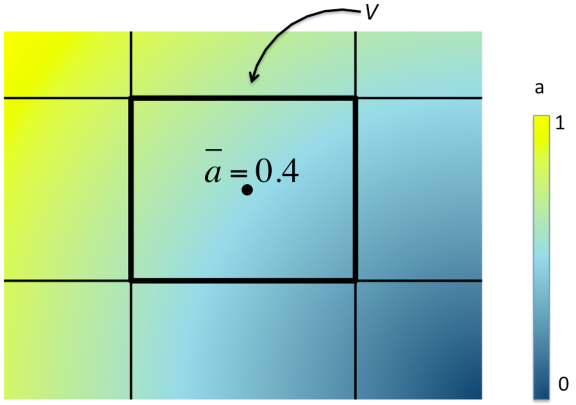


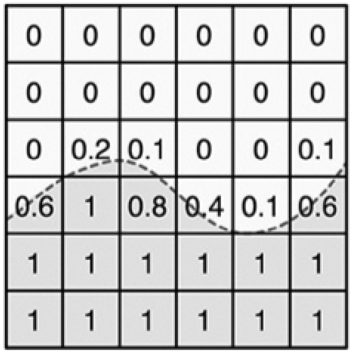
Figure 3.1: Averaging variable a in a cell of volume V to obtain numerical variable $\bar{a}$

3.1.1 Phase indicator α

In the VOF technique, the phase distribution in the domain is track by the volume fraction, α, of liquid in every cell. It is defined as the averaged value of $\mathbb{1}_1$:

$$\alpha = \overline{\mathbb{1}_1} = \frac{V_1}{V}. \tag{3.3}$$

With this definition, α is now accounting for the volume of phase 1 in each cell



0	0	0	0	0	0
0	0	0	0	0	0
0	0.2	0.1	0	0	0.1
0.6	1	0.8	0.4	0.1	0.6
1	1	1	1	1	1
1	1	1	1	1	1

Figure 3.2: α values per cell

(Figure 3.2) and is called the phase indicator function. It can be referred to as α_l for the liquid (resp. α_g for the gas), and we have $\alpha_l + \alpha_g = 1$. By definition $\nabla\alpha \neq 0$ only the cells occupied by the interface and their direct neighbors.

$$\alpha = \begin{cases} 1 & \text{if the cell is occupied by liquid} \\ 0 < \alpha < 1 & \text{if the cell is crossed by the interface} \cdot \\ 0 & \text{if the cell is occupied by gas} \end{cases} \tag{3.4}$$

As represented in Figure 3.3, the value and the gradient of α are defined at the center of each cell, whereas the normal to the interface n_{12} is a physical quantity defined everywhere at the interface. Averaging theorems give us a direct relation between the gradient of the phase indicator and the average normal to the interface [64]:

$$\nabla\alpha = -\frac{1}{V}\int_{A_{12}} \mathbf{n}_{12} dA. \tag{3.5}$$

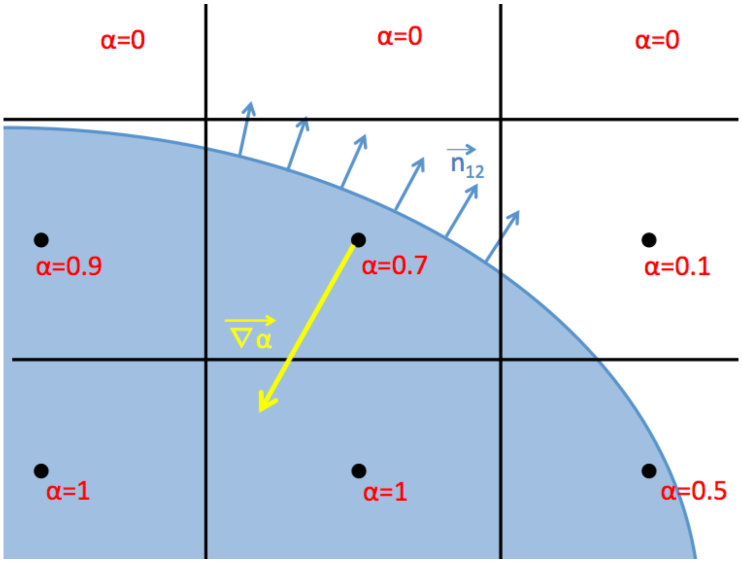


Figure 3.3: Illustration of the values of α and $\nabla\alpha$

3.1.2 Global variables

Variables of the numerical models are the global variables: average over the entire cell of a quantity. They are defined as:

$$\begin{aligned}
&\bar{\mathsf{v}} = \frac{1}{V}\int_V \mathsf{v}dV = \alpha\bar{\mathsf{v}}_1^1 + (1-\alpha)\,\bar{\mathsf{v}}_2^2 = \bar{\mathsf{v}}_1 + \bar{\mathsf{v}}_2,\\
&\bar{p} = \frac{1}{V}\int_V pdV = \alpha\bar{p}_1^1 + (1-\alpha)\bar{p}_2^2 = \bar{p}_1 + \bar{p}_2,\\
&\bar{C} = \frac{1}{V}\int_V CdV = \alpha\bar{C}_1^1 + (1-\alpha)\bar{C}_2^2 = \bar{C}_1 + \bar{C}_2,\\
&\rho = \alpha\rho_1 + (1-\alpha)\rho_2,\\
&\mu = \alpha\mu_1 + (1-\alpha)\mu_2.
\end{aligned} \tag{3.6}$$

For the definition of ρ and μ we use the fact that density and viscosity are constant in a phase, therefore $\bar{\rho}_i^i = \rho_i$ and $\bar{\mu}_i^i = \mu_i$.

For all those variables, the same method is applied to derive an equation for the global variables : the physical equations are averaged over a numerical cell, for each phase, which introduces volumetric terms in the cell occupied by the interface fluid/fluid. Those terms are then simplified using the boundary conditions at the interface.

3.2 Volume-of-fluid formulation

The key idea of the Volume of fluid method is to create a model using only the global variables α, $\bar{p}$, $\bar{\mathsf{v}}$ and $\bar{\omega}$. Those variables are defined at the center of each cell, over the entire domain, and allow us to solve a global equation over the domain rather than an equation in each phase. The idea is thereofore to average the equation of the physical model over a cell volume. For a scalar quantity a_i and a vectorial quantity $\mathbf{b}_i$, both defined in phase i, the following relations hold when averaging over the volume of the cell [64]:

$$\begin{aligned}\overline{\nabla a_i} &= \nabla \bar{a}_i + \frac{1}{V}\int_{A_{ij}} a_i \mathbf{n}_{ij} dA,\\ \overline{\nabla.\mathbf{b}_i} &= \nabla.\bar{\mathbf{b}}_i + \frac{1}{V}\int_{A_{ij}} \mathbf{b}_i \mathbf{n}_{ij} dA,\\ \overline{\frac{\partial a_i}{\partial t}} &= \frac{\partial \bar{a}_i}{\partial t} - \frac{1}{V}\int_{A_{ij}} a_i \mathbf{n}_{ij}.\mathbf{w} dA,\end{aligned} \tag{3.7}$$

where $\mathbf{n}_{ij}$ is the normal from phase i to the other. These equations enable us to convert a relation valid at every point into a relation on the numerical variables. Boundary conditions at the interface can therefore be integrated into the integral term, and considered as volumetric terms in our equation, as we show in the derivation.

3.2.1 Mass conservation

Since fluids are incompressible, the mass of liquid in a cell is proportional to the volume of liquid : $M(t) = \rho_l V_l(t)$. We apply the average operator defined in Eq. 3.1

to the mass conservation Eq. 2.3. For phase 1 this gives:

$$\begin{aligned}
\overline{\frac{\partial \rho_1}{\partial t} + \nabla.\left(\rho_1 \mathsf{v}_1\right)} &= \bar{0} = 0 \\
\overline{\frac{\partial \rho_1}{\partial t} + \nabla.\left(\rho_1 \mathsf{v}_1\right)} &= \overline{\frac{\partial \rho_1}{\partial t}} + \overline{\nabla.\left(\rho_1 \mathsf{v}_1\right)} \\
&= \frac{\partial \overline{\rho_1}}{\partial t} - \frac{1}{V}\int_{A_{12}} \rho_1 \mathbf{n}_{12}.\mathsf{w} dV + \nabla.\left(\overline{\rho_1 \mathsf{v}_1}\right) + \frac{1}{V}\int_{A_{12}} \rho_1 \mathbf{n}_{12} \mathsf{v}_1 dV \\
&= \rho_1 \frac{\partial \alpha}{\partial t} + \nabla.\left(\rho_1 \bar{\mathsf{v}}_1\right) + \frac{1}{V}\int_{A_{12}} \rho_1 \mathbf{n}_{12}.\left(\bar{\mathsf{v}}_1 - \mathsf{w}\right) dV,
\end{aligned} \tag{3.8}$$

where we used $\bar{\rho}_1 = \frac{1}{V}\int_{A_{12}} \rho_1 dV = \frac{V_1}{V}\frac{1}{V_1}\int_{A_{12}} \rho_1 dV = \alpha \bar{\rho}_1^1 = \alpha \rho_1$. Symmetrically for phase 2 we have:

$$\overline{\frac{\partial \rho_2}{\partial t} + \nabla.\left(\rho_2 \mathsf{v}_2\right)} = 0 = \rho_2 \frac{\partial (1-\alpha)}{\partial t} + \nabla.\left(\rho_2 \bar{\mathsf{v}}_2\right) - \frac{1}{V}\int_{A_{12}} \rho_2 \mathbf{n}_{12}.\left(\bar{\mathsf{v}}_2 - \mathsf{w}\right) dV \quad , \tag{3.9}$$

because $\mathbf{n}_{12} = -\mathbf{n}_{21}$. The integral term accounts for the mass transfer between phases. We note $\dot{m}_{21}$ mass transfer from gas to liquid, and:

$$\dot{m}_{21} = \frac{1}{V}\int_{A_{12}} \rho_1 \mathbf{n}_{12}.\left(\bar{\mathsf{v}}_1 - \mathsf{w}\right) dV, \dot{m}_{12} = \frac{1}{V}\int_{A_{12}} \rho_2 \mathbf{n}_{21}.\left(\bar{\mathsf{v}}_2 - \mathsf{w}\right) dV. \tag{3.10}$$

The boundary condition at the interface (Eq. 2.5) gives $\dot{m}_{12} = -\dot{m}_{21}$. By convention we note $\dot{m} = \dot{m}_{21}$ the mass transfer to phase 1 (usually the liquid one). This mass transfer depends on the physics we want to solve, in the case of simple two phase flow there is no phase change and $\dot{m}$ is set to 0. In the case of ebulition for example, $\dot{m}$ will be driven by the temperature and the saturation pressure. We use the notation $\alpha_1 = \alpha$ and $\alpha_2 = (1-\alpha)$. Overall we obtain the mass conservation for both phases:

$$\begin{aligned}
\frac{\partial \alpha_1}{\partial t} + \nabla.\left(\alpha_1 \bar{\mathsf{v}}_1\right) &= \frac{\dot{m}}{\rho_1} \\
\frac{\partial \alpha_2}{\partial t} + \nabla.\left(\alpha_2 \bar{\mathsf{v}}_2\right) &= -\frac{\dot{m}}{\rho_2}
\end{aligned} \tag{3.11}$$

We recall that $\alpha_1 + \alpha_2 = \frac{V_1}{V} + \frac{V_2}{V} = 1$. By summing the two last equations, we obtain the global mass conservation equation for the global variable:

$$\nabla.\bar{\mathsf{v}} = \dot{m}\left(\frac{1}{\rho_1} - \frac{1}{\rho_2}\right). \tag{3.12}$$

3.2.2 Phase indicator evolution

The average mass conservation of the liquid phase, or phase 1 is defined by Eq. 3.11. However, this can not be used as an equation for α since the phase quantity $\bar{\mathsf{v}}_1$ is not a variable of our numerical model (only the global variable $\bar{\mathsf{v}}$ is). Looking at the left-hand-side of the equation one can derive (we use the notation $\alpha_1 = \alpha$ and $\alpha_2 = 1 - \alpha$):

$$\begin{aligned}
\frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1 \bar{\mathsf{v}}_1) &= \frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1(\alpha_1 + \alpha_2)\bar{\mathsf{v}}_1) \\
&= \frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1\alpha_1\bar{\mathsf{v}}_1) + \nabla.(\alpha_1\alpha_2\bar{\mathsf{v}}_1) \\
&= \frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1\alpha_1\bar{\mathsf{v}}_1) + \nabla.(\alpha_1\alpha_2\bar{\mathsf{v}}_1) + (\nabla.(\alpha_1\alpha_2\bar{\mathsf{v}}_2) - \nabla.(\alpha_1\alpha_2\bar{\mathsf{v}}_2)). \\
&= \frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1(\alpha_1\bar{\mathsf{v}}_1 + \alpha_2\bar{\mathsf{v}}_2)) + \nabla.(\alpha_1\alpha_2(\bar{\mathsf{v}}_1 - \bar{\mathsf{v}}_2)) \\
&= \frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1\bar{\mathsf{v}}) + \nabla.(\alpha_1\alpha_2\bar{\mathsf{v}}_r)
\end{aligned} \tag{3.13}$$

The global equation for the phase indicator function is:

$$\frac{\partial \alpha}{\partial t} + \nabla.(\alpha\bar{\mathsf{v}}) + \nabla.(\alpha\,(1-\alpha)\,\bar{\mathsf{v}}_r) = \frac{\dot{m}}{\rho_l}, \tag{3.14}$$

where $\bar{\mathsf{v}}_r = \bar{\mathsf{v}}_1 - \bar{\mathsf{v}}_2$ is the vector of compressive velocity. Sometimes described as artificially added in order to compress the interface and insures the stiffness of the interface, we see that it is in fact directly derived from the mass conservation equation of phase 1. However as mentioned before the solver do not have access to values of the velocity in each phase, and the quantity $\bar{\mathsf{v}}_r$ is approximated on a face as:

$$\bar{\mathsf{v}}_{r,f} = \mathbf{n}_f \min\left[C_\alpha \frac{|\phi|}{|S_f|}, \max_F\left(\frac{|\phi|}{|S_F|}\right)\right], \tag{3.15}$$

where $\mathbf{n}_f$ is the face unit vector, ϕ the flux through the face, and S_f its surface. Typically the interface (continuous values of α between 0 and 1) will be spread over 3 or 4 cells, however the parameters of the simulation give the user some control on how sharp the interface has to be kept ($C_\alpha = 0$ for no compression, $C_\alpha > 1$ for enhanced compression).

3.2.3 Momentum equilibrium

We do not go through the details of the derivation, but instead, we give the main steps leading to the equation of momentum equilibrium for multiphase flow in the Volume of fluid method, as described by Hirt [43]. Indeed, it requires some approximations and linearizations that we do not justify here. This section's objective is not to give a rigorous derivation, but to show as a general idea that by averaging the equations and inserting the interface boundary conditions, one can obtain an equation for the global variables of our system.

Averaged equation for one phase

The momentum equilibrium of one phase described in Eq. 2.7 is averaged over the phase 1 and 2:

$$\overline{\rho_i \frac{\partial \mathsf{v}_i}{\partial t} + \rho_i . \nabla . \mathsf{v}_i \mathsf{v}_i} = \overline{-\nabla p_i + \nabla . \left(\mu_i \left(\nabla \mathsf{v}_i +^t \nabla \mathsf{v}_i \right) \right)}. \tag{3.16}$$

- The averaging the left-hand side of Eq. 3.16 is obtained with the average operators and theorems described in Eq. 3.1 and 3.7.

$$\overline{\frac{\partial \rho_i \mathsf{v}_i}{\partial t} + \nabla . \left(\rho_i \mathsf{v}_i \mathsf{v}_i \right)} = \rho_i \frac{\partial \overline{\mathsf{v}}_i}{\partial t} - \frac{1}{V} \int_{A_{12}} \left(\mathsf{w} \rho_i \mathsf{v}_i \right) . \mathbf{n}_{12} dA + \nabla . \left(\rho_i \overline{\mathsf{v}_i \mathsf{v}_i} \right) + \frac{1}{V} \int_{A_{12}} \left(\rho_i \mathsf{v}_i \mathsf{v}_i \right) . \mathbf{n}_{12} dA, \tag{3.17}$$

- Averaging the right-hand side of Eq. 3.16 for phase 1 gives,

$$\begin{aligned}
&\overline{-\nabla p_1 + \nabla.\left(\mu_1\left(\nabla \mathsf{v}_1 +^t \nabla \mathsf{v}_1\right)\right)} \\
&\quad = -\nabla \bar{p}_1 - \frac{1}{V}\int_{A_{12}} p_1 \mathbf{n}_{12} dA + \mu_1 \nabla.\left(\overline{\left(\nabla \mathsf{v}_1 +^t \nabla \mathsf{v}_1\right)}\right) + \frac{1}{V}\int_{A_{12}} \mu_1 \left(\nabla \mathsf{v}_1 +^t \nabla \mathsf{v}_1\right).\mathbf{n}_{12} dA, \\
&\quad = -\nabla \bar{p}_1 + \mu_1 \nabla.\left(\overline{\left(\nabla \mathsf{v}_1 +^t \nabla \mathsf{v}_1\right)}\right) + \frac{1}{V}\int_{A_{12}} \left(-\mathsf{I} p_1 + \mu_1\left(\nabla \mathsf{v}_1 +^t \nabla \mathsf{v}_1\right)\right).\mathbf{n}_{12} dA.
\end{aligned} \tag{3.18}$$

- Overall our equation for phase 1 is:

$$\rho_1 \frac{\partial \bar{\mathsf{v}}_1}{\partial t} + \nabla.\left(\rho_1 \overline{\mathsf{v}_1 \mathsf{v}_1}\right) = -\nabla \bar{p}_1 + \mu_1 \nabla.\left(\overline{\left(\nabla \mathsf{v}_1 +^t \nabla \mathsf{v}_1\right)}\right) + \mathbf{F}_{12}, \tag{3.19}$$

where $\mathbf{F}_{12} = \frac{1}{V}\int_{A_{12}} \left(-\rho_1\left(\mathsf{v}_1 - \mathsf{w}\right)\mathsf{v}_1 - \mathsf{I} p_1 + \mu_1\left(\nabla \mathsf{v}_1 +^t \nabla \mathsf{v}_1\right)\right).\mathbf{n}_{12} dA$ is an integral term accounting for the forces at the interface.

Volume-of-fluid formulation of the momentum equation

The following approximations are made : $\rho \overline{\mathsf{v}\mathsf{v}} \approx \rho_1 \overline{\mathsf{v}_1 \mathsf{v}_1} + \rho_2 \overline{\mathsf{v}_2 \mathsf{v}_2}$, and $\rho \bar{\mathsf{v}} \approx \rho_1 \bar{\mathsf{v}}_1 + \rho_2 \bar{\mathsf{v}}_2$, the last approximation being due to averaging error. Moreover, we approximate $\overline{\mu_1\left(\nabla \mathsf{v}_1 +^t \nabla \mathsf{v}_1\right)} + \overline{\mu_2\left(\nabla \mathsf{v}_2 +^t \nabla \mathsf{v}_2\right)} \approx \mu\left(\nabla \bar{\mathsf{v}} +^t \nabla \bar{\mathsf{v}}\right)$. Summing the relation given in Eq. 3.19 for both phases gives:

$$\rho \frac{\partial \bar{\mathsf{v}}}{\partial t} + \rho \nabla.\left(\bar{\mathsf{v}}\bar{\mathsf{v}}\right) = -\nabla \bar{p} + \nabla.\mu\left(\nabla \bar{\mathsf{v}} +^t \nabla \bar{\mathsf{v}}\right) + F_{12} + F_{21} \tag{3.20}$$

We note the interfacial forces due to surface tension:

$$\mathbf{F}_c = \frac{1}{V}\int_{A_{12}} \sigma H_{12} \mathbf{n}_{12} dA \tag{3.21}$$

where where σ is the surface tension, H_{12} the local curvature of the interface, $\mathbf{n}_{12}$ vector normal to the interface.

According to the force equilibrium boundary condition Eq. 2.8, $F_{12} + F_{21} = \mathbf{F}_c$. The pressure and viscous forces canceled each other out. Overall we obtain the equation of momentum equilibrium for mutliphase flow in the Volume of fluid method [43]:

$$\rho\left(\frac{\partial \bar{\mathsf{v}}}{\partial t} + \bar{\mathsf{v}}.\nabla \bar{\mathsf{v}}\right) = -\nabla p + \nabla.\mu\left(\nabla \bar{\mathsf{v}} +^t \nabla \bar{\mathsf{v}}\right) + \mathbf{F}_c, \tag{3.22}$$

where p is the hydrostatic pressure including the gravity effects $p = \bar{p} - \rho\mathbf{g}.\mathbf{z}$. The interfacial forces $\mathbf{F}_c$ can not be computed as such. Brackbill [44] developed an approximation of the interfacial forces referred as the Continuum Surface Forces (CSF) model. $\mathbf{F}_c$ is expressed as $\delta_I \sigma \bar{H} \bar{\mathbf{n}}$ where is δ_I a dirac indicating position of the interface, $\bar{H}$ is the average curvature in the cell, $\bar{\mathbf{n}}$ is the average normal to the interface. The local curvature is directly calculated from the local normal in $H_{12} = -\nabla.\mathbf{n}_{12}$. The average of the normal is calculated with the gradient of α in Eq. 3.5. Overall Brackbill's approximation of the surface tension term is:

$$\mathbf{F}_c = \sigma \nabla. \left(\frac{\nabla \alpha}{\|\nabla \alpha\|} \right) \nabla \alpha. \tag{3.23}$$

3.2.4 Concentration of the miscible component

Equation

In order to incorporate the concentration into the model, it needs to be formulated as a single relation satisfied by for the locally average concentration ω_A. The model we implemented for the evolution of the concentration field is based on the original method proposed by Haroun et al. [1] more recently referred to as Continuum Species Transfer (CST) model [56, 57]. In their paper, Haroun et al. proposed a method consistent with the VOF approach to solve an equation for the evolution of the concentration field while satisfying simultaneously Henry's law at the gas/liquid interface and the continuity of fluxes across the interface. For the derivation we drop the underscript A indicating the particular species. We apply the averaging operator to the mass conservation of the component in phase 1 (Eq. 2.11), and use the relations in Eq. 3.7 to obtain:

$$
\begin{aligned}
0 &= \overline{\frac{\partial C_1}{\partial t} + \nabla.\left(C_1 \mathsf{v}_1\right) - \nabla.\left(D_1 \nabla C_1\right)} \\
&= \left(\frac{\partial \bar{C}_1}{\partial t} - \frac{1}{V}\int_{A_{12}} C_1 \mathbf{n}_{12} \mathsf{w} dA\right) + \left(\nabla.\left(\overline{C_1 \mathsf{v}_1}\right) + \frac{1}{V}\int_{A_{12}} C_1 \mathsf{v}_1 . \mathbf{n}_{12} dA\right) \\
&\quad - \left(\nabla.\left(\overline{D_1 \nabla C_1}\right) + \frac{1}{V}\int_{A_{12}} D_1 \nabla C_1 . \mathbf{n}_{12} dA\right) \\
&= \frac{\partial \bar{C}_1}{\partial t} + \nabla.\left(\overline{C_1 \mathsf{v}_1}\right) - \nabla.\left(\overline{D_1 \nabla C_1}\right) + \frac{1}{V}\int_{A_{12}} \left(C_1 (\mathsf{v}_1 - \mathsf{w}) - D_1 \nabla C_1\right) . \mathbf{n}_{12} dA.
\end{aligned}
\tag{3.24}
$$

Identically for the phase 2. By summing those equations and using the continuity of fluxes (Eq. 2.12) we have:

$$
0 = \frac{\partial \left(\bar{C}_1 + \bar{C}_2\right)}{\partial t} + \nabla.\left(\overline{C_1 \mathsf{v}_1} + \overline{C_2 \mathsf{v}_2}\right) - \nabla.\left(\overline{D_1 \nabla C_1} + \overline{D_2 \nabla C_2}\right). \tag{3.25}
$$

The first term immediately becomes $\frac{\partial \bar{C}}{\partial t}$. Following the procedure proposed by Haroun et al. [1] we will make several approximations. Using the physical quantities we note that $C\mathsf{v} = \mathbb{1}_1 C_1 \mathsf{v}_1 + (1 - \mathbb{1}_1) C_2 \mathsf{v}_2$, so that $\overline{C_1 \mathsf{v}_1 + C_2 \mathsf{v}_2} = \overline{C\mathsf{v}} \approx \bar{C}\bar{\mathsf{v}}$, the last approximation being due to averaging error. This leaves us with the final flux term $F = \nabla.\left(D_1 \overline{\nabla C_1} + D_2 \overline{\nabla C_2}\right)$. First the diffusion coefficient is approximated to its average D, which is always true away from the interface. The averaging theorems Eq. 3.7 then give:

$$
F = \nabla.\left(D\left(\nabla \bar{C}_1 + \nabla \bar{C}_2 + \frac{1}{V}\int_{A_{12}} \left(C_1 - C_2\right) \mathbf{n}_{12} . dA\right)\right). \tag{3.26}
$$

Finally we have:

$$
\frac{\partial \bar{C}}{\partial t} + \nabla.\left(\bar{C}\bar{\mathsf{v}}\right) - \nabla.\left(D \nabla \bar{C}\right) - \nabla.\left(D \underbrace{\frac{1}{V}\int_{A_{12}} \left(C_1 - C_2\right) \mathbf{n}_{12} . dA}_{\mathbf{\Phi}}\right) = 0, \tag{3.27}
$$

where $\mathbf{\Phi}$ is an additional flux associated to the thermodynamics equilibrium at the interphase. We need several additional approximations in order to evaluate the flux $\mathbf{\Phi}$, wich is calculated at the interface A_{lg}. The above equation is an approximation over a cell, supposedly very small, and we will approximate the local quantity C_i at the interface with their average over their phases : $C_1 \approx \bar{C}_1^1$ and $C_2 \approx \bar{C}_2^2$, consequently we assume that Henry's law is valid for those average $H\bar{C}_1^1 \approx \bar{C}_2^2$. Recalling that $\bar{C} = \alpha\bar{C}_1^1 + (1-\alpha)\,\bar{C}_2^2$, we obtain the approximation, for a cell including the interface that $C_1 - C_2 \approx \frac{1-H}{\alpha+(1-\alpha)H}\bar{C}$, and the additional flux becomes:

$$\mathbf{\Phi} = \frac{1-H}{\alpha+(1-\alpha)H}\bar{C}\left(\frac{1}{V}\int_{A_{12}}\mathbf{n}_{12}dA\right). \tag{3.28}$$

The last integral is an average of the normal to the interface. As seen in the derivation of the Continuum Surface Forces (section 3.2.3), this term can be calculated from the gradient of the phase indicator using Eq. 3.5. The overall equation for the concentration for species A is finally:

$$\frac{\partial\bar{C}_A}{\partial t} + \nabla.\left(\bar{C}_A\bar{\mathrm{v}}\right) = \nabla.\left(D_A\nabla\bar{C}_A\right) + \nabla.\left(D_A\underbrace{\frac{1-H_A}{\alpha+(1-\alpha)H_A}\bar{C}\nabla\alpha}_{\mathbf{\Phi}_A}\right). \tag{3.29}$$

The additional flux in Eq. (3.29), $\mathbf{\Phi}_A$, results from the jump at the gas/liquid interface. It transforms the solubility condition, Eq. (2.14), into a volumetric term, the CST (Continuum Species Transfer) term, under the framework of the VOF formulation [1]. The presence of the CST term suffices to ensure both the jump of concentration at the interface and the continuity of fluxes since it is derived from those properties. In that sense it is a very simple formulation of the problem and leads to an easier implementation, adding a single equation to the model. It is somehow reminiscent of the continuum surface force (CSF) used for the modeling of the surface tension between two fluids [44]. Note that the CST term is only present at the interface where $\nabla\alpha \neq 0$. Additionally if $H_A = 1$, then $\mathbf{\Phi}_A = 0$ and the jump condition vanishes and that, for a large value of H_A, the concentration of acid in gas tends towards zero.

Concerning the diffusion coefficient, we take:

$$D_A = \frac{D_{1,A} D_{2,A}}{\alpha D_{1,A} + (1-\alpha) D_{2,A}}, \tag{3.30}$$

Haroun et al. [1] and Diesing et al. [57] have demonstrated that this harmonic formulation is more robust than a simple mixing rule, $D_A = (\alpha D_{1,A} + (1-\alpha) D_{2,A})$.

Flux

The flux at the interface in a cell is:

$$\dot{m}_A = \frac{1}{V} \int_{A_{12}} (C_{A,1}(\mathsf{v}_1 - \mathsf{w}) - D_{A,1} \nabla C_{A,1}) \mathbf{n}_{12} dA. \tag{3.31}$$

The volumetric flux at the interface of component A can be calculated with the same technique and formulas as previously and we obtain:

$$\dot{m}_A = \left(\bar{C}_A(\bar{\mathsf{v}} - \mathsf{w}) - D_A \nabla \bar{C}_A\right) . \nabla \alpha + \frac{1 - H_A}{\alpha + (1-\alpha) H_A} D_A \bar{C}_A \nabla \alpha, \tag{3.32}$$

This account for the mass transfer at the interface between phase 1 and 2, $\dot{m}$, as defined in section 3.2.1. Therefore for all components in our system we have the global mass transfer at the interface:

$$\dot{m} = \sum_A \dot{m}_A. \tag{3.33}$$

Boundary condition with the solid

The model previously described was used to simulate mass transfer in the case of a liquid film, where one phase was totally wetting. Here we extend the model to more complex cases, where triple point boundary between the two fluids and the solid occur. It will be particularly useful in dynamic simulation, for example during processes of drainage or imbibition. We assume no interaction and no chemical reaction of the component with the solid. In each phase at the boundary with the solid we, therefore, have $\nabla C_{i,A} . \mathbf{n}_s = 0$, where $\mathbf{n}_s$ is the normal to the surface of the solid. Using the same derivation as previously for the fluxes in the concentration equation, we can see that the boundary condition with the solid for the global variable becomes:

$$\nabla \bar{C}_A.\mathbf{n}_s = \frac{(1-H_A)}{\alpha + (1-\alpha)H_A} \bar{C}_A \nabla \alpha.\mathbf{n}_s. \tag{3.34}$$

This is the expression of the CST term for the boundary condition. Active only at the triple point ($\nabla\alpha \neq 0$), it accounts for the change in fluxes due to the jump of concentration at the interface.

3.3 Solvers description

Our final system of equation for our global variables is as follow:

$$\begin{aligned}
&\rho\left(\frac{\partial \bar{\mathsf{v}}}{\partial t} + \bar{\mathsf{v}}.\nabla\bar{\mathsf{v}}\right) = -\nabla\bar{p} + \nabla.\mu\left(\nabla\bar{\mathsf{v}} +^t \nabla\bar{\mathsf{v}}\right) + \mathbf{F}_c, \\
&\nabla.\bar{\mathsf{v}} = \dot{m}\left(\frac{1}{\rho_1} - \frac{1}{\rho_2}\right), \\
&\frac{\partial\alpha}{\partial t} + \nabla.(\alpha\bar{\mathsf{v}}) + \nabla.(\alpha\,(1-\alpha)\,\bar{\mathsf{v}}_r) = \frac{\dot{m}}{\rho_l}, \\
&\frac{\partial\bar{C}_A}{\partial t} + \nabla.\left(\bar{C}_A\bar{\mathsf{v}}\right) = \nabla.\left(D_A\nabla\bar{C}_A\right) + \nabla.\left(D_A\frac{1-H_A}{\alpha+(1-\alpha)H_A}\bar{C}\nabla\alpha\right).
\end{aligned} \tag{3.35}$$

Depending on the equations taken into account, and the calculation of $\dot{m}$, one can use this system to model different physics.

3.3.1 Two-phase flow solver

With no miscible component and no phase change $\dot{m} = 0$, and only the three first equations of the system 3.35 are taken into account. This is the VOF formulation of two-phase flow for immiscible phases, under isothermal conditions.

$$\begin{aligned}
&\rho\left(\frac{\partial\bar{\mathsf{v}}}{\partial t}+\bar{\mathsf{v}}.\nabla\bar{\mathsf{v}}\right)=-\nabla\bar{p}+\nabla.\mu\left(\nabla\bar{\mathsf{v}}+^{t}\nabla\bar{\mathsf{v}}\right)+\mathbf{F}_c,\\
&\nabla.\bar{\mathsf{v}}=0, \qquad (3.36)\\
&\frac{\partial\alpha}{\partial t}+\nabla.(\alpha\bar{\mathsf{v}})+\nabla.(\alpha\left(1-\alpha\right)\bar{\mathsf{v}}_r)=0,
\end{aligned}$$

3.3.2 Species transport solver

One can simulate a miscible component not affecting the flow, for example a pollution or a colored tracer. Since their concentration is small, we neglect their effect on the mass of each phase and have $\dot{m}=0$. Here, the component is inert towards the flow : its motion is dependent on the flow through the interface location, or the velocity, however equations are not coupled, and the concentration C_A does not affect the flow in any way. We will refer to this solver as species transport.

$$\begin{aligned}
&\rho\left(\frac{\partial\bar{\mathsf{v}}}{\partial t}+\bar{\mathsf{v}}.\nabla\bar{\mathsf{v}}\right)=-\nabla\bar{p}+\nabla.\mu\left(\nabla\bar{\mathsf{v}}+^{t}\nabla\bar{\mathsf{v}}\right)+\mathbf{F}_c,\\
&\nabla.\bar{\mathsf{v}}=0,\\
&\frac{\partial\alpha}{\partial t}+\nabla.(\alpha\bar{\mathsf{v}})+\nabla.(\alpha\left(1-\alpha\right)\bar{\mathsf{v}}_r)=0, \qquad (3.37)\\
&\\
&\frac{\partial\bar{C}_A}{\partial t}+\nabla.\left(\bar{C}_A\bar{\mathsf{v}}\right)=\nabla.\left(D_A\nabla\bar{C}_A\right)+\nabla.\left(D_A\frac{1-H_A}{\alpha+(1-\alpha)H_A}\bar{C}\nabla\alpha\right).
\end{aligned}$$

3.3.3 Phase change solver

Finally, we would like to simulate a phase change induced by diffusive mass transfer as in the supercritical CO_2 experiment described in Figure 1.3 where the supercritical CO_2 (gas phase) dissolves into the surrounding brine (liquid phase), up to disappearance. Clearly, the interface location, and therefore the flow, is affected by the mass transfer at the interface (see Figure 3.4), and the mass change due to component crossing the interface can not be neglected anymore. The phase change solver is described by the full equations in Eq. 3.35 and $\dot{m}$ is driven by the flux of miscible components:

$$\dot{m} = \sum_A \dot{m}_A. \tag{3.38}$$

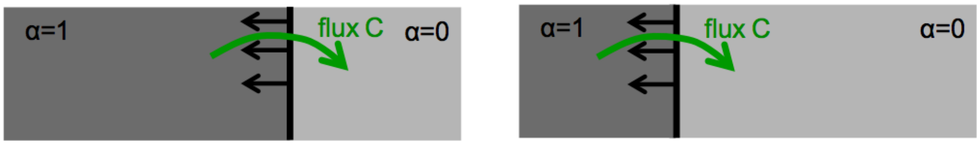


Figure 3.4: Illustration of the effect of the mass transfer on the interface

For simplicity of notation, we will further refer to our global variables without the brackets, as p, v and C_A.

Chapter 4

Numerical implementation

The previous chapter described the governing equations for the numerical variables. Here we present how they are implemented in the code, and we insist on some particular aspects such as the pressure-velocity coupling, the phase change equation solver, and the implementation of the contact angle at the solid surface.

4.1 OpenFOAM® existing solvers

We use an open source software of Computational Fluids Dynamics, OpenFOAM® (www.OpenFOAM.org). The VOF formulation described in section 3.3 is implemented in OpenFOAM in the solver *interFoam* [65]. This is the starting point of our own solver, which can be considered an extension of *interFoam*. The phase change implementation is, however, more complicated, and we used the example of another solver: *interPhaseChangeFoam*. In this solver, the same element is present in the liquid and the gas form, and phase change (calculus of the flux $\dot{m}$) is induced by the temperature. The delicate matter is the transport of the phase indicator function, in particular in the presence of phase change. A specific solver as been developed in OpenFOAM, and is described in section 4.3.

Although OpenFOAM is widely used in the CFD community, precise and accurate documentation about its solvers is not always available. In the appendix, a complete detail of the code, and the parallel with the mathematical model is made.

4.2 PISO algorithm for two-phase flow

Due to the coupling of the momentum equilibrium and the mass conservation equation, a predictor-corrector algorithm is implemented in OpenFOAM: Pressure Implicit with Splitting of Operators (PISO) [66]. Here we describe the general case with $\dot{m}$ not necessarily equal to 0. The PISO implemented in *interFoam* simply takes $\dot{m} = 0$. The general flow chart of the algorithm is presented in Figure 4.1. In order to make a clear parallel with the code implemented in OpenFOAM, we tried to be as consistent as possible with the implementation notation.

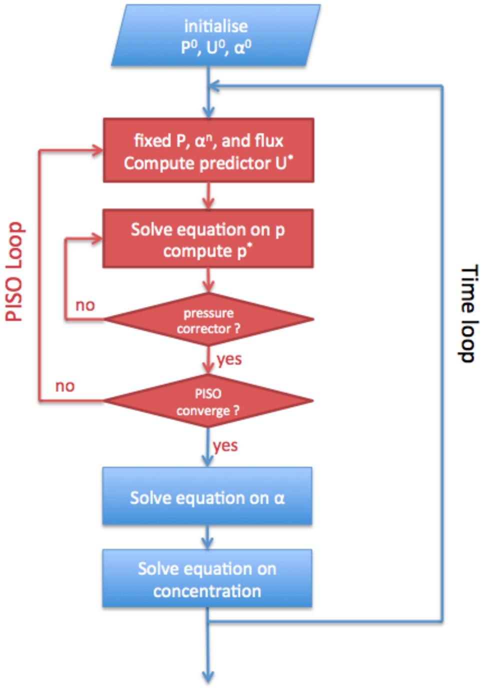


Figure 4.1: Flow chart of the multiphase flow algorithm

4.2.1 Discretization of the Navier-Stokes equation

Over a cell, the Navier-Stokes equation can be discretized in time (n) and space (P is the cell, NP the neighbor cells) implicitly:

$$\mathcal{V}\left(\frac{\rho_P^{n+1}\mathsf{v}_P^{n+1}-\rho_P^{n}\mathsf{v}_P^{n}}{\delta t}\right)=-a'_P\mathsf{v}_P^{n+1}+\sum_{NP}h_{NP}\mathsf{v}_{NP}^{n+1}-(\nabla p)_P-\rho^n\mathbf{g}+(\mathbf{F}_c)_P^n\,, \tag{4.1}$$

where phase indicator values are fixed from previous time-step. $\mathcal{V}$ is the volume of the cell, and δt is the time-step. a'_P marks the contribution of the very cell to the change in momentum, whereas h_{NP} coefficient indicates the contribution of the neighboring cells (convection and diffusion). The inertia terms are linearized in the process, and exact values of the coefficients a'_p and h_{NP} depend on the scheme of discretization chosen by the user. We define $a_P = a'_P + \frac{\mathcal{V}}{\delta t}\rho_P$. The operator a is the diagonal matrix holding the coefficients a_P, and $\mathbf{H}$ is the affine operator defined such that $\left(\mathbf{H}\left(\mathsf{v}\right)\right)_P = \sum_{NP} h_{NP}\mathsf{v}_{NP} + \frac{\mathcal{V}}{\delta t}\rho_P\mathsf{v}_P^n$. The semi-discretized Navier-Stokes equation for the velocity becomes:

$$a\mathsf{v}^{n+1} = \mathbf{H}\mathsf{v}^{n+1} - \nabla p^n + S, \tag{4.2}$$

where $S = -\rho^n\mathbf{g} + (\mathbf{F}_c)^n$.

4.2.2 Momentum predictor of the velocity

The velocity is predicted by fixing the pressure and the phase indicator, and solving the Navier-Stokes equation. Therefore the predicted velocity v^* is solution to:

$$\rho\left(\frac{\mathsf{v}^*-\mathsf{v}^n}{\delta t}+\mathsf{v}^n.\nabla\mathsf{v}^*\right)-\nabla.\mu\left(\nabla\mathsf{v}^* +^t \nabla\mathsf{v}^*\right)=-\nabla p^n-\rho^n\mathbf{g}+\mathbf{F}_c^n \tag{4.3}$$

that can be recast into:

$$a\mathsf{v}^* = \mathbf{H}\mathsf{v}^* - \nabla p + S, \tag{4.4}$$

4.2.3 Derivation of an equation for the pressure

The main problem with our guess on the velocity is that it does not necessarily obey the mass conservation equation (Eq. 3.12). However this guess on the velocity can be substitute in the discretized Navier-Stokes equation, and since a is a diagonal matrix, it is easily invertible:

$$\mathsf{v}^* = a^{-1}\mathbf{H}\mathsf{v}^* + a^{-1}\left(-\nabla p - \rho^n \mathbf{g} + \mathbf{F}_c^n\right). \tag{4.5}$$

As v^* is fixed, the pressure is then calculated such that the velocity obeys the mass conservation equation. Equation on p is therefore:

$$\nabla.\left(a^{-1}\nabla p\right) = \nabla.\left(a^{-1}\mathbf{H}\mathsf{v}^* + a^{-1}\left(-\rho^n \mathbf{g} + \mathbf{F}_c^n\right)\right) - \dot{m}\left(\frac{1}{\rho_1} - \frac{1}{\rho_2}\right). \tag{4.6}$$

4.2.4 Velocity-pressure coupling

The overall PISO loop for the pressure-velocity coupling is therefore:

1. Obtain a predictor v^* of the velocity solving the Navier Stokes equation 4.4, using the phase indicator on the previous time step, and pressure and flux of the previous guess (if first iteration this is pressure and flux of the previous time-step)

2. With this velocity predictor, compute the diagonal matrix a, its inverse a^{-1}, the affine operator $\mathbf{H}$ applied to the current predictor, and the fluxes at the faces $f\,:\, F = \left(a^{-1}\left(\mathbf{H}\left(\mathsf{v}^*\right) + S\right)\right)_f .S_f$ (where S_f is the normal to the face)

3. Solve the pressure equation 4.6 and obtain a guess for the pressure p^*

4. Find the final flux at the faces correcting the approximated flux by the pressure effect : $\tilde{F} = F - \left[a^{-1}\nabla p^*\right]_f .S_f$

5. Correct the cell-centered velocity due to the new pressure distribution and obtain a second guess on the velocity v^{**}

$$\mathsf{v}^{**} = a^{-1}\mathbf{H}\left(\mathsf{v}^*\right) - a^{-1}\nabla p^* - a^{-1}S \tag{4.7}$$

6. Go to step 2 and repeat `nCorrectors` times.

Typically, two repetitions are enough to converge [66]. The velocity-pressure coupling is not unconditionally stable and requires strict restrictions on the time-step, which is usually achieved by imposing a Courant–Friedrichs–Lewy (CFL) condition.

4.3 Phase indicator equation : MULES

Solving the equation on the phase indicator requires extra work since it has to guarantee the boundedness of α between 0 and 1. MULES (Multidimensional Universal Limiter for Explicit Solution) has been implemented in OpenFOAM since the version 1.4, as a very effective method to solve this equation explicitly. However, this comes at a cost, explicitness involves a very strict limitation on the CFL (`AlphaCo` in OpenFOAM), and therefore on the time-step limit, despite the introduction of time-step sub-cycling. Nonetheless, it is less restrictive than the pressure-velocity coupling described previously. A semi-implicit version of MULES was introduced in the version 2.3 of OpenFOAM and has been gradually improved. We only present here the explicit formulation, used in OpenFOAM 2.2.

MULES solves hyperbolic equations for a quantity β:

$$\frac{\partial \beta}{\partial t} + \nabla . \vec{F} = \beta S_p + S_u, \tag{4.8}$$

where:

- β is the scalar field of interest, defined at the center of the cells
- $\vec{F}$ is the flux of β
- S_p is the implicit source term
- S_u is the explicit source term

If β is bounded by 0 and 1, MULES can be called in OpenFOAM by taking in argument:

$$\texttt{MULES}\left(\beta, \phi, \vec{F}, S_p, S_u, 1, 0\right). \tag{4.9}$$

where ϕ is a flux on the faces used to determine the upwind direction for discretization of the flux terms in the MULES internal algorithm. In the case where source terms are all equal to zero, one can simply call `MULES` $\left(\beta, \phi, \vec{F}, 1, 0\right)$. An explicit source term S can be artificially counted implicitly by dividing it by β : $S \approx \beta \left(\frac{1}{\beta+\epsilon} S\right)$ where ϵ is a small parameter preventing from dividing by 0. Additional information on the internal algorithm of MULES can be found in the thesis of Santiago Marquez [67].

4.4 Concentration equation

The pressure, velocity, and phase indicator are fixed. Concentration is solved fully implicitly. The discretization of the time-derivative is forward Euler. The convective term $\nabla.(C\mathsf{v})$ and the CST additional flux $\mathbf{\Phi}$ are discretized with Gauss Van Leer scheme, a total variation diminishing (TVD) scheme. The diffusion term $\nabla.(D\nabla C)$ is discretized with a Gauss linear limited corrected scheme which second order and conservative.

4.5 Boundary condition : constant contact angle

After solving the equation on the phase indicator α we obtain a displacement of the interface, in particular at the boundary with the solid. However the resolution of the equation does not take into account the contact angle condition, and the interface after displacement forms an angle $\theta_I \neq \theta_0$ with the solid.

$$\mathbf{n}_{12} = \cos\theta_I \mathbf{n}_w + \sin\theta_I \mathbf{t}_w. \tag{4.10}$$

We define its normal at the triple point $\mathbf{n}_{12}$. The impact of such deviation from the right contact angle appears at the next time step, in the equation on the velocity, when calculating the interfacial forces $\mathbf{F}_c = \sigma\nabla.\left(\frac{\nabla\alpha}{\|\nabla\alpha\|}\right)\nabla\alpha$. At that time, the normal to the interface over the domain is calculated regularly, but at the triple point with the solid a correction to the normal is introduced, $\tilde{\mathbf{n}}_{12}$, such that it satisfies the boundary

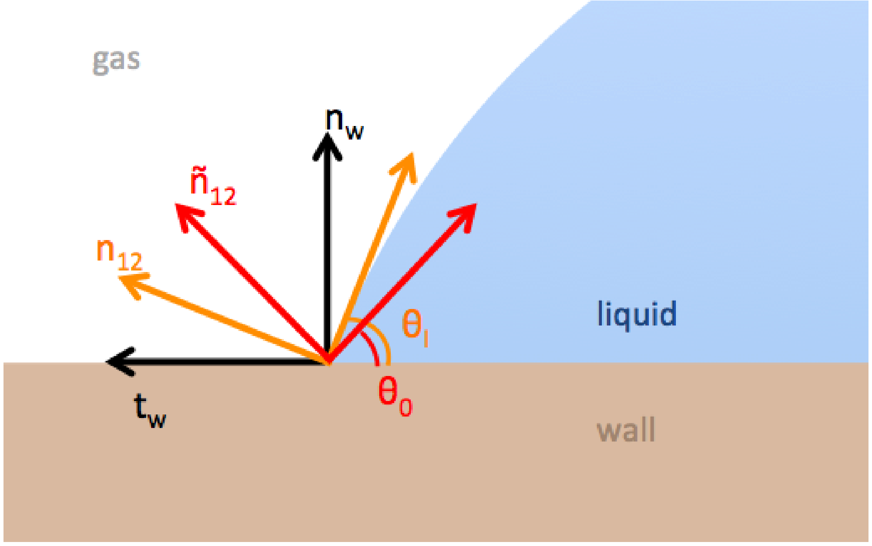


Figure 4.2: Schematic of the contact anglebinary i. θ_I is the angle after displacement of the interface, θ_0 is the boundary condition we want to impose

condition:

$$\tilde{\mathbf{n}}_{12} = \cos\theta_0 \mathbf{n}_w + \sin\theta_0 \mathbf{t}_w. \tag{4.11}$$

The corrected normal at the triple point is computed in OpenFOAM with quantities the solver has access to: the numerical normal $\mathbf{n}_{12}$ obtained by solving the equation on α, and the normal to the interface $\mathbf{n}_w$. Eq. 4.10 becomes:

$$\mathbf{t}_w = \frac{1}{\sin\theta_I}\mathbf{n}_{12} - \frac{\cos\theta_I}{\sin\theta_I}\mathbf{n}_w. \tag{4.12}$$

By substituting this expression in the definition of the corrected normal in Eq.

4.11, we derive:

$$\begin{aligned}
\tilde{\mathbf{n}}_{12} &= \cos\theta_0 \mathbf{n}_w + \sin\theta_0 \left(\frac{1}{\sin\theta_I}\mathbf{n}_{12} - \frac{\cos\theta_I}{\sin\theta_I}\mathbf{n}_w \right) \\
&= \frac{\sin^2\theta_I \cos\theta_0 - \sin\theta \sin\theta_I \cos\theta_I}{\sin^2\theta_I}\mathbf{n}_w + \frac{\sin\theta_0}{\sin\theta_I}\mathbf{n}_{12} \\
&= \frac{(1-\cos^2\theta_I)\cos\theta_0 - \sin\theta_0 \sin\theta_I \cos\theta_I}{\sin^2\theta_I}\mathbf{n}_w + \frac{\sin\theta_0 \sin\theta_I}{\sin^2\theta_I}\mathbf{n}_{12} \\
&= \frac{\cos\theta_0 - \cos\theta_I(\cos\theta_I \cos\theta_0 - \sin\theta_I \sin\theta_0)}{\sin^2\theta_I}\mathbf{n}_w + \frac{\cos(\theta_I - \theta_0)}{\sin^2\theta_I}\mathbf{n}_{12} \\
&= \frac{\cos\theta_0 - \cos\theta_I \cos(\theta_I - \theta_0)}{1-\cos^2\theta_I}\mathbf{n}_w + \frac{\cos(\theta_I - \theta_0)}{1-\cos^2\theta_I}\mathbf{n}_{12}.
\end{aligned} \tag{4.13}$$

With this expression, the corrected normal to the interface at the triple point can be calculated directly from the normal to the wall, the normal to the interface calculated after the displacement of the interface, and the angles θ_I and θ_0.

Chapter 5

Simulation results

Three solvers were presented: immiscible two-phase flow, species transport (immiscible two-phase flow with an inert species), mass transfer (immiscible two-phase flow with phase change induced by mass transfer). Here we present some of the results we obtained with those three models. First, two-phase flow is simulated in the pore-space of a porous medium, both for drainage and imbibition. Second, the species transport is validated by comparison with analytical solutions and then used to up-scale mass transfer coefficient. Finally, we present preliminary results of phase change simulations.

5.1 Drainage and imbibition in porous media

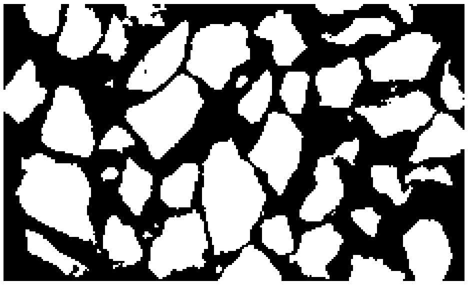


Figure 5.1: Binary image of a 2D porous medium. Black being the pore-space, and white are the solid grains

We want to simulate drainage and imbibition in a porous medium at the pore-scale. Figure 5.1 corresponds to the pore-space of interest. This pore-space first has to be discretized, which is achieved using the preprocessing tools of OpenFOAM. Then the simulations are performed with *interFoam*, the VOF solver of OpenFOAM.

5.1.1 Simulation setup

Mesh

The pore-space is meshed using `blockMesh` and `snappyHexMesh` in OpenFOAM. The domain size is 850x400 µm, meshed with 180,000 cells.

Time parameters of the simulation

Starting at a time $t = 0$, the simulation uses a time-step of $\delta t = 1 \times 10^{-5}\,\mathrm{s}$. However, in order to speed up the simulation, the time-step is adjustable, as long as it satisfies the condition on the CFL number : $CFL < 0.2$.

Initial conditions

At time $t = 0$, the pore-space is occupied by liquid only, with zero velocity.

Boundary conditions

Boundary :	left (input)	right (output)	top/bottom & solid
U	$U = 0.01 m.s^{-1}$	$\mathbf{n}.\nabla U = 0$	$\mathbf{n}.U = 0$ (no slip)
p	$\mathbf{n}.\nabla p = 0$	$p = 0$	`fixedFluxPressure`
α	$\alpha = 0$ or 1	$\mathbf{n}.\nabla \alpha = 0$	contact angle : $\theta_0 = 45°$

Table 5.1: Boundary conditions for the numerical model

In a mathematical model of the PDEs, one only has to specify a boundary condition for either the velocity or the pressure at each boundary. Numerically, however, one has to specify boundary conditions for all quantities at every boundary, as summarized in Table 5.1.

- At all wall boundaries, the `fixedFluxPressure` boundary condition is applied to the pressure field, which adjusts the pressure gradient so that the boundary flux matches the no-slip boundary condition for the velocity.

- Our model only involves the gradient of the pressure, therefore the pressure is defined up to a constant. This leads to numerical errors (matrices are ill-conditioned). There are several solution to define the pressure. Here we set it to zero on the right boundary. It may lead to unphysical behaviors of the pressure field when the second phase reaches the outlet, so one can also either set the mean value of the pressure at the outlet, or let $\mathbf{n}.\nabla p = 0$ be the boundary condition and specify the pressure in one cell in the middle of the mesh.

- Finally, On the right free boundary for the phase indicator we set $\mathbf{n}.\nabla \alpha = 0$. This tends to lead to unphysical behaviors of the phase indicator close to the output. The typical way of solving this problem is to add a small portion of void space at the output of our domain (2D or 3D) and ignore this portion in further post-processing calculations.

5.1.2 Imibibition

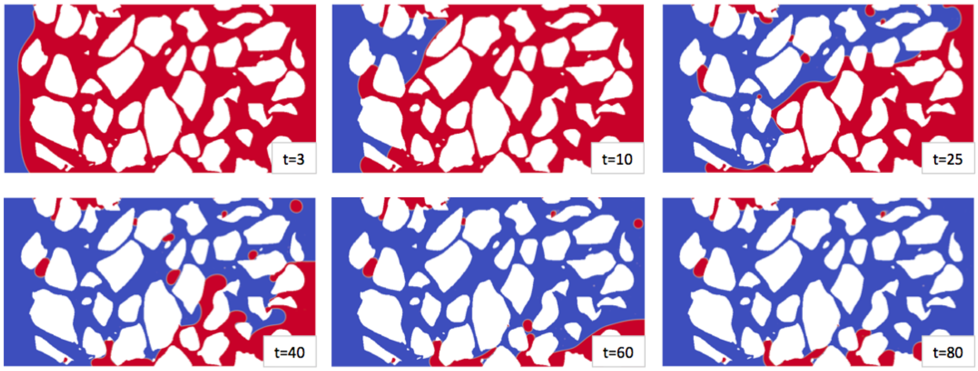


Figure 5.2: Imbibition of the oil in the water

We simulate the imibition of a wetting fluid (water) in a non wetting fluid (oil). We take $\mu_g = \mu_l = 6 \times 10^{-2}\,\mathrm{kg\,m^2\,s^{-1}}$, $\rho_g = \rho_l = 1000\,\mathrm{kg\,m^{-3}}$, the surface tension

σ =0.097 kg s^{-2}, and the contact angle $\theta_0 = 45°$. The injection rate is 0.01 m s^{-1}. Running the simulation for about 10 hours on 6 processors we obtained the results presented in Figure 5.2. Breakthgouth occurs at 27ms. After reaching steady-state, residual oil occupies 4% of the pore-space.

5.1.3 Drainage

In our base case, we simulate the drainage of a non wetting fluid (say gas or oil) in a wetting fluid (oil or water). We take $\mu_g = \mu_l$ =6 × 10^{-2} kg m^2 s^{-1}, $\rho_g = \rho_l$ =1000 kg m^{-3}, the surface tension σ =0.097 kg s^{-2}, and the contact angle $\theta_0 = 45°$. The injection rate is 0.01 m s^{-1}. Running the simulation for about 10 hours on 6 processors we obtained the results presented in Figure 5.3. Breakthrough occurs at $t = 57$ ms, and the flow reaches steady state at approximately $t = 120$ ms. Two fingers have formed, leaving 35% of residual water (S_w).

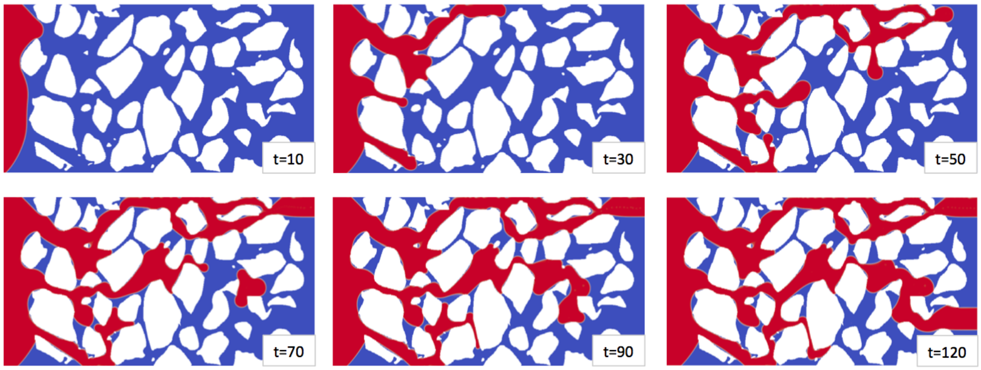
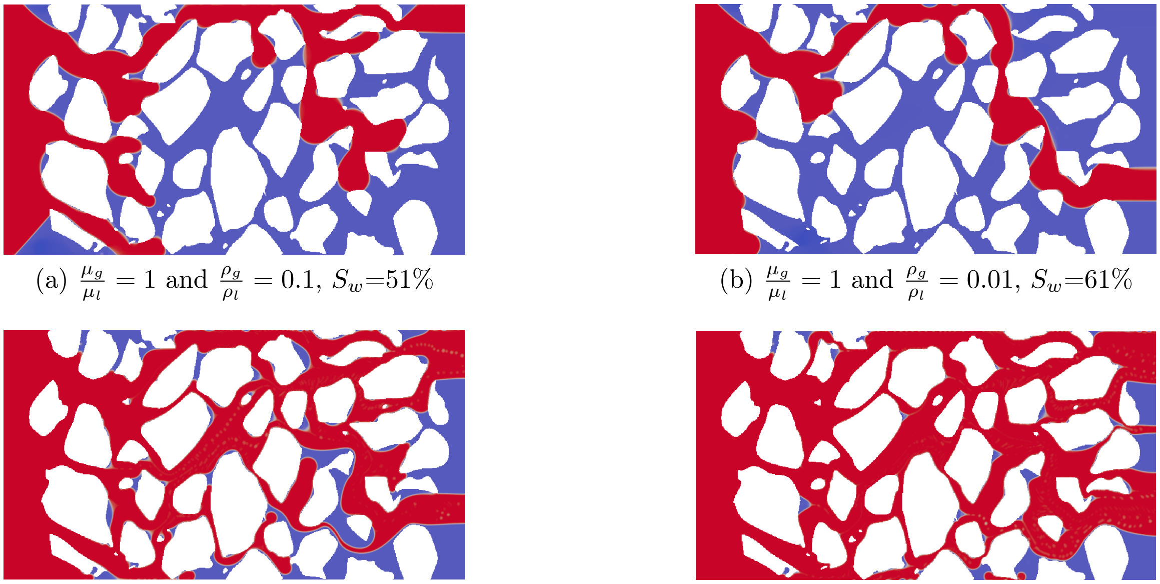


Figure 5.3: Drainage of oil in water, time is in ms. At 120 ms, the flow has reached steady-state.

We run the same simulation, but changing the ratio of density or of viscosities between the two fluids (properties of the wetting fluid remain unchanged). Results at steady state are presented in Figure 5.4. Breakthrough occurs in all cases between 50 and 60 ms. We observe that for a lower density of the gas (cases a and b), the gas tends to form fewer fingers, and the residual water is bigger. On the contrary, when the viscosity of the injecting fluid increases, the residual water decreases.



(a) $\frac{\mu_g}{\mu_l} = 1$ and $\frac{\rho_g}{\rho_l} = 0.1$, S_w=51%

(b) $\frac{\mu_g}{\mu_l} = 1$ and $\frac{\rho_g}{\rho_l} = 0.01$, S_w=61%

(c) $\frac{\mu_g}{\mu_l} = 10$ and $\frac{\rho_g}{\rho_l} = 1$, S_w=21%

(d) $\frac{\mu_g}{\mu_l} = 100$ and $\frac{\rho_g}{\rho_l} = 1$, S_w=10%

Figure 5.4: Results of the drainage at 300ms for different ratios of viscosity or density

5.2 Analytical solutions for the species transport model

We want to validate our species transport model for two-phase flow in porous media. This is achieved by comparing the simulation results with analytical solutions in simple cases. Two cases are presented here, one steady-state and one transient solution. For both the velocity is equal to zero and pure diffusion is observed.

5.2.1 Steady-state analytical solution in a tube

We consider a tube of dimension 1mm×0.2mm, with solid on the top and bottom. Each half of the tube is occupied by one phase, their densities and viscosities are equal, the contact angle is fixed at 90°. This is an equilibrium and the velocity is zero in the tube. Starting with a concentration of 0 in the tube, we fix an inlet concentration of $1\,\mathrm{kg\ m^{-3}}$ and an outlet concentration of $0\,\mathrm{kg\ m^{-3}}$ (see Figure 5.6). We set $l = 5 \times 10^{-4}$m, $H = 2$, D_{liq}=$2\,\mathrm{m^2\ s^{-1}}$ D_{gas}=$1\,\mathrm{m^2\ s^{-1}}$

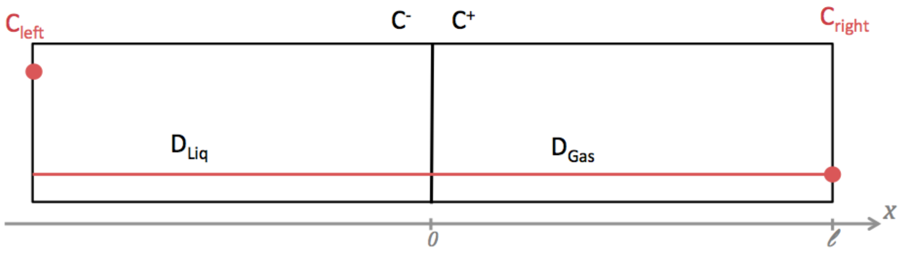


Figure 5.5: Finite tube with two phases, fixed concentration at the boundary (red dots), and initial conditions (red line)

Simulation

The domain is meshed in 2D with $300 \times 60 = 18,000$ cells. The simulation is run until $t = 100$s, with a time-step of $\delta t =$1 $\times\ 10^{-7}$ s.

Analytical solution

Although the tube is 2D, it is invariant along the y axis. Indeed the interface is perfectly vertical because of the 90°conctact angle. The concentration equation, in 1D and with diffusion only becomes: $\frac{\partial^2 C}{\partial x^2} = 0$. Therefore we have $C_{liq} = \gamma_0 x + \gamma_1$ in the liquid, and $C_{gas} = \beta_0 x + \beta_1$ in the gas. The boundary conditions with the solid and at the interface become:

$$\begin{cases} C_{liq}(-l) = C_{left} \\ C_{gas}(+l) = C_{right} \end{cases} \Rightarrow \begin{cases} -\gamma_0 l + \gamma_1 = C_{left} \\ \beta_0 l + \beta_1 = C_{right} \end{cases} \tag{5.1}$$

$$\begin{cases} C^- = HC^+ \\ D_l \nabla C^- = D_g \nabla C^+ \end{cases} \Rightarrow \begin{cases} \gamma_1 = H\beta_1 \\ D_{liq}\gamma_0 = D_{gas}\beta_0 \end{cases} \tag{5.2}$$

Solving this system we obtain the analytical solution:

$$\begin{aligned} C_{liq}(x) &= \frac{1}{H + D_{liq}/D_{gas}} \left[\frac{D_{liq}}{D_{gas}} (HC_R - C_L)\frac{x}{l} + HC_L + H\frac{D_{liq}}{D_{gas}} C_R \right], \\ C_{gas}(x) &= \frac{1}{H + D_{liq}/D_{gas}} \left[(HC_R - C_L)\frac{x}{l} + C_L + \frac{D_{liq}}{D_{gas}} C_R \right]. \end{aligned} \tag{5.3}$$

Comparison

We compare the analytical solution and the simulation results at steady state in Figure 5.6. We obtain an excellent match and the relative error is inferior to 1.5%.

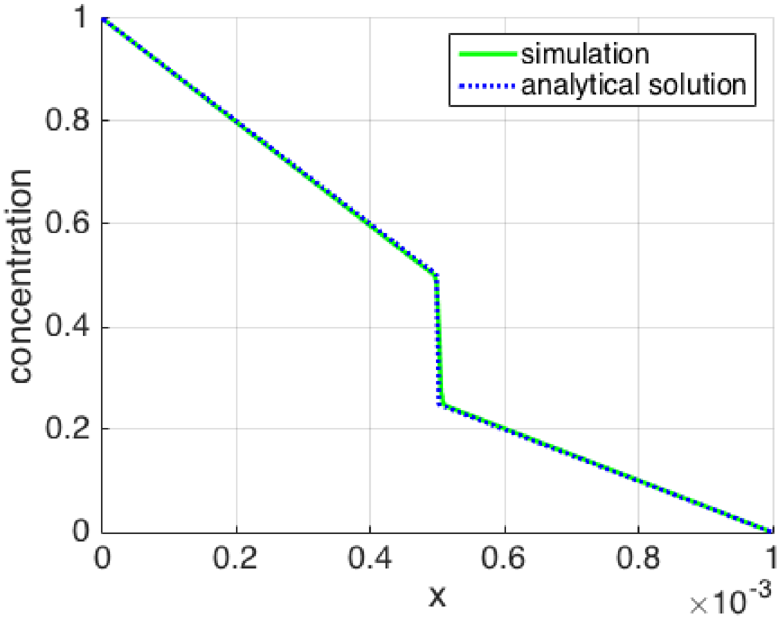


Figure 5.6: Comparaison of the concentration profile for the analytical solution (line) and the simulation results (dots) at steady state

5.2.2 Transient analytical solution in a tube

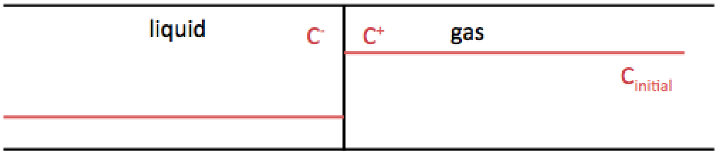


Figure 5.7: Initial conditions of the concentration in the tube (red lines)

We consider the same tube as in the previous section, when the two phases share the tube, and the contact angle is set to 90°. At time $t = 0$s, the concentration is

1 kg m^{-3} in the gas, and 0 kg m^{-3} in the liquid (see Figure 5.8). We set the diffusion parameters to $H = 2$ and $D_{liq} = D_{gas} = 1 \times 10^{-4}\,\mathrm{m}^2\ \mathrm{s}^{-1}$.

Simulation

A particular precision is required at the interface to obtain good results in the simulation, therefore the tube is meshed with a $12,000 \times 40 = 48,000$ cells grid. The time-step is $\delta t = 1 \times 10^{-7}\,\mathrm{s}$.

Analytical solution

In 1D and with no convection, the equation for the concentration in each phase becomes:

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2}. \tag{5.4}$$

We introduce the quantity $\eta = \frac{x}{\sqrt{Dt}}$, and we assume the solution is of the form $C(x,t) = F(\eta)$. Therefore, C is solution if and only if:

$$-\frac{x}{2\sqrt{D}t^{3/2}}F' = \frac{1}{Dt}F''. \tag{5.5}$$

$$\Rightarrow -\frac{1}{2}\eta F' = F'' \tag{5.6}$$

We integrate this equation and obtain the form of the concentration in each phase:

$$C = \begin{cases} \beta_0 erf(\eta_{gas}/2) + \beta_1 & \text{in the gas } x > 0 \\ \gamma_0 erf(\eta_{liq}/2) + \gamma_1 & \text{in the liquid } x < 0 \end{cases} \tag{5.7}$$

We consider the tube to be infinite. This is a fair assumption if we only look at very early times when the change in concentration is located very close to the interface and very far away from the boundaries. Boundary conditions (Henry's law and equality of fluxes) and initial conditions define our coefficients γ and β through the system of equations:

$$\begin{cases} C^{-}(t) = HC^{+}(t) \\ D_l \nabla C^{-}(t) = D_g \nabla C^{+}(t) \end{cases} \qquad \begin{cases} C(x < 0, t = 0) = C_0^{liq} \\ C(x > 0, t = 0) = C_0^{gas} \end{cases} \tag{5.8}$$

We have:

$$\begin{array}{ll} \lim_{\eta \to -\infty} erf(\eta/2) = -1 & \\ & \Rightarrow \\ \lim_{\eta \to +\infty} erf(\eta/2) \quad = 1 & \end{array} \begin{array}{l} -\gamma_0 + \gamma_1 = C_0^{liq} \\ \\ \beta_0 + \beta_1 = C_0^{gas} \end{array} \tag{5.9}$$

And the boundary conditions at the interface give:

$$\begin{gathered} \gamma_1 = H\beta_1 \\ \frac{2}{\sqrt{\pi}} \frac{1}{\sqrt{D_{liq}}} \gamma_0 = \frac{2}{\sqrt{\pi}} \frac{1}{\sqrt{D_{gas}}} \beta_0 \end{gathered} \tag{5.10}$$

Overall we find the coefficients defining the concentration profile in Eq. 5.7:

$$\begin{aligned} \gamma_0 &= \sqrt{\frac{D_{liq}}{D_{gas}}} \frac{HC_0^{gas} - C_0^{liq}}{H + \sqrt{\frac{D_{liq}}{D_{gas}}}} \\ \gamma_1 &= H \frac{\sqrt{\frac{D_{liq}}{D_{gas}}} C_0^{gas} + C_0^{liq}}{H + \sqrt{\frac{D_{liq}}{D_{gas}}}} \end{aligned} \tag{5.11}$$

$$\begin{aligned} \beta_0 &= \frac{HC_0^{gas} - C_0^{liq}}{H + \sqrt{\frac{D_{liq}}{D_{gas}}}} \\ \beta_1 &= \frac{\sqrt{\frac{D_{liq}}{D_{gas}}} C_0^{gas} + C_0^{liq}}{H + \sqrt{\frac{D_{liq}}{D_{gas}}}} \end{aligned} \tag{5.12}$$

Comparison

Simulation results and analytical solution are presented in Figure 5.8. As expected we immediately reach the jump of concentration at the interface. Simulation results are very close to the analytical solution, with an average relative error of 5%, mainly due to error near the interface: our interface is spread over the width of minimum one cell, and therefore not perfectly stiff.

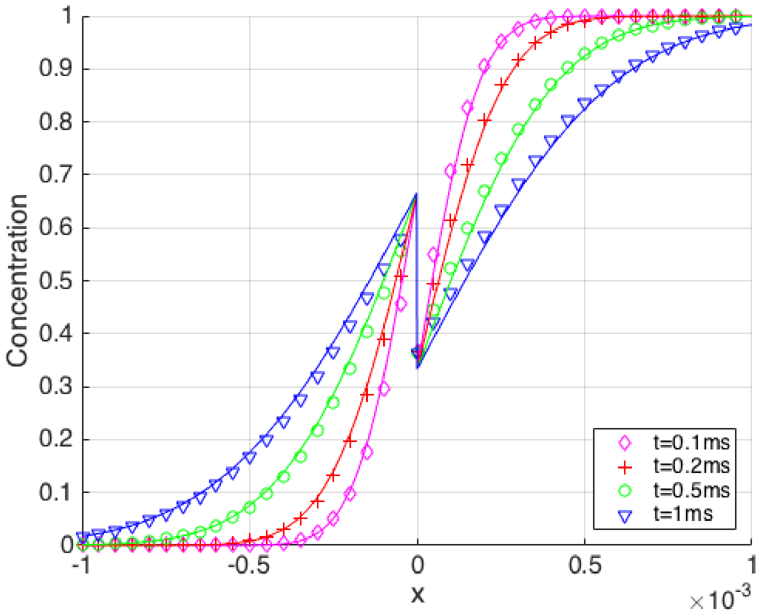


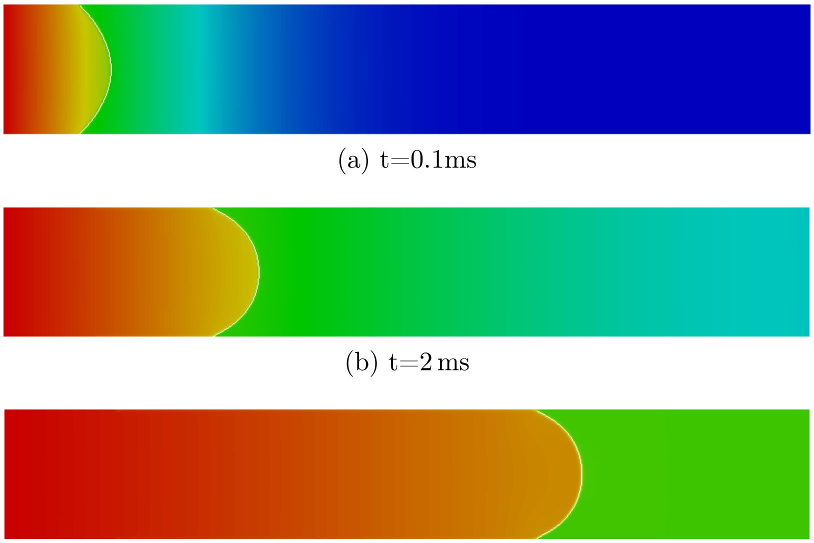
Figure 5.8: Comparison of analytical solution (straight lines) and simulation results (markers) for very early times (zoom in)

5.3 Injection of gas in a tube with concentration

The species transport solver can simulate two-phase with a miscible component, and in presence of a triple point solid/fluid/fluid. We simulate here the case of a tube, of dimension 0.1mm×0.01mm, with solid walls, initially occupied by water, and containing no tracer component. At time $t = 0$, gas is injected from the left, with a component of concentration $C = 1\,\mathrm{kg\ m^{-3}}$. Fluids have the following properties :

$\nu_l = 6 \times 10^{-5}\,\mathrm{m^2\ s^{-1}}$, $\nu_g = 6 \times 10^{-4}\,\mathrm{m^2\ s^{-1}}$, $\rho_l = \rho_g = 1000\,\mathrm{kg\ m^{-3}}$, $\sigma = 0.097\,\mathrm{kg\ s^{-2}}$ and the contact angle $\theta_0 = 45°$. The component diffusion has the following properties : $D_l = D_g = 1 \times 10^{-6}\,\mathrm{m^2\ s^{-1}}$ and $H = 0.7$.

The tube is meshed with a 200×30 grid. The simulation is run up to $t = 0.01$s, with a time-step δt=1×10^{-5} s. Results of the simulations are presented in Figures 5.9 and 5.10.



(a) t=0.1ms

(b) t=2 ms

(c) t=6 ms

Figure 5.9: Injection of gas with a miscible component in a tube. The concentration field is represented in color (blue for 0, red for $1\,\mathrm{kg\ m^{-3}}$), and the interface in white

We can see that at every time-step, the jump of concentration at the interface is respected, even close to the solid boundary. The simulation of this case is made possible by the implementation of the boundary condition for the concentration Eq. 3.34, with the CST term. This allows us to investigate more complex cases of two-phase flow, with simultaneously a triple point and a miscible in both phases.

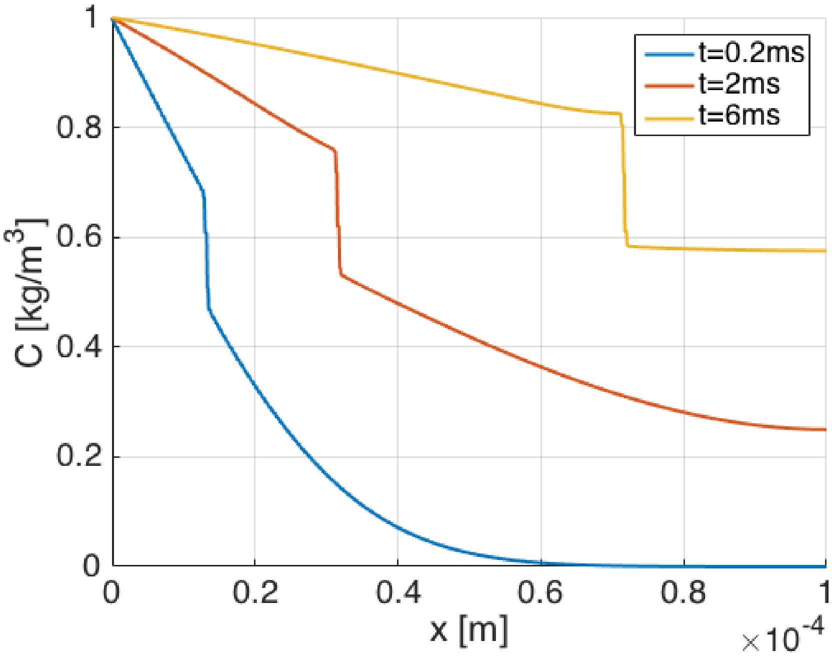


Figure 5.10: Concentration profile in the tube for different time. At every time, the jump of concentration at the interface ($H = 0.7$) is respected

5.4 Mass transfer coefficient for the injection of gas in a tube

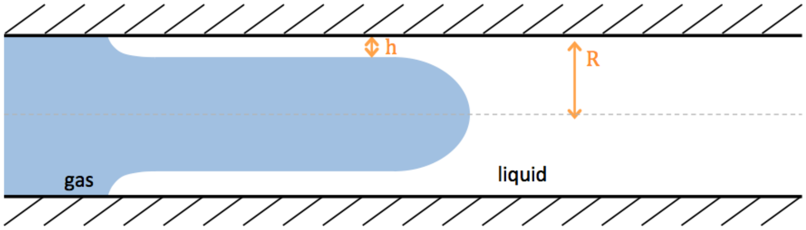
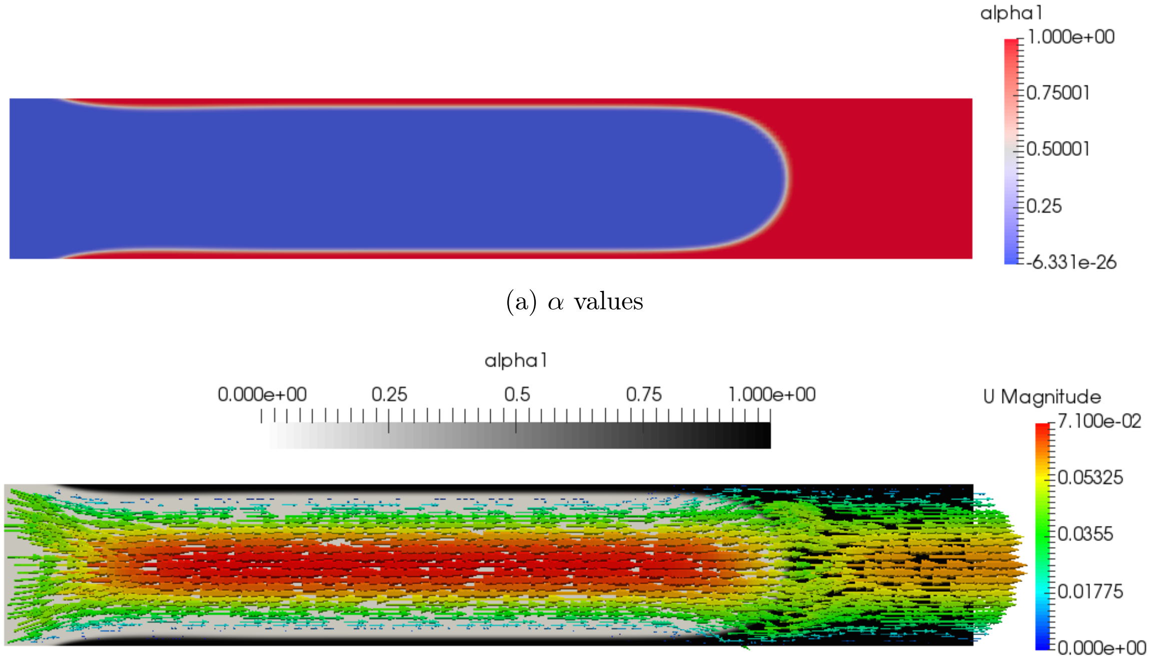


Figure 5.11: Illustration of the injection of a gas in a viscous liquid, leading to a remaining thin layer of liquid on the walls

When gas is injected into a tube filled with a more viscous fluid, it pushes the fluid out. Depending on the Reynolds number and the capillary number, it can lead to the deposition of a thin film of viscous fluid on the walls (see Figure 5.11). When the injected gas contains a miscible component, it will cross the interface and remain trapped in the thin film. We investigate the mass transfer between the flowing gas and the thin film and show that in this setting it is dependent on the Péclet number.

5.4.1 Fingering of the gas in a tube

We ran this simulation with the following parameters for the fluids $\nu_l = 6 \times 10^{-5}\,\mathrm{m}^2\ \mathrm{s}^{-1}$, $\nu_g = 6 \times 10^{-8}\,\mathrm{m}^2\ \mathrm{s}^{-1}$, $\rho_l = \rho_g = 1000\,\mathrm{kg\ m}^{-3}$, $\sigma = 0.097\,\mathrm{kg\ s}^{-2}$ and the contact angle is set at 20°. The tube's dimensions are $l = 12\,\mathrm{mm}$ and $R = 1\,\mathrm{mm}$. Gas is injected at a rate of $U_{left} = 0.04\,\mathrm{m\ s}^{-1}$. The main difference with the previous part, leading to the deposition of the thin film, is the value of the contact angle $\theta_0 = 20°$. Results at $t = 0.1\,\mathrm{s}$ are presented in Figure 5.12.



(a) α values

(b) Velocity field

Figure 5.12: Injection of gas in a tube

Taylor deduced from experiments an empirical law linking the ratio of layer width

h by the radius R directly to the capillary number Ca [68]:

$$\frac{h}{R} = \frac{1.34Ca^{2/3}}{1 + 3.35Ca^{2/3}}, \tag{5.13}$$

where the Capillary number is the ratio of the viscous forces to the capillary forces :

$$Ca = \frac{\mu_l \bar{U}_{gas}}{\sigma}, \tag{5.14}$$

where μ_l is the viscosity of the viscous fluid, σ is the surface tension, and $\bar{U}_{gas}$is the average velocity of the bubble of gas. In our particular simulation Taylor's law indicates $h/R = 9.59$ whereas we find $h/R = 9.36$, a relative error of 2%. Error possibly comes from the fact that the simulation was run on a 2D tube, whereas Taylor's experiments used cylindrical tubes.

5.4.2 Mass transfer coefficient

One application of the species transport model is the possibility to quantify the mass flux and the mass loss in dynamic case. We put in the steady case of the fingering in the tube: the thin film is stable. At time 0, a component with concentration $C = 1\,\text{kg m}^{-3}$ is injected from the left boundary. Henry's coefficient is 1. The flux from the flowing phase in the thin film is calculated from Eq. 3.32. The component moves by advection and diffusion in the flowing gas, but also crosses the interface and accumulates in the thin film.

We would like to quantify the mass transfer at the scale of the whole tube, this could help for example to improve a pore-network model of mass transfer with a thin film. To avoid boundary conditions effects, we only make calculations on the central part in the tube defined by the region Ω (Figure 5.13).

In Ω, we calculate the average velocity of the gas $\langle U_g \rangle^g$, the average concentration in the gas $\langle C_g \rangle^g$, the average concentration in the liquid $\langle C_l \rangle^l$, the total flux from the gas to the liquid F, and the effective interfacial area a_f. Most large-scale models of mass transfer try to relate the change in concentration to the difference of

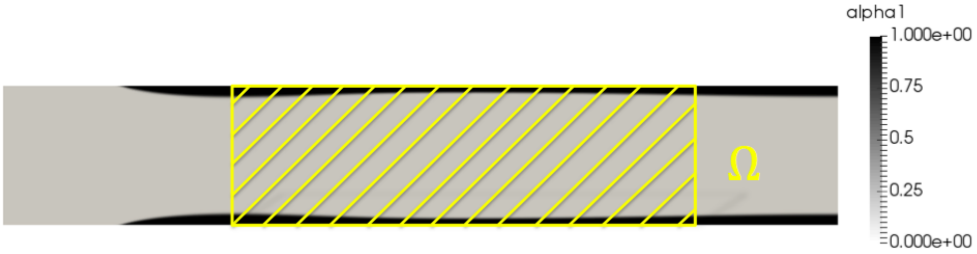


Figure 5.13: Phase distribution in the tube at steady state (gray scale), and zone of interest Ω or mass transfer calculations (yellow)

concentration through a mass exchange coefficient K :

$$F = K\left(H\langle C_g\rangle^g - \langle C_l\rangle^l\right) \tag{5.15}$$

The mass transfer is summed over the interface, and the mass transfer coefficient is expected to be of the form $K = a_f k$. For a diffusion coefficient equal in each phase to $1 \times 10^{-7}\,\mathrm{m}^2\,\mathrm{s}^{-1}$, we obtain the results plotted in Figure 5.14.

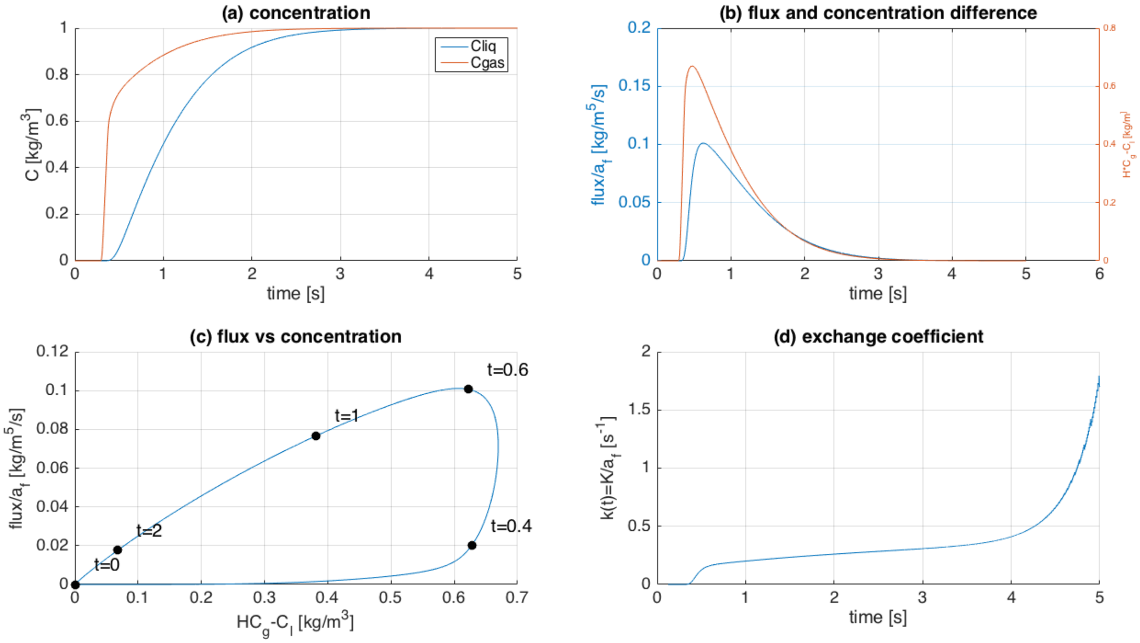


Figure 5.14: Concentration difference $HC_g - C_l$ and flux per interfacial area for D=$1 \times 10^{-7}\,\mathrm{m}^2\,\mathrm{s}^{-1}$

In Figure 5.14.(b) one can observe the similar behavior of the flux compared to the concentration difference. There seem to be several regimes. First for early times, the component has not reached yet the central part of the tube, all concentrations are equal to zero. Concentration reaches the area Ω at time 0.4s. The velocity of the gas is responsible for an increase of concentration in the gas phase. Until the first element of the component reaches the end of Ω, all the injected component either goes into the liquid or the gas, and the advection-diffusion ratio is distorted. After the breakthrough of the component, one can see in the plot (c) that the flux can be approximated linearly in function of the concentration difference, as in Figure 5.15. This gives an approximation of the mass exchange coefficient k. The high values of the mass exchange coefficient in the late time is due to computational error for very small numbers : both the flux and the concentration difference tend to zero.

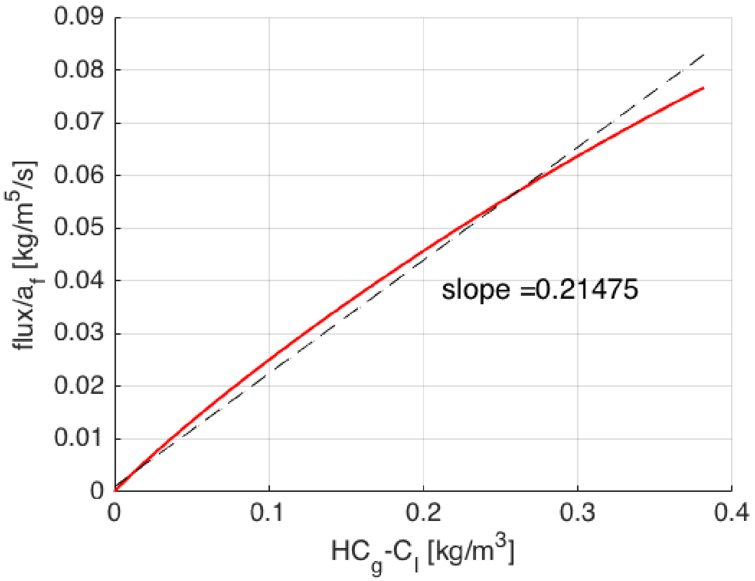


Figure 5.15: Flux per interfacial area vs. Concentration difference

The mass exchange coefficient is highly dependent on the velocity of the gas (advection) and the diffusion coefficient (diffusion). We ran the same simulation as previously for several values of the diffusion coefficient and compare them to the Péclet number. The Péclet number is defined as:

$$\mathrm{Pe} = \frac{\text{advective transport rate}}{\text{diffusive transport rate}} = \frac{LU}{D}, \tag{5.16}$$

where L is a characteristic length of the problem, U is the average velocity of the gas, and D the diffusion coefficient in the thin film. We observe that the mass exchange coefficient in this particular case is linearly dependent on the diffusion coefficient, and k is a function of the Péclet number in Figure 5.16.

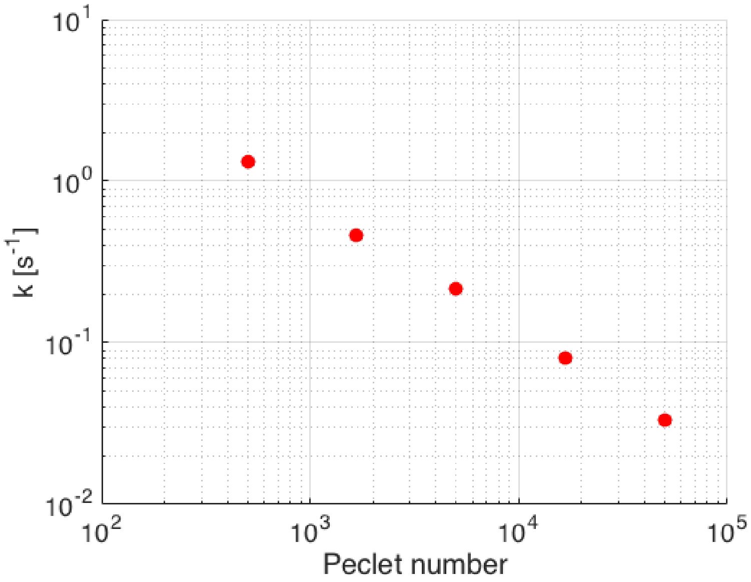
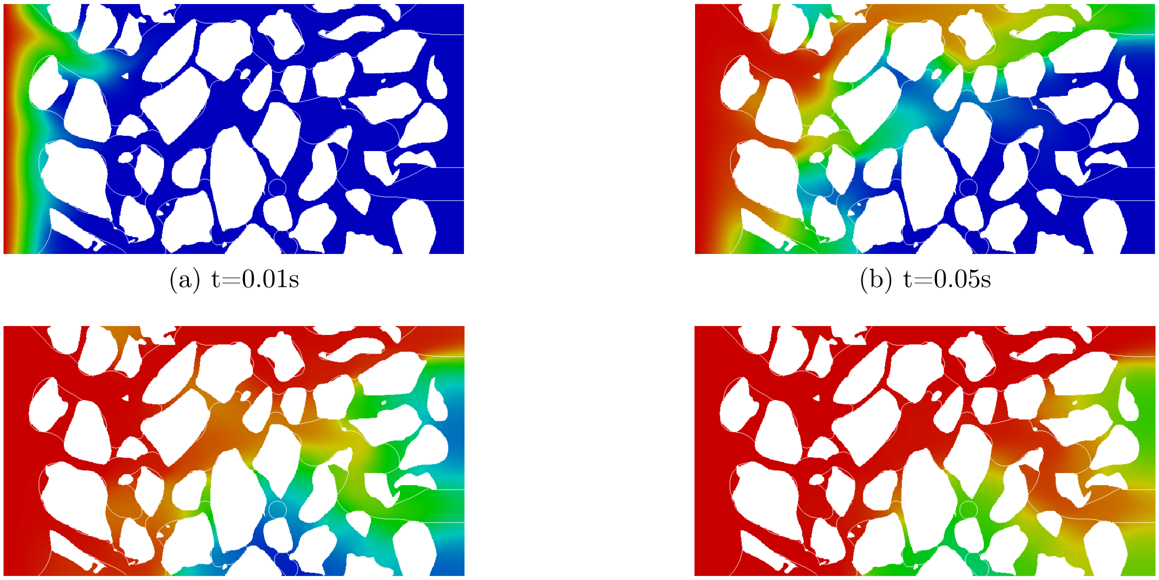


Figure 5.16: Mass exchange coefficient k in function of the Péclet number for the thin film in a tube case

5.5 Mass transfer coefficient in a complex porous media

We would like to evaluate the mass exchange coefficient for a complex porous medium. We use the same simulation as presented in Section 5.1. In the base case, we simulate the drainage of a non wetting fluid (say gas or oil) in a wetting fluid (oil or water). We have $\mu_g = \mu_l = 6 \times 10^{-2}\,\mathrm{kg\ m^2\ s^{-1}}$, $\rho_g = \rho_l = 1000\,\mathrm{kg\ m^{-3}}$, the surface tension $\sigma = 0.097\,\mathrm{kg\ s^{-2}}$, and the contact angle $\theta_0 = 45°$. The domain size is 850x400 µm. Steady-state for the flow is reached as presented in Figure 5.3.

For this, at time 0 a miscible component of concentration $1\,\mathrm{kg\,m^{-1}}$ is injected in the gas from the left. Henry's coefficient is $H_A = 1$, and the diffusion coefficient in both phases is $\mathrm{D} = 6 \times 10^{-5}\,\mathrm{m^2\ s^{-1}}$. Field concentration is presented in Figure 5.17.


(a) t=0.01s

(b) t=0.05s

(c) t=0.2s

(d) t=0.3s

Figure 5.17: Concentration field over time in a porous media with steady-state two-phase flow

The flux and difference of concentration in the two phases are plotted in Figure 5.18.

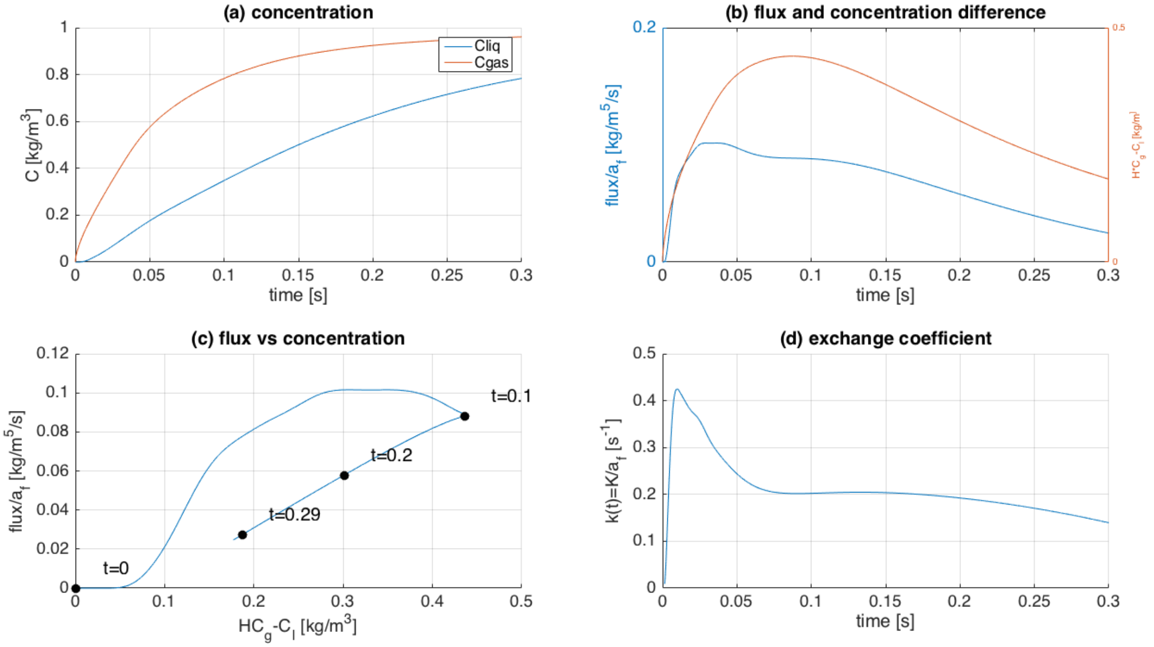


Figure 5.18: Concentration difference $HC_g - C_l$ and flux per interfacial area

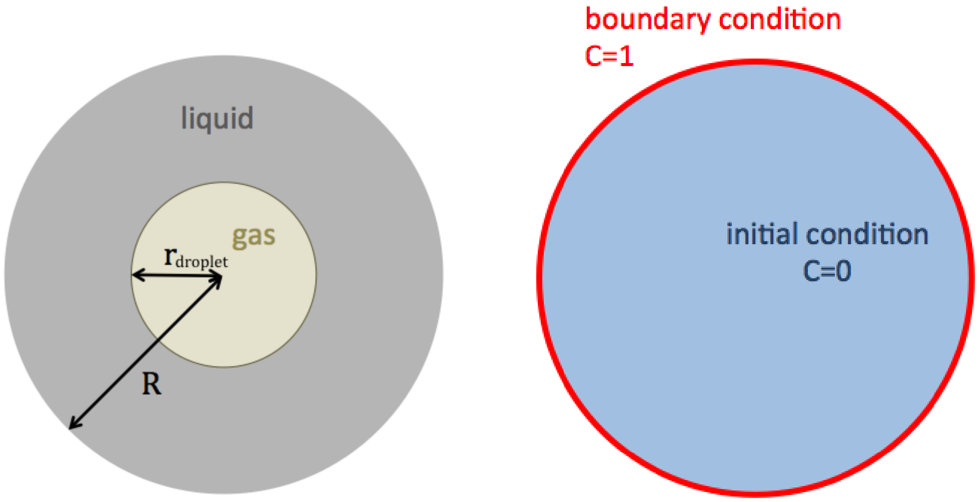
As for the tube and thin film setting, in early times, the flux behaves very nonlinearly. Until the first concentration reaches the outlet (t=0.1s), all the component going in is contributing to the concentration in the gas, so the ratio of advection per diffusion is distorted, and we ignore the data up to t=0.1s. From then, the mass exchange coefficient can be calculated from a regression on the flux per interfacial area vs. $\big(H\langle C_g\rangle^g - \langle C_l\rangle^l\big)$. We find :

$$k = 0.25\,\mathrm{s}^{-1} \tag{5.17}$$

5.6 Growth of a droplet of gas due to phase change

We want here to use the solver simulating phase change due to mass transfer at the interface. The setting is as simple as possible in order to obtain simplified PDEs, solvable numerically (for example with Matlab).

A droplet of gas is surrounded by liquid, at the equilibrium. The mesh is circular to have a cylindrical symmetry, and the boundary corresponds to a free flow (no solid, and no triple point). At the beginning of the simulation, the concentration is equal to zero in both fluids and is imposed at the boundary. Initial and boundary conditions are represented in Figure 5.19.



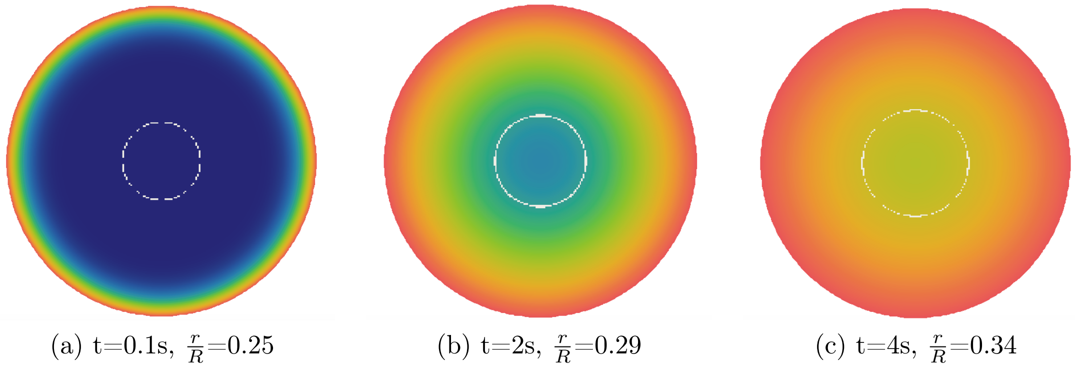
(a) initial repartition of phases

(b) IC and BC of concentration

Figure 5.19: Initial and boundary conditions

Not meant to be realistic, parameters are such as the equations governing the physics simplify. Gas and liquid have the same viscosity ($\mu = 6 \times 10^{-2}\,\mathrm{kg\ m^2\ s^{-1}}$) and density ($\rho = 1000\,\mathrm{kg\ m^{-3}}$). Since there is no flow, and both fluids are incompressible, these initial conditions would be stable, if not for the miscible component.

Concentration is fixed on the boundary condition at C=1 kg m^{-3}. In both fluids, the diffusion coefficient is $D = 1 \times 10^{-3}\,\mathrm{m^2\ s^{-1}}$, and Henry's coefficient is equal to 1. In that case, all CST terms (additional flux in the concentration equation, and the coefficient in the boundary condition on the concentration) are equal to 0, there is no jump at the interface, and the concentration is independent on the interface location. Moreover, there is no velocity due to the flow, and the concentration evolution is only driven by diffusion.



(a) t=0.1s, $\frac{r}{R}$=0.25 (b) t=2s, $\frac{r}{R}$=0.29 (c) t=4s, $\frac{r}{R}$=0.34

Figure 5.20: OpenFOAM simulation results of the growth of the droplet. Color represents the concentration (blue for 0, red for 1), and in white is the interface gas/liquid

5.6.1 VOF simulation

Results of the simulation are presented in Figure 5.20. Due to diffusion, concentration field increases in the liquid and crosses the interface to diffuse in the gas. The mass transfer induces a growth of the droplet of gas. We want to use another method to simulate this growth, in order to compare it with our VOF simulation.

5.6.2 Semi-analytical solution of the problem

The concentration evolution being driven solely by diffusion, and in the cylindrical geometry, the concentration C depends only on the radius r and the full PDE governing the concentration is:

$$\frac{\partial C}{\partial t} = D\frac{1}{r}\frac{\partial}{\partial r}r\frac{\partial C}{\partial r}, \tag{5.18}$$

$$C(r, t = 0) = C_0 \text{ initial condition}, \tag{5.19}$$

$$C(R, t) = C_R \text{ external boundary condition}, \tag{5.20}$$

$$\nabla C(0, t) = 0 \text{ center boundary condition}, \tag{5.21}$$

the last boundary condition being due to the cylindrical symmetry and the equality of fluxes. By separating the variables $C(r, t) = P(r)T(t)$ one obtains an analytical solution to this PDE:

$$C(r,t) = \sum_{m=1}^{+\infty} c_m e^{-\left(\frac{\alpha_m}{R}\right)^2 Dt} J_0\left(\alpha_m \frac{r}{R}\right), \tag{5.22}$$

where the (α_m) are the roots of the first Bessel function J_0. Coefficient c_m are determined by the equation at time 0 : $C_0 = \sum_{m=1}^{+\infty} c_m J_0\left(\alpha_m \frac{r}{R}\right)$. This analytical solution is obviously not easy to compute, and we prefer to solve the PDE, Eq. 5.18, numerically. It is discretized in 1D over N points with the boundary conditions. C_i^n is the concentration at time $t_n = n\Delta t$ at distance $r_i = i\Delta r$ from the center of the mesh:

$$\begin{aligned} \frac{\partial C_i}{\partial t} &= \frac{1}{r_i \delta r}\left[\left(r\frac{\partial C}{\partial r}\right)_{i+1/2} - \left(r\frac{\partial C}{\partial r}\right)_{i-1/2}\right] \\ &= \frac{1}{r_i \Delta r}\left[r_{i+1/2}C_{i+1} - \left(r_{i+1/2} + r_{i-1/2}\right)C_i + r_{i-1/2}C_{i-1}\right], \end{aligned} \tag{5.23}$$

$$\begin{aligned} C_{N+1} &= C_R \\ C_0 - C_1 &= 0 \end{aligned} \tag{5.24}$$

and is solved implicitly, which gives us a field of concentration. From this field, the ingoing flux can be calculated (positive flux of concentration going towards to center):

$$f_{i+1/2}^n = D\frac{C_{i+1}^n - C_i^n}{\Delta r}, \tag{5.25}$$

and then integrated over a circle of equal radius to obtain the total ingoing flux F in a disk of radius r_i:

$$F_{i+1/2}^n = 2\pi r_{i+1/2} D \frac{C_{i+1}^n - C_i^n}{\Delta r}. \tag{5.26}$$

The fluid being incompressible, the radius of the bubble is directly related to its mass : $M(t) = \rho\pi r_{bubble}^2(t)$. Therefore the change in radius of the droplet can be linked to the mass through $\frac{dM(t)}{dt} = F(r_{bubble}(t))$, which by substitution gives $\rho\pi r_{bubble}^2(t_{n+1}) - \rho\pi r_{bubble}^2(t_n) = \Delta t F_j^n$ where $j = \min_i \{r_{bubble}(t_n) < r_i\rangle$, and overall:

$$r_{bubble}(t_{n+1}) = \sqrt{r_{bubble}^2(t_n) + \frac{\Delta t}{\pi\rho}F_j^n} \tag{5.27}$$

5.6.3 Comparaison

The radius of the bubble in function of the time is plotted for the OpenFOAM simulation and the numerical PDE solution in Figure 5.21.

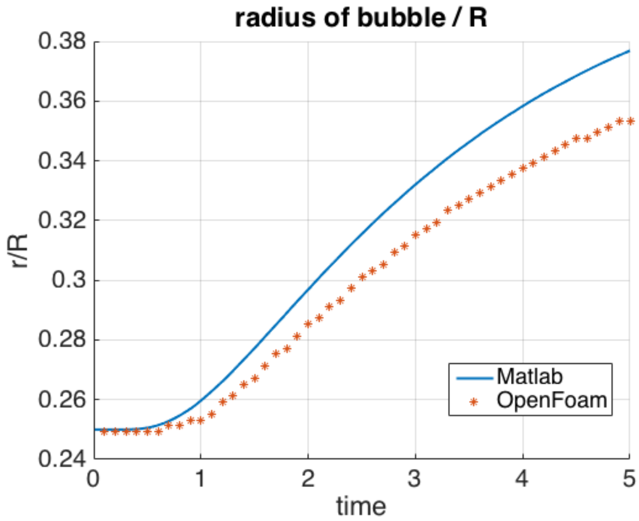


Figure 5.21: Comparaison of numerical PDE solution (line) and simulation results (dots) for the radius of the droplet

The general trend of evolution is the same in both cases, however, the droplet growth is slower in the OpenFOAM simulation. It looks that this difference does not come from an error in the computation of the flux, but a difference in the concentration field. We plot the concentration in function of the time for the midpoint (where $r = R/2$) in Figure 5.22. Concentration field differs from the solution of the PDE. We obtain similar results with different meshing (cylindrical symmetry, and finer grid), and when imposing a velocity to zero (to kill the spurious currents at the interface).

The general trend and order of magnitude of the evolution of the droplet are the expected one according to the equation we implemented. However, this does not tell us whether this model corresponds to the physics of the dissolution of a gas in a liquid. Indeed, here a chemical component dissolves in the gas and contributes to the gas phase, but the gas phase is not pure.

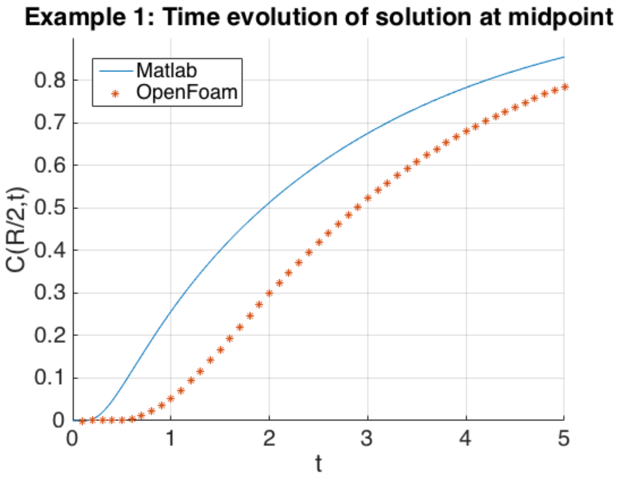


Figure 5.22: Midpoint (r=R/2) concentration for the PDE solution (line) and the simulation results (dots)

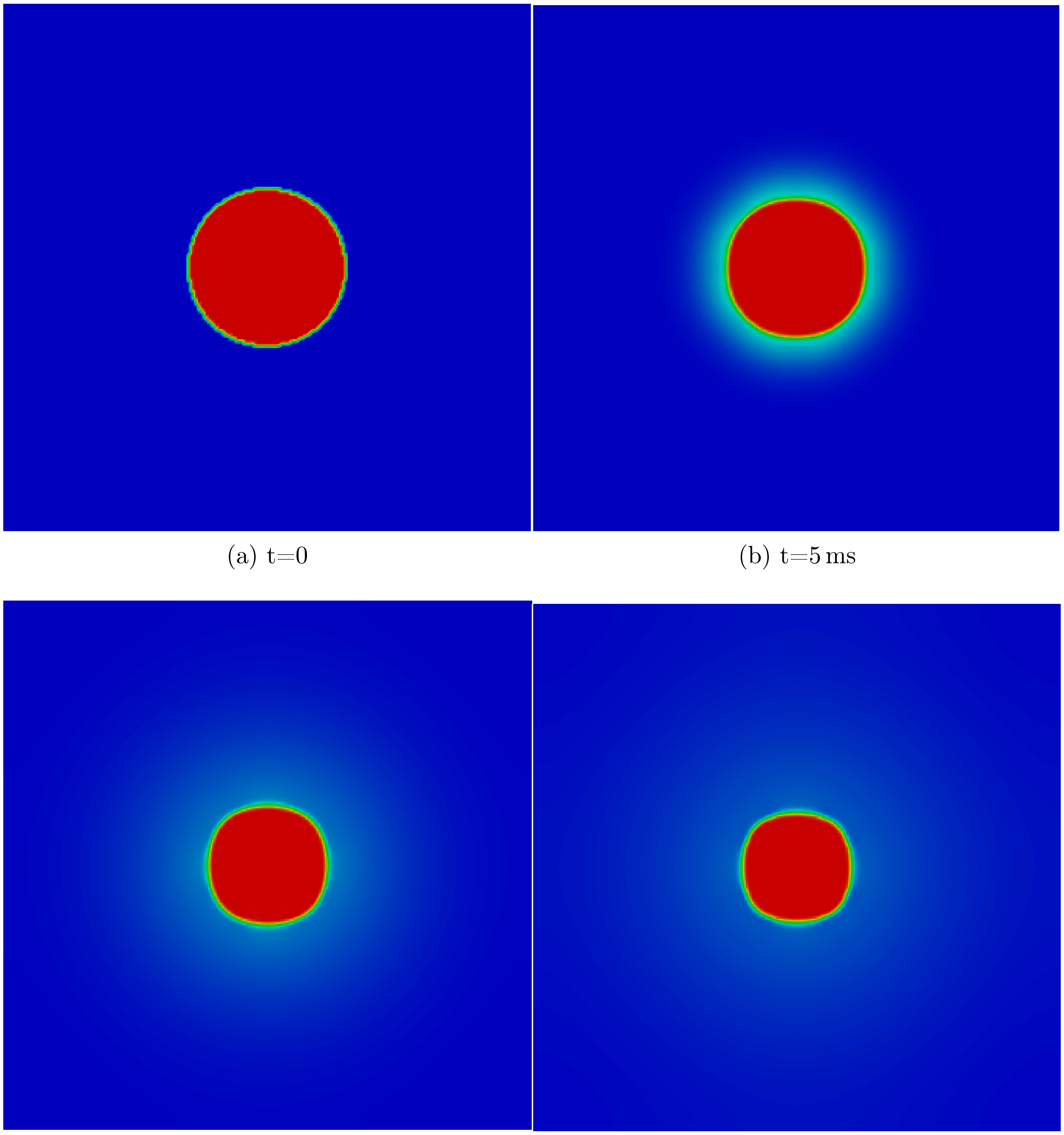
5.7 Dissolution of a droplet

This part is still an on-going work, and results are preliminary. However, the general trend of the dissolution of a pure gas is observed. In the case of a gas dissolving in the surrounding fluid, the concentration of the component in the droplet is equal to the density of the gas (it is the same chemical species). In that setting, the diffusion coefficient in the gas phase does not have any physical sense, the phase being pure. However, by setting it to zero, or a very low value $D_g << \min(D_l, 1)$, the equation on the concentration in the gas phase becomes:

$$\frac{\partial C_g}{\partial t} + \nabla . \left(\mathsf{v}_g C_g\right) = 0, \tag{5.28}$$

which is the same than the mass conservation equation for the density of the gas. Therefore, we expect the concentration to behave like the density in the droplet, and to behave like a solute element in the surrounding liquid. The results of the dissolution are presented in Figure 5.23.

Those are very preliminary results, and the dissolution of a pure gas in the surrounding fluid with this method has to be more investigated, with analytical solutions, and comparison with experimental results.



(a) t=0

(b) t=5 ms

(c) t=50 ms

(d) t=100 ms

Figure 5.23: Droplet diffusion over time. The concentration field is represented in color (blue for 0, red for 1 kg m^{-3})

Chapter 6

Conclusions

Two solvers were implemented, using the open source software OpenFOAM, to simulate two immiscible phases sharing the pore-space, and a miscible component present in both phase crossing the interface. OpenFoam uses the Volume-of-fluid formulation for two-phase flow. Following the guidelines of Haroun et al. [1], a model of a miscible component, with a discontinuity of concentration and equality of fluxes at the interface was developed, consistent with the VOF framework. It was extended to include the triple point boundary solid/fluid/fluid, and the phase change induced by mass transfer. The upscaling technique used to derive the equation on the concentration can be applied in a wide range of situations when a quantity is defined in two immiscible phases and has specific boundary conditions at the interface (possibly the solid/fluid interface).

The first solver simulates species transport when the component is considered inert and in a small quantity such that in does not affect the flow, like a tracer, of a chemical pollutant in the sub-surface. It gives results consistent with analytical solutions, giving concentration profiles with the adequate jump of concentration and the equality of fluxes at the interface. It is a powerful tool to quantify mass transfer in particular cases at the pore-scale: a tube with a thin film for pore-network models, porous medium from a real scanned rock, etc.

The phase change induced by mass transfer, like the dissolution of a gas phase in the surrounding liquid, has been implemented but is still an on-going work. Indeed, it needs further validation, both analytical and experimental, and is yet too unstable to be used as such. However, it could help us understand and investigate phenomena like Ostwald ripening in porous media.

A natural future development of this work would be the study of the effects of chemical components on the wettability. Indeed, based on the species transport solver, one could implement a model where the contact angle or the surface tension change with the concentration at the interface. Finally, the long-term objective is to upscale those changes at the reservoir scale and quantify how the flow is affected.

Appendices

Appendix A

Code structure in OpenFOAM

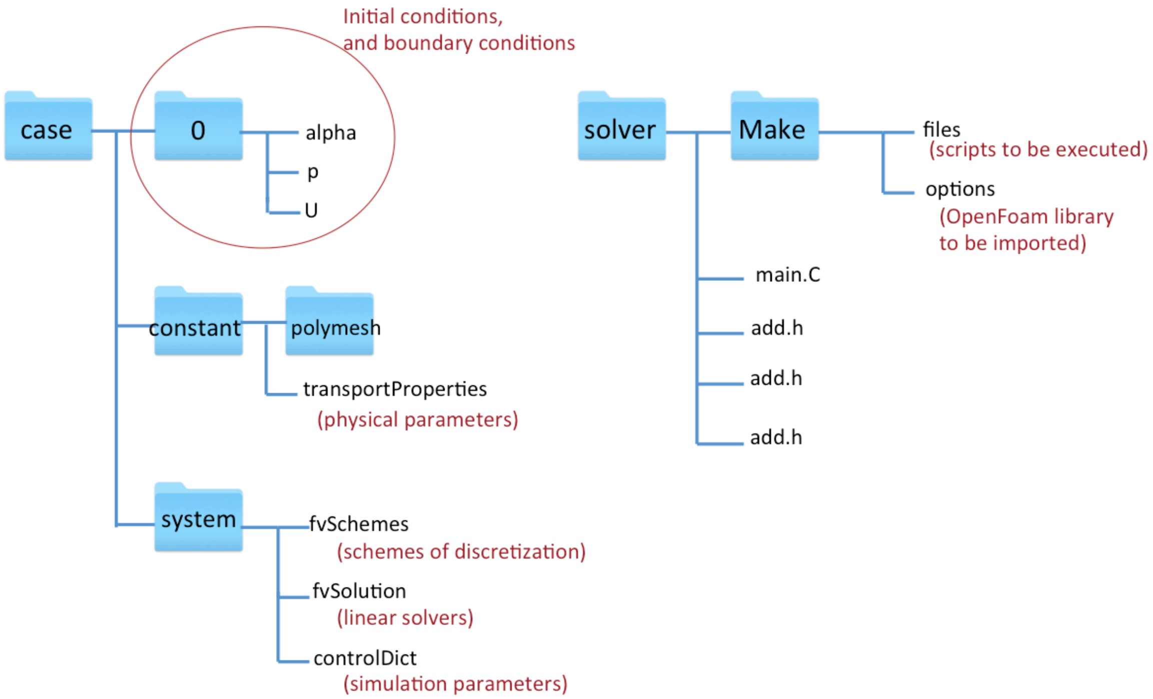


Figure A.1: Structure of a case and a solver in OpenFOAM

The OpenFOAM code is separated into the solvers which are specific of the physics to be solved, the algorithm to be used (multiphase flow, scalar transport, compressible gas, etc) and the cases which are specific domain and mesh, boundary and initial conditions, the scheme of discetizations, and parameters of the simulation (porous medium, turbine, tube, cavity, etc.).

On a case, one can use several solvers, however, each solver requires the user to input particular schemes of discretization, parameters of simulation, or physical quantities. The case, therefore, has to be adapted to the solver we want to use. As mentioned before, we based our work on the existing two-phase flow solver in OpenFOAM: *interFoam*. The objective of this appendix is to explain how the mathematical model presented in part 1 is implemented. First, we go through the code *interFoam*, then we describe our implementation of the concentration equation and the boundary condition for the concentration in the solver. The implementation of the phase change is also presented. Finally, we'll give a typical example of a case, on which the new solver can be launched (with, or without phase change).

A.1 Classes in OpenFOAM

Different classes are implemented in OpenFOAM. We only briefly describe some of them that are particularly used:

- `dimensionedScalar` : number, associated with a dimension. OpenFOAM checks dimension consistency and reports possible dimension errors.
- `volScalarField` : scalar defined at the center of each cell. It also includes its dimension (defined for example during the initiation of quantities), and its boundary conditions.
- `volVectorField` : Vector defined at the center of each cell, including its dimension and its boundary conditions
- `surfaceScalarField` : Scalar defined at the center of each faces, including its dimension and its boundary conditions

One can go from one class to the other with the operator `interpolate` (average at the faces of the centered values), `reconstruct` (average at the center from the faces values) and the inner product with the normal to the face times the surface of the face `& mesh.Sf()` (to compute fluxes)

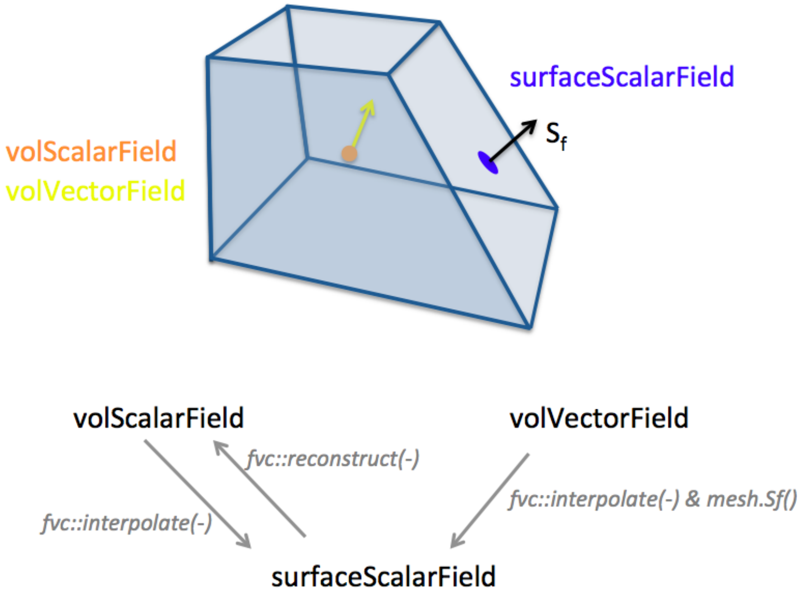


Figure A.2: classes location and operators

Appendix B

interFoam

We based our code on interFoam, the two-phase flow solver implemented in OpenFOAM. This appendix's objective is to make the parallel between the numerical equations, the PISO algorithm and the lines of code from OpenFOAM. We go through some of the scripts in the interFoam solver, mostly those that will need to be modified when implementing the phase change solver.

B.1 interFoam.C : main script

```
#include "fvCFD.H"
#include "MULES.H"
#include "subCycle.H"
#include "interfaceProperties.H"
#include "incompressibleTwoPhaseMixture.H"
#include "turbulenceModel.H"
#include "pimpleControl.H"
#include "fvIOoptionList.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
   * * * * * * * //

int main(int argc, char *argv[])
{
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"

    pimpleControl pimple(mesh);

    #include "initContinuityErrs.H"
    #include "createFields.H"
    #include "readTimeControls.H"
    #include "correctPhi.H"
    #include "CourantNo.H"
    #include "setInitialDeltaT.H"
```

Import useful libraries, set the `CourantNo` (the CFL), the initial time-step (when adaptative time-step).

In particular, it calls `createFields.H` that reads the initial conditions, boundary conditions and physical parameters of our case.

```
    Info<< "\nStarting␣time␣loop\n" << endl;

    while (runTime.run())
    {
        #include "readTimeControls.H"
        #include "CourantNo.H"
        #include "alphaCourantNo.H"
        #include "setDeltaT.H"

        runTime++;
```

Starts the time loop, calculate the flow CFL number, the alpha CFL number and set the time-step.

```
        #include "alphaEqnSubCycle.H"
        interface.correct();
```

Solves the equation on the phase indicator alpha

```
        // --- Pressure-velocity PIMPLE corrector loop
        while (pimple.loop())
        {
            #include "UEqn.H"

            // --- Pressure corrector loop
            while (pimple.correct())
            {
                #include "pEqn.H"
            }

        }
```

This corresponds to the PISO loop for the pressure-velocity coupling. The predictor `v`* is obtained in the `UEqn.h` then the pressure is deduced and the velocity is corrected in `pEqn.H`.

```
        runTime.write();

        Info<< "ExecutionTime␣=␣" << runTime.elapsedCpuTime()
           << "␣s"
            << "␣␣ClockTime␣=␣" << runTime.elapsedClockTime()
               << "␣s"
            << nl << endl;
    }

    Info<< "End\n" << endl;
```

Output the information about the current time-step.

B.2 createFields.H

This script is used once in the solver, at the very beginning. It assesses memory and creates quantities of interest. It basically reads the information in the case at time 0 (boundary and initial conditions) and in the `transportProperties` file.

```
volVectorField U
(
    IOobject
    (
        "U",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```

As an example, here is the reading of the velocity quantity U, that later will be saved at each time-step of interest.

```
Info<< "Reading␣transportProperties\n" << endl;
incompressibleTwoPhaseMixture twoPhaseProperties(U, phi);

volScalarField& alpha1(twoPhaseProperties.alpha1());
volScalarField& alpha2(twoPhaseProperties.alpha2());

const dimensionedScalar& rho1 = twoPhaseProperties.rho1();
const dimensionedScalar& rho2 = twoPhaseProperties.rho2();
```

Reads the transport properties parameters (density, viscosity) and the initial values for the phase indicator.

```
Info<< "Calculating␣field␣g.h\n" << endl;
volScalarField gh("gh", g & mesh.C());
surfaceScalarField ghf("ghf", g & mesh.Cf());

volScalarField p
(
    IOobject
    (
        "p",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    p_rgh + rho*gh
);
```

Finally, constructs the corrected pressure from the pressure and the gravity field $p' = p - \rho g h$. This is the pressure used and solved in the solver.

B.3 UEqn.H : predictor on the velocity

This script creates and solves the equation defining the predictor for the velocity.

```
fvVectorMatrix UEqn
(
    fvm::ddt(rho, U)
  + fvm::div(rhoPhi, U)
  + turbulence->divDevRhoReff(rho, U)
);
```

Left-hand side of the equation. Corresponds to $\rho\frac{\partial U}{\partial t}+\rho U.\nabla U-\nabla.\left(\mu\left(\nabla U+^{t}\nabla U\right)\right)$ however in a linearized form, to be solvable. The non-linear term is linearized by taking the mass flux at the center of the faces (`rhoPhi` $= (\rho \mathsf{v})_f \,.S_f$) from the previous time-step (or guess).

```
    {
        solve
        (
            UEqn
         ==
            fvc::reconstruct
            (
                (
                    fvc::interpolate(interface.sigmaK())*fvc::
                       snGrad(alpha1)
                  - ghf*fvc::snGrad(rho)
                  - fvc::snGrad(p_rgh)
                ) * mesh.magSf()
            )
        );
    }
```

Solves the equation for the velocity to find the predictor v^*. It computes the fluxes at the faces, (`interpolate`) then reconstruct their value at the center of the cells (`reconstruct`) to be used in the equation.

$$\begin{aligned}
\texttt{fvc::interpolate(interface.sigmaK())*fvc::snGrad(alpha1)} &\leftrightarrow \left(\sigma\nabla.\left(\frac{\nabla\alpha}{\|\nabla\alpha\|}\right)\nabla\alpha\right)_f \\
\texttt{- ghf*fvc::snGrad(rho)} &\leftrightarrow (-\rho^n\mathbf{g})_f \\
\texttt{- fvc::snGrad(p\_gh)} &\leftrightarrow (-\nabla p^n)_f \\
\texttt{mesh.magSf()} &\leftrightarrow .S_f
\end{aligned}$$

B.4 Note on explicit / implicit discretization

The discretization of a term can be implicit (any term of the velocity in the previous equation), or explicit (computation of flux). This in OpenFOAM is dealt with two libraries : `fvm` and `fvc`

- `fvc` library has operators of explicit derivation. In particular is create a field where everything is calculated from known quantities, for example, `fvc::snGrad(p_rgh)` calculates the explicit gradient of the pressure.
- `fvm` library has operator of implicit derivation. The quantity `fvm::ddt(rho, U)` does not contain any actual value, since U is the unknown of the equation. It's a matrix, containing the coefficients due to spatial or temporal discretization.

B.5 pEqn.H : pressure guess and velocity corrector

`phi` always refers to a flux computed at the center of the faces. We have `phi` $= (\nabla.U)_f .S_f$ and `rhoPhi` $= (\nabla.(\rho U))_f .S_f$.

```
volScalarField rAU("rAU", 1.0/UEqn.A());
surfaceScalarField rAUf("Dp", fvc::interpolate(rAU));

volVectorField HbyA("HbyA", U);
HbyA = rAU*UEqn.H();

surfaceScalarField phiHbyA
(
    "phiHbyA",
    (fvc::interpolate(HbyA) & mesh.Sf())
  + fvc::ddtPhiCorr(rAU, rho, U, phi)
);
```

Since a is a diagonal matrix, it can also be seen as a vector with one value per cell (the diagonal one), and therefore be described by a quantity at the center of the cells (`rAU`), and then be interpolated to the faces (`rAUf`). Similarly, `HbyA` is the operator H applied to the velocity, with the inverse of a as a factor. It is also then be interpolated to the faces.

$$\begin{aligned} \texttt{rAU} \Leftrightarrow &\quad a^{-1} \\ \texttt{rAUf} \Leftrightarrow &\quad \left(a^{-1}\right)_f .S_f \\ \texttt{HbyA} \Leftrightarrow &\quad a^{-1}\mathbf{H}(\mathsf{v}) \\ \texttt{phiHbyA} \Leftrightarrow &\quad \left(a^{-1}\mathbf{H}(\mathsf{v})\right)_f .S_f \end{aligned} \tag{B.1}$$

```
surfaceScalarField phig
(
    (
        fvc::interpolate(interface.sigmaK())*fvc::snGrad(
           alpha1)
      - ghf*fvc::snGrad(rho)
    )*rAUf*mesh.magSf()
);

phiHbyA += phig;
```

In the pressure equation, the remaining terms are computed as fluxes at the center of the faces, and named `phig`. Summing them gives us the final HbyA.

$$\texttt{phiHbyA} \Leftrightarrow \quad \left(a^{-1}\mathbf{H}(\mathsf{v})\right)_f .S_f + a^{-1}\left(-\rho^n\mathbf{g} + \left(\sigma\nabla.\left(\frac{\nabla\alpha}{\|\nabla\alpha\|}\right)\nabla\alpha\right)^n\right)_f .S_f \tag{B.2}$$

```
    fvScalarMatrix p_rghEqn
    (
        fvm::laplacian(rAUf, p_rgh) == fvc::div(phiHbyA)
    );
```

Solve the pressure equation 4.6.

```
    if (pimple.finalNonOrthogonalIter())
    {
        phi = phiHbyA - p_rghEqn.flux();

        U = HbyA + rAU*fvc::reconstruct((phig - p_rghEqn.
           flux())/rAUf);
        U.correctBoundaryConditions();
        fvOptions.correct(U);
    }
```

Correct the fluxes due to pressure effects $\tilde{F} = F - \left[a^{-1}\nabla p^*\right]_f .S_f$

Correct the cell-centered velocity due to the new pressure distribution: $\mathsf{v}^{**} = a^{-1}\mathbf{H}\left(\mathsf{v}^{*}\right) - a^{-1}\nabla p^*$

B.6 alphaEqn.H

Due to the stiffness of the interface, time-step is usually too large to directly solve the equation on alpha. Therefore, time-step is divided into `nAlphaSubCycles` cycles, where alpha is solved at each iteration. Contributions to the flux are summed at every sub-cycles.

The equation on alpha by itself is included in the `alphaEqn.H` script, in particular this approximation of the compressive velocity v_r defined in 3.15

```
word alphaScheme("div(phi,alpha)");
word alpharScheme("div(phirb,alpha)");

surfaceScalarField phic(mag(phi/mesh.magSf()));
phic = min(interface.cAlpha()*phic, max(phic));
surfaceScalarField phir(phic*interface.nHatf());
```

$$\begin{aligned}
\texttt{phic} \Leftrightarrow & \qquad \frac{|\phi|}{|S_f|} \\
\texttt{phic} \Leftrightarrow & \quad \min\left[C_\alpha \frac{|\phi|}{|S_f|}, \max_F\left(\frac{|\phi|}{|S_F|}\right)\right] \\
\texttt{phic} \Leftrightarrow & \qquad \left(\mathsf{v}_r\right)_f . S_f
\end{aligned} \tag{B.3}$$

```
surfaceScalarField phiAlpha
(
    fvc::flux
    (
        phi,
        alpha1,
        alphaScheme
    )
  + fvc::flux
    (
        -fvc::flux(-phir, alpha2, alpharScheme),
        alpha1,
        alpharScheme
    )
);

MULES::explicitSolve(alpha1, phi, phiAlpha, 1, 0);
```

Flux term in the alpha equation is computed : `phiAlpha` $\Leftrightarrow \nabla.(\alpha\bar{\mathsf{v}})+\nabla.(\alpha\,(1-\alpha)\,\bar{\mathsf{v}}_r)$. Finally, the equation on the phase indicator is solved using the solver MULES.

Appendix C

myComponentFoam

The species transport model is implemented in OpenFoam in a new solver named myComponentFoam. Changes from interFoam are highlighted in red in Figure C.1. Since myComponentFoam only describe species transport and not phase change, the flow is not affected by the concentration, and the equations on the velocity, the pressure and the phase indicator don't need to be modified.

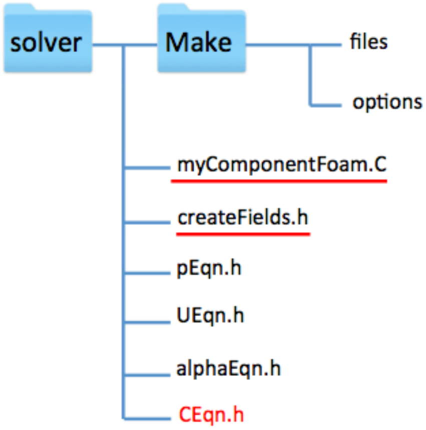


Figure C.1: myComponentFoam structure

C.1 myComponentFoam.C : main script

The equation on C has to be included in the main time loop. Since it depends on the velocity and the interface location, it was put just after the PISO loop.

```
// --- Pressure-velocity PIMPLE corrector loop
while (pimple.loop())
{

    #include "UEqn.H"

    // --- Pressure corrector loop
    while (pimple.correct())
    {
        #include "pEqn.H"
    }

}

#include "CEqn.H"
```

C.2 createFields.h

However, to include the concentration in the solver, one has to define this variable, along with the physical quantities associated (diffusion coefficients and Henry's law coefficient). One has to add to following pieces of code to `createFields.h` :

```
Info<< "Reading␣field␣C\n" << endl;

volScalarField C
(
    IOobject
    (
        "C",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```

Goes read in the case directory at time 0 the initial and boundary conditions of the concentration.

```
volScalarField my_flux  ("my_flux", rho*((C*U - Diff*fvc::
   grad(C)) & fvc::grad(alpha2)));
```

Allocating memory for the flux (the initial value is not important, as long as it is defined with the right dimensions, because it is recomputed later).

```
Info<< "Reading␣transportProperties\n" << endl;

    IOdictionary transportProperties
(
    IOobject
    (
        "transportProperties",
        runTime.constant(),
        mesh,
        IOobject::MUST_READ_IF_MODIFIED,
        IOobject::NO_WRITE
    )
);

    incompressibleTwoPhaseMixture twoPhaseProperties(U, phi);

volScalarField& alpha1(twoPhaseProperties.alpha1());
volScalarField& alpha2(twoPhaseProperties.alpha2());

const dimensionedScalar& rho1 = twoPhaseProperties.rho1();
const dimensionedScalar& rho2 = twoPhaseProperties.rho2();

    dictionary phase1 = transportProperties.subDict("phase1");
const dimensionedScalar nu1 = phase1.lookup("nu");
dictionary phase2 = transportProperties.subDict("phase2");
const dimensionedScalar nu2 = phase1.lookup("nu");
```

```
        dimensionedScalar DT1
    (
        transportProperties.lookup("DT1")
    );

    dimensionedScalar DT2
    (
        transportProperties.lookup("DT2")
    );

    dimensionedScalar H
    (
        transportProperties.lookup("H")
    );

        volScalarField Diff ("Diff", DT1*alpha1 + DT2*alpha2);

        dimensionedScalar Cs
    (
        transportProperties.lookup("Cs")
    );
```

When reading the transportProperties file in the case, not only gets the solver the values for the two-phase flow, but it also gets the diffusion coefficient and the Henry coefficient. The scalar C_S is a smoothing parameter that can help reduce the spurious current at the interface.

```
    Info<< "Reading␣field␣chi\n" << endl;
volScalarField chi
(
    IOobject
    (
        "chi",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```

Finally, we define a new indicator χ. This value remains unchanged in the code, it only specifies a particular region of the mesh where we want to do our calculations (for example, in a tube, taking into account only the central part, and leave out the input and output zone). If no further calculation is needed, one can simply set it to the uniform value of 1.

The rest of `createFields.H` remains unchanged.

C.3 CEqn.h

Finally, the equation on the concentration is introduced and solved fully implicitly.

```
Diff = DT1*alpha1 + DT2*(1.-alpha1);

    volScalarField phiH ("phiH",rho*Diff*(H1-H2)/((1.-
       alpha2)*H2+(alpha2)*H1));

    surfaceScalarField phiHf ("phiHf", fvc::interpolate(
       phiH)*fvc::snGrad(alpha2)*mesh.magSf());
```

First, the global diffusion coefficient is computed. Then the CST term is computed at the center of the cell, then extrapolated to the faces.

$$\begin{aligned}\texttt{Diff} &= \alpha D_1 + \left(1-\alpha\right) D_2 \\ \texttt{phiH} &= \rho D \frac{\omega_A \left(H_A - 1\right)}{\alpha + H_A \left(1-\alpha\right)} \\ \texttt{phiHf} &= \left(\rho D \frac{\omega_A \left(H_A - 1\right)}{\alpha + H_A \left(1-\alpha\right)} \nabla\alpha\right)_f .S_f\end{aligned} \tag{C.1}$$

```
    fvScalarMatrix CEqn
    (
        fvm::ddt(rho, C)
      + fvm::div(rhoPhi, C)
      - fvm::laplacian(rho*Diff, C)
              + fvm::div(phiHf,C)
    );

    CEqn.solve();
```

The equation on the concentration is discretized fully implicitly and solved.

```
        surfaceScalarField phiCC ("phiCC", fvc::interpolate(rho
           *Diff*(1.-H)*C  /(alpha1+(1.-alpha1)*H)) * fvc::
           snGrad(alpha1) * mesh.magSf());

        my_flux = fvc::reconstruct(CEqn.flux() + phiCC) & fvc::
           grad(alpha1);
```

The flux of mass of the miscible component at the interface $\dot{m}$ is calculated from the new value of the concentration.

$$
\begin{aligned}
\texttt{phiCC} &\Leftrightarrow \left(\rho D \frac{1-H}{\alpha+(1-\alpha)H}\nabla\alpha\right)_f .S_f \\
\texttt{my\_flux} &\Leftrightarrow \dot{m} \Leftrightarrow \left(C(\mathsf{v}-\mathsf{w}) - D\nabla C\right).\nabla\alpha + \frac{H-1}{\alpha H + (1-\alpha)} DC\nabla\alpha
\end{aligned}
\tag{C.2}
$$

```
        if(runTime.outputTime())
        {
                my_flux.write();
        }
```

The flux values are stored, so they can be post-processed if necessary (in preview for example).

C.4 Boundary condition

The last element of the model is to implement the new boundary condition. Boundary conditions are quite deeper in the OpenFoam code, and adding one requires several changes. However, one does not need to go in the OpenFoam classes, we will add an artificial new class in our current solver.

It has to be included in the Make/file.

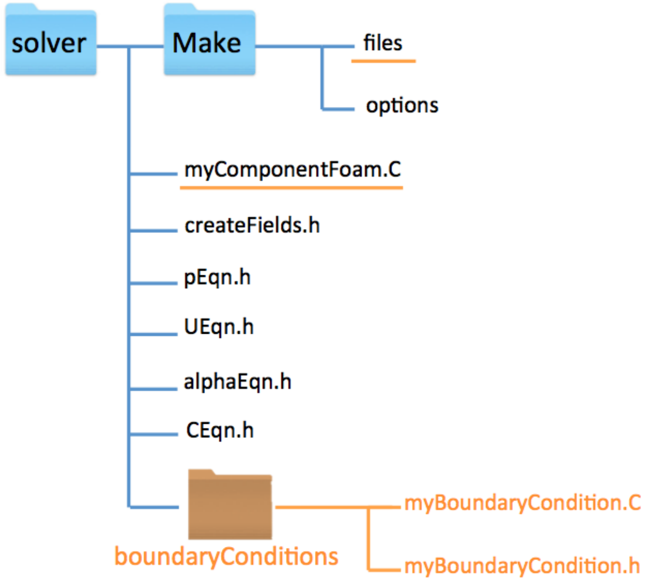


Figure C.2: Change in the solver to implement a new boundary condition

Listing C.1: Make/file

```
boundaryConditions/myBoundaryCondition.C
myComponentFoam.C

EXE = $(FOAM_USER_APPBIN)/myComponentFoam
```

And as a header in the main script.

Listing C.2: myComponentFoam.C

```
#include "boundaryConditions/myBoundaryCondition.H"
```

Appendix D

myinterPhaseChangeFoam

The phase change is implemented on the same base than the species transport (solver myComponentFoam). However, phase change induces several additional terms in the equations on the pressure and the phase indicator. Most of this work is related to the implementation of the phase change due to the temperature in the OpenFoam solver (interPhaseChangeFoam). Changes from myComponentFoam are highlighted in green in figure D.1.

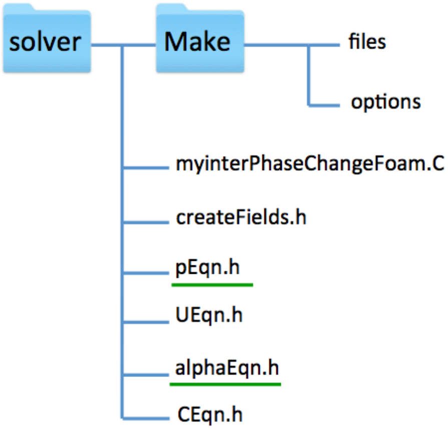


Figure D.1: myComponentFoam structure

D.1 pEqn.H

```
        fvScalarMatrix p_rghEqn
        (
            fvc::div(phiHbyA) - fvm::laplacian(rAUf, p_rgh) -
               flux*(1.0/rho1 - 1.0/rho2)

        );
```

The equation on the pressure now includes the additional term from the equation described in the PISO algorithm (eq 4.6). As in the species transport solver, the flux is calculated directly after solving the equation on the concentration in CEqn.H.

D.2 alphaEqn.H

```
                volScalarField Sp ("Sp", (1.0/rho1 * flux) / (
                   alpha1 +SMALL));
                volScalarField Su ("Su", 0*divU );

                MULES::explicitSolve
                (
                        geometricOneField(),
                        alpha1,
                        phi,
                        phiAlpha,
                        Sp,
                        Su,
                        1,
                        0
                );
```

The source term in the alpha equation is taken into account implicitly, since in our case the gas is dissolving into the liquid, so that the flux $\dot{m}$ from liquid to gas is negative. It's then solves using the MULES solver.

Appendix E

Running a case

When installing OpenFoam, a folder of tutorials comes along. In particular, for interFoam, the case of a damBreak is available. We describe here what has to be modified or added to a case on which interFoam runs (like the damBreak), for it to be usable by myComponentFoam and myinterPhaseChangeFoam. Mostly it requires new initial and boundary conditions (for the concentration), physical parameters (Henry coefficient, diffusion coefficients), but also schemes of discretization (for concentration and flux related quantities).

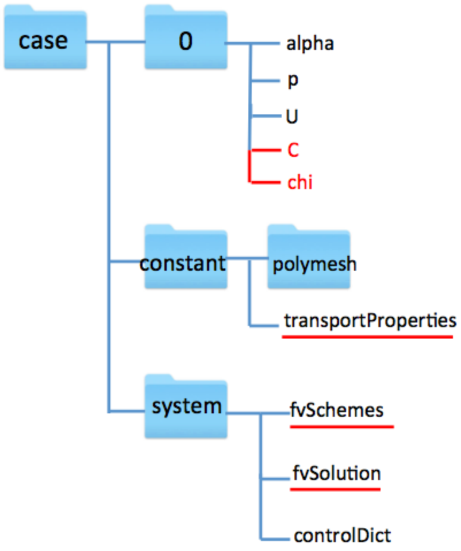


Figure E.1: case structure

E.1 transportProperties

In addition to the surface tension and the two phases properties, Henry coefficient, the diffusion coefficient and the smoothing coefficient C_S have to be defined :

```
H                H [0 0 0 0 0 0 0] 1;

DT1              DT1 [0 2 -1 0 0 0 0] 1e-7;

DT2              DT2 [0 2 -1 0 0 0 0] 1e-7;

Cs               Cs [0 0 0 0 0 0 0] 0;
```

E.2 fvSchemes

Definition of the schemes of discretization for the concentration related quantities:

```
divSchemes
{
    default              none;
    div(rhoPhi,U)        Gauss linearUpwind grad(U);
    div(phi,omega)       Gauss linearUpwind grad(omega);
    div(phi,k)           Gauss linearUpwind grad(k);
    div(phi,alpha)       Gauss vanLeer;
    div(phirb,alpha)     Gauss interfaceCompression;
    div((muEff*dev(T(grad(U))))) Gauss linear;
        div(phiHf,C)       Gauss vanLeer;
    div(rhoPhi,C)      Gauss vanLeer;
    div(rho*phi,C)         Gauss vanLeer;
        div(rho*phi,U)          Gauss upwind;
        div(phi,rho1)           Gauss upwind;
    div(phi,rho2)               Gauss upwind;
}
```

And the flux of component can be directly computed from the equation on the concentration by adding:

```
fluxRequired
{
        C;
}
```

E.3 fvSolution

Finally, since we defined new variables, we have to define a new solver for the concentration in the solver section in the fvSolution file:

```
ddtSchemes
{
    default               Euler;
}

interpolationSchemes
{
    default               linear;
}
```

Bibliography

[1] Y. Haroun, D. Legendre, and L. Raynal. Volume of fluid method for interfacial reactive mass transfer: Application to stable liquid film. *Chemical Engineering Science*, 65(10):2896‚Äì2909, May 2010.

[2] Carl I Steefel, Lauren E Beckingham, and Gautier Landrot. Micro-continuum approaches for modeling pore-scale geochemical processes. *Rev Mineral Geochem*, 80:217–246, 2015.

[3] Douglas Wendle Waples. *Organic geochemistry for exploration geologists*. Burgess Pub. Co., 1981.

[4] Yossi Cohen and Daniel H Rothman. Mechanisms for mechanical trapping of geologically sequestered carbon dioxide. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 471, page 20140853. The Royal Society, 2015.

[5] Sergi Molins. Reactive interfaces in direct numerical simulation of pore scale processes. *Rev Mineral Geochem*, 80:461–481, 2015.

[6] Cyprien Soulaine and Hamdi A. Tchelepi. Micro-continuum approach for pore-scale simulation of subsurface processes. *Transport In Porous Media*, 2016.

[7] Crist Khachikian and Thomas C Harmon. Nonaqueous phase liquid dissolution in porous media: Current state of knowledge and research needs. *Transport in Porous Media*, 38(1-2):3–28, 2000.

[8] Berken Agaoglu, Nadim K Copty, Traugott Scheytt, and Reinhard Hinkelmann. Interphase mass transfer between fluids in subsurface formations: A review. *Advances in Water Resources*, 79:162–194, 2015.

[9] Michel Quintard and Stephen Whitaker. Convection, dispersion, and interfacial transport of contaminants: Homogeneous porous media. *Advances in Water Resources*, 17(4):221 – 239, 1994.

[10] Persefoni E. Kechagia, Ioannis N. Tsimpanogiannis, Yanis C. Yortsos, and Peter C. Lichtner. On the upscaling of reaction-transport processes in porous media with fast or finite kinetics. *Chemical Engineering Science*, 57(13):2565 – 2577, 2002.

[11] F.A. Coutelieris, M.E. Kainourgiakis, A.K. Stubos, E.S. Kikkinides, and Y.C. Yortsos. Multiphase mass transport with partitioning and inter-phase transport in porous media. *Chemical Engineering Science*, 61(14):4650 – 4661, 2006.

[12] Cyprien Soulaine, Gérald Debenest, and Michel Quintard. Upscaling multi-component two-phase flow in porous media with partitioning coefficient. *Chemical Engineering Science*, 66(23):6180–6192, 2011.

[13] Stephen Whitaker. Flow in porous media i: A theoretical derivation of darcy's law. *Transport in porous media*, 1(1):3–25, 1986.

[14] Yildiray Cinar and Amir Riaz. Carbon dioxide sequestration in saline formations: Part 2?review of multiphase flow modeling. *Journal of Petroleum Science and Engineering*, 124:381–398, 2014.

[15] Roland Lenormand, Eric Touboul, and Cesar Zarcone. Numerical models and experiments on immiscible displacements in porous media. *Journal of fluid mechanics*, 189:165–187, 1988.

[16] Lewis and Whitman. Principles of gas absorption. *Ind. Eng. Chem.*, 16:1215–1220, 1924.

[17] Higbie. The rate of absorption of a pure gas into a still liquid during short periods of exposure. *Trans. AIChE*, 35, 1935.

[18] Danckwerts. *Gas-Liquid Reactions*. 1970.

[19] Cass T. Miller, Michele M. Poirier-McNeil, and Alex S. Mayer. Dissolution of trapped nonaqueous phase liquids: Mass transfer characteristics. *Water Resources Research*, 26(11):2783–2796, 1990.

[20] Christopher A Kennedy and William C Lennox. A pore-scale investigation of mass transport from dissolving dnapl droplets. *Journal of contaminant hydrology*, 24(3):221–246, 1997.

[21] Susan E Powers, Indumathi M Nambi, and Garrey W Curry. Non–aqueous phase liquid dissolution in heterogeneous systems: Mechanisms and a local equilibrium modeling approach. *Water Resources Research*, 34(12):3293–3302, 1998.

[22] C Jia, K Shing, and YC Yortsos. Visualization and simulation of non-aqueous phase liquids solubilization in pore networks. *Journal of Contaminant Hydrology*, 35(4):363–387, 1999.

[23] NA Sahloul, MA Ioannidis, and I Chatzis. Dissolution of residual non-aqueous phase liquids in porous media: pore-scale mechanisms and mass transfer rates. *Advances in water resources*, 25(1):33–49, 2002.

[24] Cheema Chomsurin and Charles J Werth. Analysis of pore-scale nonaqueous phase liquid dissolution in etched silicon pore networks. *Water resources research*, 39(9), 2003.

[25] Susan E Powers, Linda M Abriola, and Walter J Weber. An experimental investigation of nonaqueous phase liquid dissolution in saturated subsurface systems: Steady state mass transfer rates. *Water Resources Research*, 28(10):2691–2705, 1992.

[26] Sophie Roman, Cyprien Soulaine, Moataz Abu AlSaud, Anthony Kovscek, and Hamdi Tchelepi. Particle velocimetry analysis of immiscible two-phase flow in micromodels. *Advances in Water Resources*, 2015.

[27] ML Johns and LF Gladden. Magnetic resonance imaging study of the dissolution kinetics of octanol in porous media. *Journal of colloid and interface science*, 210(2):261–270, 1999.

[28] Changyong Zhang, Charles J Werth, and Andrew G Webb. A magnetic resonance imaging study of dense nonaqueous phase liquid dissolution from angular porous media. *Environmental science & technology*, 36(15):3310–3317, 2002.

[29] Leslie A Dillard and Martin J Blunt. Development of a pore network simulation model to study nonaqueous phase liquid dissolution. *Water Resources Research*, 36(2):439–454, 2000.

[30] Leslie A Dillard, Hedeff I Essaid, and Martin J Blunt. A functional relation for field-scale nonaqueous phase liquid dissolution developed using a pore network model. *Journal of contaminant hydrology*, 48(1):89–119, 2001.

[31] Rudolf J Held and Michael A Celia. Pore-scale modeling and upscaling of nonaqueous phase liquid mass transfer. *Water Resources Research*, 37(3):539–549, 2001.

[32] W Zhao and MA Ioannidis. Pore network simulation of the dissolution of a single-component wetting nonaqueous phase liquid. *Water resources research*, 39(10), 2003.

[33] Weishu Zhao and Marios A Ioannidis. Gas exsolution and flow during supersaturated water injection in porous media: I. pore network modeling. *Advances in Water Resources*, 34(1):2–14, 2011.

[34] Dengen Zhou, Leslie A Dillard, and Martin J Blunt. A physically based model of dissolution of nonaqueous phase liquids in the saturated zone. *Transport in Porous Media*, 39(2):227–255, 2000.

[35] Weishu Zhao and Marios A Ioannidis. Convective mass transfer across fluid interfaces in straight angular pores. *Transport in porous media*, 66(3):495–509, 2007.

[36] KÅre Langaas and Paul Papatzacos. Numerical investigations of the steady state relative permeability of a simplified porous medium. *Transport in Porous Media*, 45(2):241–266, 2001.

[37] Alexandre M Tartakovsky and Paul Meakin. Pore scale modeling of immiscible and miscible fluid flows using smoothed particle hydrodynamics. *Advances in Water Resources*, 29(10):1464–1478, 2006.

[38] Nicos S Martys and Hudong Chen. Simulation of multicomponent fluids in complex three-dimensional geometries by the lattice boltzmann method. *Physical review E*, 53(1):743, 1996.

[39] Michael R Swift, E Orlandini, WR Osborn, and JM Yeomans. Lattice boltzmann simulations of liquid-gas and binary fluid systems. *Physical Review E*, 54(5):5041, 1996.

[40] Chad E Knutson, Charles J Werth, and Albert J Valocchi. Pore-scale modeling of dissolution from variably distributed nonaqueous phase liquid blobs. *Water Resources Research*, 37(12):2951–2963, 2001.

[41] Antoine Riaud, Shufang Zhao, Kai Wang, Yi Cheng, and Guangsheng Luo. Lattice-boltzmann method for the simulation of multiphase mass transfer and reaction of dilute species. *Phys. Rev. E*, 89:053308, May 2014.

[42] Paul Meakin and Alexandre M Tartakovsky. Modeling and simulation of pore-scale multiphase fluid flow and reactive transport in fractured and porous media. *Reviews of Geophysics*, 47(3), 2009.

[43] Cyril W Hirt and Billy D Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981.

[44] JU Brackbill, Douglas B Kothe, and C1 Zemach. A continuum method for modeling surface tension. *Journal of computational physics*, 100(2):335–354, 1992.

[45] Duong A. Hoang, Volkert van Steijn, Luis M. Portela, Michiel T. Kreutzer, and Chris R. Kleijn. Benchmark numerical simulations of segmented two-phase flows in microchannels using the volume of fluid method. *Computers & Fluids*, 86(0):28 – 36, 2013.

[46] Pierre Horgue, Frédéric Augier, Paul Duru, Marc Prat, and Michel Quintard. Experimental and numerical study of two-phase flows in arrays of cylinders. *Chemical Engineering Science*, 102:335–345, 2013.

[47] Andrea Ferrari and Ivan Lunati. Direct numerical simulations of interface dynamics to link capillary pressure and total surface energy. *Elsevier Advances in Water Resources*, 2013.

[48] Andrea Ferrari and Ivan Lunati. Inertial effects during irreversible meniscus reconfiguration in angular pores. *Advances in Water Resources*, 74(0):1 – 13, 2014.

[49] Andrea Ferrari, Joaquin Jimenez-Martinez, Tanguy Le Borgne, Yves Méheust, and Ivan Lunati. Challenges in modeling unstable two-phase flow experiments in porous micromodels. *Water Resources Research*, pages n/a–n/a, 2015.

[50] Carla Santiago, Shahin Ghomeshi, Sergey Kryuchkov, and Apostolos Kantzas. Pore level modeling of imbibition in heavy oil saturated media. *Journal of Petroleum Science and Engineering*, 140:108–118, 2016.

[51] Bruno Lafaurie, Carlo Nardone, Ruben Scardovelli, Stéphane Zaleski, and Gianluigi Zanetti. Modelling merging and fragmentation in multiphase flows with surfer. *Journal of Computational Physics*, 113(1):134–147, 1994.

[52] Ali Q Raeini, Martin J Blunt, and Branko Bijeljic. Modelling two-phase flow in porous media at the pore scale using the volume-of-fluid method. *Journal of Computational Physics*, 231(17):5653–5668, 2012.

[53] T Abadie, J Aubin, and D Legendre. On the combined effects of surface tension force calculation and interface advection on spurious currents within volume of fluid and level set frameworks. *Journal of Computational Physics*, 297:611–636, 2015.

[54] Dieter Bothe, Mario Koebe, Kerstin Wielage, and Hans-Joachim Warnecke. Vof-simulations of mass transfer from single bubbles and bubble chains rising in aqueous solutions. In *ASME/JSME 2003 4th Joint Fluids Summer Engineering Conference*, pages 423–429. American Society of Mechanical Engineers, 2003.

[55] Alexandru Aurelian Onea. *Numerical simulation of mass transfer with and without first order chemical reaction in two-fluid flows*. Citeseer, 2007.

[56] Holger Marschall, Korbinian Hinterberger, Christian Schüler, Florian Habla, and Olaf Hinrichsen. Numerical simulation of species transfer across fluid interfaces in free-surface flows using openfoam. *Chemical engineering science*, 78:111–127, 2012.

[57] Daniel Deising, Holger Marschall, and Dieter Bothe. A unified single-field model framework for volume-of-fluid simulations of interfacial species transfer applied to bubbly flows. *Chemical Engineering Science*, 139:173–195, 2016.

[58] Y Haroun, D Legendre, and L Raynal. Direct numerical simulation of reactive absorption in gas–liquid flow on structured packing using interface capturing method. *Chemical Engineering Science*, 65(1):351–356, 2010.

[59] Yacine Haroun, Ludovic Raynal, and Dominique Legendre. Mass transfer and liquid hold-up determination in structured packing by cfd. *Chemical engineering science*, 75:342–348, 2012.

[60] Rutherford Aris. *Vectors, tensors and the basic equations of fluid mechanics*. Courier Corporation, 2012.

[61] RG Cox. The dynamics of the spreading of liquids on a solid surface. part 1. viscous flow. *Journal of Fluid Mechanics*, 168:169–194, 1986.

[62] Robert Jack Hansen and TY Toong. Dynamic contact angle and its relationship to forces of hydrodynamic origin. *Journal of Colloid and Interface Science*, 37(1):196–207, 1971.

[63] Ross Taylor and Rajamani Krishna. *Multicomponent mass transfer*, volume 2. John Wiley & Sons, 1993.

[64] Stephen Whitaker. *The method of volume averaging*, volume 13. Springer Science & Business Media, 1998.

[65] Henrik Rusche. *Computational fluid dynamics of dispersed two-phase flows at high phase fractions.* PhD thesis, Imperial College London (University of London), 2003.

[66] RI Issa, B Ahmadi-Befrui, KR Beshay, and AD Gosman. Solution of the implicitly discretised reacting flow equations by operator-splitting. *Journal of Computational Physics*, 93(2):388–410, 1985.

[67] Santiago Marquez Damian. An extended mixture model for the simultaneous treatment of short and long scale interfaces. *Universidad Nacional del Litoral, Argentinien. Krepper, E., Lucas, D., Frank, T., Prasser, H.-M.*, 238(7):1690–1702, 2013.

[68] GI Taylor. Deposition of a viscous fluid on the wall of a tube. *Journal of Fluid Mechanics*, 10(02):161–165, 1961.