



A cut-cell method for the numerical simulation of 3D multiphase flows with strong interfacial effects

Alexandre Caboussat ^{a,*}, Julien Hess ^{b,c}, Alexandre Masserey ^b, Marco Picasso ^c

^a Geneva School of Business Administration, University of Applied Sciences and Arts Western Switzerland (HES-SO), Rue de la Tambourine 17, Carouge, 1227, Geneva, Switzerland

^b Ycoor Systems SA, Sierre, 3960, Switzerland

^c Institute of Mathematics, Ecole polytechnique fédérale de Lausanne, Lausanne, 1015, Switzerland

ARTICLE INFO

In memory of Professor Roland Glowinski

Keywords:

Multiphase flows
Operator splitting
Two-grid method
Mesh cutting
Surface tension
Emulsion

ABSTRACT

We present a numerical model for the approximation of multiphase flows with free surfaces and strong interfacial effects. The model relies on the multiphase incompressible Navier-Stokes equations, and includes surface tension effects on the interfaces between phases, and contact angles. The volume-of-fluid approach is used to track the interfaces and the free surfaces between the various phases and the ambient air. The numerical method relies on an operator splitting strategy. The space discretization relies on a two-grid approach that uses an unstructured finite element mesh for diffusion phenomena and a structured Cartesian grid for advection phenomena. An adaptive mesh refinement algorithm is incorporated to better track the interfaces and free surfaces on the finite element mesh, and approximate more accurately surface forces. The model is validated through numerical experiments, in particular for emulsion problems.

1. Introduction

Multiphase flows are ubiquitous in nature, and industrial applications are numerous. The objective of this work is ultimately to address the numerical simulation of emulsion processes in food engineering [1–4]. These processes typically involve the creation of small droplets of one liquid dispersed into another. It includes strong interfacial effects, such as surface tension, contact angles, and boundary layers. For instance, the numerical simulation of multiphase flows includes contact angles in [5,6]. Mesh dependency, including non-consistent modeling of contact angles in capillary flows, has been emphasized in [7]. The interaction between droplets and interfacial effects has been treated in, e.g., [8,9], in particular in T-junction geometries.

The main characteristic of the method presented here is to allow the simulation of multiphase flows with strong interfacial effects on the interfaces between phases. A mathematical model for the numerical simulation of Newtonian fluids with free surfaces has been presented and validated in [10–12]. Surface tension effects have been added in [13], but only when considering one single liquid phase surrounded by void. The model has been extended to the simulation of multiphase flows in [14], for macroscopic effects, but without surface tension effects. Here the interfacial effects are added to the algorithm by means of a mesh cutting algorithm to increase the accuracy of the approximation method at the interfaces.

We consider a domain containing several liquid phases, potentially with free surfaces in contact with the ambient air considered as void. The incompressible Navier-Stokes equations are solved in the whole liquid domain. The surface tension and the contact

* Corresponding author.

E-mail address: alexandre.caboussat@hesge.ch (A. Caboussat).

<https://doi.org/10.1016/j.jcp.2024.112846>

Received 31 January 2023; Received in revised form 14 January 2024; Accepted 8 February 2024

Available online 23 February 2024

0021-9991/© 2024 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

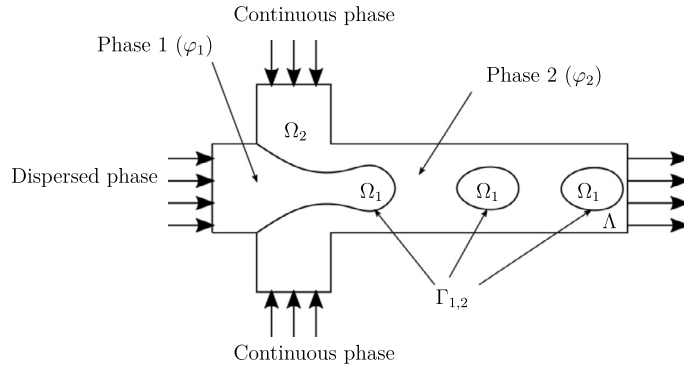


Fig. 1. 2D sketch of the geometrical domain for the simulation of multiphase flows with strong interfacial effects. The case of the geometry for emulsion processes, where droplets of a dispersed phase (index $\ell = 1$) injected horizontally are created by a continuous phase (index $\ell = 2$) injected laterally from top and bottom.

angles on each interface (liquid/liquid, liquid/air and solid/liquid) are taken into account. The system of equations is supplemented with appropriate boundary and initial conditions. The novelty of the proposed approach lies in the choice of a dedicated numerical method. In particular, a local adaptive mesh cutting technique [11,15] is applied at the interfaces to provide an accurate description of the surface effects and forces. An operator splitting approach is advocated to decouple the diffusion operators and the advection operators (transport of quantities) [16,17]. The volume fraction approach is used to track the position of the liquid phases [18]. The space discretization relies on a two-grid method that uses an unstructured finite element mesh for the diffusion phenomena, and a structured Cartesian grid for the advection phenomena [19].

The article is structured as follows. In Section 2, we describe the mathematical model. Section 3 details the time discretization via an operator splitting algorithm. Section 4 describes the space discretization, using a two-grid approach. A special emphasis is made on the novel adaptive local mesh cutting technique. The results of numerical experiments are presented in Section 5, to validate the numerical methods, and to finally address the microfluidic process of producing emulsions.

2. Mathematical model

Let us consider $\Lambda \subset \mathbb{R}^d$, $d = 2, 3$, a bounded domain containing $L \geq 2$ immiscible and incompressible liquid phases, and let $T > 0$ be the final time of the simulation. For any time $t \in (0, T)$, the subdomain of Λ occupied by the ℓ -th phase is denoted by $\Omega_\ell(t)$, $\ell = 1, \dots, L$, such that $\Omega(t) := \bigcup_{\ell=1}^L \Omega_\ell(t)$ represents the whole liquid domain at time t .

For convenience, we denote by $\Omega_0(t) := \Lambda \setminus \Omega(t)$ the domain considered as vacuum; it is assumed to have no influence on the liquid phases. The interfaces between different media are expressed as $\Gamma_{i,j}(t) := \partial\Omega_i(t) \cap \partial\Omega_j(t)$, for $0 \leq i, j \leq L$, $i \neq j$. The interfaces $\Gamma_{0,j}(t)$, $j = 1, \dots, L$ between the phases and the vacuum are called free surfaces. A sketch of the geometrical domain for the emulsion process (in two space dimensions) is illustrated in Fig. 1.

The evolution of the liquid domains is modeled by means of a volume-of-fluid method (VOF). Let $\varphi_\ell : \Lambda \times (0, T) \rightarrow \{0, 1\}$ be the characteristic function of domain $\Omega_\ell(t)$, such that

$$\Omega_\ell(t) = \{ \mathbf{x} \in \Lambda : \varphi_\ell(\mathbf{x}, t) = 1 \}, \quad \ell = 1, \dots, L.$$

It follows that $\varphi := \sum_{\ell=1}^L \varphi_\ell$ is the characteristic function of the liquid domain $\Omega(t)$. We define by $\varphi_0 = 1 - \varphi$ the characteristic function of $\Omega_0(t)$. Note that the emulsion process illustrated in Fig. 1 does not contain any void domain $\Omega_0(t)$ since the computational domain is full of liquid ($\Omega(t) = \Lambda$). Denoting respectively by ρ_ℓ and μ_ℓ the mass density and the dynamic viscosity of the ℓ -th phase (both assumed to be constant in each liquid phase), the density and the viscosity of the liquid may then be expressed as

$$\rho = \sum_{\ell=1}^L \varphi_\ell \rho_\ell, \quad \text{and} \quad \mu = \sum_{\ell=1}^L \varphi_\ell \mu_\ell.$$

Let Q_T denote the restricted space-time domain containing the liquid:

$$Q_T = \{ (\mathbf{x}, t) \in \Lambda \times (0, T) : \mathbf{x} \in \Omega(t) \} = \{ (\mathbf{x}, t) \in \Lambda \times (0, T) : \varphi(\mathbf{x}, t) = 1 \}.$$

Assuming that Q_T is sufficiently regular, the velocity field $\mathbf{v} : Q_T \rightarrow \mathbb{R}^d$, the pressure field $p : Q_T \rightarrow \mathbb{R}$ and the VOF functions $\varphi_\ell : \Lambda \times (0, T) \rightarrow \{0, 1\}$ are assumed to satisfy the following system of equations:

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} \cdot \nabla) \mathbf{v} - \nabla \cdot (\sigma(\mathbf{v}, p)) = \mathbf{F}, \quad (1)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (2)$$

$$\frac{\partial \varphi_\ell}{\partial t} + \mathbf{v} \cdot \nabla \varphi_\ell = 0, \quad \ell = 1, \dots, L, \quad (3)$$

where $\sigma(\mathbf{v}, p) = 2\mu\boldsymbol{\varepsilon}(\mathbf{v}) - p\mathbf{I}$ is the total stress tensor, $\boldsymbol{\varepsilon}(\mathbf{v}) = \frac{1}{2}(\nabla\mathbf{v} + \nabla\mathbf{v}^T)$ is the strain rate tensor and \mathbf{F} is the vector of external forces (e.g., the gravity forces $\mathbf{F} = \rho\mathbf{g}$ in most cases).

Equations (1)(2) are the incompressible, time-dependent, Navier-Stokes equations. The transport equations (3) describe the kinematics of the liquid domains (and their interfaces) and must be interpreted in a weak sense, as \mathbf{v} in $(\Lambda \times (0, T)) \setminus Q_T$ is a regular extension of \mathbf{v} inside Q_T (see, e.g., [19,20]). The continuity of the velocity field is enforced on each interface, namely

$$[\mathbf{v}]_{ij} = \mathbf{0} \quad \text{on} \quad \Gamma_{i,j}(t), \quad 1 \leq i, j \leq L, \quad i \neq j, \quad (4)$$

where $[\cdot]_{ij}$ is the jump operator on $\Gamma_{i,j}(t)$ defined as

$$[\mathbf{w}]_{ij} := \mathbf{w}|_{\Omega_i(t)} - \mathbf{w}|_{\Omega_j(t)}, \quad (5)$$

for any field $\mathbf{w} : Q_T \rightarrow \mathbb{R}^d$. Note that, when writing the weak formulation of the Navier-Stokes equations with variable densities and viscosities, the explicit formulation of boundary terms leads to enforcing the continuity of velocities and forces at the interfaces between phases. This continuous velocity is then used to transport the fluid particles in the volume-of-fluid model. Let us denote by $\mathbf{n}_{\Gamma_{ij}}$ the unit normal vector to $\Gamma_{i,j}(t)$ (towards $\Omega_j(t)$), for $0 \leq i, j \leq L, i \neq j$. Surface tension effects are taken into account via a force term imposed on each interface, namely

$$[\sigma\mathbf{n}_{\Gamma_{ij}}]_{ij} = \gamma_{ij} H_{ij} \mathbf{n}_{\Gamma_{ij}} \quad \text{on} \quad \Gamma_{i,j}(t), \quad 0 \leq i, j \leq L, \quad i \neq j, \quad (6)$$

where γ_{ij} is the surface tension coefficient between phases i and j , and H_{ij} is the mean curvature of $\Gamma_{i,j}(t)$.

The equations in (6) must be completed with boundary conditions including *contact angles* between the interface $\Gamma_{i,j}(t)$ and the wall $\partial\Omega(t)$. This effect is treated following the approach advocated in [21,22], namely by adding an integral over $\Gamma_{i,j}(t)$ in the weak formulation, using a surface version of the divergence theorem.

The model is completed with initial and boundary conditions. The volume fraction of each phase $\varphi_\ell(\cdot, 0)$ is given at initial time $t = 0$ in Λ , which is equivalent to defining the initial liquid regions $\Omega_\ell(0)$, for $\ell = 1, \dots, L$. The initial (divergence free) velocity field $\mathbf{v}(0) = \mathbf{v}_0$ is then prescribed in $\Omega(0)$.

The Navier-Stokes equations (1)(2) are completed with no-slip or pure-slip boundary conditions imposed on impermeable walls $\partial\Omega(t) \cap \partial\Lambda$. Inflow boundary conditions for both the velocity \mathbf{v} and the volume fractions φ_ℓ are enforced on the inlet part of $\partial\Lambda$ (if any, namely, for all $(\mathbf{x}, t) \in \partial\Lambda \times (0, T)$ such that $\mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}_{\partial\Lambda} < 0$). Outflow boundary conditions are imposed at the exit of the domain (if any), i.e. enforcing a zero normal force and a zero tangential velocity.

3. Time discretization

The implicit splitting algorithm of order one described in [12,13,19] for Newtonian free surface flows, and in [14] for multiphase flows, is extended here to multiphase flows with interfacial effects. It relies on an *operator splitting* strategy that allows to decouple advection and diffusion phenomena.

Let N_T be the total number of time steps, $0 = t^0 < t^1 < t^2 < \dots < t^{N_T} = T$ be a subdivision of the time interval $[0, T]$, and $\tau^n := t^{n+1} - t^n$ be the $(n + 1)$ -th time step, $n = 0, 1, 2, \dots, N_T - 1$.

Assume that $\varphi_\ell^n : \Lambda \rightarrow \mathbb{R}$ is an approximation of φ_ℓ at time t^n , which defines the domain occupied by the ℓ -th phase $\Omega_\ell^n = \{\mathbf{x} \in \Lambda : \varphi_\ell^n(\mathbf{x}) = 1\}$, for $\ell = 1, \dots, L$. The overall liquid domain at time t^n is thus given by $\Omega^n = \bigcup_{\ell=1}^L \Omega_\ell^n$, and $\Omega_0^n = \Lambda \setminus \Omega^n$. Let \mathbf{v}^n, p^n be known approximations of \mathbf{v}, p in Ω^n at time t^n . The approximations $\varphi_\ell^{n+1}, \Omega_\ell^{n+1}, \Omega^{n+1}, \mathbf{v}^{n+1}, p^{n+1}$ at time t^{n+1} are computed by means of a splitting algorithm as illustrated in Fig. 2.

3.1. Diffusion operator

First, a generalized Stokes problem is solved in order to obtain the predicted velocity $\mathbf{v}^{n+1/2}$ and the pressure p^{n+1} in the overall liquid domain Ω^n . The continuous Stokes system reads:

$$\rho \frac{\partial \mathbf{v}}{\partial t} - \nabla \cdot \sigma(\mathbf{v}, p) = \mathbf{F}, \quad (7)$$

$$\nabla \cdot \mathbf{v} = 0. \quad (8)$$

An implicit Euler scheme is used for the time discretization of this Stokes system in Ω^n :

$$\rho^n \frac{\mathbf{v}^{n+1/2} - \mathbf{v}^n}{\tau^n} - \nabla \cdot \sigma^n(\mathbf{v}^{n+1/2}, p^{n+1}) = \mathbf{F}^n, \quad (9)$$

$$\nabla \cdot \mathbf{v}^{n+1/2} = 0, \quad (10)$$

with no-slip or pure-slip conditions on the boundary of the cavity $\partial\Omega^n \cap \partial\Lambda$, and surface tension forces on each interface:

$$[\sigma^n(\mathbf{v}^{n+1/2}, p^{n+1})\mathbf{n}_{\Gamma_{ij}^n}]_{ij} = \gamma_{ij} H_{ij}^n \mathbf{n}_{\Gamma_{ij}^n} \quad \text{on} \quad \Gamma_{i,j}^n, \quad 0 \leq i, j \leq L, \quad i \neq j, \quad (11)$$

where H_{ij}^n is the mean curvature of $\Gamma_{i,j}^n = \partial\Omega_i^n \cap \partial\Omega_j^n$. Here the density and the viscosity are respectively defined by

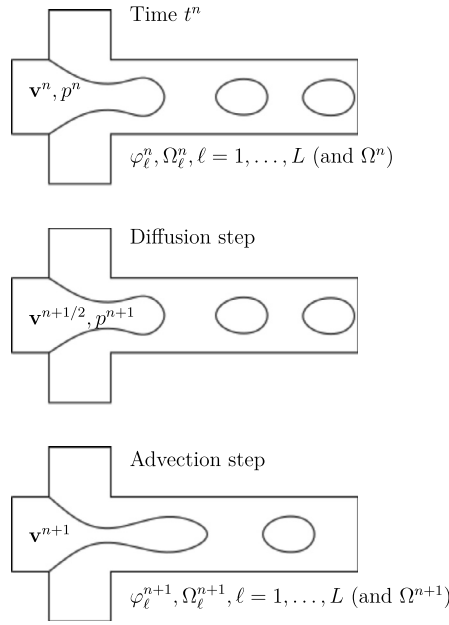


Fig. 2. Operator splitting algorithm (from top to bottom). At each time step n , we first solve a generalized Stokes problem, including interfacial effects, in order to obtain the predicted velocity $\mathbf{v}^{n+1/2}$ and the pressure p^{n+1} . Then, advection problems are solved in order to obtain the new volume fractions φ_ℓ^{n+1} (and thus the new liquid domain Ω_ℓ^{n+1}), and the corrected velocity \mathbf{v}^{n+1} .

$$\rho^n = \sum_{\ell=1}^L \varphi_\ell^n \rho_\ell, \quad \text{and} \quad \mu^n = \sum_{\ell=1}^L \varphi_\ell^n \mu_\ell.$$

This implies in particular that $\sigma^n(\mathbf{w}, q) = 2\mu^n \varepsilon(\mathbf{w}) - q\mathbf{I}$. Finally, $\mathbf{F}^n = \mathbf{F}(t^n)$.

3.2. Advection operator

Then, considering the advection operators in (1) and (3), the second step consists in using the prediction of the velocity $\mathbf{v}^{n+1/2}$ in order to transport the volume fractions φ_ℓ^n and the velocity $\mathbf{v}^{n+1/2}$ itself. This step allows to obtain the new volume fractions φ_ℓ^{n+1} (and thus the new liquid domains Ω_ℓ^{n+1} and Ω^{n+1}) and the corrected velocity \mathbf{v}^{n+1} . More precisely, this advection step consists in solving, between t^n and t^{n+1} the following system of $L + 1$ equations:

$$\frac{\partial \varphi_\ell}{\partial t} + \mathbf{v} \cdot \nabla \varphi_\ell = 0, \quad \ell = 1, \dots, L, \tag{12}$$

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = 0, \tag{13}$$

with initial conditions φ_ℓ^n and $\mathbf{v}^{n+1/2}$, respectively. This system of hyperbolic equations is solved with a forward characteristics method, so that φ_ℓ^{n+1} and \mathbf{v}^{n+1} are respectively given by

$$\varphi_\ell^{n+1}(\mathbf{x} + \tau^n \mathbf{v}^{n+1/2}(\mathbf{x})) = \varphi_\ell^n(\mathbf{x}), \quad \ell = 1, \dots, L, \tag{14}$$

$$\mathbf{v}^{n+1}(\mathbf{x} + \tau^n \mathbf{v}^{n+1/2}(\mathbf{x})) = \mathbf{v}^{n+1/2}(\mathbf{x}), \tag{15}$$

for all $\mathbf{x} \in \Omega^n$. The new domain Ω^{n+1} is then defined as $\Omega^{n+1} = \bigcup_{\ell=1}^L \Omega_\ell^{n+1}$, where $\Omega_\ell^{n+1} = \{\mathbf{x} \in \Lambda : \varphi_\ell^{n+1}(\mathbf{x}) = 1\}$, and $\Omega_0^{n+1} = \Lambda \setminus \Omega^{n+1}$.

4. Space discretization

The space discretization relies on the two-grid approach presented in [12,13,19] and illustrated in Fig. 3, but incorporates a novel mesh cutting algorithm inspired by [11,15] and illustrated in Fig. 4. At each time step, it relies on:

- (i) The restriction of a coarse, unstructured tetrahedral finite element discretization $\mathcal{T}_{h_{fe}}$ of the whole cavity Λ (with typical mesh size h_{fe}), onto the overall liquid domain Ω^n at each time step. The Stokes problem (9)(10) is solved on that mesh with low order continuous finite elements.
- (ii) A regular grid $\mathcal{C}_{h_{cells}}$ of (smaller) structured cells (with typical cell size $h_{cells} < h_{fe}$) covering the cavity Λ , in order to solve the advection problems (12)(13) with a forward characteristics method.

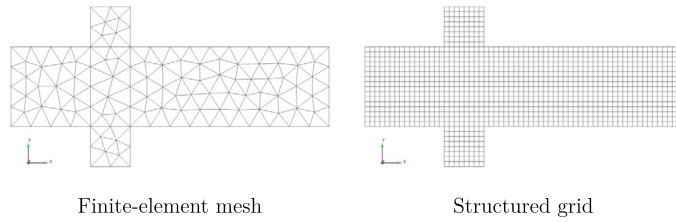


Fig. 3. Two-grid method (2D sketch). The diffusion problem is solved on an unstructured finite element mesh of typical size h_{fe} (left). The advection problems are solved on a structured grid of smaller cubic cells of typical size h_{cells} (right).

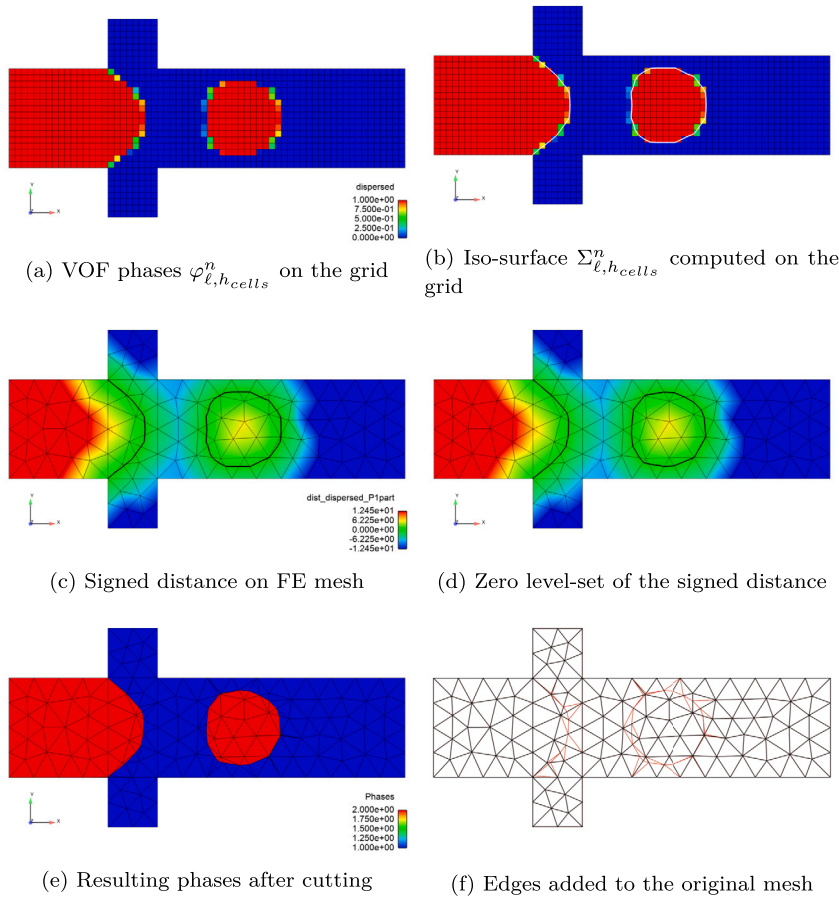


Fig. 4. Detailed cut-cell algorithm for the dispersed VOF phase: (a) Approximations $\varphi_{\ell, h_{cells}}^n$ of the VOF phase; (b) Computation of the iso-surface $\{x \in \Lambda : \varphi_{\ell}^n(x) = 1/2\}$ with the *Marching Cubes 33* algorithm; (c) Computation of the signed distance on the finite element mesh; (d) Computation of the zero level-set of the signed distance; (e) Resulting phases on the finite element mesh, and (f) Visualization of the added edges on the finite element mesh.

The rationale behind using a two-grid method is, on the one hand, to use a finer structured grid to increase the accuracy of the approximation of φ_{ℓ}^{n+1} (which are discontinuous at the interfaces), and thus of the approximations of the interfaces. Even with small grid cells, the forward characteristic method remains tractable in terms of computational cost. Despite being a low order method, the SLIC method is sufficient to obtain appropriate linear convergence for the global algorithm, while providing for a simple implementation. On the other hand, the computational cost of solving parabolic problems implicitly (such as the Stokes problem) is kept reasonable by using a coarser mesh and low order finite elements. Following [19,20], we typically advocate $3 \leq h_{fe}/h_{cells} \leq 5$ for a reasonable trade-off between accuracy and computational efficiency. The initialization is achieved by enforcing piecewise constant values of the characteristic functions for each liquid phase, and of the velocity, on each grid cell.

In addition, in order to increase the accuracy of the approximation of interfacial effects, we introduce a mesh cutting algorithm that allows to generate vertices on the interfaces. This procedure increases the number of elements in the neighborhood of interfaces and free surfaces so that the mesh is adapted to the interfaces. It allows to obtain a smoother and more accurate approximation of these interfaces. The complete numerical method to locally refine the mesh near the interfaces is detailed in Section 4.2.

4.1. General space-discretized operator-splitting algorithm

The complete space-discretized operator-splitting algorithm reads as follows. Let us denote by $\varphi_{\ell,h_{cells}}^n, \mathbf{v}_{h_{cells}}^n$ the piecewise constant approximations of $\varphi_{\ell}^n, \mathbf{v}^n$ on $C_{h_{cells}}$, and by $\mathbf{v}_{h_{fe}}^n, p_{h_{fe}}^n$ the finite element approximations of \mathbf{v}^n, p^n on $\mathcal{T}_{h_{fe}}^n$. At each time step t^n , we assume that $\varphi_{\ell,h_{cells}}^n$ and $\mathbf{v}_{h_{cells}}^n$ are given, and we proceed as follows:

- (1) **(Adapt)** Using $\varphi_{\ell,h_{cells}}^n$, we construct an adapted (and restricted) finite element discretization $\mathcal{T}_{h_{fe}}^n$ of $\mathcal{T}_{h_{fe}}$ in order to obtain a more accurate approximation of the interfaces. Then, we interpolate $\mathbf{v}_{h_{cells}}^n$ from $C_{h_{cells}}$ onto $\mathcal{T}_{h_{fe}}^n$ to obtain $\mathbf{v}_{h_{fe}}^n$ (see Section 4.2.5).
- (2) **(Diffusion)** We solve the Stokes problem on $\mathcal{T}_{h_{fe}}^n$ to obtain a prediction of the velocity $\mathbf{v}_{h_{fe}}^{n+1/2}$ and the pressure $p_{h_{fe}}^{n+1}$, with the introduction of the interfacial terms. Then we interpolate $\mathbf{v}_{h_{fe}}^{n+1/2}$ from $\mathcal{T}_{h_{fe}}^n$ onto $C_{h_{cells}}$ to obtain $\mathbf{v}_{h_{cells}}^{n+1/2}$.
- (3) **(Advection)** We solve the transport equations on $C_{h_{cells}}$ in order to obtain the corrected velocity $\mathbf{v}_{h_{cells}}^{n+1}$ and $\varphi_{\ell,h_{cells}}^{n+1}$ (see Section 4.4).

The overall convergence rate of the numerical algorithm is order one. Thus, when dividing h_{fe}, h_{cells} and τ by two, the error is divided by two. Note that when removing the interfacial terms, and thus not adapting the mesh in **(Adapt)**, the algorithm corresponds to the one presented in [14].

Let us denote by K a generic element (tetrahedron) of $\mathcal{T}_{h_{fe}}$ with barycenter \mathbf{p}_K , and by C a generic cell of $C_{h_{cells}}$ with barycenter \mathbf{q}_C . At each time step n , the finite element approximation spaces for the approximations of the velocity and pressure respectively are defined as follows:

$$V_{h_{fe}}^n = \left\{ v_{h_{fe}} \in C^0(\Omega^n) : v_{h_{fe}}|_K \in \mathbb{P}_1^B, \forall K \in \mathcal{T}_{h_{fe}}^n \right\}, \tag{16}$$

$$Q_{h_{fe}}^n = \left\{ q_{h_{fe}} \in C^0(\Omega^n) : q_{h_{fe}}|_K \in \mathbb{P}_1, \forall K \in \mathcal{T}_{h_{fe}}^n \right\}, \tag{17}$$

where \mathbb{P}_1^B is the space of first order polynomials on K enriched with a bubble function (mini-elements) [23], and

$$W_{h_{cells}} = \left\{ \theta_{h_{cells}} \in L^2(\Lambda) : \theta_{h_{cells}}|_C \in \mathbb{P}_0, \forall C \in C_{h_{cells}} \right\}. \tag{18}$$

4.2. Adaptive finite element discretization and mesh cutting

In order to reduce numerical diffusion, we construct an adapted finite element mesh $\mathcal{T}_{h_{fe}}^n$ that is refined around the interfaces and free surfaces of Ω^n . The approach presented in [11] is improved here by using a mesh cutting approach that relies on the piecewise constant approximations $\varphi_{\ell,h_{cells}}^n$ on the finer grid $C_{h_{cells}}$. In addition to the accuracy improvement, it allows the extension to multiple phases, and to increase accuracy around the interfaces between liquid phases as well. The underlying idea is to take advantage of the two-grid method, and of the finer structured mesh, to generate a more accurate approximation of the interfaces.

Fig. 4 details the steps of the cut-cell algorithm. Starting with the piecewise constant approximations $\varphi_{\ell,h_{cells}}^n$ of the VOF phases (a), we compute approximations of φ_{ℓ}^n at the vertices of each cell by averaging. Based on these values at the vertices, an approximation of the iso-surface $\left\{ \mathbf{x} \in \Lambda : \varphi_{\ell}^n(\mathbf{x}) = 1/2 \right\}$ is constructed with the algorithm *Marching Cubes 33* [15] (b). This iso-surface consists of an independent surface mesh. A signed distance with respect to this iso-surface is computed on the vertices of the finite element mesh (c), and its zero level-set is determined (d). Based on this zero level-set, a cut-cell method allows to refine the finite element mesh by adding new edges [11] (e)-(f).

Note that the algorithm does not need to compute intersections between the iso-surface constructed with the *Marching Cubes 33* algorithm and the finite element mesh.

4.2.1. Extraction of iso-surfaces on the structured grid

For each phase $\ell = 1, \dots, L$, the $1/2$ -iso-surface

$$\Sigma_{\ell}^n := \left\{ \mathbf{x} \in \Lambda : \varphi_{\ell}^n(\mathbf{x}) = \frac{1}{2} \right\}$$

is approximated by using the *Marching Cubes 33* algorithm [15]. Originally designed for computer graphics, the purpose of the *Marching Cubes 33* algorithm is to extract a polygonal mesh of an iso-surface from a three-dimensional discrete scalar field. In three dimensions of space, the algorithm proceeds through the list of cubic cells of the structured grid to determine the ones intersecting the iso-surface defined by $\left\{ \mathbf{x} \in \Lambda : \varphi_{\ell}(\mathbf{x}, t) = \frac{1}{2} \right\}$, at some time t . For each cell containing a section of the iso-surface, a triangular mesh approximating the trilinear interpolant within the cell is constructed. In the end, a triangular mesh is obtained that approximates the iso-surface. We use the fast and memory efficient implementation of the *Marching Cubes 33* algorithm described in [15]. The resulting iso-surface approximations $\Sigma_{\ell,h_{cells}}^n$ are 2D surface finite element triangulations. The *Marching Cubes 33* algorithm requires, as an input, values of $\varphi_{\ell,h_{cells}}^n$ in the corners of each cell of $C_{h_{cells}}$. These values are denoted by $\tilde{\varphi}_{\ell,h_{cells}}^n$, and obtained for each vertex

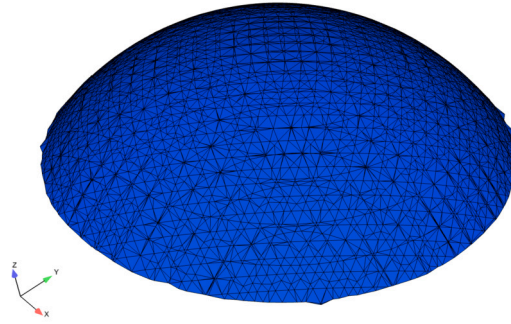


Fig. 5. Visualisation of the interface mesh (zero level-set) after cutting the finite element mesh, in the case of a hydrophilic droplet lying on a horizontal plane.

$\mathbf{q} \in C_{h_{\text{cells}}}$ as $\tilde{\varphi}_{\ell, h_{\text{cells}}}^n(\mathbf{q}) := |\mathcal{N}_{\mathbf{q}}|^{-1} \sum_{C \in \mathcal{N}_{\mathbf{q}}} \varphi_{\ell, h_{\text{cells}}}^n(C)$, where $\mathcal{N}_{\mathbf{q}} := \{C \in C_{h_{\text{cells}}} : \mathbf{q} \in C\}$ is the set of neighbor cells that share vertex \mathbf{q} and $|\mathcal{N}_{\mathbf{q}}|$ is its cardinality.

4.2.2. Computation of signed distances on the finite element mesh

Consider $\mathcal{T}_{h_{fe}}$ the initial unstructured tetrahedral finite element discretization of the whole cavity Λ . For each phase $\ell = 0, \dots, L$ (including the void), we define the piecewise linear signed distance $d_{\ell}^n : \mathcal{T}_{h_{fe}} \rightarrow \mathbb{R}$ with respect to the iso-surface approximation $\Sigma_{\ell, h_{\text{cells}}}^n$ as follows:

$$d_{\ell}^n(\mathbf{p}) := \text{sgn}(\tilde{\varphi}_{\ell, h_{\text{cells}}}^n(\mathbf{p}) - \frac{1}{2}) \min_{T \in \Sigma_{\ell, h_{\text{cells}}}^n} d(\mathbf{p}, T),$$

for each vertex $\mathbf{p} \in \mathcal{T}_{h_{fe}}$, where $\text{sgn}(\cdot)$ is the usual sign function, $d(\cdot, \cdot)$ is the standard point-to-triangle Euclidean distance [24], and $\tilde{\varphi}_{\ell, h_{\text{cells}}}^n$ is a trilinear interpolation of $\tilde{\varphi}_{\ell, h_{\text{cells}}}^n$ at \mathbf{p} (as it is not necessarily a vertex of $C_{h_{\text{cells}}}$).

Note that $d_{\ell}^n > 0$ in phase ℓ and $d_{\ell}^n < 0$ elsewhere. Furthermore, the signed distance d_0^n relying on φ_0^n is the opposite of the signed distance based on $\varphi^n = \sum_{\ell=1}^L \varphi_{\ell}^n$.

4.2.3. Iterative mesh cutting algorithm

The locally adapted finite element discretization $\mathcal{T}_{h_{fe}}^n$ is obtained from $\mathcal{T}_{h_{fe}}$ by using an iterative version of the level-set approach presented in [11]. The mesh cutting procedure relies on the above-presented distance function built thanks to the information on $C_{h_{\text{cells}}}$, instead of the values of the volume fraction on $\mathcal{T}_{h_{fe}}$, as in [11] for one phase only.

Let $\mathfrak{T}^{-1} := \mathcal{T}_{h_{fe}}$ be the initial tetrahedral discretization of Λ , and $d_{\ell}^{n,-1} := d_{\ell}^n$ be the initial signed distances defined on \mathfrak{T}^{-1} , for $\ell = 0, \dots, L$. Then:

For $k = 0, \dots, L$,

- (i) Refine \mathfrak{T}^{k-1} by adding extra vertices on the zero level-set of $d_k^{n,k-1}$ and cut the neighboring elements to keep a conforming mesh, as in [11]. The resulting tetrahedral mesh is denoted \mathfrak{T}^k .
- (ii) For $\ell = 0, \dots, L$, update the signed distances $d_{\ell}^{n,k}$ as the linear interpolation of $d_{\ell}^{n,k-1}$ onto the new mesh \mathfrak{T}^k . Note that since the signed distances are piecewise linear functions, the zero level-sets of $d_{\ell}^{n,k}$ and $d_{\ell}^{n,k-1}$ are identical.

Note that, when cutting an element as in [11], a vertex is added on an edge only if the distance between that vertex and the edge extremities is larger than 1% of the edge length. More precisely, this distance should be larger than $\tau = \max(\tau_{\min}, \varepsilon h_{\min})$, where $\tau_{\min} = 10^{-6}$ is the global minimal edge length, $\varepsilon = 10^{-2}$, and h_{\min} is the smallest edge length in the original mesh. Furthermore, whenever a choice must be made, elements are cut in order to maximize the minimum angle in the conforming sub-elements. The final discretization \mathfrak{T}^L is an adapted finite element discretization of Λ that accurately approximates the interfaces between sub-domains occupied by each phase. Fig. 5 visualizes the resulting interface (zero level-set) after cutting the finite element mesh, in the case of a hydrophilic droplet lying on a horizontal plane.

4.2.4. Phase index identification

Each of the elements of this final discretization \mathfrak{T}^L contains one and only one phase by construction. We associate one phase to each of the elements $K \in \mathfrak{T}^L$ by defining the phase index i_K as:

$$i_K := \begin{cases} \ell & \text{if } d_{\ell}^L(\mathbf{p}_K) > 0 \text{ for some } 0 \leq \ell \leq L, \\ -1 & \text{otherwise.} \end{cases} \quad (19)$$

Note that the phase index is well-defined. The elements with $i_K = 0$ correspond to the elements in the ambient air (vacuum). The elements with $i_K = -1$ correspond to the elements that have been categorized as undetermined due to numerical approximations.

The region $\mathcal{M} := \{K \in \mathfrak{T}^L : i_K < 0\}$ is the region composed by undetermined elements. If $\mathcal{M} \neq \emptyset$, the undetermined elements get a phase index as follows. Let us assume that \mathcal{M} is composed of $M > 0$ connected components $\mathcal{M}_k, k = 1, \dots, M$ (in practice detected with a depth-first-search algorithm, see, e.g., [25]). Each connected component is attributed the phase index of its main phase, computed on the structured grid as

$$i_K := \operatorname{argmax}_{0 \leq \ell \leq L} \frac{\sum_{C \in \mathcal{C}_{h_{cells}}} |\mathcal{M}_k \cap C| \varphi_{\ell, h_{cells}}^n(C)}{|\mathcal{M}_k|}, \quad \forall K \in \mathcal{M}_k. \tag{20}$$

The characteristic functions of the sub-domains Ω_ℓ^n are therefore approximated by piecewise constant functions $\varphi_{\ell, h_{fe}}^n : \mathfrak{T}^L \rightarrow \{0, 1\}$ defined as

$$\varphi_{\ell, h_{fe}}^n(K) = \begin{cases} 1 & \text{if } i_K = \ell \\ 0 & \text{otherwise} \end{cases}, \quad \forall K \in \mathfrak{T}^L. \tag{21}$$

Finally, the restricted and locally adapted finite element discretization of Ω^n is defined as $\mathcal{T}_{h_{fe}}^n := \{K \in \mathfrak{T}^L : i_K > 0\}$, namely the union of all elements K in \mathfrak{T}^L that contain a liquid phase.

Note that, there might be some finite elements that do not contain a defined phase after the cut-cell algorithm. These undefined regions are mainly associated with triple-points, as the triple-points cannot be represented with simple signed distances. The proposed heuristics allows to fill the empty region with a given phase for robustness.

4.2.5. Interpolation/projection on the finite element mesh

The piecewise constant approximation $\mathbf{v}_{h_{cells}}^n \in (W_{h_{cells}})^d$ is interpolated onto the finite element space $(V_{h_{fe}}^n)^d$ in order to solve the Stokes problem. The value of the interpolant at vertex $\mathbf{p} \in \mathcal{T}_{h_{fe}}^n$ is

$$\mathbf{v}_{h_{fe}}^n(\mathbf{p}) = \frac{\sum_{K \in \mathcal{N}_{\mathbf{p}}} \sum_{C \in \mathcal{N}_K} \mathbf{v}_{h_{cells}}^n(C) \|\mathbf{q}_C - \mathbf{p}\|^{-2}}{\sum_{K \in \mathcal{N}_{\mathbf{p}}} \sum_{C \in \mathcal{N}_K} \|\mathbf{q}_C - \mathbf{p}\|^{-2}}, \tag{22}$$

where $\mathcal{N}_{\mathbf{p}} := \{K \in \mathcal{T}_{h_{fe}}^n : \mathbf{p} \in K\}$ is the set of elements that share vertex \mathbf{p} and $\mathcal{N}_K := \{C \in \mathcal{C}_{h_{cells}} : \varphi_{h_{cells}}^n(C) > 0, K \cap C \neq \emptyset\}$ is the set of liquid/active cells that intersect element K [26].

4.3. Stokes operator

Let us address the space discretization of (9)(10). The Stokes problem is solved with \mathbb{P}_1^B - \mathbb{P}_1 finite elements. It consists in finding the velocity $\mathbf{v}_{h_{fe}}^{n+1/2} \in (V_{h_{fe}}^n)^d$ and the pressure $p_{h_{fe}}^{n+1} \in Q_{h_{fe}}^n$, satisfying the essential boundary conditions on $\partial\Omega_{h_{fe}}^n$, and such that:

$$\begin{aligned} & \int_{\Omega_{h_{fe}}^n} \left(\rho^n \frac{\mathbf{v}_{h_{fe}}^{n+1/2} - \mathbf{v}_{h_{fe}}^n}{\tau^n} \cdot \mathbf{w} + 2\mu^n \boldsymbol{\varepsilon}(\mathbf{v}_{h_{fe}}^{n+1/2}) : \boldsymbol{\varepsilon}(\mathbf{w}) \right) d\mathbf{x} \\ & - \int_{\Omega_{h_{fe}}^n} p_{h_{fe}}^{n+1} \nabla \cdot \mathbf{w} d\mathbf{x} - \int_{\Omega_{h_{fe}}^n} q \nabla \cdot \mathbf{v}_{h_{fe}}^{n+1/2} d\mathbf{x} = \int_{\Omega_{h_{fe}}^n} \mathbf{F}^n \cdot \mathbf{w} d\mathbf{x}, \\ & + \sum_{0 \leq i, j \leq N, i \neq j} \int_{\Gamma_{ij, h_{fe}}^n} \gamma_{ij} H_{ij, h_{fe}}^n \mathbf{n}_{\Gamma_{ij, h_{fe}}^n} \cdot \nabla \mathbf{w} dS \end{aligned} \tag{23}$$

for all $\mathbf{w} \in (V_{h_{fe}}^n)^d$ and $q \in Q_{h_{fe}}^n$, compatible with the essential boundary conditions on $\partial\Omega_{h_{fe}}^n$.

We opt for the approach in [22] to automatically incorporate the surface terms, i.e., not only the surface tension effects, but also the contact angles terms. More precisely, we use a surface divergence formula in order to modify the last term of (23):

$$\int_{\Gamma_{ij, h_{fe}}^n} H_{ij, h_{fe}}^n \mathbf{n}_{\Gamma_{ij, h_{fe}}^n} \cdot \nabla \mathbf{w} dS = - \int_{\Gamma_{ij, h_{fe}}^n} \operatorname{tr} \left(\nabla_{\Gamma_{ij, h_{fe}}^n} \nabla \mathbf{w} \right) dS + \int_{\partial\Gamma_{ij, h_{fe}}^n} \nabla \mathbf{w} \cdot \mathbf{m} d\ell.$$

The corresponding linear system is solved with a sequential, preconditioned GMRES method.

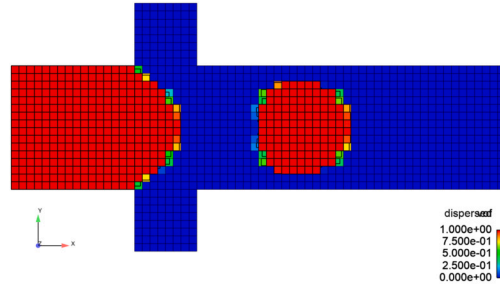


Fig. 6. Visualization of the SLIC algorithm to concentrate the VOF phases prior to the advection with the forward characteristics method.

4.3.1. Interpolation/projection on the structured grid

The continuous approximation of the velocity $\mathbf{v}_{h_{fe}}^{n+1/2} \in (V_{h_{fe}}^n)^d$ is then interpolated at the center of each grid cell in order to obtain a piecewise constant approximation $\mathbf{v}_{h_{cells}}^{n+1/2} \in (W_{h_{cells}}^n)^d$. The value of the interpolant at cell $C \in \mathcal{C}_{h_{cells}}$ is

$$\mathbf{v}_{h_{cells}}^{n+1/2}(C) = \frac{\sum_{K \in \mathcal{N}_C} \sum_{\mathbf{p} \in K} \mathbf{v}_{h_{fe}}^{n+1/2}(\mathbf{p}) \|\mathbf{q}_C - \mathbf{p}\|^{-2}}{\sum_{K \in \mathcal{N}_C} \sum_{\mathbf{p} \in K} \|\mathbf{q}_C - \mathbf{p}\|^{-2}}, \quad (24)$$

where $\mathcal{N}_C := \{K \in \mathcal{T}_{h_{fe}}^n : K \cap C \neq \emptyset\}$ is the set of elements that intersect the cell C [26].

4.4. Advection operators

Equations (14)(15) are solved using a *forward characteristics method with projection* on the structured grid $\mathcal{C}_{h_{cells}}$, together with a SLIC interface reconstruction algorithm for the approximation of $\varphi_{\ell, h_{cells}}^{n+1}$ [18], and post-processing heuristics to avoid artificial compression.

More precisely, at time t^n , the algorithm consists in moving each cell $C \in \mathcal{C}_{h_{cells}}$ along the linearized characteristics $\tau^n \mathbf{v}_{h_{cells}}^{n+1/2}(C)$, and conservatively redistributing the transported quantities $\varphi_{\ell, h_{cells}}^n$ and $\mathbf{v}_{h_{cells}}^{n+1/2}$ into the overlapped cells (proportionally to the volume of intersection between the transported cell and the overlapped cells) [19,20]. Although the forward characteristics method does not have to satisfy a CFL condition theoretically, the time step τ^n is chosen in order to enforce the CFL number to be typically between 1 and 5.

In order to enhance the quality of the approximations $\varphi_{\ell, h_{cells}}^{n+1}$, and thus reduce the numerical diffusion of the interfaces, we use a variation of the SLIC algorithm inspired by [18]. The SLIC algorithm is not used for interface reconstruction. It is used prior to the advection of each cell: instead of being diffuse in each cell, the volume fractions are geometrically condensed towards a sub-region of the cell (according to the neighboring values). Fig. 6 illustrates the SLIC algorithm for the emulsion process, and shows the condensed phase location for the cells near the interfaces. The objective is to reduce the numerical diffusion of the forward characteristic method, but not to reconstruct the interface.

Replacing the SLIC algorithm with a PLIC approach would thus lead to slightly different values of the volume fractions on the grid, but with little impact on the surface mesh constructed with the Marching Cubes 33 algorithm, and on the cut finite element mesh.

Furthermore, when dealing with multiple phases, the SLIC reconstruction, the transport of volume fractions, the creation of new edges in the cut-cell method, and the filling of voids are done independently for each phase. Therefore, the ordering of phases does not have any impact. Details can be found in [19,20].

In order to avoid the artificial compression of the fluid phases (i.e. where $\varphi_{h_{cells}}^{n+1} > 1$), a post-processing technique related to global repair algorithms [27] is implemented. This *decompression* technique redistributes the excess of liquid from over-filled cells to the cells that are partially empty, and produces final values $\varphi_{h_{cells}}^{n+1}$ which are between zero and one. The decompression algorithm is described in [14] and summarized, together with a custom correction step, in Appendix A.

The decompression algorithm favors conservation over causality. Fig. 7 illustrates the cells that are impacted by the final correction step. It emphasizes that i) these cells are strictly located at the interfaces and ii) the magnitude of the correction is of order 0.1%.

5. Numerical experiments

Numerical experiments are presented to validate the numerical methods with successive benchmark cases, and finally focus on microfluidics and the emulsion processes. We highlight two aspects of the proposed numerical framework: the geometrical reconstruction of the interfaces, and the enforcement of surface forces on the interfaces. All numerical experiments have been

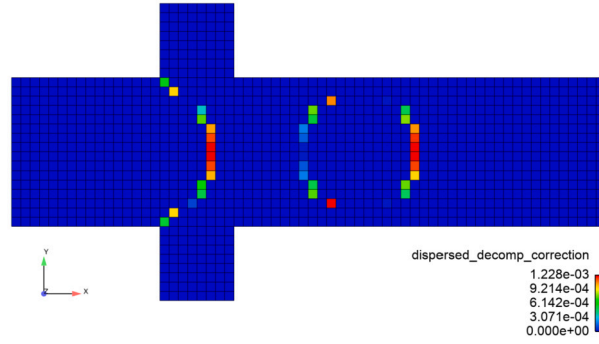


Fig. 7. Visualization of the cells impacted by the correction step applied after the redistribution algorithm (Step 4. of Appendix A). Location of the cells in which a small amount of the continuous phase has been extracted and replaced by the dispersed phase in order to empty the buffer of the dispersed phase. The correction is of the order of 0.1% in each individual cell.

performed on the same workstation, based on 64 Core AMD EPYC 7742 2.25GHz Processor, (Turbo up to 3.4GHz), and 256GB DDR4 3200MHz ECC Memory.

5.1. Interface reconstruction via multiphase mesh cutting algorithm

In a first step, the interface reconstruction and the mesh cutting algorithm are validated with pure geometric configurations, involving three or more phases with interfaces and free surfaces, see, e.g., [28–31]. The accuracy of the interfaces approximation is quantified by the *relative volume error* defined as:

$$\varepsilon := \sum_{\ell=1}^L \frac{|V_{\ell} - V_{\ell,ex}|}{V_{\ell,ex}}, \quad (25)$$

where $V_{\ell} = |\Omega_{\ell}|$ is the approximated volume of the domain containing phase ℓ and $V_{\ell,ex}$ is the exact geometrical volume. We consider configurations in two space dimensions within a unit square discretized with N mesh elements per side. The first configuration is visualized in Fig. 8 and consists of two phases in concentric circles surrounded by a third phase. It shows that the mesh cutting algorithm accurately tracks the interfaces. Fig. 9 shows the convergence of the approximation of the volumes, which is slightly better than first-order convergence.

The second configuration is visualized in Fig. 10, and includes a triple-point [28,29]. With triple-points, the reconstruction of the interfaces is problematic and a void \mathcal{M} is created (top row). This issue vanishes when adding the correction step described in Section 4.2 (bottom row). Thus this correction step to fill potential voids is applied for all numerical results presented hereafter. Fig. 11 shows again a convergence for the relative volume error that is slightly better than first-order convergence.

Finally, the third configuration is visualized in Fig. 12, and includes four phases. Flat horizontal interfaces are accurately reconstructed also with unstructured meshes, as well as curved boundaries and triple-points. Note that the mesh cutting procedure appears to be smoother with unstructured meshes. Fig. 13 shows again a kind of super-convergence for the relative volume error.

Even though some of the examples presented in this Section include triple- or quadruple-points, they will not be discussed further in the sequel, as the main application of this work (emulsion processes) does not include such configurations.

5.2. Advection of four-material disk in a swirling motion

We consider the two-dimensional transport of multiple phases within a *given velocity field* (stretching flow, see, e.g., [32,33]). Within our framework, such a pure advection problem is solved only on the structured grid, therefore the cut-cell algorithm on the underlying finite-element mesh acts as a post-processing only. In the sequel of [14], we verify here the behavior of the cut-cell method.

The initial liquid domain consists of a disk of radius 0.15 centered at the point (0.5, 0.75). It is split into four liquid phases, each of them filling a quarter of the disk, as illustrated in Fig. 14. The advection velocity is given by:

$$\mathbf{u}(x, y, t) = \begin{cases} 2 \sin(\pi y) \cos(\pi y) \sin(\pi x) \sin(\pi x) \cos(\pi t/T), \\ 2 \sin(\pi x) \cos(\pi x) \sin(\pi y) \sin(\pi y) \cos(\pi t/T). \end{cases} \quad (26)$$

This prescribed velocity being periodic in time, the initial liquid domain is recovered at time $T = 2$. The computational domain is $\Lambda = [0, 1]^2$, with a discretization of $h_{fe} = 1/100$, $h_{cells} = 1/400$ and $\tau = 1/400$.

Fig. 14 illustrates the position of the liquid phases at initial time ($t = 0$), at the time of maximal deformation ($t = 1$), and at final time ($t = 2$). One can see that the cut-cell method creates an undefined region near the quadruple point that is filled with one phase only. No special emphasis is made on the reconstruction of this central quadruple point. In the test cases presented hereafter, this situation will not appear as we consider only two-fluid cases, with a final focus on emulsion processes.

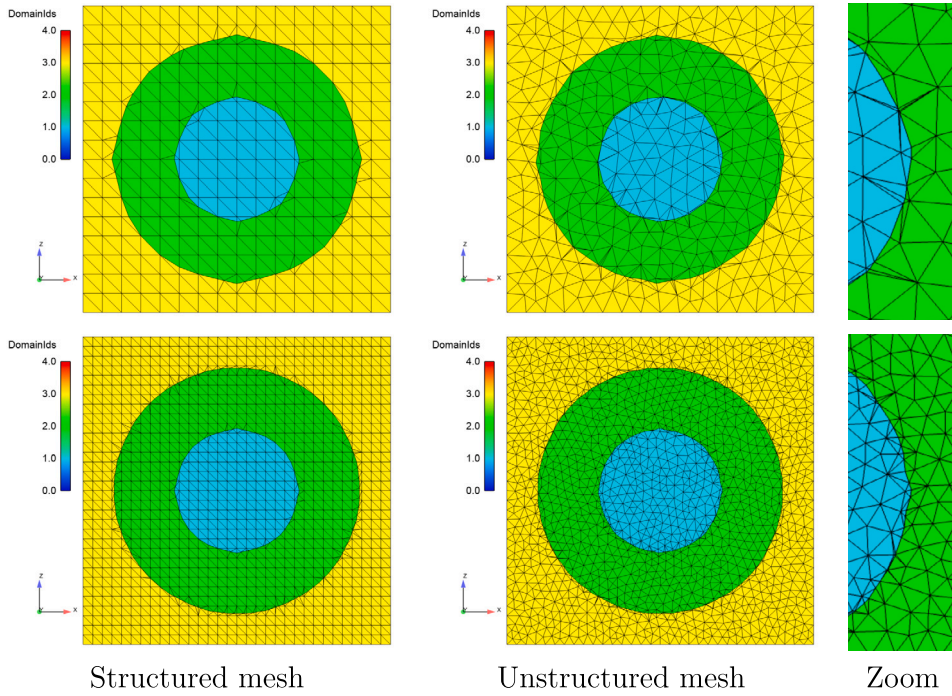


Fig. 8. Interface reconstruction with mesh cutting. Configuration 1. Top row: coarse mesh ($N = 16$); bottom row: fine mesh ($N = 32$). Left: structured mesh; right: unstructured mesh (with zoom on the right part of the interface).

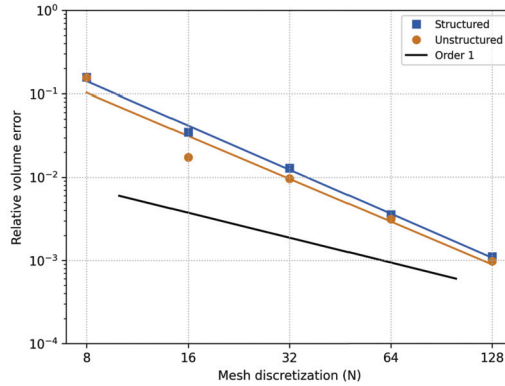


Fig. 9. Interface reconstruction with mesh cutting. Configuration 1. Convergence of relative volume error.

5.3. Lock-exchange experiment

As in, e.g., [14,34], we perform here numerical simulations of a *lock-exchange flow* with a high density ratio. We consider the setup described in [35]. A rectangular tank $\Lambda = (-1.5, 3.0) \times (0, 0.3) \times (0, h_{fe})$ is completely filled (i.e. $\Omega(t) = \Lambda$) with two immiscible fluids of densities ρ_1 and ρ_2 ($\rho_1 < \rho_2$) respectively, with ratio $r := \rho_2/\rho_1$. This setup is a pseudo-2D geometry, i.e. in a thin-layered domain. Both fluids are initially at rest and separated by a vertical gate located at $x = 0$.

Here we consider a ratio $r = 9.93$, with $\rho_1 = 0.169454$ [kg m⁻³], $\rho_2 = 1.68268$ [kg m⁻³], $\mu_1 = 1.864 \cdot 10^{-5}$ [Pa s] and $\mu_2 = 2.1 \cdot 10^{-5}$ [Pa s]. A sharp front remains visible during the experiment reported in [35] as well as in some numerical simulations [34]. No surface tension is enforced in this particular case.

We consider four different discretizations of the computational domain Λ with $N = 10, 20, 40$ and 80 mesh elements in the vertical part of the tank, such that $h_{fe} = 0.3/N$. We choose $h_{cells} = h_{fe}/4$ and $\tau^n = 0.2/N$. As in [35] we perform two distinct experiments: we start the simulation with the denser (resp. lighter) fluid in the left part of the cavity, with the interface located at one-third of the length of the cavity. Then we observe the progression of the front along the bottom (resp. top) part of the cavity when the front is moving to the right.

Fig. 15 shows snapshots of the evolution of the two liquids on the structured grid (when $N = 80$). As expected in the absence of turbulence modeling, numerical instabilities are observed at some point. Convergence results are illustrated in Figs. 16 and 17.

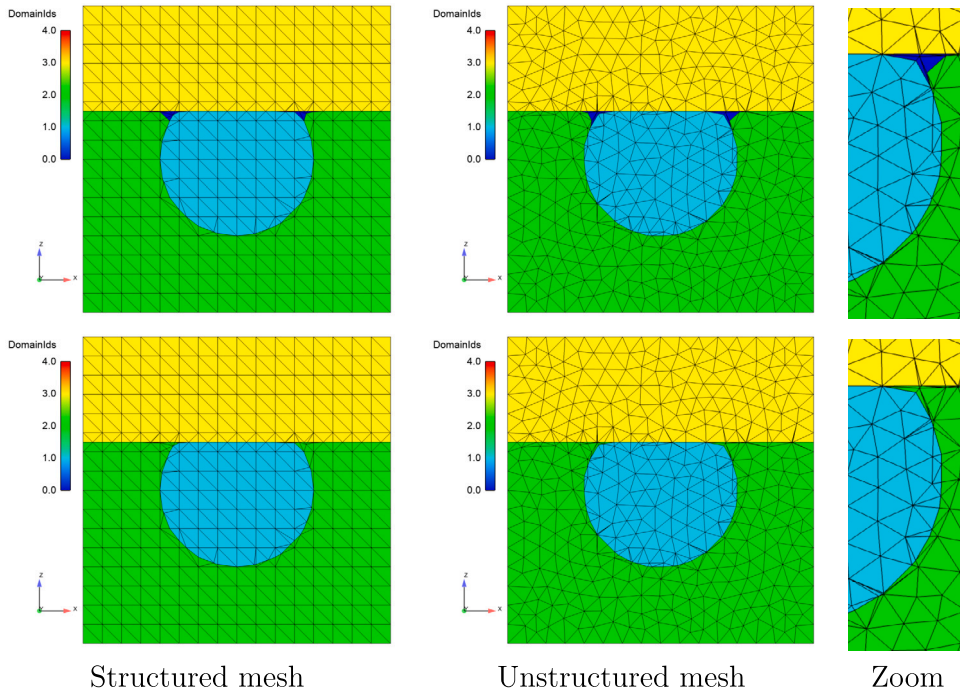


Fig. 10. Interface reconstruction with mesh cutting. Configuration 2. Coarse mesh ($N = 16$). Top row: without the filling heuristic of the undetermined elements in \mathcal{M} ; bottom row: with the filling heuristic of the undetermined elements in \mathcal{M} . Left: structured mesh; right: unstructured mesh (with zoom on the triple-point).

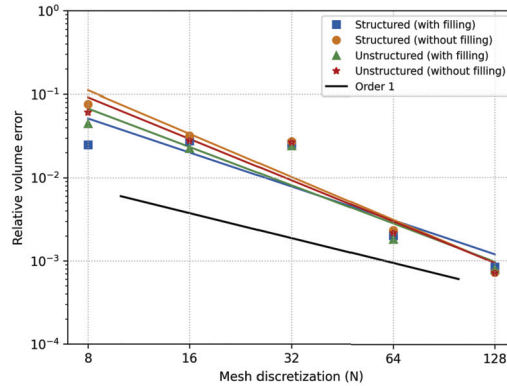


Fig. 11. Interface reconstruction with mesh cutting. Configuration 2. Convergence of relative volume error.

Fig. 16 illustrates the time evolution of the front, and compares with experimental data [35], for the dense (left column) and light (right column) fronts evolution (with slip boundary conditions). We observe that, when the discretization becomes finer, the fronts move faster. The evolution of the light front is more homogeneous, while the dense front shows more variability for various values of N . Moreover, the slope of the light front evolution is more accurate. We note that the influence of the mesh cutting algorithm is very limited. Fig. 17 shows the convergence orders, and shows that the fronts are approximated with order one. The mesh cutting algorithm does not influence the accuracy of the dense front evolution. However, the accuracy of the light front gets better with adaptive mesh refinement. Fig. 15 visualizes the numerical approximations of φ_1 and φ_2 on the structured grid of small cells. It presents little inclusions of the light material inside the heavier one, which do not appear on the finite element mesh after interpolation from the fine grid to the coarser mesh.

5.4. Surface tension and contact angles for static droplets

We address here the case of hydrophilic and hydrophobic droplets resting on a horizontal plane. We enforce contact angles via the approach advocated in [22], with angles ranging from 10 to 170 degrees.

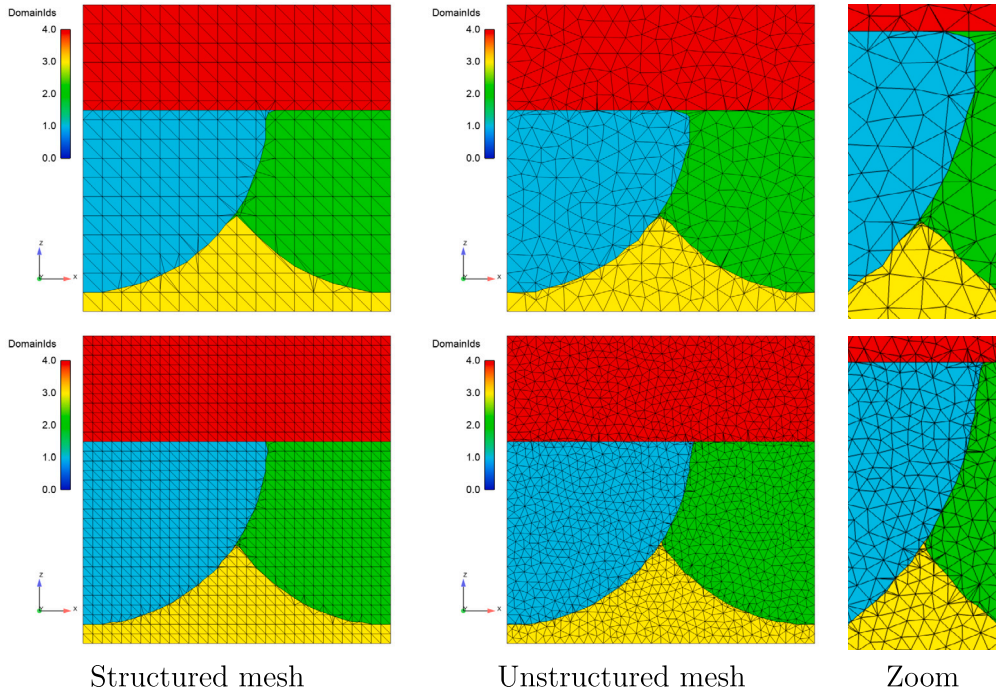


Fig. 12. Interface reconstruction with mesh cutting. Configuration 3. Top row: coarse mesh ($N = 16$); bottom row: fine mesh ($N = 32$). Left: structured mesh; right: unstructured mesh (with zoom on the triple-point).

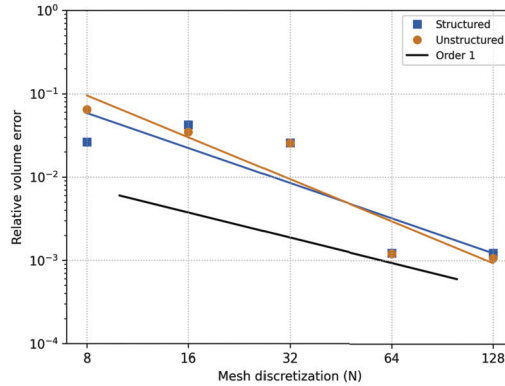


Fig. 13. Interface reconstruction with mesh cutting. Configuration 3. Convergence of relative volume error.

In a first step, we focus on a 2D geometrical setup in a domain $\Lambda = (-4, 4) \times (0, 2)$ [mm]. We consider a structured and symmetric finite element mesh with N discretization points along the x -direction such that $h_{fe} = 8/N$ and $h_{cells} = h_{fe}/4$; we are interested in the stationary solution of an evolutive problem approximated with adaptive time steps such that $CFL = \max(|\mathbf{v}_{h_{cells}}^n| \tau^n / h_{cells}) \leq 1$.

We consider a single fluid phase with density $\rho = 10^{-6}$ [kg m⁻³] and viscosity $\mu = 10^{-4}$ [Pa s] and the droplet is initialized as a half disk of radius 1 [mm] centered at the origin. The surface tension coefficient is set to $\gamma_{01} = 0.07$ [N/m].

Fig. 18 shows the profile of the stationary interface $\Gamma_{0,1}$ for various angles and discretization parameters. We can observe that the algorithm can reproduce a large range of shapes for various contact angles. Fig. 19 shows the error between the expected and the approximate values of the contact angles at stationary solutions. The approximate contact angle is computed as $\theta = \arcsin(d/(2R))$, where d and R are respectively the droplet diameter and the radius of the circle that fits (in a least-squares sense) the free surface. One can see that the algorithm reaches a very good accuracy for angles ranging from 50 to 130 degrees. Then the accuracy deteriorates for larger/smaller angles. The lack of accuracy is coming from the mesh approximation and does not vanish, even when refining the meshes. The numerical approximation of the contact angle is always more conservative than the real value. The order of magnitude of the walltime for such 2D calculations is about 8 [min] (when $N = 32$), 16 [min] (when $N = 64$), and 48 [min] (when $N = 128$).

In a second step, we focus on a full 3D geometrical setup in a domain $\Lambda = (-4, 4) \times (-4, 4) \times (0, 2)$ [mm] with the same fluid properties and discretization parameters. The droplet is initialized as the hemisphere of radius 1 [mm] centered at the origin.

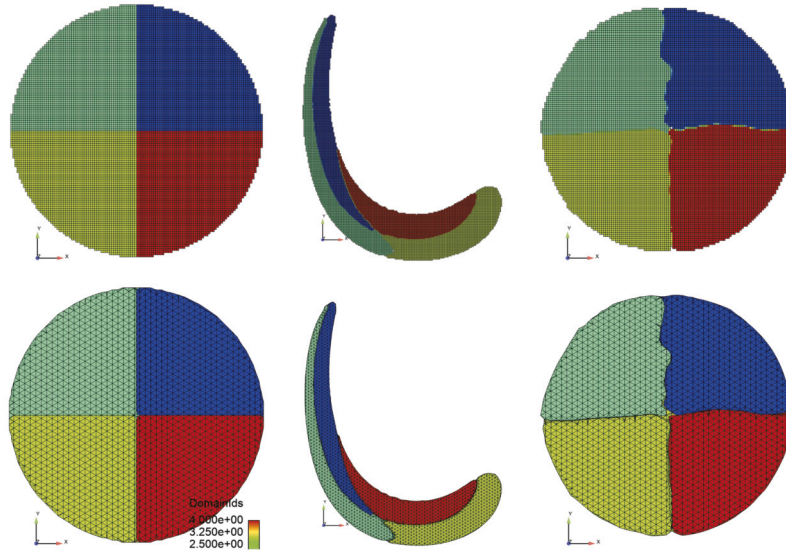


Fig. 14. Single vortex test case with period $T = 2$. Representation of the computed interfaces at times $t = 0$ (initial shape, left), $t = 1$ (maximal deformation, middle) and $t = 2$ (return to initial shape, right). First row : visualization on the structured grid; Second row : visualization on the finite element mesh after the cut-cell post-processing.



Fig. 15. Lock-exchange experiment. Snapshots of the numerical approximation of φ_1 and φ_2 at times $t = 0, 0.1, 0.2, 0.5, 1.0, 2.0, 3.0$ [s] with $N = 80$. Left column: evolution of the dense front when the denser fluid is located on the left; right column: evolution of the light front when the lighter fluid is located on the left.

Fig. 20 visualizes snapshots of the approximated stationary droplet surface for various values of the contact angle. Fig. 21 (first row) shows again the profile of the stationary interface for various contact angles and discretization parameters, in a vertical cut. Conclusions are similar as those for the 2D case. Fig. 21 (second row) shows the re-centered trace of the contact surface of the droplet on the horizontal plane at stationary solution. We can observe that this contact surface is indeed formed by concentric circular shapes and that its accuracy increases when N increases. Finally, Fig. 22 shows the error between the expected and the approximate values of the contact angle. We reach the same conclusion that the algorithm provides very good accuracy for contact angles ranging from 50 to 130 degrees. The order of magnitude of the walltime for such 3D calculations is about 16 [min] (when $N = 32$) and 300 [min] (when $N = 64$).

5.5. 2D emulsion processes

Finally, we address the simulation of emulsion processes in a cross-junction microchannel as illustrated in Fig. 1. We consider the same setup as in [3], where the subscript ‘d’ refers to the dispersed phase (φ_1) injected in the main horizontal channel of width $w_d = 200$ [μm] and the subscript ‘c’ refers to the continuous phase (φ_2) introduced in the two lateral channels of width $w_c = 100$ [μm]. The length of the channels before the cross-junction is w_d and w_c respectively, while the length of the channel downstream the cross-junction is equal to $17w_c = 1700$ [μm]. The goal is to investigate the droplet formation mechanism and compare our results with those presented in [3].

For a given geometry, the process parameters are the inlet volumetric flow rates (Q_d and Q_c), the fluid densities (ρ_d and ρ_c) and viscosities (μ_d and μ_c), and the surface tension γ_{12} . The droplet diameter $d = d(Ca, Q, \lambda)$, defined as the diameter of the disk that has the same surface as the droplet, is governed by three dimensionless numbers [3]: the capillary number $Ca = \frac{u_c \mu_c}{\gamma_{12}} = \frac{Q_c \mu_c}{2w_c \gamma_{12}}$

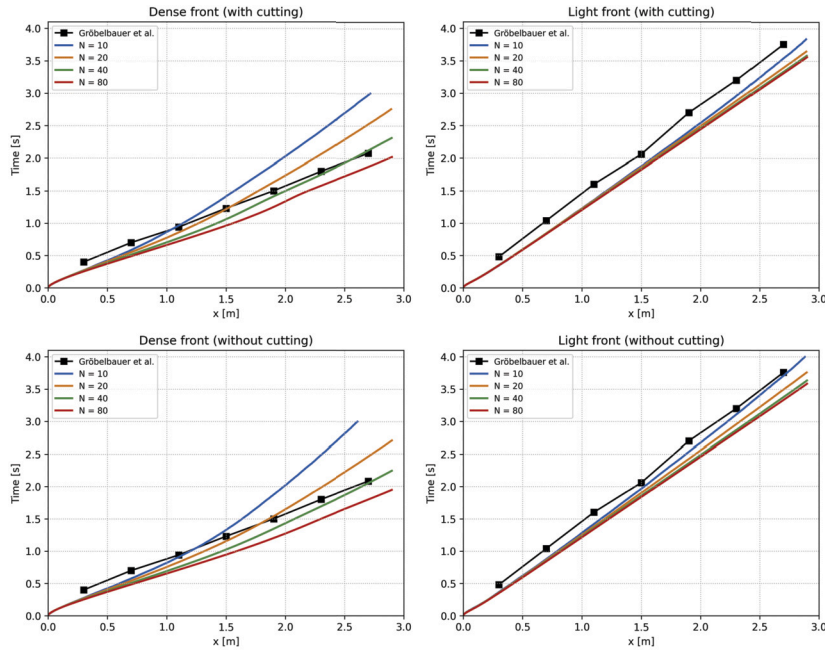


Fig. 16. Lock-exchange experiment. Time evolution of the interface fronts. Left column: dense front; right column: light front. Top row: with adaptive mesh cutting algorithm; bottom row: without adaptive mesh cutting algorithm.

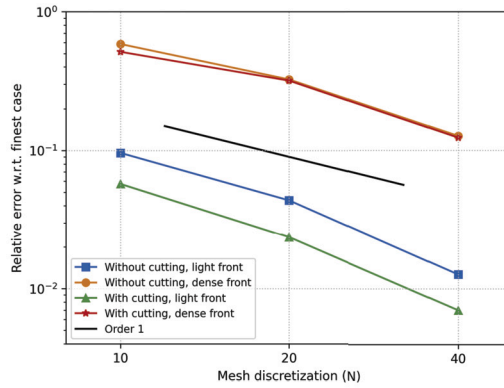


Fig. 17. Lock-exchange experiment. Convergence orders for the approximation of the time evolution of the dense and light fronts, with and without adaptive mesh cutting algorithm.

describing the ratio between viscous and surface tension forces, where u_c is the average inlet velocity of the continuous phase; the flow rate ratio $Q = \frac{Q_d}{Q_c}$, and the viscosity ratio $\lambda = \frac{\mu_d}{\mu_c}$. The Reynolds number $Re = \frac{\rho_c u_c w_c}{\mu_c}$, describing the ratio between inertial and viscous forces, remains small ($Re < 2$) and therefore has negligible effect on the droplet formation for the flow regimes considered in the present study.

The geometry is the one illustrated in Fig. 1. It is discretized with $N = 11$ finite elements in the width of the main channel such that $h_{fe} = w_d/N$. We choose $h_{cells} = h_{fe}/4$ and the time step is adapted in order to target a unitary CFL number.

Numerically, parabolic velocity profiles are imposed on the three channel inlets, a free outflow condition is enforced on the main channel outlet, and no-slip conditions are enforced on the rest of the boundary. At time $t = 0$, the horizontal inlet tube is filled with the dispersed phase ($\varphi_d^0(x) = 1$ if $x \leq -\frac{W_c}{2}$, zero otherwise), while the rest of the domain Λ (two vertical inlet tubes and horizontal outlet tube) is filled with the continuous phase. The velocity field is initialized to zero. Piecewise constant values of those fields are enforced on each grid cell. In order to be consistent with [3], in the following the working units are $L_0 = 10^{-5}$ [m] (length scale), $T_0 = 10^{-6}$ [s] (time scale) and $M_0 = 10^{-12}$ [kg] (mass scale).

5.5.1. The effect of the capillary number

In a first step, we study the effect of the capillary number on the droplet diameter. As in [3] we consider $\rho_d = \rho_c = 1$, $\mu_c = 0.08$, $\lambda = 1/4$ and $\gamma_{12} = 0.016$. For fixed values of Q_d , we control Ca by only varying Q_c .

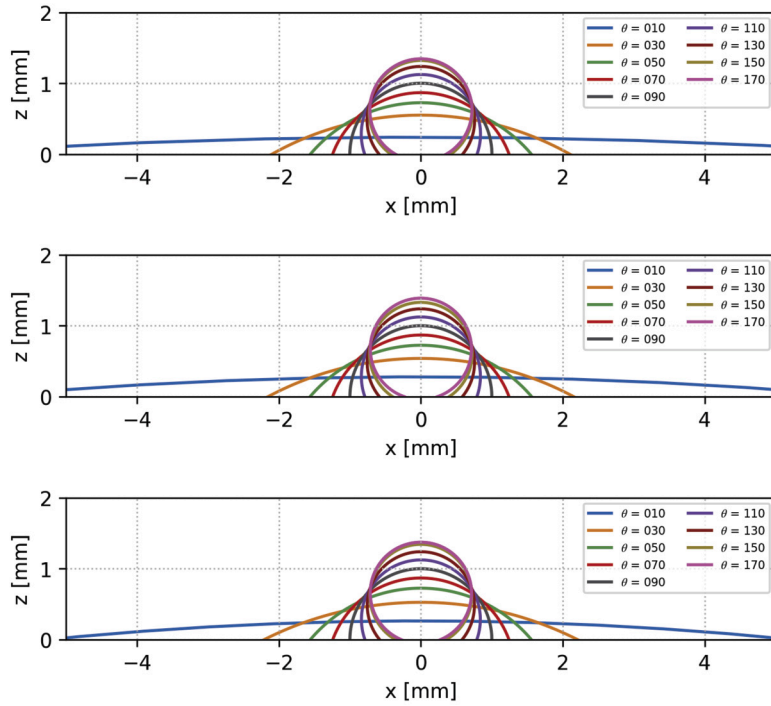


Fig. 18. Contact angles in a 2D setup. Snapshots of the stationary droplet shape for several values of the contact angle. Top: $N = 32$; middle: $N = 64$; bottom: $N = 128$.

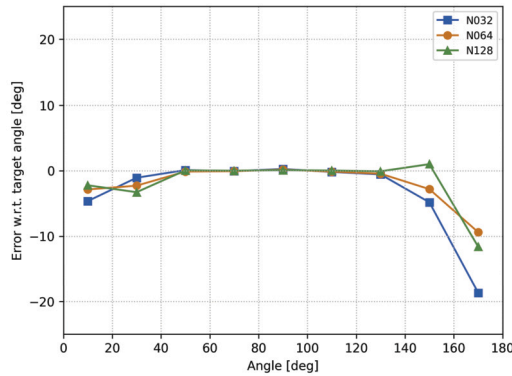


Fig. 19. Contact angles in a 2D setup. Approximation error between the exact and approximate values of the contact angle.

The formation of one droplet is illustrated in Fig. 23, where snapshots of the emulsion process are shown at several time steps for $Q_d = 0.004$ and $Ca = 0.02$ and 0.06 . We can observe the three stages of droplet formation, identified in [3] as expansion (row 1), necking (rows 2-3) and figuration (rows 4-6). As expected, a smaller capillary number generates larger droplets.

Fig. 24 illustrates the diameter and spacing of droplets over time, for several flow rates of the dispersed phase and various capillary numbers. The droplets are identified as connected components of the dispersed phase (computed with a depth-first-search algorithm). Each point corresponds to the moment at which a new droplet is created. One can observe that the diameter and the frequency between the droplets are quite regular in all regimes.

Fig. 25 illustrates the average droplet diameter as a function of the capillary number, for several values of Q_d . The numerical results are compared with those presented in [3]. At the moment, our numerical model is not capable of imposing complete wetting/non-wetting conditions on the boundary of the main channel, and therefore it is impossible to generate big droplets for which $d > w_d$ (i.e. when $Ca < 0.015$) as they eventually enter in contact with the boundary of the computational domain. When the capillary number is between 0.015 and 0.035 , our numerical results are appropriately overlapping with those in [3]. Numerical results can also be provided for larger values of Ca and show a coherent fit with the literature results.

Fig. 26 illustrates the average time interval between successive droplets breakups, as a function of the capillary number and for several values of Q_d . We observe that the interval decreases when Ca increases, and is nearly constant with respect to Ca when Q_d is large.

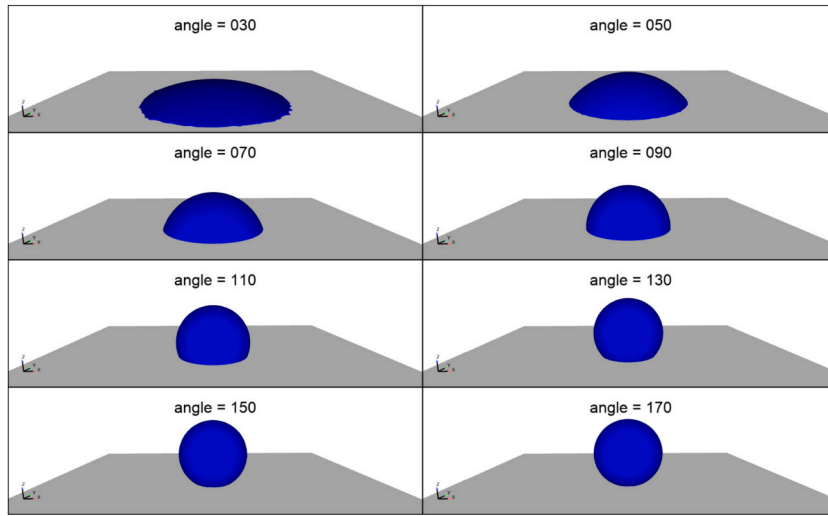


Fig. 20. Contact angles in a 3D setup. Snapshots of the approximated stationary droplet surface for various values of the contact angle. ($N = 64$).

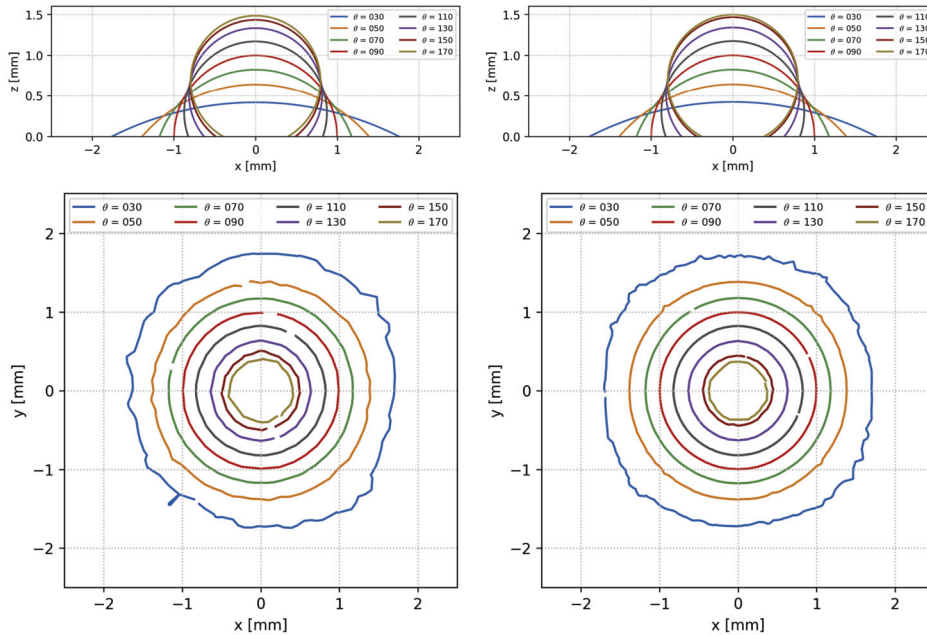


Fig. 21. Contact angles in a 3D setup. Top: droplet shape for several values of the contact angle. Bottom: Contact surface of the droplet on the horizontal plane. Left: $N = 32$; right: $N = 64$.

Fig. 27 illustrates the location of the droplets breakup (on the horizontal axis), as a function of the capillary number and for several values of Q_d . When the capillary number increases (larger lateral velocity, smaller droplets), one can see that the breakup location is constant for all values of the dispersed flow rate Q_d . When it becomes smaller, the breakup location has a tendency to be further away from the cross-junction of the channels.

5.5.2. The effect of the flow rates ratio

In a second step, we study the effect of the flow rate ratio Q . As in [3], we keep the same density and viscosity for both fluids but we now consider $\gamma_{12} = 0.010$. For fixed values of the flow rate ratio Q , we control Ca by varying Q_c and deduce Q_d accordingly.

Droplet regimes are illustrated in Fig. 28, where snapshots are shown for two values of the capillary number and three values of Q . The color code corresponds to the index of the connected components of the dispersed phase, when tracking the various droplets. We observe that, when Q decreases, the time interval between droplets' breakups increases. On the other hand, the capillary number governs the size and frequency of the droplets. Fig. 29 illustrates the diameter and frequency of droplets over time, for $Ca = 0.16$

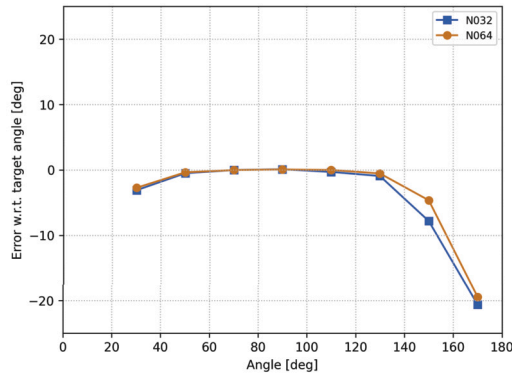


Fig. 22. Contact angles in a 3D setup. Approximation error between the exact and approximate values of the contact angle.

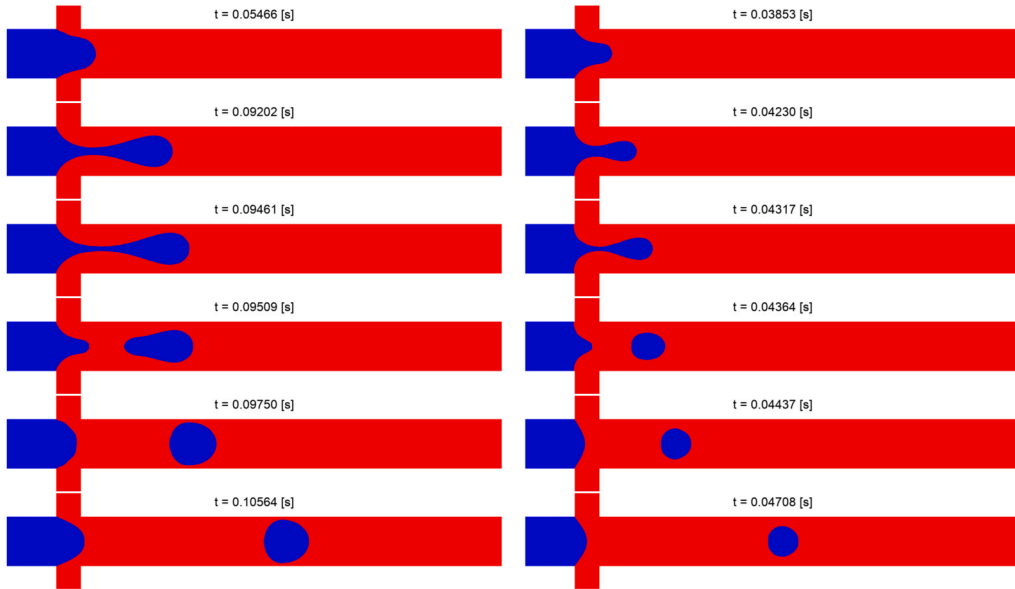


Fig. 23. Emulsion in 2D configuration. Snapshots of the numerical approximations of φ_1 and φ_2 when $Q_d = 0.004$ and $Ca = 0.02$ (left) or $Ca = 0.06$ (right).

and several values of the flow rate ratio. The droplet regime is very regular for larger values of Q , but a remarkable pattern can be observed when $Q = 1/16$ as two smaller droplets are consistently and repeatedly following a larger one.

Fig. 30 visualizes the relationship between the average droplet diameter and the capillary number, including a comparison with the results from [3]. Again, our numerical results are only obtained for droplets whose diameter is smaller than the main channel diameter (i.e. for $Ca > 0.02$). The results overlap for $Ca < 0.04$, and the trend is identical. The droplet diameters remain different for different values of Q . Moreover, another limitation of our model is that we only observe droplet breakups for $Q < 1/6$. Finally, Fig. 31 visualizes the average time interval between successive droplets (left) and the average breakup location (right). We remark that the time interval increases when the capillary number decreases (lower inflow velocities), but the general trend is independent of Q when Ca takes larger values. Similarly, the location of the breakup is further from the cross-junction when the capillary number decreases, meaning that the droplet takes more space to breakup. This location depends strongly on Q for small values of Ca .

5.6. 3D emulsion processes

Finally, let us expand our numerical results to a full 3D geometry. We now consider two cylinders intersecting each other perpendicularly as visualized in Fig. 32. Both cylinders have equal diameter $w_d = w_c = 0.1$ [mm]. All inlet channels have a length of 0.05 [mm], while the outlet channel has a length of 0.85 [mm].

The inflow rates are given by $Q_d = 0.048$ [mm³/s] and $Q_c = 0.768$ [mm³/s]. The physical parameters of the two phases are given by $\rho_d = \rho_c = 1000$ [kg/m³], $\mu_d = 0.05$ [Pa s], $\mu_c = 0.001$ [Pa s] and $\gamma_{12} = 0.01$ [N/m]. No-slip boundary conditions are enforced on the lateral walls. Fig. 33 illustrates snapshots of the numerical solution (dispersed phase) at two time steps Fig. 34 illustrates snapshots of the numerical solution (dispersed phase) at several time steps, and from a top and side views. One can observe that the numerical solution on rows number 4 and number 8 are very similar, which illustrates the periodicity of the process.

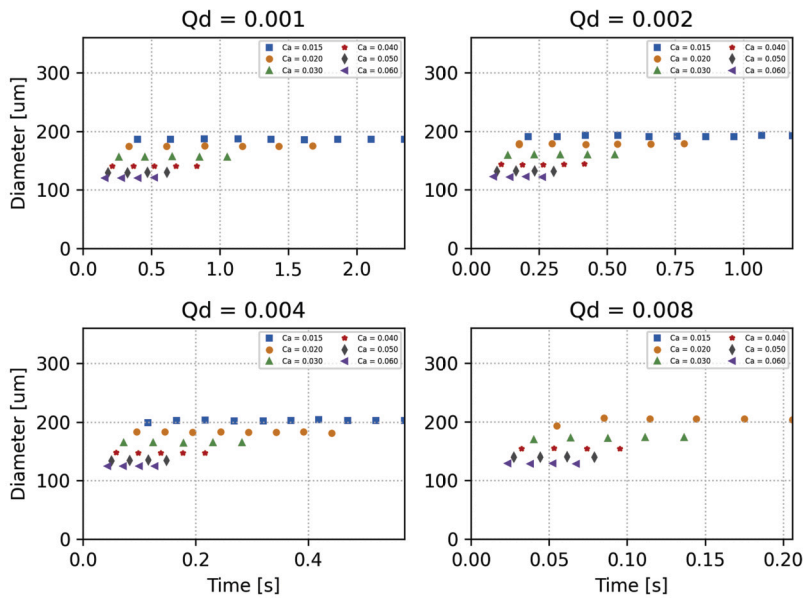


Fig. 24. Emulsion in 2D configuration. Droplets diameter over time for several flow rates of the dispersed phase and various capillary numbers.

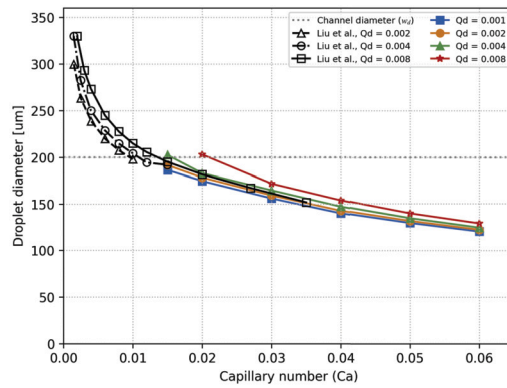


Fig. 25. Emulsion in 2D configuration. Droplet diameter as a function of the capillary number for several flow rates of the dispersed phase. Comparison with results from [3].

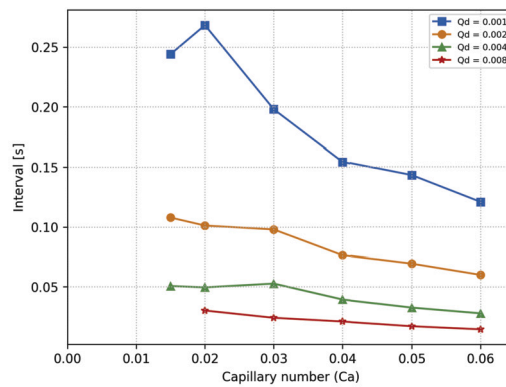


Fig. 26. Emulsion in 2D configuration. Influence of capillary number on the average time interval between successive droplets.

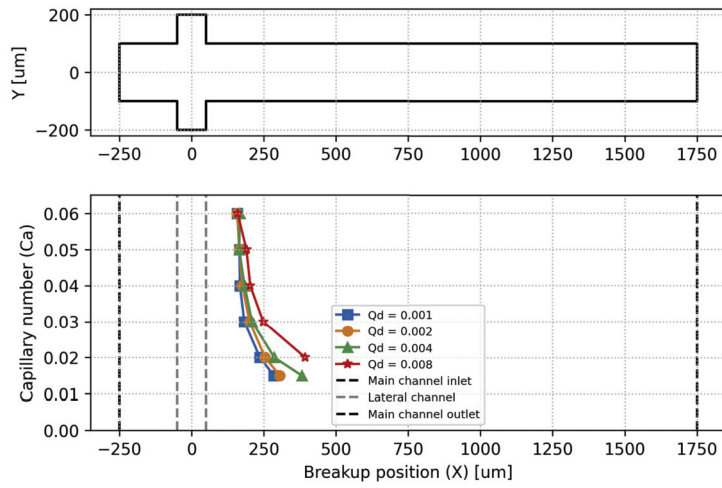


Fig. 27. Emulsion in 2D configuration. Influence of capillary number on the estimated breakup position.

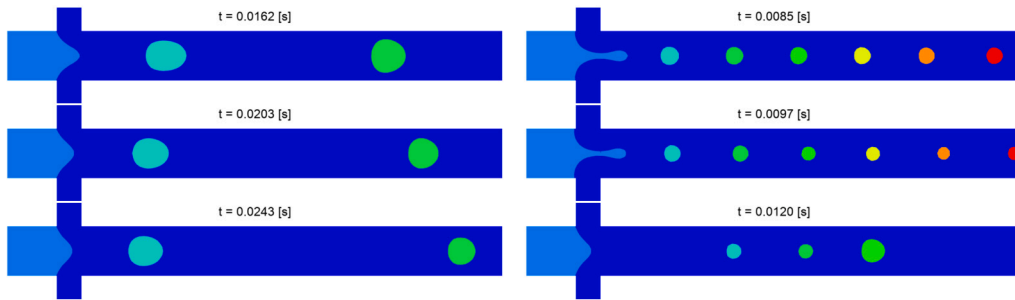


Fig. 28. Emulsion in 2D configuration. Snapshots of the numerical approximation of Ω_1 and Ω_2 . The color code corresponds to the index of the connected components of the dispersed phase, when tracking the various droplets. Left: $Ca = 0.08$; right: $Ca = 0.16$. From top to bottom: $Q = 1/8, Q = 1/12, Q = 1/16$.

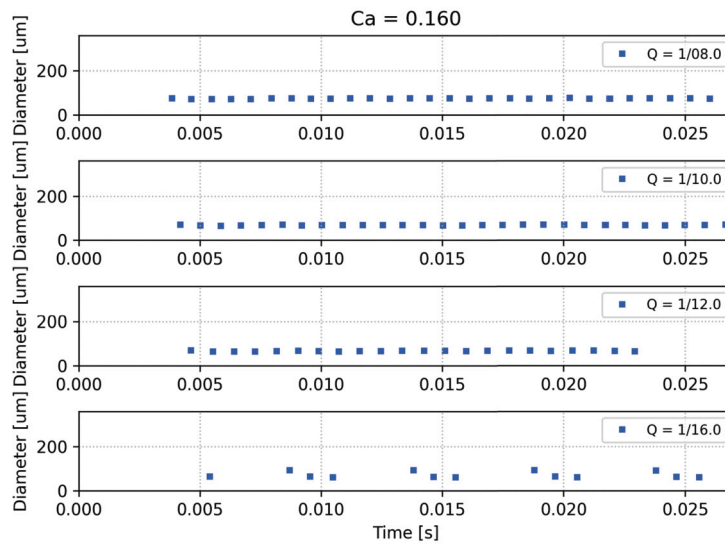


Fig. 29. Emulsion in 2D configuration. Droplet diameter over time for $Ca = 0.16$ and several flow rate ratios.

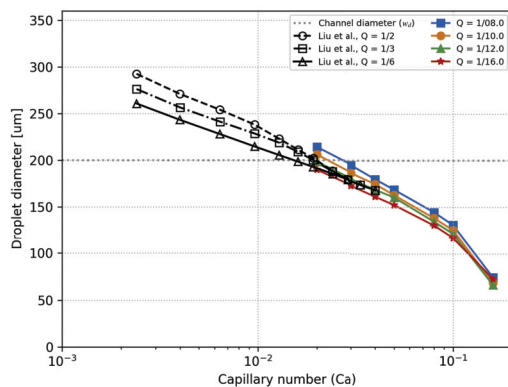


Fig. 30. Emulsion in 2D configuration. Droplet diameter as a function of the capillary number for several flow rate ratios. Comparison with results from [3].

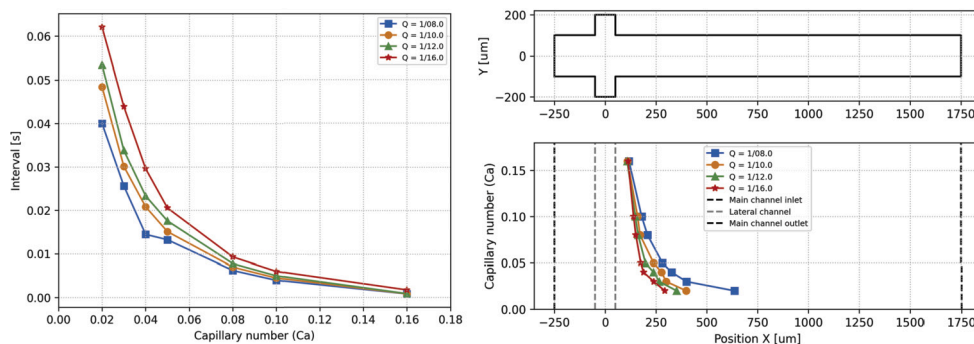


Fig. 31. Emulsion in 2D configuration. Left: influence of capillary number on the averaged time interval between successive droplets. Right: influence of capillary number on the averaged breakup position.

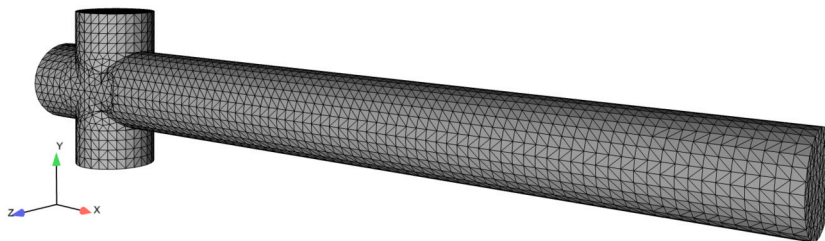


Fig. 32. Emulsion in 3D configuration. Sketch of the geometrical setup and illustration of the mesh.

6. Conclusion and perspectives

A mathematical model for the approximation of multiphase flows with free surfaces and strong interfacial effects has been presented. A numerical method relying on operator splitting and a two-grid method has been detailed. Emphasis has been put on a mesh cutting algorithm to locally adapt the mesh around the interfaces and thus approximate more accurately the surface forces. Numerical experiments have allowed to validate the model, and a sensitivity analysis of the parameters in emulsion processes has been presented. At this point, the numerical drawback of the algorithm is that there is no droplet breakup if the ratio between inflow velocities is too small, and the lack of pure wetting/non-wetting conditions does not allow to generate larger droplets filling the width of the channel. Moreover, there is no explicit treatment of triple-points in the algorithm, especially in the cut-cell method, and the current heuristics could be adapted in the future. Perspectives also include the numerical simulation of such multiphase flows but with several, non-Newtonian, rheologies.

CRediT authorship contribution statement

Alexandre Caboussat: Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Validation, Writing – original draft, Writing – review & editing. **Julien Hess:** Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Alexandre Masserey:** Conceptualization, Investigation, Validation,

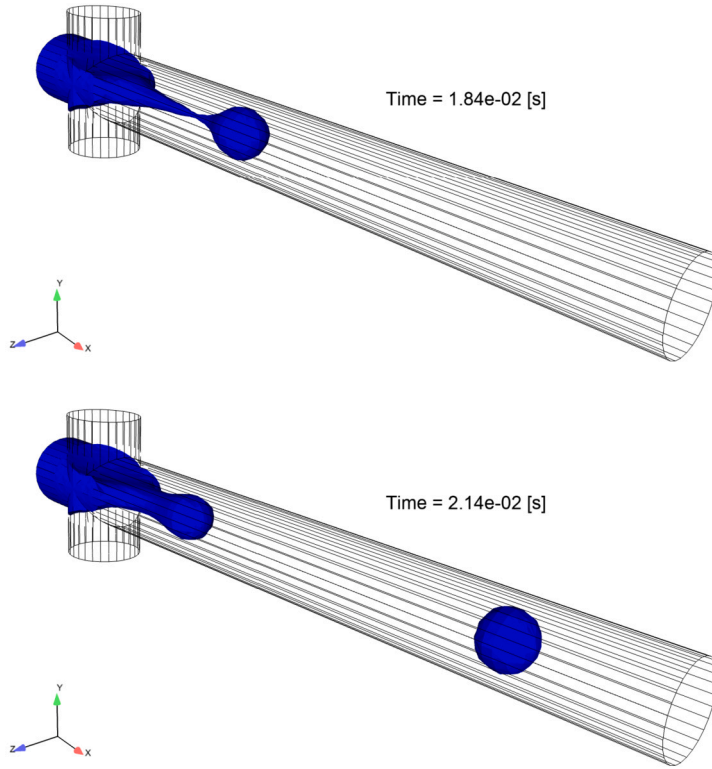


Fig. 33. Emulsion in 3D configuration. Snapshot of the numerical approximation of the dispersed phase.

tion, Visualization, Writing – review & editing. **Marco Picasso:** Conceptualization, Funding acquisition, Methodology, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors thank David Vega who graciously provided us with support to use his implementation of the Marching Cubes 33 algorithm [15]. All the computations were performed using the software `cfSFLOW++` developed by EPFL and Ycoor Systems SA.

Appendix A. Redistribution-decompression algorithm

1. **Extraction of the fluid excess from over-filled cells.** Let B_ℓ^{n+1} denote a buffer for each phase $\ell = 1, \dots, L$. For each cell $C \in C_{h_{\text{cells}}}$ such that $\varphi_{h_{\text{cells}}}^{n+1}(C) > 1$:

- (i) Compute excess of the phase ℓ : $e_\ell := (\varphi_{h_{\text{cells}}}^{n+1}(C) - 1) \frac{\varphi_{\ell, h_{\text{cells}}}^{n+1}(C)}{\varphi_{h_{\text{cells}}}^{n+1}(C)}$;
- (ii) Update phase VOF: $\varphi_{\ell, h_{\text{cells}}}^{n+1}(C) \leftarrow \varphi_{\ell, h_{\text{cells}}}^{n+1}(C) - e_\ell$;
- (iii) Update phase buffer: $B_\ell^{n+1} \leftarrow B_\ell^{n+1} + e_\ell$.

The phase buffers B_ℓ^{n+1} are completed with leftovers from the previous time step (B_ℓ^n) or with fluid lost in cells that flew out of the domain during the convection step at iteration $n + 1$.

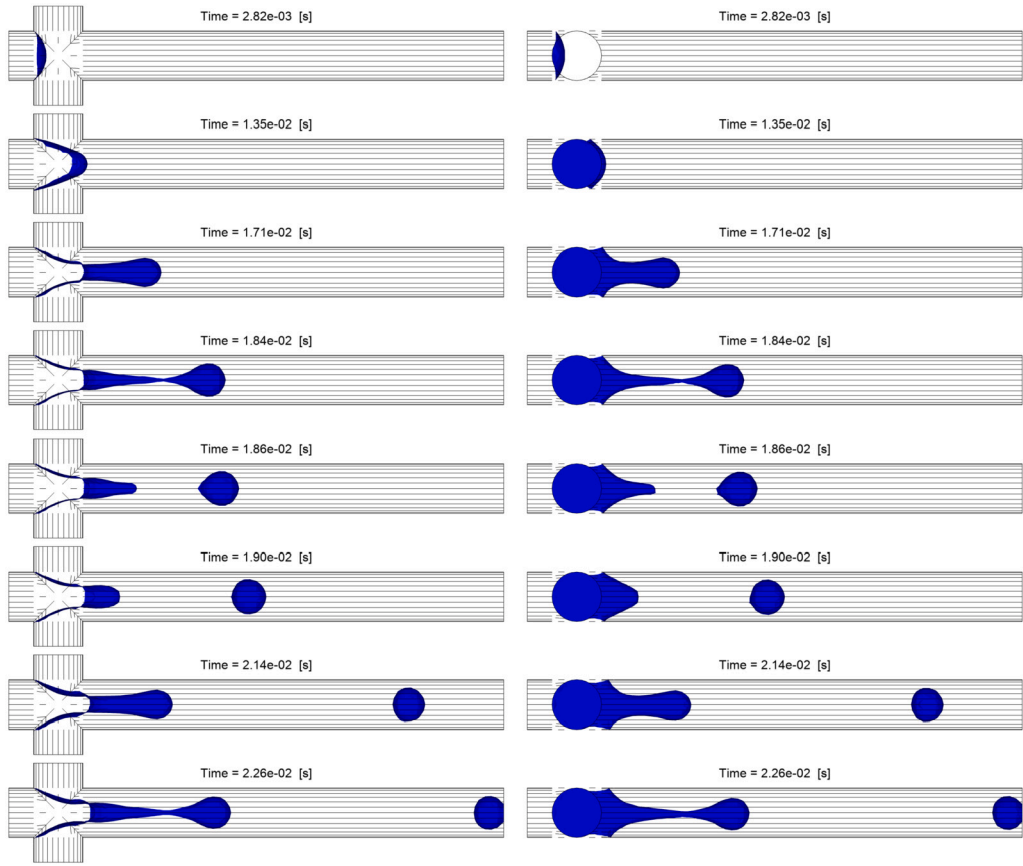


Fig. 34. Emulsion in 3D configuration. Snapshots of the numerical solution at several time steps. Left: lateral view; right: top view.

2. **Redistribution of the excess according to phase ratio.** For each cell $C \in C_{h_{cells}}$ such that $0 < \varphi_{h_{cells}}^{n+1}(C) < 1$, in decreasing order:

- (i) Compute the amount of phase ℓ redistributed in cell C : $r_\ell := \min(B_\ell^{n+1}, (1 - \varphi_{h_{cells}}^{n+1}(C)) \frac{\varphi_{\ell, h_{cells}}^{n+1}(C)}{\varphi_{h_{cells}}^{n+1}(C)})$;
- (ii) Update phase VOF: $\varphi_{\ell, h_{cells}}^{n+1}(C) \leftarrow \varphi_{\ell, h_{cells}}^{n+1}(C) + r_\ell$;
- (iii) Update phase buffer: $B_\ell^{n+1} \leftarrow B_\ell^{n+1} - r_\ell$.

3. **Redistribution of the excess according to phase order.** If needed, for each cell $C \in C_{h_{cells}}$ such that $0 < \varphi_{h_{cells}}^{n+1}(C) < 1$, in decreasing order:

- (i) Compute the amount of phase ℓ redistributed in cell C : $r_\ell := \min(B_\ell^{n+1}, (1 - \varphi_{h_{cells}}^{n+1}(C)))$;
- (ii) Update phase VOF: $\varphi_{\ell, h_{cells}}^{n+1}(C) \leftarrow \varphi_{\ell, h_{cells}}^{n+1}(C) + r_\ell$;
- (iii) Update phase buffer: $B_\ell^{n+1} \leftarrow B_\ell^{n+1} - r_\ell$.

When some phase is less present than the other ones, or when the domain is full of liquid ($\Omega(t) = \Lambda$), it may happen that $B_\ell^{n+1} > 0$ for some ℓ after the above decompression algorithm. We present a correction step to completely redistribute the buffer of some phases chosen a priori (at the expense of other phases).

Let $\mathcal{L} := \{1, \dots, L\}$ denote the set of all phase indices, $\mathcal{I} \subset \mathcal{L}$ be the subset of phases whose excess must be completely redistributed, and $\bar{\mathcal{I}} := \mathcal{L} \setminus \mathcal{I}$ be the subset of remaining phases. The correction step is implemented as follows:

4. **Correction to favor some phases.** Let $\varepsilon \ll 1$ be a given tolerance. For each phase $\ell \in \mathcal{I}$ and as long as $B_\ell^{n+1} > \varepsilon$,

- (a) Let $C_\ell := \{C \in C_{h_{cells}} : 0 < \varphi_{\ell, h_{cells}}^{n+1}(C) < 1\}$ be the set of cells that already contain and are not full of phase ℓ .

- (b) Let $O_\ell := \sum_{C \in \mathcal{C}_\ell} \varphi_{\ell, h_{\text{cells}}}^{n+1}(C)$ denote the total space occupied by phase ℓ .
- (c) Let $A_\ell := \sum_{C \in \mathcal{C}_\ell} \sum_{i \in \tilde{I}} \varphi_{i, h_{\text{cells}}}^{n+1}(C)$ denote the total space available for phase ℓ .
- (d) If $A_\ell < \varepsilon$, **break**.
- (e) For each cell $C \in \mathcal{C}_\ell$,
- (i) Let $a_\ell := \sum_{i \in \tilde{I}} \varphi_{i, h_{\text{cells}}}^{n+1}(C)$ be the available space for phase ℓ in C .
 - (ii) Let $r_\ell := \min(a_\ell, B_\ell^{n+1} \cdot \varphi_{\ell, h_{\text{cells}}}^{n+1}(C)/O_\ell)$ be the amount of phase ℓ that will be redistributed in cell C .
 - (iii) For each phase $i \in \tilde{I}$,
 - (a) Let $e_i := r_\ell \cdot \varphi_{i, h_{\text{cells}}}^{n+1}(C)/a_\ell$ be the amount of phase i that will be extracted from cell C in order to make room for phase ℓ .
 - (b) Update $\varphi_{i, h_{\text{cells}}}^{n+1}(C) \leftarrow \varphi_{i, h_{\text{cells}}}^{n+1}(C) - e_i$
 - (c) Update $B_i^{n+1} \leftarrow B_i^{n+1} + e_i$
 - (iv) Update $\varphi_{\ell, h_{\text{cells}}}^{n+1}(C) \leftarrow \varphi_{\ell, h_{\text{cells}}}^{n+1}(C) + r_\ell$
 - (v) Update $B_\ell^{n+1} \leftarrow B_\ell^{n+1} - r_\ell$

Finally, if there exists $i \in \tilde{I}$ such that $C_i \neq \emptyset$, apply a second time the decompression algorithm from [14] (Step 1.-3. above).

References

- [1] M. Azarmanesh, M. Farhadi, P. Azizian, Double emulsion formation through hierarchical flow-focusing microchannel, *Phys. Fluids* 28 (2016) 032005, <https://doi.org/10.1063/1.4944058>.
- [2] E. Hughes, A.A. Maan, S. Acquistapace, A. Burbidge, M.L. Johns, D.Z. Gunes, P. Clausen, A. Syrbe, J. Hugo, K. Schroen, V. Miralles, T. Atkins, R. Gray, P. Homewood, K. Zick, Microfluidic preparation and self diffusion PFG-NMR analysis of monodisperse water-in-oil-in-water double emulsions, *J. Colloid Interface Sci.* 389 (1) (2013) 147–156, <https://doi.org/10.1016/j.jcis.2012.07.073>.
- [3] H. Liu, Y. Zhang, Lattice Boltzmann simulation of droplet generation in a microfluidic cross-junction, *Commun. Comput. Phys.* 9 (5) (2011) 1235–1256, <https://doi.org/10.4208/cicp.231009.101110s>.
- [4] G. Vladislavjević, R. Nuumani, S.A. Nabavi, Microfluidic production of multiple emulsions, *Micromachines* 8 (2017) 75, <https://doi.org/10.3390/mi8030075>.
- [5] S. Dong, Wall-bounded multiphase flows of n immiscible incompressible fluids: consistency and contact-angle boundary condition, *J. Comput. Phys.* 338 (2017) 21–67, <https://doi.org/10.1016/j.jcp.2017.02.048>.
- [6] Z. Huang, G. Lin, A.M. Ardekani, Implementing contact angle boundary conditions for second-order phase-field models of wall-bounded multiphase flows, *J. Comput. Phys.* 471 (2022) 111619, <https://doi.org/10.1016/j.jcp.2022.111619>.
- [7] S. Castonguay, T. Gervais, A simple static contact angle-based mesh-dependency correction for 3d capillary flow simulations, *Comput. Fluids* 228 (2021) 105060, <https://doi.org/10.1016/j.compfluid.2021.105060>.
- [8] S.M. Mousavi, F. Sotoudeh, B.J. Lee, M.-R. Paydari, N. Karimi, Effect of hybrid wall contact angles on slug flow behavior in a t-junction microchannel: a numerical study, *Colloids Surf. A, Physicochem. Eng. Asp.* 650 (2022) 129677, <https://doi.org/10.1016/j.colsurfa.2022.129677>.
- [9] J. Yin, S. Kuhn, Numerical simulation of droplet formation in a microfluidic t-junction using a dynamic contact angle model, *Chem. Eng. Sci.* 261 (2022) 117874, <https://doi.org/10.1016/j.ces.2022.117874>.
- [10] A. Bonito, A. Caboussat, M. Picasso, J. Rappaz, A numerical method for fluid flows with complex free surfaces, in: R. Glowinski, P. Neittaanmäki (Eds.), *Partial Differential Equations, in: Computational Methods in Applied Sciences, vol. 16*, Springer, Netherlands, 2008, pp. 187–208.
- [11] A. Caboussat, P. Clausen, J. Rappaz, Numerical simulation of two-phase flow with interface tracking by adaptive Eulerian grid subdivision, *Math. Comput. Model.* 55 (2012) 490–504, <https://doi.org/10.1016/j.mcm.2011.08.027>.
- [12] A. Caboussat, M. Picasso, J. Rappaz, Numerical simulation of free surface incompressible liquid flows surrounded by compressible gas, *J. Comput. Phys.* 203 (2) (2005) 626–649, <https://doi.org/10.1016/j.jcp.2004.09.009>.
- [13] A. Caboussat, A numerical method for the simulation of free surface flows with surface tension, *Comput. Fluids* 35 (10) (2006) 1205–1216, <https://doi.org/10.1016/j.compfluid.2005.08.004>.
- [14] N. James, S. Boyaval, A. Caboussat, M. Picasso, Numerical simulation of 3D free surface flows, with multiple incompressible immiscible phases. Applications to impulse waves, *Int. J. Numer. Methods Fluids* 76 (12) (2014) 1004–1024, <https://doi.org/10.1002/flid.3967>.
- [15] D. Vega, J. Abache, D. Coll, A fast and memory saving Marching Cubes 33 implementation with the correct interior test, *J. Comput. Graph. Tech.* 8 (3) (2019) 1–18, <http://jcggt.org/published/0008/03/01/>.
- [16] A. Bonito, A. Caboussat, M. Picasso, *Operator Splitting Algorithms for Free Surface Flows: Application to Extrusion Processes*, Springer International Publishing, Cham, 2016, pp. 677–729.
- [17] G.I. Marchuk, Splitting and alternating direction methods, in: *Handbook of Numerical Analysis, Vol. I*, North-Holland, Amsterdam, 1990, pp. 197–462.
- [18] W.F. Noh, P. Woodward, SLIC (simple line interface calculation), in: *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics*, vol. 59, 1976, pp. 330–340.
- [19] V. Maronnier, M. Picasso, J. Rappaz, Numerical simulation of three dimensional free surface flows, *Int. J. Numer. Methods Fluids* 42 (7) (2003) 697–716, <https://doi.org/10.1002/flid.532>.
- [20] V. Maronnier, M. Picasso, J. Rappaz, Numerical simulation of free surface flows, *J. Comput. Phys.* 155 (1999) 439–455, <https://doi.org/10.1006/jcph.1999.6346>.
- [21] J.-F. Gerbeau, T. Lelièvre, C. Le Bris, Simulations of MHD flows with moving interfaces, *J. Comput. Phys.* 184 (1) (2003) 163–191, [https://doi.org/10.1016/S0021-9991\(02\)00025-6](https://doi.org/10.1016/S0021-9991(02)00025-6).
- [22] J.-F. Gerbeau, T. Lelièvre, Generalized Navier boundary condition and geometric conservation law for surface tension, *Comput. Methods Appl. Mech. Eng.* 198 (2009) 644–656, <https://doi.org/10.1016/j.cma.2008.09.011>.
- [23] D. Arnold, F. Brezzi, M. Fortin, A stable finite element for the Stokes equations, *Calcolo* 21 (4) (1984) 337–344, <https://doi.org/10.1007/BF02576171>.

- [24] N. Aspert, D. Santa-Cruz, T. Ebrahimi, MESH: Measuring Errors Between Surfaces Using the Hausdorff Distance, Proceedings. IEEE International Conference on Multimedia and Expo, vol. 1, IEEE, 2002, pp. 705–708.
- [25] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 4th edition, The MIT Press, Chichester, 2022, <http://mitpress.mit.edu/9780262046305/introduction-to-algorithms/>.
- [26] R. Franke, Scattered data interpolation: test of some methods, Math. Comput. 38 (1982) 181–200, <https://doi.org/10.2307/2007474>.
- [27] M. Shashkov, B. Wendroff, The repair paradigm and application to conservation laws, J. Comput. Phys. 198 (1) (2004) 265–277, <https://doi.org/10.1016/j.jcp.2004.01.014>.
- [28] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2) (2007) 2096–2132, <https://doi.org/10.1016/j.jcp.2007.06.033>.
- [29] A. Caboussat, M. Francois, R. Glowinski, D. Kothe, J. Sicilian, A numerical method for interface reconstruction of triple points within a volume tracking algorithm, Math. Comput. Model. 48 (11) (2008) 1957–1971, <https://doi.org/10.1016/j.mcm.2008.05.009>.
- [30] S.P. Schofield, R.V. Garimella, M.M. Francois, R. Loubère, A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations, J. Comput. Phys. 228 (3) (2009) 731–745, <https://doi.org/10.1016/j.jcp.2008.09.023>.
- [31] S.P. Schofield, R.V. Garimella, M.M. Francois, R. Loubère, Material order-independent interface reconstruction using power diagrams, Int. J. Numer. Methods Fluids 56 (6) (2008) 643–659, <https://doi.org/10.1002/flid.1544>.
- [32] B.Y. Choi, M. Bussmann, A piecewise linear approach to volume tracking a triple point, Int. J. Numer. Methods Fluids 53 (6) (2007) 1005–1018, <https://doi.org/10.1002/flid.1317>.
- [33] S.P. Schofield, M.A. Christon, V. Dyadechko, R.V. Garimella, R.B. Lowrie, B.K. Swartz, Multi-material incompressible flow simulation using the moment-of-fluid method, Int. J. Numer. Methods Fluids 63 (8) (2010) 931–952.
- [34] J. Étienne, E. Hopfinger, P. Saramito, Numerical simulations of high density ratio lock-exchange flows, Phys. Fluids 17 (2005) 036601, <https://doi.org/10.1063/1.1849800>.
- [35] H. Gröbelbauer, T. Fanneløp, R. Britter, The propagation of intrusion fronts of high density ratios, J. Fluid Mech. 250 (1993) 669–687, <https://doi.org/10.1017/S0022112093001612>.