Journal of Computational Physics 348 (2017) 271–297

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Level-set simulations of soluble surfactant driven flows

Charles Cleret de Langavant [a,*], Arthur Guittet [a], Maxime Theillard [c], Fernando Temprano-Coleto [a], Frédéric Gibou [a,b]

[a] Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070, United States
[b] Department of Computer Science, University of California, Santa Barbara, CA 93106-5110, United States
[c] Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA 92093, United States

ARTICLE INFO

Article history:
Received 20 October 2015
Received in revised form 9 May 2017
Accepted 2 July 2017
Available online 8 July 2017

Keywords:
Navier–Stokes
Incompressible
Soluble surfactants
Surfactant driven flows
Marangoni forces
Quad/Octrees
Adaptive mesh refinement
Stable projection method

ABSTRACT

We present an approach to simulate the diffusion, advection and adsorption–desorption of a material quantity defined on an interface in two and three spatial dimensions. We use a level-set approach to capture the interface motion and a Quad/Octree data structure to efficiently solve the equations describing the underlying physics. Coupling with a Navier–Stokes solver enables the study of the effect of soluble surfactants that locally modify the parameters of surface tension on different types of flows. The method is tested on several benchmarks and applied to three typical examples of flows in the presence of surfactant: a bubble in a shear flow, the well-known phenomenon of tears of wine, and the Landau–Levich coating problem.

1. Introduction

Many problems in fluid dynamics exhibit a different behavior in the presence of surfactants at the interface between two phases or at a free surface. Surfactants are amphiphilic molecules, i.e. they possess both hydrophilic and hydrophobic groups, giving them properties that can affect the flow through two different mechanisms that have the same origin. First, surfactants at an interface lower its surface tension, thus altering the pressure jump at the interface. Second, when the surfactant concentration is not constant along the interface, it generates the so-called Marangoni forces, which are tangential stresses that can critically perturb the flow near the interface.

The computational modeling of the diffusion, the advection, the adsorption and the desorption processes of a material quantity on a moving interface, as well as its possible effects on a fluid flow, is a challenging task. Several computational studies have been undertaken since the early 1990s, involving more or less sophisticated numerical methods and more or less comprehensive physical models depending, for example, on whether or not the surfactant is considered soluble, which implies the presence of adsorption–desorption coupling between the bulk and the interface; whether the effect of surfactants on the surface tension are modeled by a linear or a nonlinear equation of state; or whether the full Navier–Stokes equations or the Stokes equations are considered.

In 1989, Stone and Leal [34] developed a boundary integral method for studying the effects of insoluble surfactant on drop deformation and breakup. One year later, He, Dagan and Maldarelli [14] investigated, using uniform retardation

[*] Corresponding author.
E-mail address: charlesdelangavant@hotmail.com (C. Cleret de Langavant).

http://dx.doi.org/10.1016/j.jcp.2017.07.003
0021-9991/© 2017 Elsevier Inc. All rights reserved.

perturbation schemes, the retarding effect of the Marangoni forces due to soluble surfactant on the motion of a fluid sphere in creeping translation inside a tube. In 1992, Borhan and Mao [3] developed a boundary element method to study the effects of surfactant on drop deformation. The same year, Milliken, Stone and Leal [22] developed a boundary integral method for studying the evolution of viscous drops under the hypothesis of Stokes flow with arbitrary viscosity ratios in the presence of surfactant and using a nonlinear equation of state for the interfacial tension. In 1994, Leveque and Li [17] designed an immersed interface method for elliptic equations with discontinuous coefficients and singular source terms, which could be applied to the study of the effects of Marangoni forces on fluid flows. In 1998, Eggleton and Stebe [11] investigated the effects of a soluble surfactant on the deformation of a droplet using a boundary integral method for Stokes flows. In 2003, Drumright-Clarke and Renardy [10] used the full Navier–Stokes equations to study the effects of insoluble surfactant on the motion and breakup of viscous drops in a shear flow, using a volume-of-fluid method. In 2004, James and Lowengrub [15] developed a finite-elements solver using a surfactant-conserving volume-of-fluid interface-capturing method to investigate the effects of insoluble surfactant on various interfacial flows. In 2006, Xu, Li, Lowengrub and Zhao [41] developed a level-set method on uniform grids for solving the surfactant equation coupled to an external Stokes flow solver, and applied it to investigate the effects of insoluble surfactant on single drops, on drop–drop interactions and on the interactions of many drops. It was the first time the level-set method was used to simulate fluid interfaces with surfactant. In 2007, Muradoglu and Tryggvason [27] developed a front tracking method to investigate the axisymmetric motion and deformation of a viscous drop with soluble surfactant moving in a circular tube. This approach introduced an adsorption layer in order to model the adsorption/desorption terms. In 2009, Teigen et al. [38] presented a diffuse-interface approach for modeling the advection, the diffusion and the adsorption–desorption of material quantities on a deformable interface, using finite-difference schemes on a block-structured adaptive grid.

In this paper, we present a level-set method on adaptive Quad/Octree grids to study the effects of both soluble and insoluble surfactant on free surface flows in two and three spatial dimensions. We use the full Navier–Stokes equations and a nonlinear equation of state for the surface tension. After describing the governing equations in section 2 we will describe the numerical approach in section 3. Section 4 is dedicated to the verification in both in two and three spatial dimensions, mainly using variants of the test examples proposed in [27]. In Section 5, we present three examples of application of our method: the deformation of a bubble in a shear flow in the presence of soluble surfactant, the well-known phenomenon of tears of wine, and the effect of the presence of surfactant on the flow near the meniscus in the Landau–Levich coating problem. Section 6 gathers a few concluding remarks on the topic of the simulation of surfactant-driven flows using our method.

2. Governing equations

The governing equations are those describing the evolution of the surfactant concentration, both at the interface and in the bulk as well as their coupling. In addition, the surfactant concentration and the fluid's velocity field are coupled through the surfactant's advection and the surface tension effects.

2.1. Evolution of a material quantity defined on an interface

A scalar quantity Γ, here the surface surfactant concentration, diffusing along an interface Σ and being advected by an external velocity field $\mathbf{u}$ is described by the following time-dependent equation [33]:

$$\Gamma_t + \nabla_s \cdot (\Gamma \mathbf{u}) = D_s \Delta_s \Gamma, \tag{1}$$

where D_s is the diffusion constant of Γ along Σ. Using the definition of the surface divergence $\nabla_s \cdot \mathbf{v} = \nabla \cdot \mathbf{v} - \mathbf{n} \cdot [(\nabla\ \mathbf{v}) \cdot \mathbf{n}]$, where $\mathbf{n}$ is the outward normal to Σ, and that of the surface Laplacian $\Delta_s f = \Delta f - \frac{\partial^2 f}{\partial n^2} - \kappa \frac{\partial f}{\partial n}$, where $\kappa = \nabla \cdot \mathbf{n} = \nabla_s \cdot \mathbf{n}$ is the mean curvature of the interface, Equation (1) yields:

$$\Gamma_t + \mathbf{u} \cdot \nabla_s \Gamma + \Gamma \nabla \cdot \mathbf{u} - \Gamma \mathbf{n} \cdot [(\nabla \mathbf{u}) \cdot \mathbf{n}] = D_s \left(\Delta \Gamma - \frac{\partial^2 \Gamma}{\partial n^2} - \kappa \frac{\partial \Gamma}{\partial n} \right). \tag{2}$$

Splitting the velocity into its normal u_n and its tangential $\mathbf{u}_s$ components and using the fact that $\mathbf{n} \cdot \nabla_s f = 0$ for any scalar quantity f, Equation (1) can be further written as:

$$\Gamma_t + \mathbf{u}_s \cdot \nabla_s \Gamma + \Gamma \nabla \cdot \mathbf{u}_s - \Gamma \mathbf{n} \cdot [(\nabla \mathbf{u}_s) \cdot \mathbf{n}] + \Gamma \kappa u_n = D_s \left(\Delta \Gamma - \frac{\partial^2 \Gamma}{\partial n^2} - \kappa \frac{\partial \Gamma}{\partial n} \right).$$

Such relations and their variants, along with their rigorous demonstration, can also be found in [6] and [9].

2.2. Coupling with a quantity defined in the bulk

When considering the problem of how the surfactant concentration evolves in time on an interface immersed in a fluid bulk, it often occurs that the surfactants, despite its tendency to accumulate preferentially on the interface, can also dissolve in the bulk Ω^-. The surfactant concentration at the interface $\Gamma(\mathbf{x},t)$ is then coupled with a surfactant concentration in the bulk, $c(\mathbf{x},t)$, with a diffusion constant D. The exchange of material between the interface and the bulk is usually modeled by an adsorption–desorption source/sink term S that depends on the kinetics considered (see [7]). For the case of Langmuir kinetics:

$$S\,(c,\Gamma) = k_a c\,(\Gamma_\infty - \Gamma) - k_b \Gamma, \tag{3}$$

where k_a and k_b are namely the adsorption and desorption coefficients, and Γ_∞ represents the saturation level of surfactant on the interface. The system of governing equations for the evolution of the surfactant concentration in the entire domain can thereafter be written as:

$$\begin{cases} \Gamma_t + \mathbf{u}\cdot\nabla_s\Gamma + \Gamma\nabla\cdot\mathbf{u} - \Gamma\mathbf{n}\cdot[(\nabla\mathbf{u})\cdot\mathbf{n}] = D_s\left(\Delta\Gamma - \partial^2\Gamma/\partial n^2 - \kappa\,\partial\Gamma/\partial n\right) + S \quad \forall x \in \Sigma, \\ c_t + \mathbf{u}\cdot\nabla c = D\Delta c \quad \forall x \in \Omega, \\ D\,\nabla c\cdot\mathbf{n} = -S \quad \forall x \in \Sigma. \end{cases} \tag{4}$$

2.3. Coupling with the Navier–Stokes equations

The evolution of the concentration of surfactant governed by (4) and the fluid velocity, described by the Navier–Stokes equations, have reciprocal influences: On the one hand, the incompressible Navier–Stokes equations give the fluid's velocity $\mathbf{u}$ and pressure p, under which the interface is stretched and distorted, and under which the quantities c and Γ are advected. On the other hand the presence of surfactant influences the fluid velocity by modifying the surface tension along the interface, and therefore the boundary conditions for the Navier–Stokes solver.

A common model for describing the effect on the surface tension γ by the interface concentration Γ is the Szyszkowski equation:

$$\gamma(\Gamma) = \gamma_0 + n\,R\,T\,\Gamma_\infty \ln\left(1 - \frac{\Gamma}{\Gamma_\infty}\right), \tag{5}$$

where R is the ideal gas constant, T is the absolute temperature, γ_0 is the surface tension of a clean interface, and n is a constant that depends on the type of surfactant and on the presence of excess electrolytes.

It is important to note that both the equation of state and the adsorption–desorption term must be congruent. For instance, the equation of state (5) and the kinetic term (3) are both derived from the Langmuir isotherm, and are therefore congruent. Other choices for this pair are also possible (see [7] for a detailed review of different equations of state and kinetics). For instance, in the case of Frumkin kinetics:

$$\begin{cases} \gamma(\Gamma) = \gamma_0 + n\,R\,T\,\Gamma_\infty \left[\ln\left(1 - \frac{\Gamma}{\Gamma_\infty}\right) + \frac{A}{2}\left(\frac{\Gamma}{\Gamma_\infty}\right)^2\right], \\ S\,(c,\Gamma) = k_a c\,(\Gamma_\infty - \Gamma) - k_b\Gamma\exp\left(-A\,(\Gamma/\Gamma_\infty)\right), \end{cases} \tag{6}$$

where A is an interaction parameter that depends on the surfactant considered. Other alternatives (like, for example, the 'generalized Frumkin' pair reported in [19] for some nonionic polyoxyethylene surfactants and alcohols) can be found in specialized literature. While in our implementation it is possible to consider any pair of kinetic term and equation of state, in what follows we will assume Langmuir kinetics (Equations (3) and (5)), unless otherwise indicated.

The presence of a varying surface tension implies the following boundary condition on the jump of the stress tensor $\underline{\underline{\sigma}} = -p\mathbb{1} + \mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)$ across the interface:

$$\left[\underline{\underline{\sigma}}\right]\cdot\mathbf{n} = \gamma\kappa\mathbf{n} - \nabla_s\gamma.$$

The last term in the above boundary condition, whose consequences are only made visible when the surface tension is non-constant, is at the origin of the Marangoni force [41]. The complete system of equations describing a surfactant-driven incompressible flow, with a soluble surfactant in the bulk, as well as the coupling between the interface and the bulk, is given by the following system of partial differential equations:

$$\begin{cases} \rho\,(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}) = \nabla\cdot\underline{\underline{\sigma}} + \mathbf{f} \quad \forall x \in \Omega^-, \\ \nabla\cdot\mathbf{u} = 0 \quad \forall x \in \Omega^-, \\ \Gamma_t + \mathbf{u}\cdot\nabla_s\Gamma + \Gamma\nabla\cdot\mathbf{u} - \Gamma\mathbf{n}\cdot[(\nabla\mathbf{u})\cdot\mathbf{n}] = D_s\left(\Delta\Gamma - \partial^2\Gamma/\partial n^2 - \kappa\,\partial\Gamma/\partial n\right) + S \quad \forall x \in \Sigma, \\ c_t + \mathbf{u}\cdot\nabla c = D\Delta c \quad \forall x \in \Omega^-, \end{cases} \tag{7}$$

completed by the following interface conditions on Σ:

$$\begin{cases} \quad\quad S &= k_a c\,(\Gamma_\infty - \Gamma) - k_b \Gamma, \\ \quad\quad \gamma &= \gamma_0 + n\; R\; T\; \Gamma_\infty \ln\left(1 - \dfrac{\Gamma}{\Gamma_\infty}\right), \\ D\;\nabla c \cdot \mathbf{n} &= -S, \\ \left[\underline{\underline{\sigma}}\right] \cdot \mathbf{n} &= \gamma \kappa \mathbf{n} - \nabla_s \gamma. \end{cases} \tag{8}$$

3. Numerical method

3.1. General framework: the level-set approach on Quad/Octrees

3.1.1. Overview

The level-set method is a general computational method to capture efficiently an interface. It is based on the use of a level-set function ϕ that divides the computational domain Ω into (at most) three parts: $\Omega^+ = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x}) > 0\}$, $\Omega^- = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x}) < 0\}$ and the interface $\Sigma = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x}) = 0\}$. As a convention, the domain of interest for our computation will be Ω^-. The main great advantage of this method is that it naturally captures the changes in the topology of the interface that can occur in many physical situations, such as the pinching of a droplet. The main level-set equation, which evolves the interface according to a velocity field $\mathbf{u}$, is:

$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0. \tag{9}$$

Although the algorithms we propose could be employed on a uniform Cartesian mesh, we employ here a Quadtree data structure in two spatial dimensions and an Octree data structure in three spatial dimensions for increased efficiency. The construction of the Quad/Octree is recursive: starting from the root cell, which represents the whole computational domain, children are added to the different cells: four of them in the case of a Quadtree data structure (eight for an Octree data structure), corresponding to the splitting of the cell in four (eight for Octrees) new smaller cells. The level of a cell is defined as the depth of the leaf characterizing the corresponding cell in the tree. Graded grids limit the difference of level between two adjacent cells in the grid to one, whereas in non-graded grids, this difference can be arbitrary. The advantage of using non-graded grids is that they are more straightforward to build, and for a same maximum level (corresponding to the depth of the tree), the number of necessary nodes is smaller. An illustration of the principle of the application of Quadtrees to the storage of non-graded adaptive grids in two dimensions is given in Fig. 1.

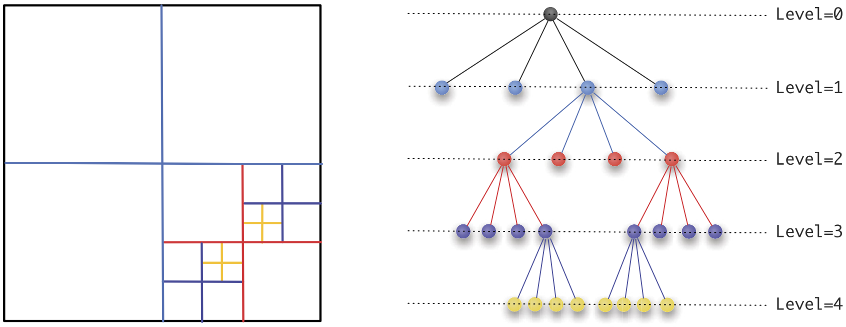


Fig. 1. An example of the partition of a square into a non-graded adaptive grid (left) and the Quadtree used for its storage.

3.1.2. Refinement criteria

Being mainly interested in refining the grid close to where the interface lies, we apply the following simple refinement criteria proposed in [35,25]: Split any cell $\mathcal{C}$ if

$$\min_{v \in \text{vertices}(\mathcal{C})} |\phi(v)| \leq \text{Lip}\,(\phi) \times \text{diag-size}(\mathcal{C}),$$

where diag-size$(\mathcal{C})$ is the length of the diagonal of the current cell $\mathcal{C}$, Lip (ϕ) is the Lipschitz constant of ϕ and v refers to a vertex (i.e. a node) of the current cell.

3.1.3. Reinitialization of the level-set function

Any Lipschitz continuous function can be chosen for the level-set function ϕ; however, the numerical results are known to be more robust, to induce less mass loss and to provide better geometrical information when computing geometrical quantities when ϕ is taken as the so-called signed distance function. An arbitrary level-set function ϕ_0 can be reinitialized

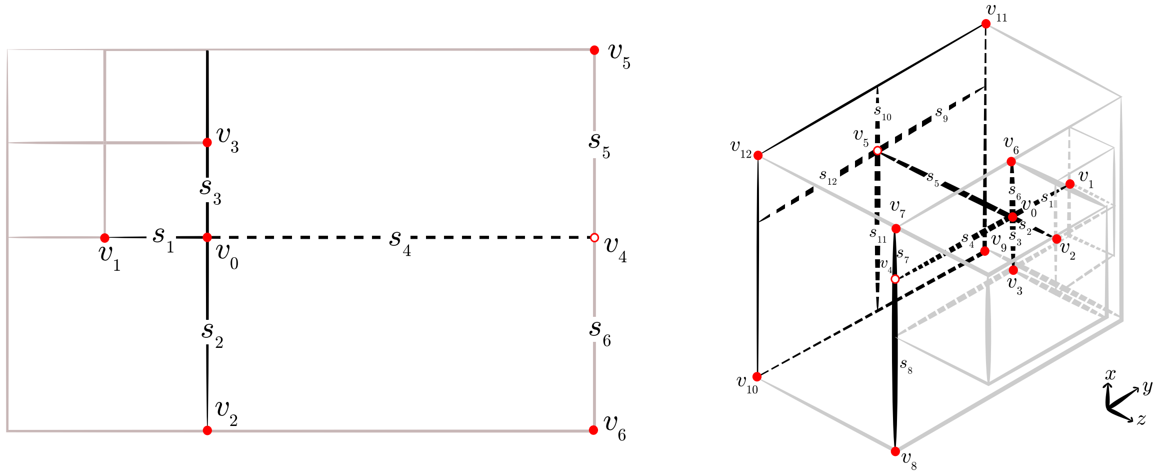


Fig. 2. General configuration of a T-junction centered on the node v_0 in two (left) and three (right) spatial dimensions.

into a signed distance function that describes the same interface Σ by iterating a few steps in fictitious time τ (using a TVD RK-2 scheme) the following reinitialization equation [37]:

$$\phi_\tau + \operatorname{sgn}(\phi_0)\,(|\nabla\phi| - 1) = 0. \tag{10}$$

Here τ is a fictitious time. The outward normal $\mathbf{n}$ to the interface Σ and the mean curvature κ are defined as:

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \quad \text{and} \quad \kappa = \nabla\cdot\mathbf{n}.$$

In this work, we use the algorithms described in [25] to solve Equations (9), (10), and to compute the normal and curvature.

3.1.4. Computation of spatial derivatives

In order to compute such spatial derivatives using standard finite-difference formulas, we are bound to introduce ghost nodes where the mesh is locally non-uniform (the so called T-junctions; see Fig. 2). We use the framework of [25]. Considering a node-sampled function $f : v_i \longrightarrow \mathbb{R}$, its value at the ghost node v_4 in two dimensions is defined by the following third-order interpolation (denoting by f_i the value of $f(v_i)$ and by s_i the distance between grid nodes as given in Fig. 2):

$$f_4^G = \frac{f_5 s_6 + f_6 s_5}{s_5 + s_6} - \frac{s_5 s_6}{s_2 + s_3}\left(\frac{f_2 - f_0}{s_2} + \frac{f_3 - f_0}{s_3}\right),$$

whereas in three spatial dimensions, the definition of the, at most two, ghost nodes are:

$$f_4^G = \frac{f_7 s_8 + f_8 s_7}{s_7 + s_8} - \frac{s_7 s_8}{s_3 + s_6}\left(\frac{f_3 - f_0}{s_3} + \frac{f_6 - f_0}{s_6}\right),$$

and

$$\begin{aligned} f_5^G = & \frac{s_{11}s_{12}f_{11} + s_{11}s_9 f_{12} + s_{10}s_{12}f_9 + s_{10}s_9 f_{10}}{(s_{10}+s_{11})(s_9+s_{12})} \\ & - \frac{s_{10}s_{11}}{s_3+s_6}\left(\frac{f_3 - f_0}{s_3} + \frac{f_6 - f_0}{s_6}\right) - \frac{s_9 s_{12}}{s_1+s_4}\left(\frac{f_1 - f_0}{s_1} + \frac{f_4^G - f_0}{s_4}\right). \end{aligned}$$

From these ghost nodes definitions, second order accurate discretizations of the standard differential operators can be derived following [25].

3.1.5. Extrapolation procedures

The algorithms described in the next sections depend on the ability to extrapolate a scalar quantity f from Ω^- to Ω^+, which is performed by iterating a few steps in fictitious time τ (using a TVD RK-2 scheme) of the following equation:

$$\frac{\partial f}{\partial \tau} + H(\phi)\,\mathbf{n}\cdot\nabla f = 0, \tag{11}$$

where H is the Heaviside function defined for every node v as follows:

$$H(\phi(\mathbf{x})) = \begin{cases} 1 & \text{if } \phi(\mathbf{x}) \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Likewise, extrapolating a scalar quantity f from the interface Σ to the entire domain in the normal direction is performed by solving in fictitious time τ the following equation:

$$\frac{\partial f}{\partial \tau} + \delta(\phi)\ \mathbf{n} \cdot \nabla f = 0, \tag{12}$$

where δ is the indicator function of the interface, defined by:

$$\delta(\phi(\mathbf{x})) = \begin{cases} 0 & \text{if } \phi(\mathbf{x}) = 0, \\ 1 & \text{otherwise.} \end{cases}$$

We refer the interested reader to [26] for the discretization formulas of H and δ, as well as how to discretize Equations (11)–(12).

3.2. Evolution of the surfactant concentration

In this section, we present how we solve the time-dependent evolution of the interfacial concentration of a soluble material quantity, including the advection, the surface diffusion and the adsorption–desorption on the interface. Coupling this solver with a Navier–Stokes solver [13] enables to complete the resolution of the full flow problem described in the previous section.

3.2.1. Solving the equation for the concentration in the bulk

The equation for the surfactant concentration in the bulk, assuming that $\mathbf{u}$ is divergence free, is given by $\big($see (4)$\big)$:

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = D \Delta c \quad \text{in } \Omega^-, \quad \text{with} \quad D\ \nabla c \cdot \mathbf{n} = -S(c, \Gamma) \ \text{ on } \Sigma. \tag{13}$$

The advective part of this equation is solved by a non-conservative semi-Lagrangian method discretized using a backward differentiation formula, while the diffusive part is solved implicitly:

$$\alpha\, \frac{c^{n+1} - c_d^n}{\Delta t_n} + \beta\, \frac{c_d^n - c_d^{n-1}}{\Delta t_{n-1}} = D \Delta c^{n+1},$$

where:

$$\alpha = \frac{2\Delta t_n + \Delta t_{n-1}}{\Delta t_n + \Delta t_{n-1}},$$

$$\beta = -\frac{\Delta t_n}{\Delta t_n + \Delta t_{n-1}},$$

and where c_d^n and c_d^{n-1} are the values of c at time t^n and t^{n-1}, respectively, taken at the departure points $\mathbf{x}_d^n$ and $\mathbf{x}_d^{n-1}$, i.e. following the characteristics back from the current node's location $\mathbf{x}^{n+1}$ under the velocity field $\mathbf{u}$ during Δt_n and $\Delta t_n + \Delta t_{n-1}$. Specifically, these points are constructed as

$$\mathbf{x}^* = \mathbf{x}^{n+1} - \frac{\Delta t_n}{2} \mathbf{u}^{\mathbf{n}}(\mathbf{x}^{\mathbf{n+1}}),$$

$$\mathbf{u}^* = \frac{2\Delta t_{n-1} + \Delta t_n}{2\Delta t_{n-1}} \mathbf{u}^{\mathbf{n}}(\mathbf{x}^*) - \frac{\Delta t_n}{2\Delta t_{n-1}} \mathbf{u}^{\mathbf{n-1}}(\mathbf{x}^*),$$

$$\mathbf{x}_{\mathbf{d}}^{\mathbf{n}} = \mathbf{x}^{\mathbf{n+1}} - \Delta t_n \mathbf{u}^*,$$

and

$$\mathbf{x}^* = \mathbf{x}^{\mathbf{n+1}} - \Delta t_n \mathbf{u}^{\mathbf{n}}(\mathbf{x}^{\mathbf{n+1}}),$$

$$\mathbf{x}_{\mathbf{d}}^{\mathbf{n-1}} = \mathbf{x}^{\mathbf{n+1}} - (\Delta t_n + \Delta t_{n-1})\, \mathbf{u}^{\mathbf{n}}(\mathbf{x}^*).$$

Since $\mathbf{x}_d^n$, $\mathbf{x}_d^{n-1}$, $\mathbf{x}^*$ are not in general located on a grid point, we use quadratic interpolation procedures to interpolate the velocity fields at these points [25].

Note that for the case of a constant timestep $\Delta t_n = \Delta t_{n-1} = \Delta t$, the above discretization reduces to:

$$\frac{1}{\Delta t}\left(\frac{3}{2}c^{n+1} - 2c_d^n + \frac{1}{2}c_d^{n-1}\right) = D\Delta c^{n+1}.$$

The equation for solving c^{n+1} implicitly is therefore:

$$\frac{\alpha}{\Delta t_n}c^{n+1} - D\Delta c^{n+1} = \left(\frac{\alpha}{\Delta t_n} - \frac{\beta}{\Delta t_{n-1}}\right)c_d^n + \frac{\beta}{\Delta t_{n-1}}c_d^{n-1}, \tag{14}$$

supplemented with the boundary condition:

$$D\,\nabla c^{n+1}\cdot \mathbf{n} = -S(c^n,\Gamma^n) \quad \forall x \in \Sigma. \tag{15}$$

Far away from the interface Σ, Equation (14) is discretized using the finite differences framework presented in section 3.1.4. Close to the interface, where the non-homogeneous Neumann boundary condition (15) needs to be enforced, a finite volume approach is employed as in [32,29,31]. In two spatial dimensions, considering a cell $\mathcal{C}_{i,j}$ in Ω^- centered at the node (i,j) and cut by the interface Σ, and using the divergence theorem on the Laplace operator of Equation (14), we have:

$$\frac{\alpha}{\Delta t_n}\int\limits_{\mathcal{C}_{i,j}\cap\Omega^-} c^{n+1}\,d\mathbf{x} - D\int\limits_{\partial\mathcal{C}_{i,j}\cap\Omega^-}\nabla c^{n+1}\cdot\mathbf{n}\,dl = \int\limits_{\mathcal{C}_{i,j}\cap\Omega^-} f_{i,j}\,d\mathbf{x} - \int\limits_{\mathcal{C}_{i,j}\cap\Sigma} S(c^n,\Gamma^n)\,dl, \tag{16}$$

where $f_{i,j} = \left[\left(\frac{\alpha}{\Delta t_n} - \frac{\beta}{\Delta t_{n-1}}\right)c_d^n + \frac{\beta}{\Delta t_{n-1}}c_d^{n-1}\right]_{i,j}$ and where we have used the boundary condition (15). We approximate the integrals using the methods detailed in [24]. These approximations yield the filling of the row of the linear system corresponding to the node (i,j). The linear system is symmetric positive definite.

3.2.2. Solving the equation for the interface concentration

Assuming that the velocity field is divergence free, the interface concentration satisfies the following equation:

$$\frac{\partial\Gamma}{\partial t} + \mathbf{u}\cdot\nabla\Gamma - \Gamma\mathbf{n}\cdot[(\nabla\,\mathbf{u})\cdot\mathbf{n}] = D_s\left(\Delta\Gamma - \frac{\partial^2\Gamma}{\partial n^2} - \kappa\frac{\partial\Gamma}{\partial n}\right) + S(c,\Gamma), \tag{17}$$

which is discretized in time using a modified non-conservative semi-Lagrangian combined with a semi-implicit backward-differentiation-formula scheme [40], treating diffusion implicitly as well:

$$\begin{aligned}\alpha\,\frac{\Gamma^{n+1}-\Gamma_d^n}{\Delta t_n} + \beta\,\frac{\Gamma_d^n - \Gamma_d^{n-1}}{\Delta t_{n-1}} = {} & D_s\Delta\Gamma^{n+1} + \eta\left(\Gamma_d^n\Big\{\mathbf{n}\cdot[(\nabla\,\mathbf{u})\cdot\mathbf{n}]\Big\}_d^n + S(c_d^n,\Gamma_d^n) + A_d^n\right)\\ & + \zeta\left(\Gamma_d^{n-1}\Big\{\mathbf{n}\cdot[(\nabla\,\mathbf{u})\cdot\mathbf{n}]\Big\}_d^{n-1} + S(c_d^{n-1},\Gamma_d^{n-1}) + A_d^{n-1}\right),\end{aligned} \tag{18}$$

where $A = D_s\left(-\frac{\partial^2\Gamma}{\partial n^2} - \kappa\frac{\partial\Gamma}{\partial n}\right)$ is the specific interfacial term of the surface Laplace operator, and where all the quantities evaluated explicitly in the right-hand side are computed at the departure points (subindex d). Also:

$$\eta = \frac{\Delta t_n + \Delta t_{n-1}}{\Delta t_{n-1}},$$

$$\zeta = -\frac{\Delta t_n}{\Delta t_{n-1}}.$$

Note that for the case of a constant timestep $\Delta t_n = \Delta t_{n-1} = \Delta t$, (18) reduces to the implicit–explicit extrapolated Gear scheme [1]:

$$\begin{aligned}\frac{1}{\Delta t}\left(\frac{3}{2}\Gamma^{n+1} - 2\Gamma_d^n + \frac{1}{2}\Gamma_d^{n-1}\right) = {} & D_s\Delta\Gamma^{n+1} + 2\left(\Gamma_d^n\Big\{\mathbf{n}\cdot[(\nabla\,\mathbf{u})\cdot\mathbf{n}]\Big\}_d^n + S(c_d^n,\Gamma_d^n) + A_d^n\right)\\ & - \left(\Gamma_d^{n-1}\Big\{\mathbf{n}\cdot[(\nabla\,\mathbf{u})\cdot\mathbf{n}]\Big\}_d^{n-1} + S(c_d^{n-1},\Gamma_d^{n-1}) + A_d^{n-1}\right).\end{aligned}$$

Following the idea of [10], all the quantities of interest are constantly extrapolated to a tube surrounding the interface Σ (see Fig. 3, left). The new surfactant concentration Γ^{n+1} is then obtained by solving the above equation in this tube with

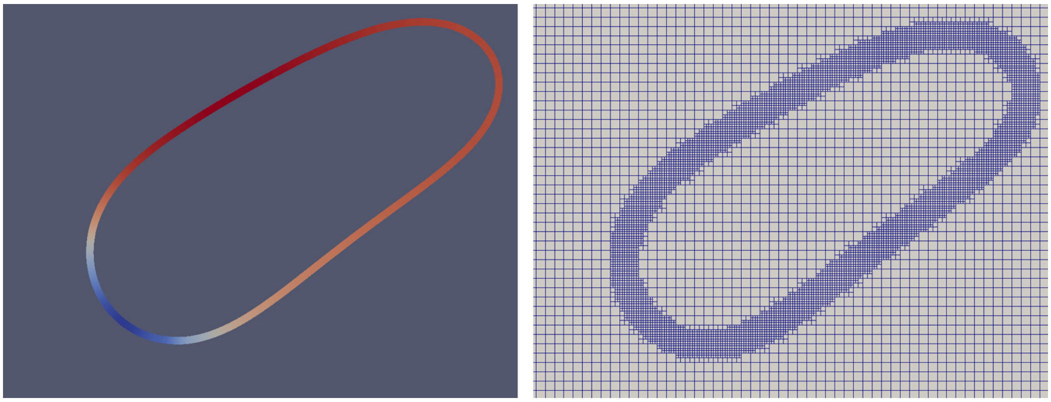


Fig. 3. Left: visualization of the geometry of the tube used to solve the equation governing the evolution of the concentration Γ of surfactant on the interface (18), in the case of a bubble in a shear flow. The tube is colored according to interface concentration (with increasing concentration from blue to red). Right: grid used to perform this simulation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

homogeneous Neumann boundary conditions. Constant extrapolation is performed following the procedures presented in [2]. A level-set function χ describing a tube of width 2δ around the interface can easily be obtained from the level-set function ϕ describing the domain, once it has been reinitialized as described in section 3.1.3, by taking simply: $\chi(\mathbf{x}) = |\phi(\mathbf{x})| - \delta$. One of the advantages of this method is that the same grid is used to solve the concentration in the bulk and the concentration on the interface (see Fig. 3, right).

Note that the specific interfacial terms of the surface Laplace operator $\left(-\dfrac{\partial^2\Gamma}{\partial n^2} - \kappa\dfrac{\partial\Gamma}{\partial n}\right)$ are here taken into account in an explicit way. However, by extending Γ from the interface to the whole domain before solving the equation, we have ensured that Γ is almost constant in the normal direction and therefore those terms are very small (here, we take them to be zero).

3.3. Coupling with a Navier–Stokes solver

A Navier–Stokes solver, described below, is used at each time step, alternating with the resolution of the governing equations for the interface concentration and the bulk concentration of surfactant. The velocity field given by the Navier–Stokes equations is used to advect c and Γ, whereas the value of Γ obtained using the procedure described in the section 3.2.2 is used to compute the value of the surface tension γ and its gradient $\nabla\gamma$, which intervene at the next iteration in the boundary conditions for the Navier–Stokes solver.

3.3.1. The projection method

The Navier–Stokes equations

$$\begin{cases} \rho\left(\mathbf{u}_t + \mathbf{u}\cdot\nabla\,\mathbf{u}\right) & = -\nabla p + \mu\Delta\mathbf{u} + \mathbf{f} \\ \qquad\qquad \nabla\cdot\mathbf{u} & = 0 \end{cases}$$

are solved using a projection method [8]. We first solve for an auxiliary velocity $\mathbf{u}^*$ using again a semi-Lagrangian/backward-differentiation-formula discretization of the momentum equation from which the pressure term has been removed:

$$\rho\left(\alpha\,\frac{\mathbf{u}^* - \mathbf{u}_d^n}{\Delta t_n} + \beta\,\frac{\mathbf{u}_d^n - \mathbf{u}_d^{n-1}}{\Delta t_{n-1}}\right) = \mu\Delta\mathbf{u}^* + \mathbf{f}. \tag{19}$$

Here the velocity components are stored at the nodes on the corners of the cells, as in [23], from which the semi-Lagrangian updates are computed. To treat the viscosity implicitly, a linear system for $\mathbf{u}^*$ is built using the finite differences techniques described in section 3.1.4 and solved using a BiCGSTAB solver and the multigrid method presented in [39] as a preconditioner. The velocity components (u, v) are then interpolated to the cells faces as in the standard MAC grid arrangement. The projection is then performed on that grid, i.e. we write $\mathbf{u}^{n+1} = \mathbf{u}^* - \nabla\Phi$ and solve the following linear system for the Hodge variable Φ:

$$\nabla\cdot\nabla\Phi = \nabla\cdot\mathbf{u}^*. \tag{20}$$

Following the standard MAC grid projection method, the Hodge variable is stored at the center of the grid cells. The gradient operator appearing in the above equation is approximated using the second order discretizations in [20]. To ensure the stability of the projection step, the divergence operator D is constructed following [13], in particular it is related to the discrete gradient G through

$$L_F G = -(L_C D)^T,$$

where L_F and L_C are two diagonal matrices. We refer the interested reader to [13] for more details about the stability. The velocity components are then interpolated back to the nodes with least-squares interpolations in order to proceed to the advection of the different quantities of interest [13].

Also, even though our algorithm does not require the pressure to be computed, we note that it is given by $p = \alpha \frac{\rho}{\Delta t_n}\Phi - \mu \Delta \Phi = \alpha \frac{\rho}{\Delta t_n}\Phi - \mu \nabla \cdot \mathbf{u}^*$.

3.3.2. Boundary conditions

Free surface boundary conditions are implemented in order to enforce the exact boundary conditions in velocity and pressure that are imposed by the presence of surfactant:

$$\left[\underline{\underline{\sigma}}\right] \cdot \mathbf{n} = \gamma \kappa \mathbf{n} - \nabla_s \gamma\ , \tag{21}$$

where the surface tension γ is computed as a function of the interface concentration Γ. Even though the coupling with the Navier–Stokes solver admits any regime, in our computational examples we will limit ourselves to Stokes flows for simplicity of the boundary conditions. Indeed, in this case the treatment is considerably more straightforward, since there is a decoupling between the pressure and the velocity (we refer the interested reader to the original derivation in [18]). The jump boundary conditions read as follows [41]:

$$\begin{cases} [p] = -\gamma\kappa, \\ [\nabla p \cdot \mathbf{n}] = \Delta_s \gamma, \\ [\mathbf{u}] = 0, \\ [\mu\, \nabla \mathbf{u} \cdot \mathbf{n}] = \nabla_s \gamma. \end{cases} \tag{22}$$

Furthermore, in our computational examples we will assume that one of the phases is dynamically negligible (i.e. the shear stress in that phase is much smaller than the shear stress in the other phase, and its pressure is constant and taken as zero). This allows for the utilization of a monophase Navier–Stokes solver, and the above four boundary conditions reduce to two (imposing the four of them with only one phase would be overconstraining the problem). Additionally, the jump boundary conditions reduce to Dirichlet (for the pressure) and Neumann (for the velocity) boundary conditions:

$$\begin{cases} \quad p &= \gamma\kappa, \\ \mu \nabla \mathbf{u} \cdot \mathbf{n} &= -\nabla_s \gamma. \end{cases} \tag{23}$$

The boundary conditions for the velocities are enforced on $\mathbf{u}^*$ during the first step of the projection algorithm (19), taking into account the fact that $\mathbf{u}^*$ is different from $\mathbf{u}^{n+1}$ by adding the gradient of the Hodge variable obtained at the previous iteration to the conditions enforced (assuming that $\nabla\Phi^{n+1}$ is not far from $\nabla\Phi^n$, which proves to be a good approximation).

The boundary condition for the pressure is enforced during the second step of the projection method, i.e. when solving for the Hodge variable (20) using the expression $p = \alpha \frac{\rho}{\Delta t_n}\Phi - \mu \nabla \cdot \mathbf{u}^*$.

3.3.3. Time step restriction

As suggested by Brackbill et al. in [4], the simulation of free surface flows is stable if it is able to resolve the capillary waves along the interface. Denoting by c the velocity of those capillary waves, this induces a restriction of:

$$c \frac{\Delta t}{\Delta x} < \frac{1}{2}, \tag{24}$$

where here Δx refers to the smallest grid size. However, the velocities c_k of those capillary waves are given as a function of the wave number k by:

$$c_k = \sqrt{\frac{\gamma k}{\rho}}. \tag{25}$$

The maximum time step allowed can be estimated in Equation (24) using the maximum wave velocity. Equation (25) shows that this maximum velocity is reached for maximum wave numbers, i.e. for minimum wave lengths. The minimum

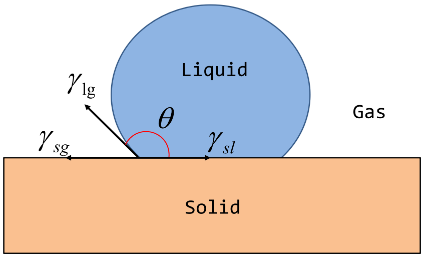


Fig. 4. Solid–liquid–gas triple point and contact angle.

wave length that can be resolved is $2\Delta x$, which corresponds to a maximum wave number $k_{max} = \frac{\pi}{\Delta x}$. Therefore, by substituting c by $c_{max} = \sqrt{\gamma\pi/\rho\Delta x}$ in Equation (24), one obtains the following time-step constraint:

$$\Delta t < \sqrt{\frac{\rho(\Delta x)^3}{2\pi\gamma}} \quad \text{or} \quad \Delta t = \mathcal{O}\left(\Delta x^{\frac{3}{2}}\right).$$

3.3.4. Contact angle

A contact angle is implemented in the Navier–Stokes solver in order to model the dynamics of the wetting of a surface. The model chosen involves another velocity boundary condition: at the triple point (the point where the wall, the bulk fluid and the external fluid, assumed to be at rest, meet), a velocity is imposed, equal to [36]:

$$\mathbf{u} = \frac{1}{\beta}(\cos(\theta_c) - \cos(\theta))\mathbf{t},$$

where β is the friction coefficient between the solid surface and the fluid (usually taken between 2 and 10), θ the angle formed by the fluid surface with the substrate, θ_c the contact angle (of about 45° for an interface between water and glass) and $\mathbf{t}$ is the tangent vector to the substrate (Fig. 4).

4. Verification of the method

4.1. Verification in two spatial dimensions

Three types of tests are designed in order to validate separately the diffusive part of the governing equation for the interfacial concentration, the advective part of this same equation, and the diffusive part of the governing equation for the bulk concentration as well as the coupling terms between the bulk and the interface. Those tests are for the most part inspired by those of Muradoglu and Tryggvason in [27], some slightly modified in order to fully validate every term of the governing equations.

4.1.1. Verification of the surfactant diffusion

In order to validate the time-dependent evolution equation for Γ:

$$\frac{\partial\Gamma}{\partial t} = D_s\Delta_s\Gamma,$$

Muradoglu and Tryggvason [27] considered the diffusion of a surfactant around a circular bubble of radius one, on which an initial surfactant distribution has been imposed. The initial concentration is written in cylindrical coordinates as: $\Gamma(\theta, t=0) = \frac{1}{2}(1+\sin(\theta))$. Writing the surface Laplace operator on a circle of radius r in cylindrical coordinates, $\Delta_s = \frac{1}{r^2}\frac{\partial^2}{\partial\theta^2}$, one can solve analytically the surface diffusion equation for Γ as:

$$\Gamma(\theta, t) = \frac{1}{2}\left(1 + e^{-D_s t/r^2}\sin(\theta)\right).$$

Fig. 5 depicts the initial and final distribution for Γ as well as a typical adaptive grid. The results of the simulation with a mesh refinement of min/max-level equal to 0/8 are given in Fig. 6.

We can also focus on the convergence of the algorithm, by starting with a simulation with a min/max-level of 0/6 and doing this simulation again after having split every cell (to obtain a min/max-level of 1/7). After having iterated this process up to a min/max-level of 3/9, we can compare the results, which are presented in Fig. 7. We also computed the L^∞ error as a function of time for every min/max-level chosen. The result is represented in Fig. 8. We observe that the L^∞ error increases fast during the first time steps, which is normal since the first time steps of our second-order-in-time discretization need to be initialized, and thus the first time steps are only of order one in time. We also observe that the error tends to decrease after some time, which can be explained by the fact that for every resolution chosen, the method converges asymptotically

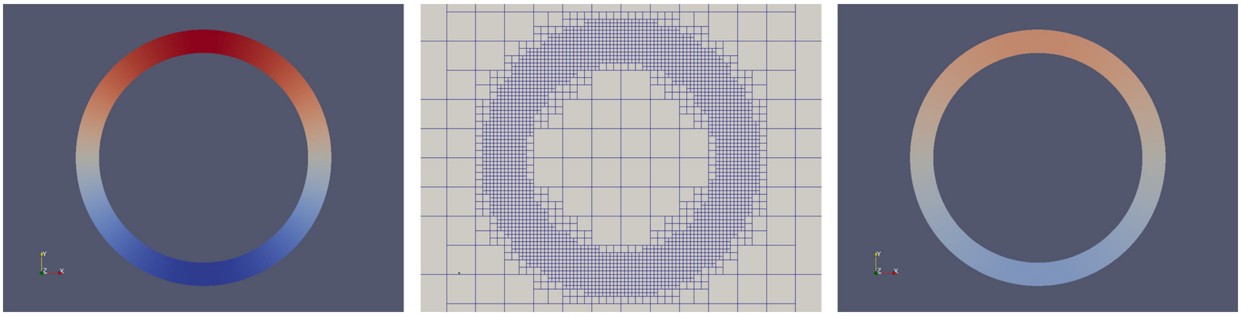


Fig. 5. Representation of the geometry chosen for the interfacial diffusion test. From left to right: (a) the computational tube at initial time, colored by interfacial concentration; (b) the mesh used for the computation with minimum level/maximum level 0/8; and (c) the computational tube at final time, with the same coloration.

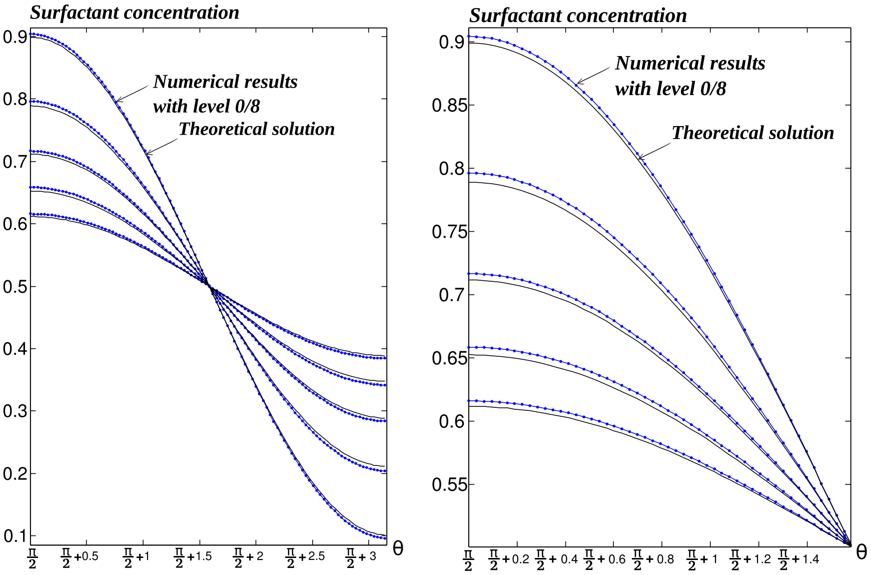


Fig. 6. Representation of the interfacial concentration as a function of the angle θ for $\theta \in \left[\frac{\pi}{2};\frac{3\pi}{2}\right]$ at different times. The blue dotted line represents the numerical results for a min/max-level of 0/8, whereas the black line represents the theoretical solution. A zoom of these curves is given in the right part of this figure. The time from the more stretched curves to the flatter ones are .25, .5, .75, 1 and 1.25. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to a uniform distribution of surfactant around the bubble, with $\Gamma(\theta, t = +\infty) = \frac{1}{2}$. Table 1 indicates that the method is first-order accurate in the L^∞-norm.

4.1.2. Verification of the surfactant advection

The test proposed by Muradoglu and Tryggvason in [27] consists of a spherical bubble with initial radius 1 that is laden with a uniform distribution of surfactant. This bubble is expanding under a radial unit velocity flow. Note that this velocity field is not divergence free, but has a divergence equal to: $\nabla \cdot \mathbf{u} = \frac{1}{r}$ in cylindrical coordinates; the advection equation that needs to be solved is therefore:

$$\frac{\partial \Gamma}{\partial t} + \mathbf{u} \cdot \nabla \Gamma + \Gamma \nabla \cdot \mathbf{u} - \Gamma\, \mathbf{n} \cdot [(\nabla\, \mathbf{u}) \cdot \mathbf{n}] = 0\,. \tag{26}$$

However, since we have chosen $\mathbf{u}$ confounded with the outward normal $\mathbf{n}$ ($\mathbf{u} = \frac{\mathbf{x}}{|\mathbf{x}|} = \mathbf{n}$), the relation $||\mathbf{n}|| = ||\mathbf{u}|| = 1$ implies that $(\nabla\, \mathbf{u}) \cdot \mathbf{n} = (\nabla\, \mathbf{n}) \cdot \mathbf{n} = 0$. Therefore, the last term on the left member of Equation (26) is always equal to zero. This test therefore does not fully test the interface advection and we modify it with a more complex geometry while keeping the radial velocity field $\mathbf{u} = \frac{\mathbf{x}}{|\mathbf{x}|}$. Specifically, we consider a star-like shape, with an equation in cylindrical coordinates given by:

$$r(\theta) = 0.75\,(1 - 0.2\sin(7\theta))\,.$$

Fig. 9 represents the geometry chosen for those advection tests. In particular, it is obvious that the radial velocity field is now not confounded anymore with the normal so the term $-\Gamma\mathbf{n} \cdot [(\nabla\, \mathbf{u}) \cdot \mathbf{n}]$ in Equation (26) now has a value that

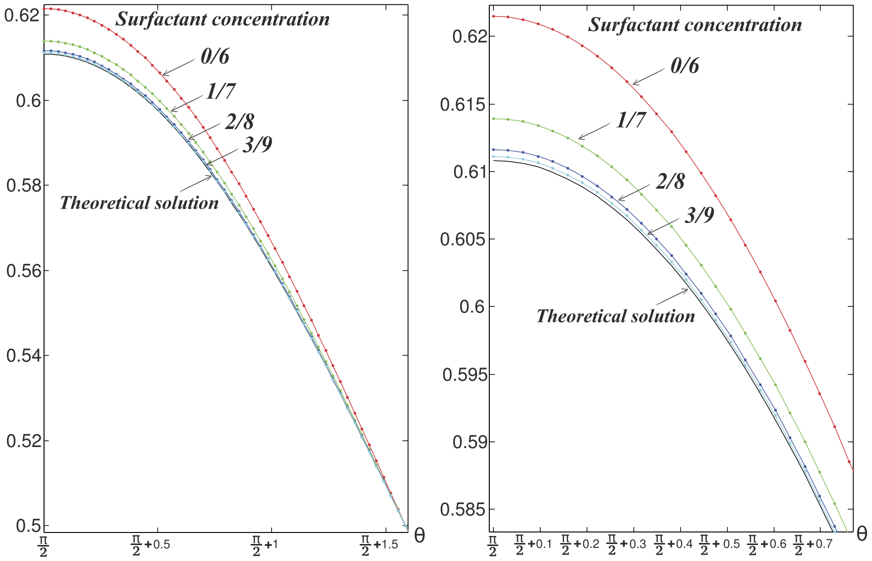


Fig. 7. Representation of the interfacial concentration as a function of the angle θ for $\theta \in \left[\frac{\pi}{2};\pi\right]$ at the same time for different min/max-level chosen for the simulation. A detail of these curves is given in the right part of this figure.

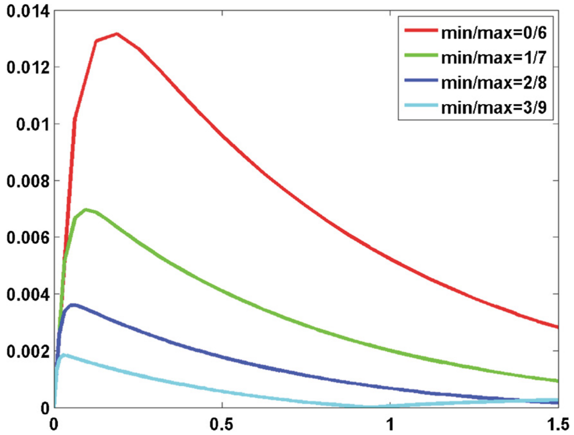


Fig. 8. L^∞ error for Example 4.1.1 as a function of time for different min/max-level.

Table 1
Order of accuracy for Example 4.1.1.

Grid level (min/max)	L^∞ error	Order
0/6	1.32×10^{-2}	
1/7	6.97×10^{-3}	.921
2/8	3.60×10^{-3}	.953
3/9	1.83×10^{-3}	.976

depends on the location on the star-shaped bubble. Finally, taking the initial surfactant concentration to be $\Gamma(\theta, t=0) = \frac{1}{2}$ and invoking the conservation principle of the interfacial surfactant, we get the following analytical description of the time-dependent evolution of the surfactant concentration on the star:

$$\Gamma(\theta, t) = \frac{1}{2} \frac{\sqrt{r^2(\theta) + (r')^2(\theta)}}{\sqrt{(t + r(\theta))^2 + (r')^2(\theta)}}.$$

The interfacial concentration as a function of the angle θ is given in Fig. 10. The numerical results are in agreement with the analytical solution. As we could have expected the error is greater in the hollows between two branches of the star (which corresponds to where the concentration reaches its maximum), because it is where the interface is less resolved.

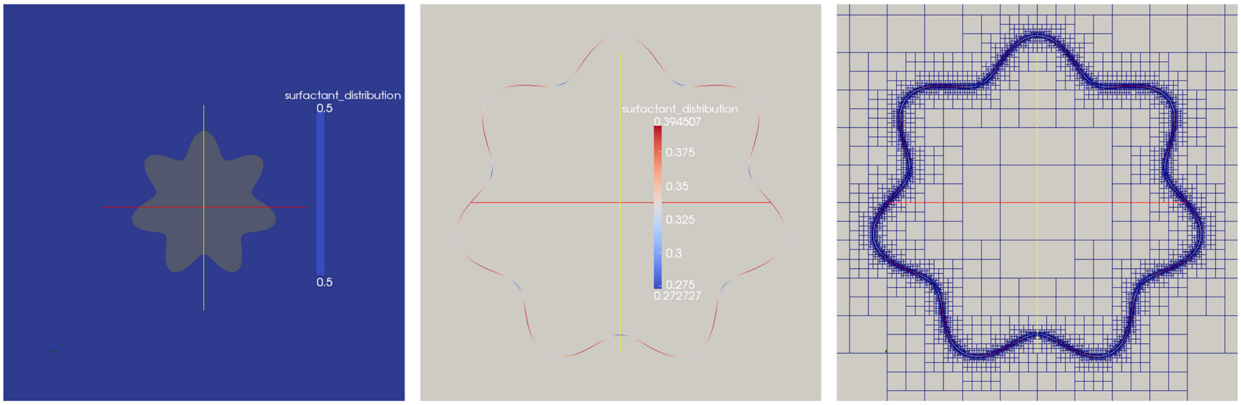


Fig. 9. Representation of the geometry chosen for the interfacial advection test: (a) the star at initial time, colored according to the interfacial concentration; (b) the concentration distribution at final time, with the same color map; and (c) the mesh used for the computation with a min/max-level of 0/9. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

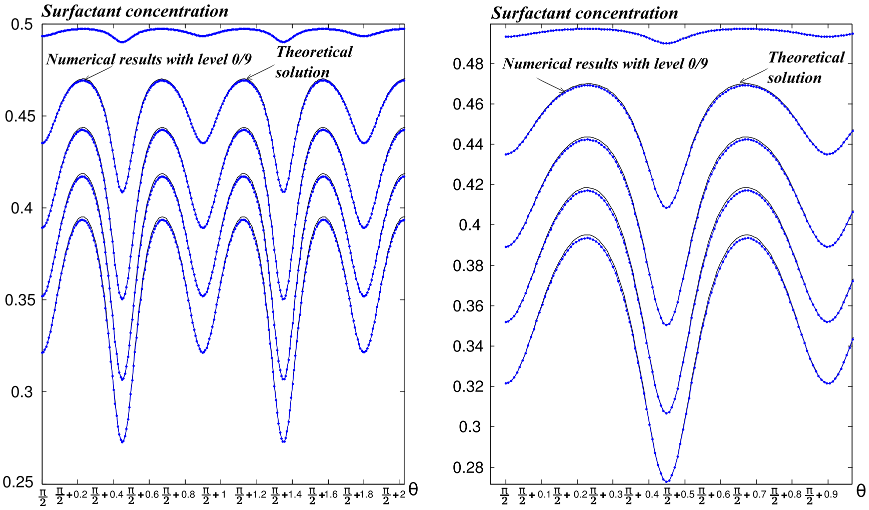


Fig. 10. Representation of the interfacial concentration as a function of the angle θ for $\theta \in \left[\frac{\pi}{2}; \frac{8\pi}{7}\right]$ at different times for a min/max-level of 0/9. The blue dotted lines correspond to the numerical results whereas the black lines represent the analytical solution. A detail of these curves is given on the right part of this figure. Times from top curve to bottom curve are .025, .125, .25, .375 and .5. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

A more challenging test that we consider is the advection of a circular bubble in a vortex. In a computational domain with dimensions $[0, 1] \times [0, 1]$, we place a circular bubble of radius 0.15 and centered at $(0.5, 0.75)$. The bubble is initially laden with a uniform surfactant concentration $\Gamma(x, y, t = 0) = \frac{1}{2}$, and is advected by the following divergence-free velocity field $\mathbf{u} = (u, v)$:

$$\begin{aligned} u(x, y) &= \quad \sin^2(\pi x) \sin(2\pi y), \\ v(x, y) &= -\sin^2(\pi y) \sin(2\pi x), \end{aligned} \tag{27}$$

for $t \in [0, \frac{1}{2}]$. At $t = \frac{1}{2}$ the velocity field is reverted so that at $t = 1$ the system should return to its initial configuration.

Fig. 11 represents the global configuration for this test, at initial time and final time. It gives an idea of how challenging this test is, since the surface of the bubble is very stretched during this advection, and it develops a very sharp extremity that tends to be under-resolved for low grid resolutions. This test is therefore an indication of the typical accuracy that one may expect in the case where an interface undergoes large deformations, especially when fine features develop (here the thin tail of the interface). It also illustrates the advantage of an adaptive grid.

In Fig. 12 we have represented the repartition of the concentration as a function of θ around the circle for different spatial resolutions ranging from 0/8 to 4/12 after having performed an advection with the velocity field $\mathbf{u}$ given by Equation (27) during a time 0.5 and having advected it back to its initial position with the velocity field $-\mathbf{u}$ during the same time interval 0.5.

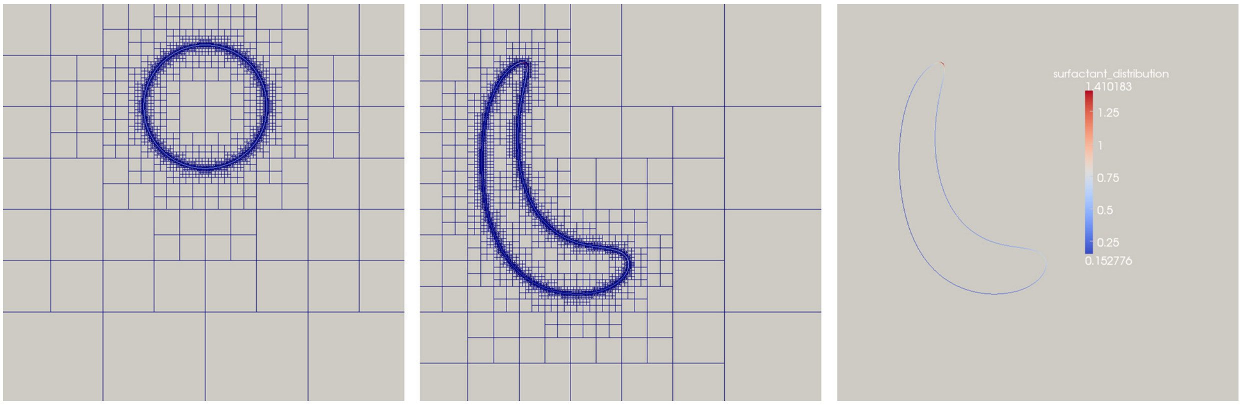


Fig. 11. Representation of the geometry chosen for the vortex advection test. From left to right: (a) the grid (with min/max-level 0/9) at initial time; (b) the same grid at time $t = 0.5$; and (c) the interface at time $t = 0.5$, colored by interfacial concentration. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

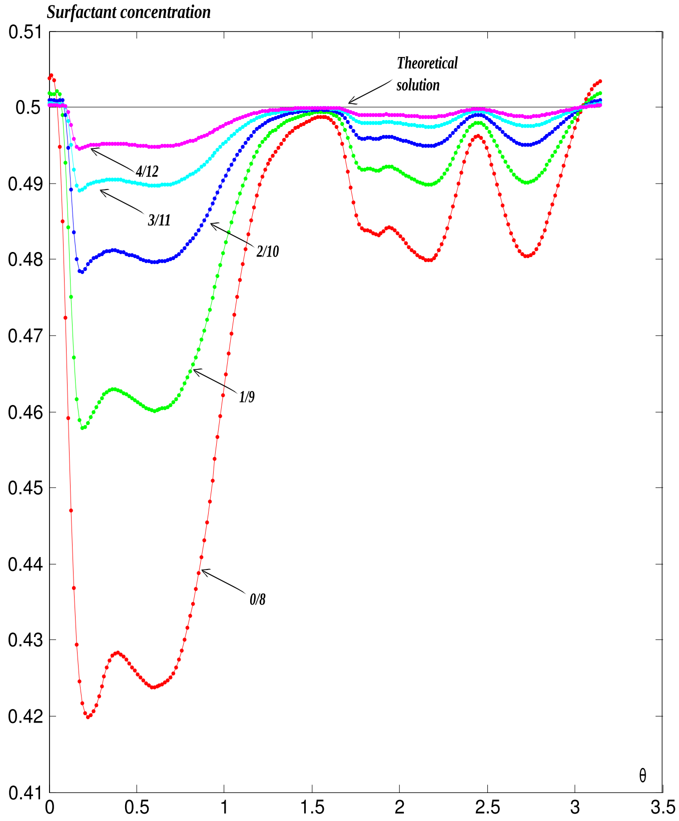


Fig. 12. Representation of the interfacial concentration as a function of the angle θ for $\theta \in [0; \pi]$ at the final time ($t = 1$) for different min/max-level chosen for the simulation. The red dotted line corresponds to a min/max-level of 0/8, the green one to 1/9, the blue one to 2/10, the cyan one to 3/11 and the magenta one to 4/12. The black line represents the theoretical solution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We can also observe the time-evolution of the surfactant loss on the bubble by integrating the surface concentration around the interface, and observing how this amount evolves with time. The integration along the interface is performed using the integration method described in [24]. The results are given in Fig. 13.

Fig. 13 indicates that the sudden change of the velocity field from $\mathbf{u}$ to $-\mathbf{u}$ at time 0.5 has no significant influence on the evolution of the total amount of surfactant. The surfactant loss at final time is about 6% for the lowest spatial resolution, which is a good result since the interface tends to be under-resolved around time 0.5 when the bubble is most distorted.

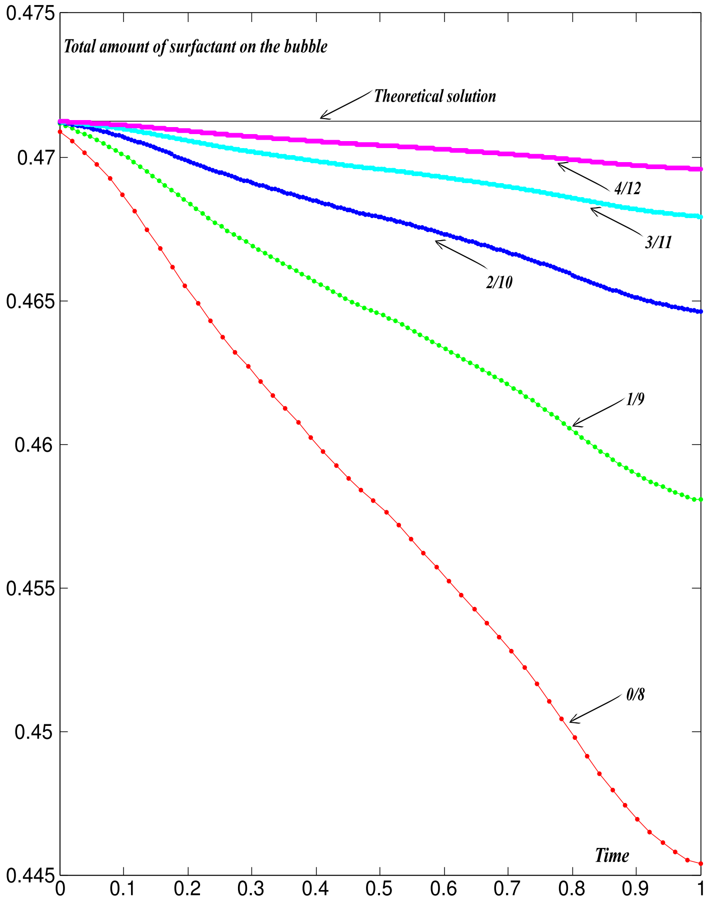


Fig. 13. Representation of the total amount of surfactant as a function of time for different min/max-level chosen for the simulation. The red dotted line corresponds to a min/max-level of 0/8, the green one to 1/9, the blue one to 2/10, the cyan one to 3/11 and the magenta one to 4/12. The black line represents the theoretical solution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1.3. *Verification of the bulk concentration diffusion and of the coupling between the bulk and the interface*

The first test is that proposed in [27]: a circular bubble of radius 1, initially clean, is placed in an infinite bulk charged with an initial constant concentration c_∞ of surfactant. The surfactant concentration c in the bulk diffuses with a diffusion constant D but is not advected, i.e. we take $\mathbf{u} = 0$ in Equation (13). A simplified coupling term S is chosen as $S = k_a c$. In this setting, [27] give an analytical solution, with the restriction that it is valid for short times only. Indeed, in an infinite domain and with a source term of the form given above, the concentration on the bubble will tend to $+\infty$, whereas for a finite computational domain the conservation of the total amount of surfactant (having implemented homogeneous Neumann boundary conditions for c at the boundary of the domain) prevents the concentration of surfactant in the bulk from diverging. In addition, it appears that even in an infinite domain this analytical solution remains an approximation, valid for short times only [28]. For these two reasons (the facts that we work with a finite domain, and that the analytical solution to which we compare our results is only an approximation, valid for short times only), we will restrict our investigations of the results of this test to short times after the beginning of the diffusion and coupling between the bulk and the interface.

Fig. 14 provides the results for the time-dependent evolution of the total amount of surfactant on the interface, with min/max-level ranging from 6/8 to 10/12. We can observe that the numerical results obtained are consistent for short times with the analytical solution proposed in [27]. However, the time scale observed is so small compared to the characteristic time scale of diffusion that the analytical solution reduces to a line, which means that we only observe the dominant order of the source term, which is: $S = k_a c_\infty + \mathcal{O}(t)$.

For our second test, we consider a clean infinite wall that is placed in front of a semi-infinite bulk domain with initial concentration c_∞ that diffuses with a diffusion coefficient D. A simplified coupling term $S = k_a c$ ensures the surfactant transfer between the bulk and the interface. The analytical solution for the time-dependent evolution of the concentration c as a function of the time t and of the distance x to the interface is given by:

$$c(x,t) = c_\infty \left[\operatorname{erf}\left(\frac{x}{2\sqrt{Dt}} \right) + e^{k_a x + k_a^2 Dt} \operatorname{erfc}\left(\frac{x}{2\sqrt{Dt}} + k_a \sqrt{Dt} \right) \right], \tag{28}$$

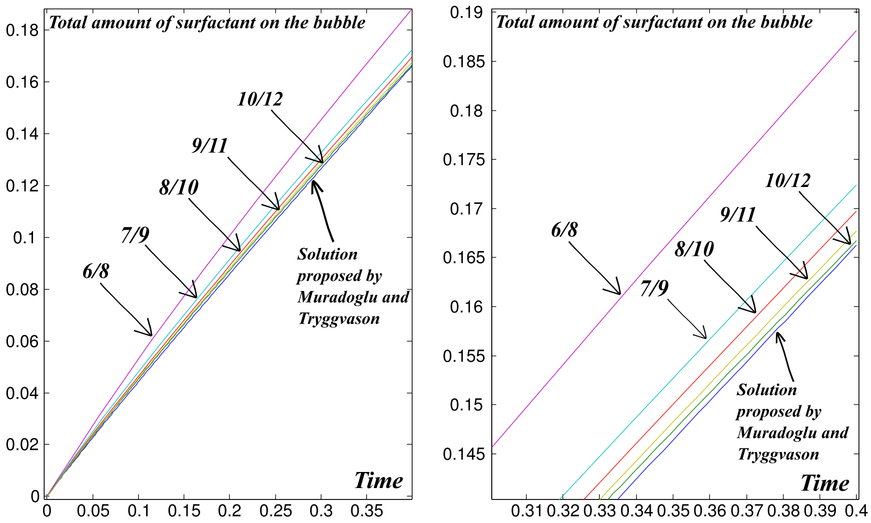


Fig. 14. Representation of the total amount of surfactant on the bubble as a function of time for short times. The magenta line corresponds to a min/max-level of 6/8, the cyan one to 7/9, the red one to 8/10, the yellow one to 9/11 and the green one to 10/12. The black line represents the analytical solution proposed by [27]. A detail of these curves is given in the right part of this figure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

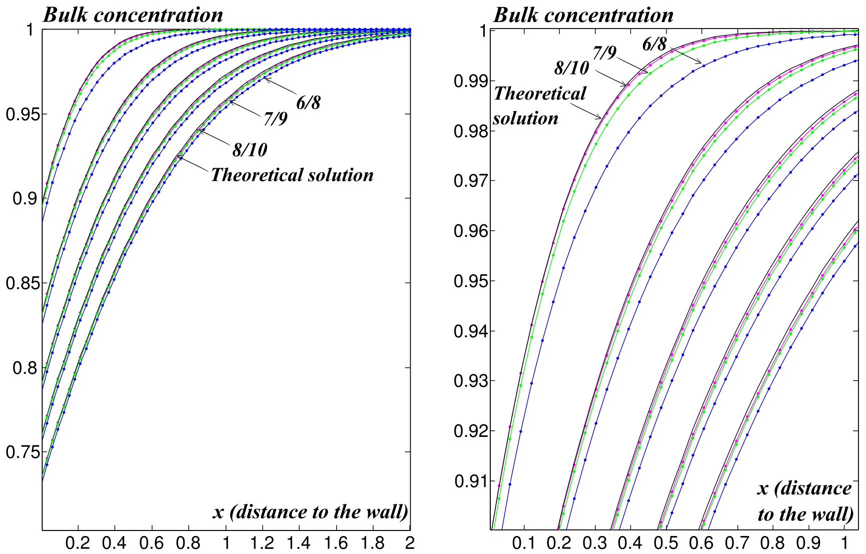


Fig. 15. Bulk concentration c as a function of the distance x to the interface. The blue dotted line corresponds to a min/max-level of 6/8, the green one to 7/9 and the magenta one to 8/10. The black line represents the analytical solution. A detail of these curves is given in the right part of this figure. The time elapsed, from top to bottom curves, is .04, .12, .2, .28 and .36. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \ ,$$

and

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-t^2} dt \ .$$

The results presented in Fig. 15 give the surfactant concentration c as a function of the distance to the wall for different times, and with different min/max-level ranging from 6/8 to 8/10. Fig. 15 shows that even for long times, the diffusion

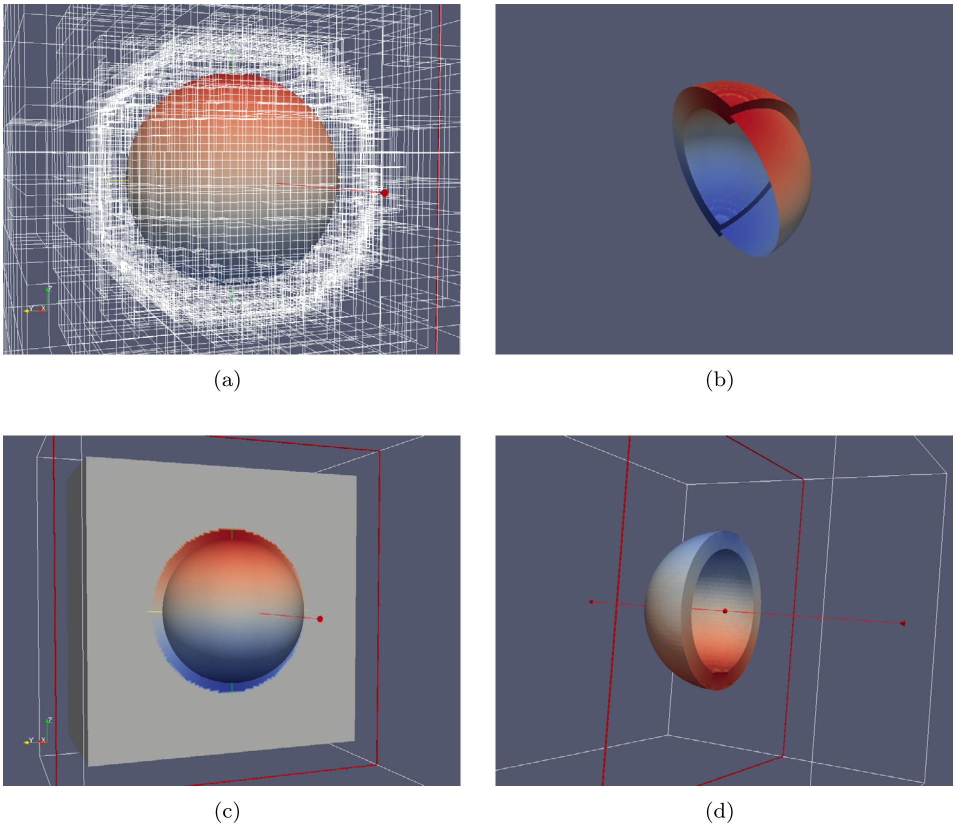

(a) (b) (c) (d)

Fig. 16. Setup for Example 4.2.1: (a) the mesh used for this computation with a min/max-level of 0/6; (b) a visualization of the tube and the solution at initial time with a min/max-level of 0/7 (c) solid visualization of the computational tube and the solution at final time with min-level/max/level 0/7, colored by the interface concentration; (d) another visualization of the tube, with the same min/max-level.

of the bulk concentration c as well as the surfactant transfer between the bulk and the interface occur as predicted by the analytical solution. As expected, though, the accuracy of the results tends to be deteriorate for long times. This can be attributed to the fact that the analytical solution was computed for an infinite domain, whereas the simulations were only performed on a finite computational domain. This explanation is consistent with the observation that the concentration in the bulk for long times tends to be lower than predicted by the analytical solution in infinite domain.

4.2. Verification in three spatial dimensions

4.2.1. Verification of the surfactant diffusion

We consider the diffusion of surfactant around a spherical bubble of radius 1, on which an initial distribution of the surfactant has been imposed. The initial concentration chosen, in spherical coordinates, is $\Gamma(\theta, t=0) = \frac{1}{2}\left(1+\sin(\theta)\right)$. The surface Laplace operator on a sphere of radius r can be written in spherical coordinates, yielding $\Delta_s = \frac{1}{r^2 \sin(\theta)} \frac{\partial}{\partial\theta}\left(\sin(\theta)\frac{\partial}{\partial\theta}\right)$. The solution of the time-dependent evolution equation for Γ,

$$\frac{\partial \Gamma}{\partial t} = D_s \Delta_s \Gamma,$$

is therefore

$$\Gamma(\theta, t) = \frac{1}{2}\left(1 + e^{-\frac{2tD_s}{r^2}} \sin(\theta)\right).$$

Fig. 16 proposes different visualizations for the geometry of this problem. The results of the simulations with a mesh refinement of min/max-level of 0/7 are given in Fig. 17. Fig. 18 provides the results of the simulation under mesh refinement.

4.2.2. Verification of the surfactant advection

For the verification of the interface advection, we use the same idea as in the two-dimensional case, performing the expansion of a shape laden with surfactant under a radial velocity field. For the same reasons as in section 4.1.2, the shape

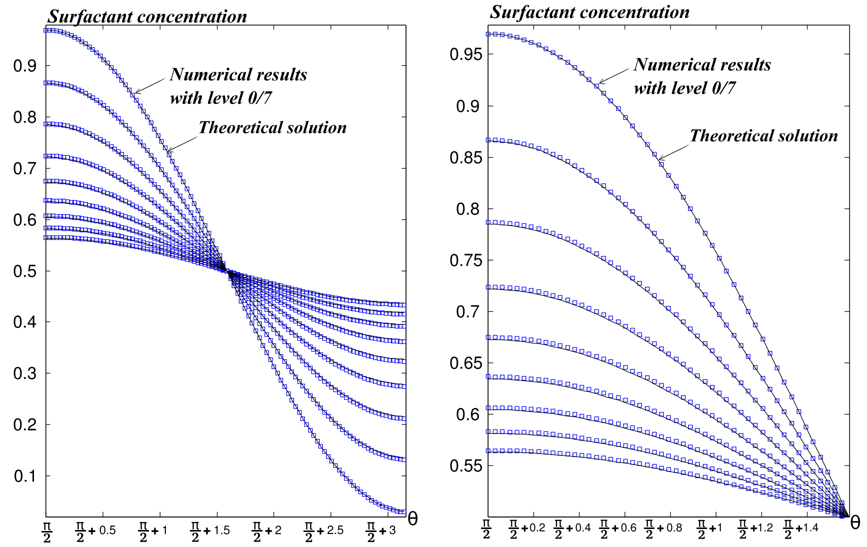


Fig. 17. Representation of the interfacial concentration as a function of the angle θ for $\theta \in \left[\frac{\pi}{2}; \frac{3\pi}{2}\right]$ at different times. The blue dotted line with square markers represents the numerical results for a min/max-level of 0/7, whereas the black line represents the theoretical solution. A detail of these curves is given in the right part of this figure. The times from the top to bottom curves are .04, .16, .28, .4, .52, .64, .76, .88 and 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

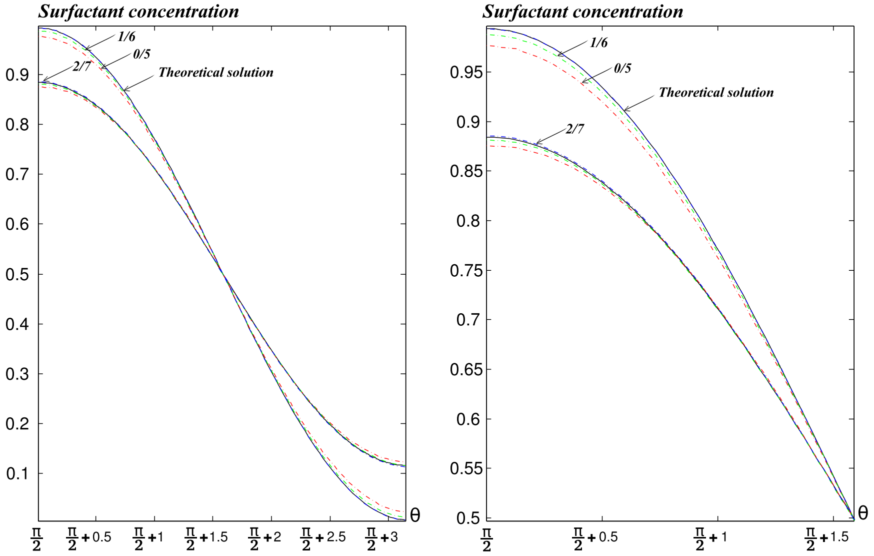


Fig. 18. Representation of the interfacial concentration as a function of the angle θ for $\theta \in \left[\frac{\pi}{2}; \pi\right]$ at two different times (from top to bottom, .02 and .14) for various min/max-level. The red dotted lines correspond to a min/max-level of 0/5, the green lines correspond to a min/max-level of to 1/6 and the blue lines correspond to a min/max-level of to 2/7. The black lines represent the theoretical solution. A detail of these curves is given in the right part of this figure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

should not be a sphere if the test is to be relevant. We thus consider a star-like shape given by the equation, in spherical coordinates,

$$r(\theta,\phi) = 0.75 + 0.2\,(1 - 0.6\cos(6\phi))\,(1 - \cos(6\theta))\,.$$

Taking the initial surfactant concentration to be $\Gamma(\theta, t=0) = \frac{1}{2}$, the conservation of the interfacial surfactant leads to the following analytical description of the time-dependent evolution of the surfactant concentration on the star:

$$\Gamma(\theta,\phi,t) = \frac{1}{2}\,\frac{\sqrt{r^2(\theta,\phi) + \left(\frac{\partial r}{\partial \theta}\right)^2(\theta,\phi)}\sqrt{r^2(\theta,\phi) + \left(\frac{\partial r}{\partial \phi}\right)^2(\theta,\phi)}}{\sqrt{(t + r(\theta,\phi))^2 + \left(\frac{\partial r}{\partial \theta}\right)^2(\theta,\phi)}\sqrt{(t + r(\theta,\phi))^2 + \left(\frac{\partial r}{\partial \phi}\right)^2(\theta,\phi)}}\,.$$

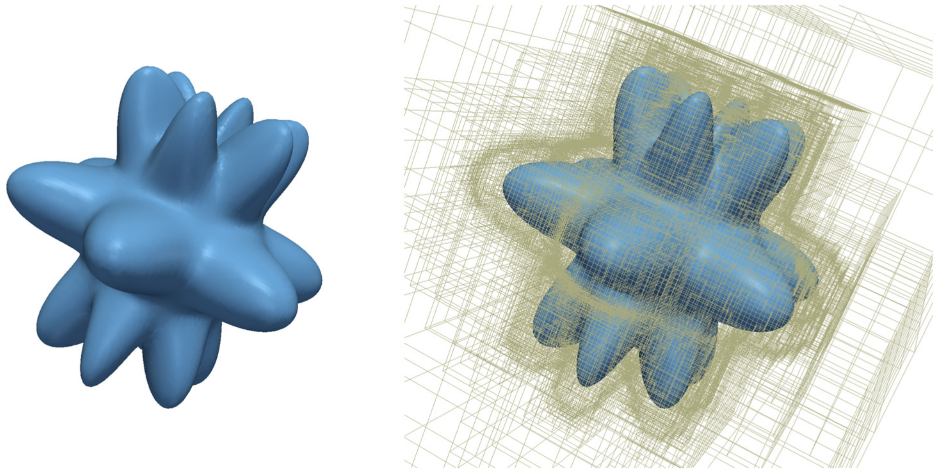


Fig. 19. Representation of the geometry chosen for the interfacial advection test in 3D. From left to right: the star at initial time and the same star with the mesh used for the computation with a min/max-level of 0/7.

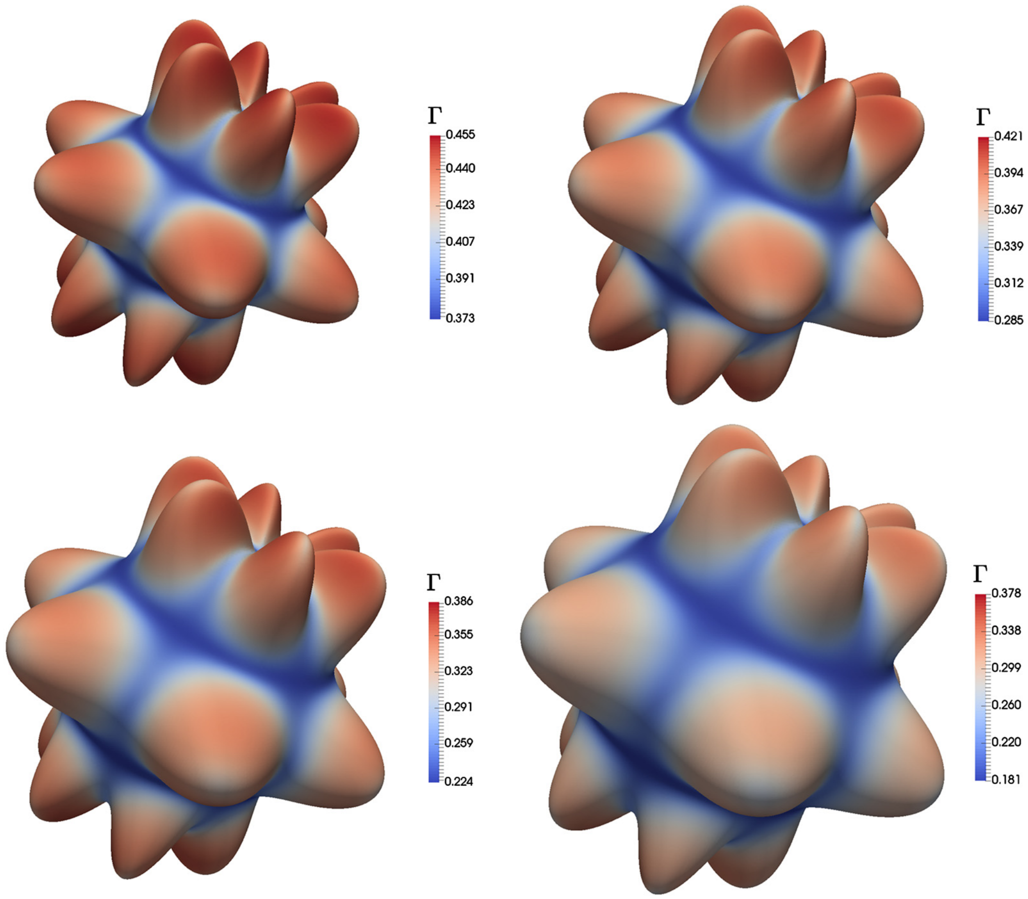


Fig. 20. Visualization of the motion and surface concentration evolution on a star advected under a radial velocity field. The time elapsed is, from top left to bottom right: .125, .25, .375 and .5. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

Fig. 19 represents the geometry, while Fig. 20 depicts the solution at different times. Similarly to the two-dimensional case, we can study the interfacial concentration as a function of the angle θ at different times. We do that in two different ways: by clipping the star by a plane that crosses the apexes of the star, or by clipping it by a plane that passes through the hollow regions (see Fig. 21). The numerical results for a clipping plane that passes through the hollow regions of the star are given in Fig. 22 whereas the results for a clipping plane crossing the apexes of the star are given in Fig. 23. The numerical

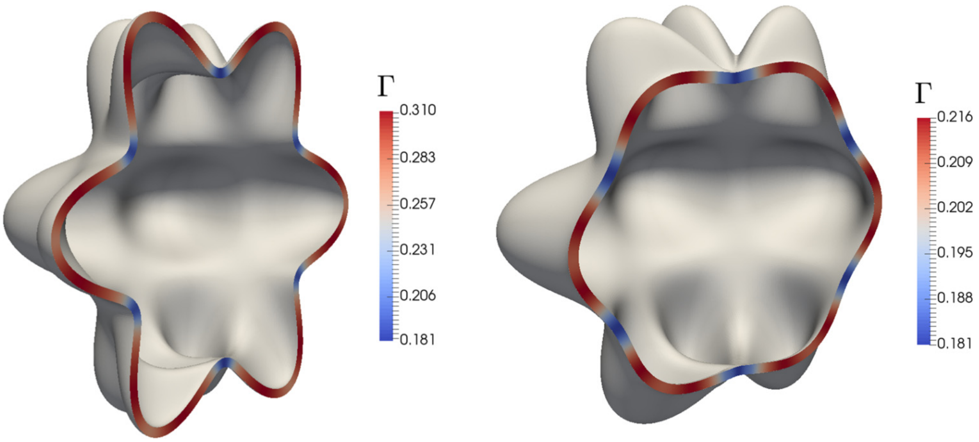


Fig. 21. Representation of two different ways to clip the star with a plane containing the z axis: by a plane that crosses the apexes of the star (left) or by a plane that passes through the hollow regions (right). Both figures show the star at $t = 0.5$, with Γ represented in the tube in which the interface transport is solved. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

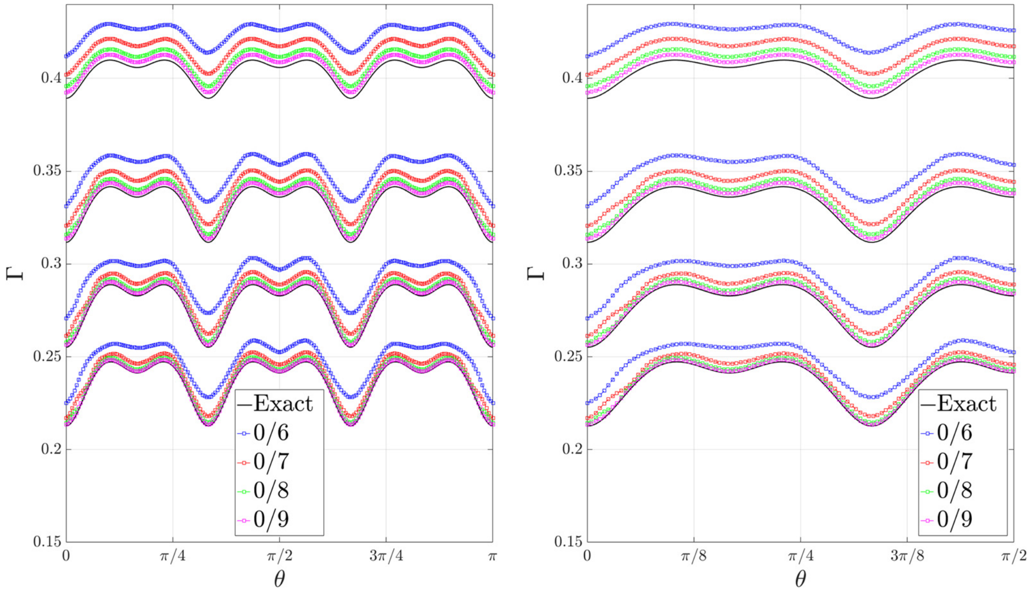


Fig. 22. Representation of the interfacial concentration as a function of the angle θ for $\theta \in [0; \pi]$ at different times (from top to bottom .1, .2, .3 and .4), clipping the star by a plane that passes through the hollow regions between the apexes. The color of the lines with square markers indicates the resolution of the simulation. The black lines represent the exact solution. A detail of these curves is given in the right part of this figure. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

results obtained are in good agreement with the analytical solution, and are convergent. As we could have expected, the errors are greater when clipping the interface by a plane that passes through the hollows between two branches of the star since the interface is less resolved in those regions. For the same reason, the results are less accurate for angles θ close to 0 or π.

5. Computational examples

In this section, we consider a few physically relevant problems that can be studied with our computational method.

5.1. Bubble in a shear flow

The problem of the deformation of bubbles or drops in a shear flow or a uniaxial extensional flow and in the presence of surfactant, has already been widely studied in the literature (see e.g. [34] and the references therein). The results of all those studies tend to prove that the presence of surfactant promotes the cleavage of drops and bubbles, which can

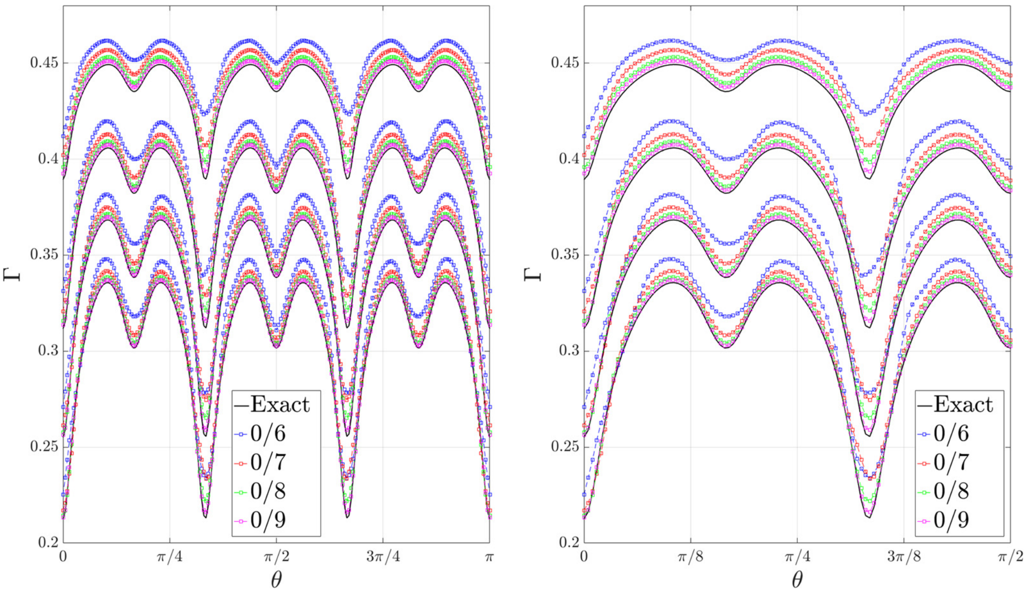


Fig. 23. Representation of the interfacial concentration as a function of the angle θ for $\theta \in [0; \pi]$ at different times (from top to bottom .1, .2, .3 and .4), clipping the star by a plane that passes through the apexes of the star. The color of the lines with square markers indicates the resolution of the simulation. The black lines represent the exact solution. A detail of these curves is given in the right part of this figure. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

be expected since the presence of surfactant lowers the surface tension. Moreover, when the coupling term between the interfacial concentration and the bulk concentration is small, or when the surfactant is supposed to be insoluble, the flow tends to advect the surfactant towards the ends of the drop, where it accumulates, causing the drop to develop pointed ends as the surface tension becomes very small at its extremities. As expected, these effects become less visible as the surface Péclet number $Pe_s = \frac{LU}{D_s}$ increases. The surface Péclet number, which measures the ratio between the advective effects and the diffusive effects of the interfacial surfactant, considers length and velocity scales L and U, respectively.

In this section, we focus on the case when the surfactant is soluble and when the interfacial and the bulk concentrations are not initially balanced, with the interfacial concentration being significantly lower than the bulk's. Fig. 24 shows the time evolution of a bubble placed in such conditions. For this simulation, we modeled the effect of the surfactant on the surface tension through Frumkin kinetics (see Equations (6)), using the parameters of the 1-nonanol ($C_9H_{19}OH$) given by Lin et al.:

$$\begin{cases} n &= 1\,, \\ A &= 3.410\,, \\ \Gamma_\infty &= 5.915 \cdot 10^{-6}\ (\mathrm{mol/m^2}), \\ k_b/k_a &= 0.3137\ (\mathrm{mol/m^3}), \\ D_s &= 6.3 \cdot 10^{-10}\ (\mathrm{m^2/s})\,, \\ T &= 295.85\ (\mathrm{K}), \end{cases}$$

and noting that the surface tension for a clean interface is in this case the value for a water–air interface $\gamma_0 = 0.072$ (N/m).

Fig. 24 reveals that the motion of the advected fluid around the bubble tends to enhance the adsorption of surfactant on the bubble's interface near the location of breakup, thus creating propitious conditions for the future breakup of the bubble. Moreover, as the surfactant accumulates near the future breakup point, driven by the flow, its presence not only weakens the surface tension around it, but also creates a gradient of surface tension from the regions of high surfactant concentration to the regions of low surfactant concentration thus generating Marangoni forces that drive the flow close to the interface from the equatorial plane of the bubble to its edges, which is another mechanism that can be held responsible for the breakup of the bubble.

After the breakup occurs, we can observe that the ends of the two daughter bubbles near the breakup point present a sharp pointed shape. This can be attributed to the fact that the high surfactant concentration near those two edges has weakened the surface tension thus allowing such shapes to exist. This is perfectly consistent with all the observations reported in the literature inventoried above, such sharp-ended shapes having already been reported in the case of bubbles in shear flows or extensional flows in the presence of surfactant.

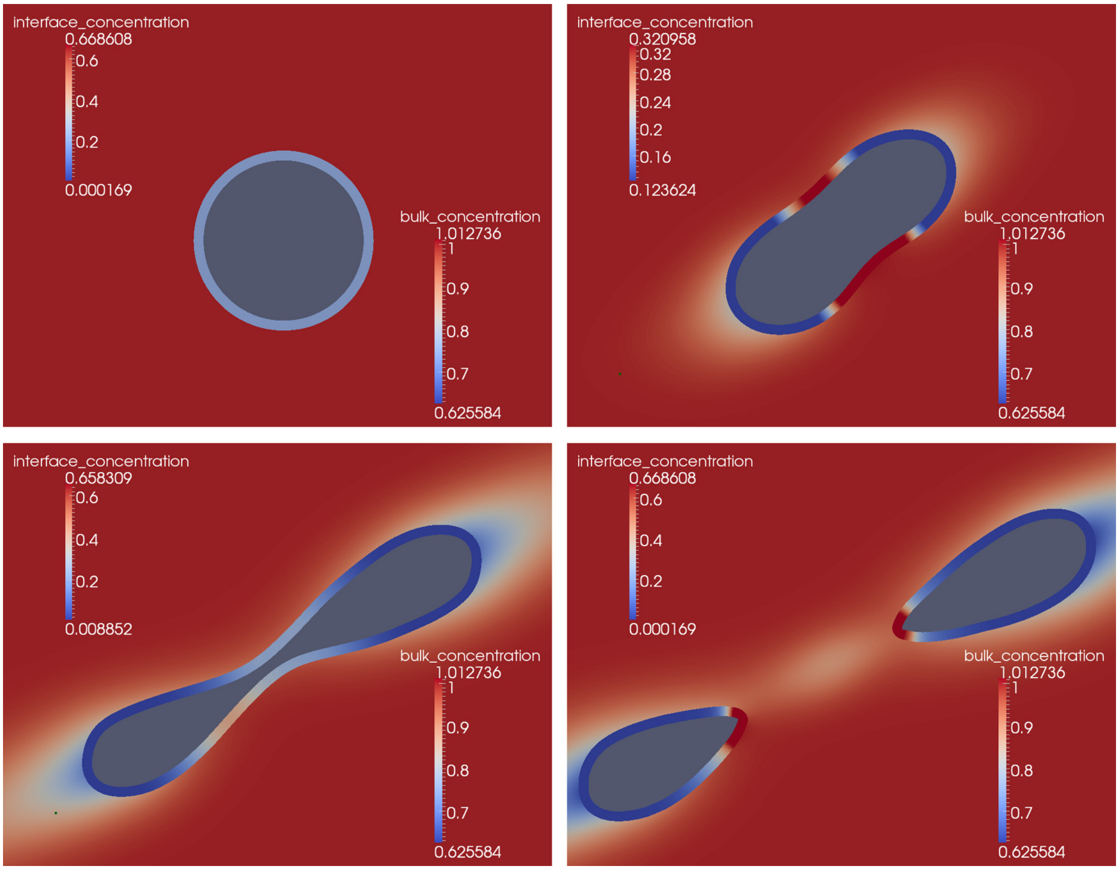


Fig. 24. Drop breakup and evolution of the normalized concentrations in the bulk and on the interface for a bubble with a low initial concentration of surfactant placed in a contaminated bulk and under a shear flow. The time elapsed, from top left to bottom right, is 0, .25, .5 and .75. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

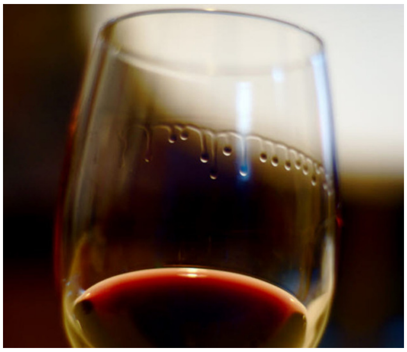


Fig. 25. Illustration of the phenomenon of tears of wine along the wall of a glass containing ordinary table wine.

5.2. Tears of wine

We consider the phenomenon of tears of wine (see Fig. 25), which is an emblematic manifestation of the Marangoni forces. The reason for the formation of the so called tears of wine is that the evaporation of ethanol at the air–fluid interface of the thin layer of wine deposited on the wall of a glass causes a huge decrease of the alcohol concentration in the film; whereas on the surface of the rest of the wine inside the glass, evaporation is compensated by diffusion inside the glass, and the concentration of ethanol near the interface remains almost constant. As the ethanol surface tension is much lower than the one of water, this creates a gradient of surface tension in the fluid film from the bottom to the top of the glass. This gradient of surface tension causes the emergence of Marangoni forces directed upwards. As the width of the film layer is very small, the fluid is not sheared very much by the gravity (this is why it tends to be static), and the Marangoni forces are strong enough to counterbalance those gravitational forces. This causes the fluid to climb along the wall of the glass, hence the formation of droplets that fall under their own weight once they have reached a critical size, evoking the shape of tears. This phenomenon has been reported for centuries but was first correctly explained by physicist James Thomson, the elder brother of Lord Kelvin, in 1855.[1]

As ethanol is not a surfactant, it is of no use to define an interfacial concentration since the only variable of interest in order to compute the surface tension is the bulk concentration near the interface. However, the evaporation rate of alcohol can be modeled by a desorption term on the interface. We can use our computational framework to simulate the evaporation of ethanol. The value of k_b is set to zero, since we neglect the contribution of the vapors of alcohol to the concentration of ethanol in the wine; and the source term is taken equal to $S(c) = k_a c$, the value of the evaporation rate k_a being given by O'Hare et al. (fitting the data reported in [30] reveals that $k_a = 5.919 \cdot 10^{-4}$ (m/s)). Of course, since the interfacial surfactant concentration has no physical meaning (despite the fact that it allows one to keep track of the evaporated ethanol and check that the total amount of ethanol remains constant in time, even though it decreases in the bulk), the surface diffusion coefficient is set to zero. The surface tension of the fluid is taken to be equal to a function of the molar fraction of ethanol, following the law that has been identified by Ghahremani et al. [12], namely:

$$\gamma\left(x_m\right) = \left[11.53\left(\log\left(x_m\right)\right)^3 + 26.54\left(\log\left(x_m\right)\right)^2 - 10.08\log\left(x_m\right) + 22.48\right] \cdot 10^{-3} \;\; (\mathrm{N/m}),$$

where x_m is the molar fraction of bulk ethanol at the interface. Note also that the molar fraction can be obtained from c as follows:

$$x_m\left(c\right) = \frac{c}{c + \dfrac{\rho_\mathrm{w}}{\mathrm{MW_w}}\left(1 - \dfrac{\mathrm{MW_e} \cdot c}{\rho_\mathrm{e}}\right)},$$

where:

$$\begin{cases} \rho_\mathrm{w} & = 1000\ (\mathrm{kg/m^3}), \\ \rho_\mathrm{e} & = 789\ (\mathrm{kg/m^3}), \\ \mathrm{MW_w} & = 0.018\ (\mathrm{kg/mol}), \\ \mathrm{MW_e} & = 0.046\ (\mathrm{kg/mol}) \end{cases}$$

are the densities and molecular weights of water and ethanol, respectively.

Fig. 26 depicts the motion of the wine in a thin layer deposited on the wall of a glass and connected to the rest of the wine in the glass. We can clearly see that as the mechanisms of evaporation and ethanol diffusion in water lower the ethanol concentration specifically in the thin layer, the gradient of surface tension thus created causes the wine to climb the side of the glass. The contact angle combined with the effects of gravity prevent the drop from climbing higher than a certain height. The wine thus accumulates at the upper extremity of the film, forming a droplet that eventually falls back under its own weight. Even though it seems difficult to define precise numerical parameters that could be measured in order to confirm the relevance of this simulation, the general behavior of the flow, the shape of the droplet that forms, and the mechanism responsible for its formation tend to show that this simulation is in good qualitative agreement with the phenomenon of tears of wine.

5.3. Coating

Dip coating, which consists in the deposition of a liquid on a solid substrate by simply withdrawing the substrate from a bath, is the simplest form of coating. Despite its numerous and important industrial applications, some of the phenomena involved are not fully understood. One of these phenomena is the film thickening due to the presence of surfactant. Landau and Levich were the first to determine the theoretical thickness h_∞ of the film deposited by the withdrawal of a flat wall of a fluid bath (with no surfactant). Under the hypothesis of low Reynolds number, this thickness is proportional to the capillary length l_c and depends on the capillary number Ca through:

$$h_\infty = 0.945\, l_c\, Ca^{\frac{2}{3}}, \tag{29}$$

where $l_c = \sqrt{\gamma/\rho g}$ and $Ca = \mu U/\gamma$.

However in the presence of surfactant, the thickness of the film given by Equation (29) proves too small. Despite numerous reports of this phenomenon of film thickening, it is only rather recently that the first experimental study of the flow field close to the coated substrate has been undertaken by Mayer and Krechetnikov in [21]. Their study features outstanding visualizations of the flow near the interface. This experimental work did not completely settle the controversy about

[1] Contrary to common beliefs, the number, width and thickness of the tears (or legs) and how slow they fall does not predict the quality of wine. Rather they are indicative of the degree of alcohol and quantity of glycerol (that increases the viscosity and thus makes the tears slower to fall).

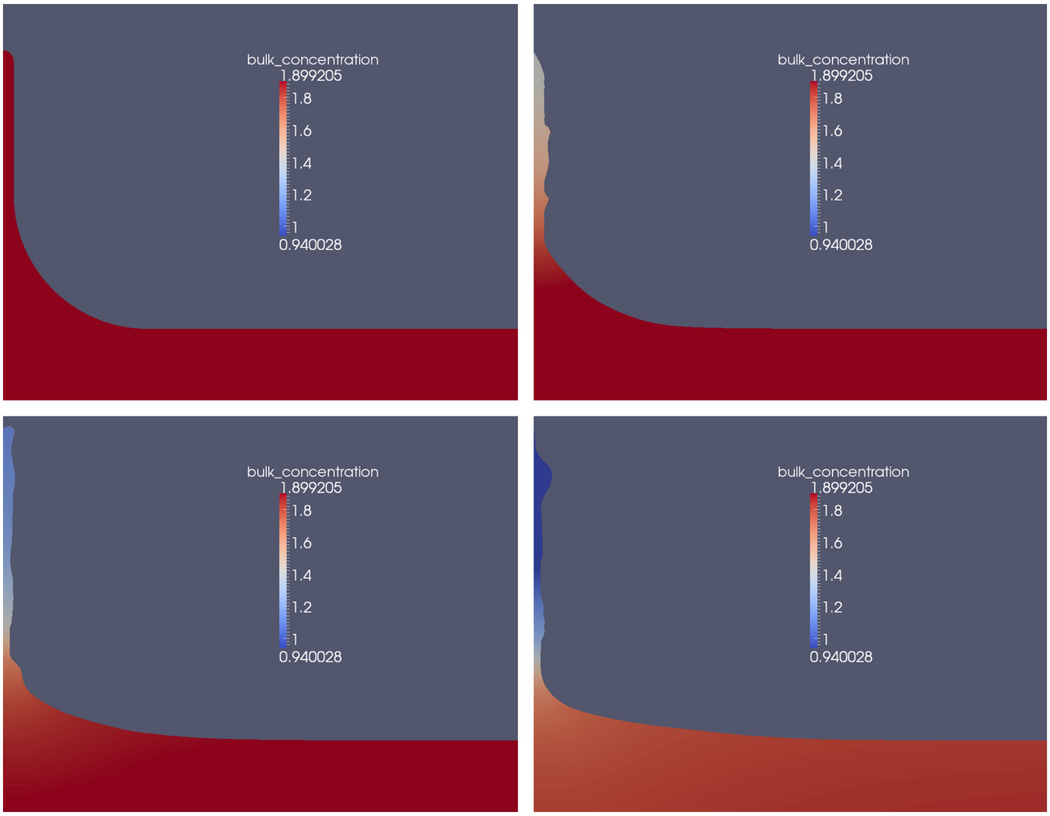
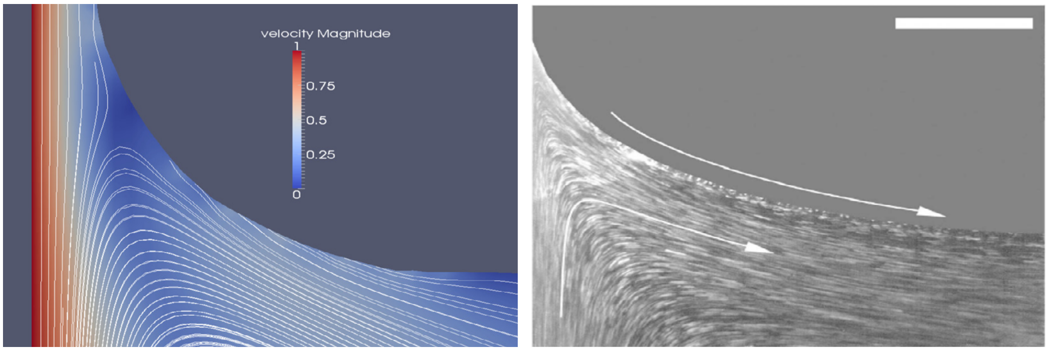


Fig. 26. Visualization of the motion and evolution of the normalized bulk concentration of ethanol in a thin layer of wine deposited on the wall of a glass and connected to a reservoir (the content of the glass itself). Time elapses from top left to bottom right. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

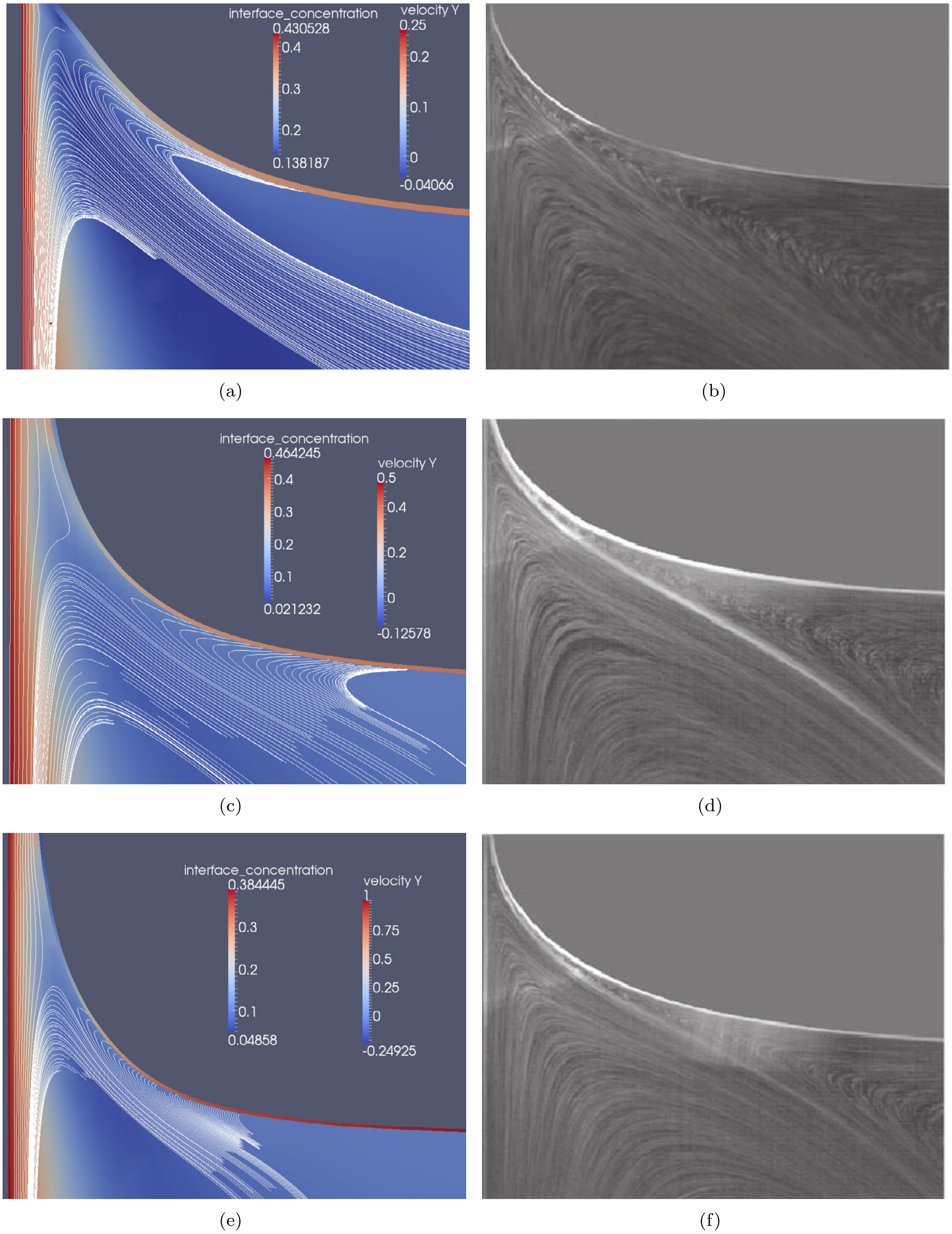


(a) (b)

Fig. 27. Visualization of the streamlines occurring during dip coating without surfactant. (a) Numerical simulations with $Ca = 5 \times 10^{-2}$. (b) Flow visualization established by Mayer and Krechetnikov in [21] at $Ca = 1.8 \times 10^{-3}$. Both visualizations clearly reveal the presence of an interfacial stagnation point. The values of velocity displayed are normalized. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

the reasons of this film thickening, but it revealed that the flow field thus identified allows Marangoni stresses to be the mechanism responsible for the film thickening.

Few numerical studies of this phenomenon have been undertaken. In [16] Krechetnikov and Homsy have conducted a study of the effects of surfactant on the film thickness in the Landau–Levich problem as well as the Bretherton problem, with a numerical algorithm based on a boundary-integral formulation. However, their simulations led to a film thinning instead of the thickening observed experimentally. More recently, in [5], Campana, Ubal, Giavedoni and Saita have solved the full hydrodynamic problem using the finite elements method on a domain tessellated into quadrangular elements and have found values of film thickening in good agreement with the experiments reported by Krechetnikov and Homsy.



(a) (b)

(c) (d)

(e) (f)

Fig. 28. Effect of increasing the capillary number on the shape of the streamlines in the context of dip coating. The first column presents the results of our simulations for different capillary numbers: (a) $Ca = 2.5 \times 10^{-2}$, (c) $Ca = 5 \times 10^{-2}$ and (e) $Ca = 10^{-1}$. The second column presents the visualizations obtained by Mayer and Krechetnikov in [21]: (b) $Ca = 5.9 \times 10^{-4}$, (d) $Ca = 1.2 \times 10^{-3}$ and (f) $Ca = 1.8 \times 10^{-3}$. The values of interface concentration and velocity displayed are normalized. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

It appears that two different phenomena could account for the film thickening observed experimentally in the Landau–Levich problem in the presence of surfactant. Equation (29) combined with the expression given above for l_c and Ca shows that the thickness of the film is proportional to $\gamma^{-\frac{1}{6}}$. By weakening the surface tension, the presence of surfactant have therefore the immediate effect of thickening the film. But another effect that must be taken into account is the Marangoni effect. As the surface of the fluid is stretched by the withdrawal of the wall at constant velocity, the surfactant concentration on the film becomes smaller. Therefore the surface tension gets bigger on the film portion of the interface than on the rest of the horizontal interface. In turn, this generates Marangoni forces that tend to advect the fluid near the interface from the reservoir to the film, creating very characteristic flow patterns, depending on the parameters of the problem. The experimental study undertaken in [21] reveals these patterns.

We have used our computational framework to simulate dip coating (using the same parameters for the surfactant as in Section 5.1) and we find that similar patterns occur. We also studied the qualitative evolution as a function of the parameters of the problem. In particular, while solving the Landau–Levich problem without any surfactant leads to a flow with an interfacial stagnation point (see Fig. 27), the presence of surfactant leads to more complex flow patterns, with the notable presence of an interior stagnation point. Fig. 28 presents the flow topologies for the Landau–Levich problem in the presence of surfactant for increasing capillary numbers, and compares them with the experimental visualization of [21]. We note that we do not consider the same Ca as in [21], as such computations would take about half a year on our workstation and we leave this exercise for when we develop a parallel approach. Nevertheless, we can observe that the fluid near the interface is advected upwards by Marangoni forces, and that both the fluid advected by the wall and the fluid advected by Marangoni forces near the interface are flowing back to the reservoir inside the meniscus; a phenomenon named in [21] the Separating Streamline (SSL). Our numerical results are consistent with the experimental visualizations of [21]. In particular we can observe in both numerical and experimental visualizations that as the capillary number increases, the SSL clearly approaches the shape of the interface and the upper recirculation region gets thinner and thinner, which could perhaps indicate a decrease of the significance of surface tension gradients for high capillary numbers.

6. Conclusion

We have introduced a numerical method for solving the governing equations of surfactant-driven flows in the case of both soluble and insoluble surfactant. Numerical examples in two and three spatial dimensions indicate that this approach can faithfully simulate the surface and bulk diffusion processes for the surfactant concentration. We then coupled these equations to a Navier–Stokes solver and showed that the method can enable the study of the effects of both soluble and insoluble surfactant on free surface flows. This methodology may be used to study the numerous interesting scientific questions, motivated by important industrial applications, that still have to be explored.

Acknowledgements

This research was supported by ONR N00014-11-1-0027 and ARO W911NF-16-1-0136.

References

[1] U.M. Ascher, S.J. Ruuth, B.T.R. Wetton, Implicit–explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (3) (1995) 797–823.
[2] T. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (2004) 349–355.
[3] A. Borhan, C.F. Mao, Effects of surfactants on the motion of drops through circular tubes, Phys. Fluids (1992).
[4] J. Brackbill, D. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[5] D.M. Campana, S. Ubal, M.D. Giavedoni, F.A. Saita, Numerical prediction of the film thickening due to surfactants in the Landau–Levich problem, Phys. Fluids (2010).
[6] P. Cermelli, E. Fried, M.E. Gurtin, Transport relations for surface integrals arising in the formulation of balance laws for evolving fluid interfaces, J. Fluid Mech. (2005).
[7] C.-H. Chang, E.I. Franses, Adsorption dynamics of surfactants at the air/water interface: a critical review of mathematical models, data, and mechanisms, Colloids Surf. A, Physicochem. Eng. Asp. 100 (1995) 1–45.
[8] A. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.
[9] R. Clift, J.R. Grace, M.E. Weber, Bubbles, Drops and Particles, Academic Press, 1978.
[10] M.A. Drumright-Clarke, Y. Renardy, The effect of insoluble surfactant at dilute concentration on drop breakup under shear with inertia, Phys. Fluids 16 (2004) 14–21.
[11] C.D. Eggleton, K.J. Stebe, An adsorption–desorption-controlled surfactant on a deforming droplet, J. Colloid Interface Sci. (1998).
[12] H. Ghahremani, A. Moradi, J. Abedini-Torghabeh, S.M. Hassani, Measuring surface tension of binary mixtures of water + alcohols from the diffraction pattern of surface ripples, Chem. Sin. (1992).
[13] A. Guittet, M. Theillard, F. Gibou, A stable projection method for the incompressible Navier–Stokes equations on arbitrary geometries and adaptive Quad/Octrees, J. Comput. Phys. 292 (2015) 215–238.
[14] Z. He, Z. Dagan, C. Maldarelli, The influence of surfactant adsorption on the motion of a fluid sphere in a tube, J. Fluid Mech. (1990).
[15] A. James, J. Lowengrub, A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant, J. Comput. Phys. 201 (2) (Dec. 2004) 685–722.
[16] R. Krechetnikov, G.M. Homsy, Surfactant effects in the Landau–Levich problem, J. Fluid Mech. (2005).
[17] R.J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.
[18] R.J. LeVeque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (3) (1997) 709–735.
[19] S.-Y. Lin, Y.-C. Lee, M.-W. Yang, H.-S. Liu, Surface equation of state of non-ionic C_mE_n surfactants, Langmuir (2008).
[20] F. Losasso, F. Gibou, R. Fedkiw, Simulating water and smoke with an octree data structure, ACM Trans. Graph. (2004) 457–462.
[21] H.C. Mayer, R. Krechetnikov, Landau–Levich flow visualization: revealing the flow topology responsible for the film thickening phenomena, Phys. Fluids (2012).
[22] W.J. Milliken, H.A. Stone, L.G. Leal, The effect of surfactants on the transient motion of Newtonian drops, Phys. Fluids (1992).

[23] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids, J. Comput. Phys. 219 (2) (Dec. 2006) 912–929.

[24] C. Min, F. Gibou, Geometric integration over irregular domains with application to level-set methods, J. Comput. Phys. 226 (2) (Oct. 2007) 1432–1443.

[25] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive Cartesian grids, J. Comput. Phys. 225 (1) (2007) 300–321.

[26] C. Min, F. Gibou, Robust second-order accurate discretizations of the multi-dimensional Heaviside and Dirac delta functions, J. Comput. Phys. 227 (22) (Nov. 2008) 9686–9695.

[27] M. Muradoglu, G. Tryggvason, A front-tracking method for computation of interfacial flows with soluble surfactants, J. Comput. Phys. 227 (4) (Feb. 2008) 2238–2262.

[28] M. Muradoglu, G. Tryggvason, Simulations of soluble surfactants in 3D multiphase flow, J. Comput. Phys. 274 (2014) 737–757.

[29] Y.T. Ng, C. Min, F. Gibou, An efficient fluid–solid coupling algorithm for single-phase flows, J. Comput. Phys. 228 (23) (Dec. 2009) 8807–8829.

[30] K.D. O'Hare, P.L. Spedding, J. Grimshaw, Evaporation rate of the ethanol and water components comprising a binary liquid mixture, Dev. Chem. Eng. Miner. Process. (1992).

[31] J. Papac, F. Gibou, C. Ratsch, Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions, J. Comput. Phys. 229 (2010) 875–889.

[32] J.W. Purvis, J.E. Burkhalter, Prediction of critical Mach number for store configurations, AIAA J. 17 (1979) 1170–1177.

[33] H.A. Stone, A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface, Phys. Fluids (1989).

[34] H.A. Stone, L.G. Leal, The effect of surfactants on drop deformation and breakup, J. Fluid Mech. (1989).

[35] J. Strain, Tree methods for moving interfaces, J. Comput. Phys. 151 (1999) 616–648.

[36] Y. Sui, H. Ding, P.D. Spelt, Numerical simulations of flows with moving contact lines, Annu. Rev. Fluid Mech. 46 (1) (2014) 97–119.

[37] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.

[38] K.E. Teigen, X. Li, J. Lowengrub, F. Wang, A. Voigt, A diffuse-interface approach for modelling transport, diffusion and adsorption/desorption of material quantities on a deformable interface, Commun. Math. Sci. (2009).

[39] M. Theillard, C.H. Rycroft, F. Gibou, A multigrid method on non-graded adaptive octree and quadtree cartesian grids, J. Sci. Comput. (2012), http://dx.doi.org/10.1007/s10915-012-9619-2.

[40] D. Wang, S.J. Ruuth, Variable step-size implicit–explicit linear multistep methods for time-dependent partial differential equations, J. Comput. Math. 26 (6) (2008) 838–855.

[41] J. Xu, Z. Li, J. Lowengrub, H. Zhao, A level-set method for interfacial flows with surfactant, J. Comput. Phys. 212 (2) (Mar. 2006) 590–616.