Fitted front tracking methods for two-phase incompressible Navier–Stokes flow: Eulerian and ALE finite element discretizations

Marco Agnese and Robert Nürnberg*
Department of Mathematics, Imperial College, London, SW7 2AZ, U.K.

Abstract

We investigate novel fitted finite element approximations for two-phase Navier–Stokes flow. In particular, we consider both Eulerian and Arbitrary Lagrangian–Eulerian (ALE) finite element formulations. The moving interface is approximated with the help of parametric piecewise linear finite element functions. The bulk mesh is fitted to the interface approximation, so that standard bulk finite element spaces can be used throughout. The meshes describing the discrete interface in general do not deteriorate in time, which means that in numerical simulations a smoothing or a remeshing of the interface mesh is not necessary. We present several numerical experiments, including convergence experiments and benchmark computations, for the introduced numerical methods, which demonstrate the accuracy and robustness of the proposed algorithms. We also compare the accuracy and efficiency of the Eulerian and ALE formulations.

Key words. incompressible two-phase flow, Navier–Stokes equations, ALE method, free boundary problem, surface tension, finite elements, front tracking

1 Introduction

Fluid flow problems with a moving interface are encountered in many applications in physics, engineering and biophysics. Typical applications include drops and bubbles, die swell, dam break, liquid storage tanks, ink-jet printing and fuel injection. For this reason, developing robust and efficient numerical methods for these flows is an important problem and has attracted tremendous interest over the last few decades.

*email: {m.agnese13,robert.nurnberg}@imperial.ac.uk

A crucial aspect of these types of fluid flow problems is that apart from the solution of the flow in the bulk domain, the position of the interface separating the two bulk phases also needs to be determined. At the interface certain boundary conditions need to be fulfilled, which specify the motion of the phase boundary. These conditions relate the variables of the bulk flow, velocity and pressure, across the two phase, taking into account external influences, such as for example surface tension. Numerically, in order to be able to compute the flow solution as well as the interface geometry, a measure to track the interface starting from an initial position needs to be incorporated. There are several strategies to deal with this problem, which can be divided into two categories depending on the viewpoint: interface capturing and interface tracking.

Interface capturing methods use an Eulerian description of the interface, which is defined implicitly. A characteristic scalar field, that is advected by the flow, is used to identify the two phases as well as the interface along the boundaries of the individual fluid domains. Depending on the method, this scalar field may be, for example, a discontinuous Heaviside function or a signed-distance function. The most important methods, which belong to this category, are the volume-of-fluid method, the level-set method and the phase-field method. In the volume-of-fluid method, the characteristic function of one of the phases is approximated numerically, see e.g. [38, 50, 49]. In the level-set method, the interface is given as the level set of a function, which has to be determined, see e.g. [54, 53, 47, 34, 35, 55]. Instead, the phase-field method works with diffuse interfaces, and therefore the transition layer between the phases has a finite size. There is no tracking mechanism for the interface, but the phase state is included implicitly in the governing equations. The interface is associated with a smooth, but highly localized variation of the so-called phase-field variable. Examples for phase-field methods applied to two-phase flow are [39, 5, 44, 15, 27, 22, 42, 1, 4, 36, 32]. Extensions of the method to multi-phase flows can be found in [24, 25, 16, 6]. The appeal of interface capturing approaches is the fact that they are usually easy to implement, and that they offer an automated way to deal with topological changes.

Interface tracking approaches, instead, use a Lagrangian description of the interface which is described explicitly. Here the interface is represented by a collection of particles or points, and this representation is transported by the bulk flow velocity. The great advantage of interface tracking approaches is that they offer an accurate and computationally efficient approximation of the evolving interface. Challenges are the mesh quality both of the interface representation and of the bulk triangulations, as well as the need to heuristically deal with topological changes. We refer e.g. to [57, 7, 56, 31, 10, 12] for further details, and to [43, 48] for the related immersed boundary method, which is used to simulate fluid-structure interactions using Eulerian coordinates for the fluid and Lagrangian coordinates for the structure.

In this paper we use a direct description of the interface using a parametrization of the unknown interface, similarly to the previous work [12]. In particular, our numerical method will be based on a piecewise linear parametric finite elements, and the description of the interface will be advected in normal direction with the normal part of the fluid velocity. The tangential degrees of freedom of the interface velocity are implicitly used to

ensure a good mesh quality, and this is a main feature of our proposed methods. But in contrast to [12], where an unfitted bulk finite element approximation was used, we will adopt a fitted mesh approach, which means that the discrete interface is composed of faces of elements from the bulk triangulation. Fitted and unfitted bulk mesh approaches are fundamentally different approximation methods, and need specialized implementation techniques. A fitted method has the advantage that discontinuity jumps at the interface are captured naturally, but it has the disadvantage that in a standard Eulerian method the velocity needs to be interpolated from an old mesh to a new mesh, unless the interface is stationary. Hence so-called Arbitrary Lagrangian Eulerian (ALE) methods are often proposed. Here the equations are posed in a moving domain framework, and an arbitrary Lagrangian velocity may be chosen to improve the quality of the bulk mesh. The original idea for ALE methods goes back to the papers [23, 40], and we refer to [46, 45, 28, 29, 31, 37, 30] for applications to two-phase Navier–Stokes flow.

In this paper we will propose both standard Eulerian and ALE finite element approximations for two-phase incompressible Navier–Stokes flow. One aim of our paper is to investigate numerically, if there is a clear advantage of ALE type methods over standard Eulerian methods. We stress that we are not aware of any detailed comparisons between fitted Eulerian and ALE front tracking methods in the literature. This paper aims to fill this gap. On the other hand, similar comparisons between ALE type interface tracking methods and various interface capturing methods have been presented in e.g. [41, 26]. We note that in the case of viscous incompressible two-phase Stokes flow, there is no need for the numerical method to interpolate the velocity from the old to the new mesh. Hence there is no need to employ an ALE method. In fact, in the case of two-phase Stokes flow, all our proposed numerical methods collapse the approximation considered by the authors in [3].

The remainder of the paper is organized as follows. In Section 2 we introduce the precise mathematical model of two-phase flow that we study, and formulate the weak formulations on which our Eulerian and ALE finite element approximations will be based. In Section 3 we introduce four different fully discrete finite element discretizations, and in Section 4 we discuss computational issues for their implementation as well as the employed solution methods. Finally, in Section 5 we present several numerical simulations for the introduced finite element approximations. In particular, we investigate their accuracy in convergence tests with the help of two exact solutions. Moreover, we compare the performance of our schemes in two well known benchmark problems.

2 Mathematical model and weak formulations

Let $\Omega \subset \mathbb{R}^d$, for $d = 2$ or $d = 3$ be a given domain that contains two different immiscible, incompressible fluids (liquid-liquid or liquid-gas) which for all $t \in [0, T]$ occupy time dependent regions $\Omega_+(t)$ and $\Omega_-(t) := \Omega \setminus \overline{\Omega}_+(t)$ and which are separated by an interface $(\Gamma(t))_{t\in[0,T]}$, $\Gamma(t) \subset \Omega$. We limit ourselves to interfaces formed by closed hypersurfaces, see Figure 1 for a pictorial representation in the case $d = 2$. For later use, we assume

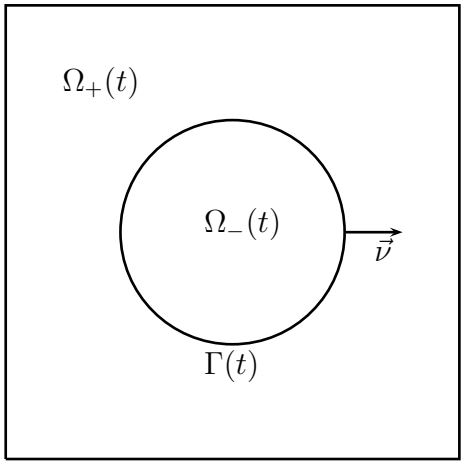


Figure 1: The domain Ω in the case $d = 2$.

that $(\Gamma(t))_{t\in[0,T]}$ is a sufficiently smooth evolving hypersurface without boundary that is parametrized by $\vec{x}(\cdot,t) : \Upsilon \to \mathbb{R}^d$, where $\Upsilon \subset \mathbb{R}^d$ is a given reference manifold such that $\Gamma(t) = \vec{x}(\Upsilon, t)$. Then

$$\vec{\mathcal{V}}(\vec{z},t) := \vec{x}_t(\vec{q},t) \quad \forall\ \vec{z} = \vec{x}(\vec{q},t) \in \Gamma(t)\,, \tag{2.1}$$

defines the velocity of $\Gamma(t)$, and $\vec{\mathcal{V}}\,.\,\vec{\nu}$ is the normal velocity of the evolving hypersurface $\Gamma(t)$, where $\vec{\nu}(t)$ is the unit normal on $\Gamma(t)$ pointing into $\Omega_+(t)$.

Denoting the velocity and pressure by $\vec{u}$ and p, respectively, we introduce the stress tensor

$$\underline{\underline{\sigma}} = \mu\,(\nabla\,\vec{u} + (\nabla\,\vec{u})^T) - p\,\underline{\underline{\mathrm{id}}} = 2\,\mu\,\underline{\underline{D}}(\vec{u}) - p\,\underline{\underline{\mathrm{id}}}\,, \tag{2.2}$$

where $\mu(t) = \mu_+\,\mathcal{X}_{\Omega_+(t)} + \mu_-\,\mathcal{X}_{\Omega_-(t)}$, with $\mu_\pm \in \mathbb{R}_{>0}$, denotes the dynamic viscosities in the two phases, $\underline{\underline{\mathrm{id}}} \in \mathbb{R}^{d\times d}$ is the identity matrix and $\underline{\underline{D}}(\vec{u}) := \frac{1}{2}\,(\nabla\,\vec{u} + (\nabla\,\vec{u})^T)$ is the rate-of-deformation tensor. Here and throughout, $\mathcal{X}_{\mathcal{D}}$ denotes the characteristic function for a set $\mathcal{D}$.

The fluid dynamics in the bulk domain Ω is governed by the two-phase Navier–Stokes model

$$\rho\,(\vec{u}_t + (\vec{u}\,.\,\nabla)\,\vec{u}) - 2\mu\,\nabla\,.\,\underline{\underline{D}}(\vec{u}) + \nabla\,p = \vec{f} \quad \text{in } \Omega_\pm(t)\,, \tag{2.3a}$$

$$\nabla\,.\,\vec{u} = 0 \quad \text{in } \Omega_\pm(t)\,, \tag{2.3b}$$

where $\rho(t) = \rho_+\,\mathcal{X}_{\Omega_+(t)} + \rho_-\,\mathcal{X}_{\Omega_-(t)}$, with $\rho_\pm \in \mathbb{R}_{>0}$, denotes the fluid density in the two phases, and where $\vec{f} := \rho\,\vec{f}_1 + \vec{f}_2$ is a possible forcing term. The velocity and stress tensor need to be coupled across the free surface $\Gamma(t)$. Viscosity of the fluids leads to the continuity condition

$$[\vec{u}]^+_- = \vec{0} \quad \text{on } \Gamma(t)\,, \tag{2.4}$$

where $[\vec{u}]^+_- := \vec{u}_+ - \vec{u}_-$ denote the jump in velocity across the interface $\Gamma(t)$. Here and throughout, we employ the shorthand notation $\vec{h}_\pm := \vec{h}\,|_{\Omega_\pm(t)}$ for a function $\vec{h} : \Omega\times[0,T] \to$

$\mathbb{R}^d$; and similarly for scalar and matrix-valued functions. The assumption that there is no change of phase leads to the dynamic interface condition

$$\vec{\mathcal{V}} \,.\, \vec{\nu} = \vec{u} \,.\, \vec{\nu} \quad \text{on } \Gamma(t)\,, \tag{2.5}$$

which means that the normal velocity of the interface needs to match the flow normal velocity across the interface $\Gamma(t)$. Moreover, the momentum conservation in a small material volume that intersects the interface leads to the stress balance condition

$$[\underline{\underline{\sigma}}\,\vec{\nu}]^+_- = -\nabla_s \,.\underline{\underline{\sigma}}_\Gamma \quad \text{on } \Gamma(t)\,, \tag{2.6}$$

where $\underline{\underline{\sigma}}_\Gamma$ is the interface stress tensor and $\nabla_s\,.$ is the surface divergence on $\Gamma(t)$. The operator ∇_s is the surface gradient on $\Gamma(t)$ and it is the orthogonal projection of the usual gradient operator ∇ on the tangent space of the surface $\Gamma(t)$. Therefore, given a smooth function h defined on a neighbourhood of $\Gamma(t)$, it is defined as

$$\nabla_s\, h(\vec{z}) = \underline{\underline{P}}(\vec{z})\, \nabla\, h(\vec{z}) = \nabla\, h(\vec{z}) \;-\; \nabla\, h(\vec{z})\,.\,\vec{\nu}(\vec{z})\,\vec{\nu}(\vec{z})\,, \quad \vec{z} \in \Gamma(t)\,, \tag{2.7}$$

with the usual projection operator $\underline{\underline{P}}$ defined as

$$\underline{\underline{P}} = \underline{\underline{\mathrm{id}}} \;-\; \vec{\nu}\,\vec{\nu}^{\,T} \quad \text{on } \Gamma(t)\,. \tag{2.8}$$

It can be shown that the definition of $\nabla_s\, h$ does not depend on the chosen extension of h, but only on the value of h on $\Gamma(t)$. See e.g. [20, §2.1] for details. For later use, we also define the surface Laplacian, also known as Laplace–Beltrami operator, Δ_s on $\Gamma(t)$ as

$$\Delta_s = \nabla_s \,.\, \nabla_s\,. \tag{2.9}$$

We restrict ourselves to the case that the only force acting on the interface is the surface tension contact force. Therefore the interface stress tensor $\underline{\underline{\sigma}}_\Gamma$ has the following constitutive equation

$$\underline{\underline{\sigma}}_\Gamma = \gamma\, \underline{\underline{P}}\,, \tag{2.10}$$

where γ is the surface tension coefficient. Substituting (2.10) in (2.6) we obtain, assuming γ is constant, the so called clean interface model for the interfacial forces

$$[\underline{\underline{\sigma}}\,\vec{\nu}]^+_- = -\gamma\,\varkappa\,\vec{\nu} \quad \text{on } \Gamma(t)\,, \tag{2.11}$$

which, in the case of Newtonian stress tensor (2.2), assumes the form

$$[2\mu\,\underline{\underline{D}}(\vec{u})\,.\,\vec{\nu} - p\,\vec{\nu}]^+_- = -\gamma\,\varkappa\,\vec{\nu} \quad \text{on } \Gamma(t)\,. \tag{2.12}$$

Here $\varkappa$ denotes the mean curvature of $\Gamma(t)$, i.e. the sum of the principal curvatures of $\Gamma(t)$, where we have adopted the sign convention that $\varkappa$ is negative where $\Omega_-(t)$ is locally convex. In particular, see e.g. [20], it holds that

$$\Delta_s\, \vec{\mathrm{id}} = \varkappa\,\vec{\nu} \qquad \text{on } \Gamma(t)\,. \tag{2.13}$$

In order to close the system, we prescribe the initial data $\Gamma(0) = \Gamma_0$ and the initial velocity $\vec{u}(0) = \vec{u}_0$. Finally, we impose the Dirichlet condition $\vec{u} = \vec{g}$ on $\partial_1\Omega$ and the free-slip condition $\vec{u}\,.\,\vec{\mathrm{n}} = 0$ and $\underline{\underline{\sigma}}\,\vec{\mathrm{n}}\,.\,\vec{\mathrm{t}} = 0$, $\forall\, \vec{\mathrm{t}} \in \{\vec{\mathrm{n}}\}^\perp$, on $\partial_2\Omega$, with $\vec{\mathrm{n}}$ denoting the outer unit

normal of $\partial\Omega$ and $\{\vec{\mathrm{n}}\}^\perp := \{\vec{\mathrm{t}} \in \mathbb{R}^d : \vec{\mathrm{t}} \,.\, \vec{\mathrm{n}} = 0\}$. As usual, it holds that $\partial\Omega = \partial_1\Omega \cup \partial_2\Omega$ and $\partial_1\Omega \cap \partial_2\Omega = \emptyset$. Therefore the total system of equations can be rewritten as follows:

$$\rho\,(\vec{u}_t + (\vec{u}\,.\,\nabla)\,\vec{u}) - 2\mu\,\nabla\,.\,\underline{\underline{D}}(\vec{u}) + \nabla\,p = \vec{f} \qquad \text{in } \Omega_\pm(t)\,, \tag{2.14a}$$

$$\nabla\,.\,\vec{u} = 0 \qquad \text{in } \Omega_\pm(t)\,, \tag{2.14b}$$

$$\vec{u} = \vec{g} \quad \text{on } \partial_1\Omega\,, \qquad \vec{u}\,.\,\vec{\mathrm{n}} = 0\,, \quad \underline{\sigma}\,\vec{\mathrm{n}}\,.\,\vec{\mathrm{t}} = 0 \quad \forall\ \vec{\mathrm{t}} \in \{\vec{\mathrm{n}}\}^\perp \qquad \text{on } \partial_2\Omega\,, \tag{2.14c}$$

$$[\vec{u}]^+_- = \vec{0} \qquad \text{on } \Gamma(t)\,, \tag{2.14d}$$

$$[2\mu\,\underline{\underline{D}}(\vec{u})\,.\,\vec{\nu} - p\,\vec{\nu}]^+_- = -\gamma\,\varkappa\,\vec{\nu} \qquad \text{on } \Gamma(t)\,, \tag{2.14e}$$

$$(\vec{\mathcal{V}} - \vec{u})\,.\,\vec{\nu} = 0 \qquad \text{on } \Gamma(t)\,, \tag{2.14f}$$

$$\Gamma(0) = \Gamma_0\,, \qquad \vec{u}(0) = \vec{u}_0\,. \tag{2.14g}$$

In what follows we introduce different weak formulations for (2.14a–g), on which our finite element approximations will be based.

2.1 Eulerian weak formulations

In order to obtain a weak formulation for (2.14a–g), we define the function spaces, for a given $\vec{b} \in [H^1(\Omega)]^d$,

$$\mathbb{U}(\vec{b}) := \{\vec{\phi} \in [H^1(\Omega)]^d : \vec{\phi} = \vec{b} \quad \text{on } \partial_1\Omega\,, \quad \vec{\phi}\,.\,\vec{\mathrm{n}} = 0 \quad \text{on } \partial_2\Omega\}\,,$$
$$\mathbb{P} := L^2(\Omega)\,, \qquad \widetilde{\mathbb{P}} := \{\eta \in \mathbb{P} : \int_\Omega \eta \,\mathrm{d}\mathcal{L}^d = 0\}\,,$$

and let, as usual, $(\cdot,\cdot)$ and $\langle\cdot,\cdot\rangle_{\Gamma(t)}$ denote the L^2–inner products on Ω and $\Gamma(t)$, respectively. In addition, we let $\mathcal{L}^d$ and $\mathcal{H}^{d-1}$ denote the Lebesgue measure in $\mathbb{R}^d$ and the $(d-1)$-dimensional Hausdorff measure, respectively.

We also need a weak form of the differential geometry identity (2.13), which can be obtained by multiplying (2.13) with a test function and performing integration by parts leading to

$$\langle \varkappa\,\vec{\nu}, \vec{\eta}\rangle_{\Gamma(t)} + \left\langle \nabla_s\,\mathrm{i}\vec{\mathrm{d}}, \nabla_s\,\vec{\eta}\right\rangle_{\Gamma(t)} = 0 \quad \forall\ \vec{\eta} \in [H^1(\Gamma(t))]^d\,. \tag{2.15}$$

Moreover, on noting (2.2) and (2.12), we have that

$$\begin{aligned}\int_{\Omega_+(t)\cup\Omega_-(t)} (\nabla\,.\,\underline{\sigma})\,.\,\vec{\xi}\,\mathrm{d}\mathcal{L}^d &= -\left(\underline{\sigma}, \nabla\,\vec{\xi}\right) - \left\langle [\underline{\sigma}\,\vec{\nu}]^+_-, \vec{\xi}\right\rangle_{\Gamma(t)} \\ &= -\left(2\,\mu\,\underline{\underline{D}}(\vec{u}) - p\,\underline{\underline{\mathrm{id}}}, \nabla\,\vec{\xi}\right) - \left\langle [2\mu\,\underline{\underline{D}}(\vec{u})\,.\,\vec{\nu} - p\,\vec{\nu}]^+_-, \vec{\xi}\right\rangle_{\Gamma(t)} \\ &= \left(p, \nabla\,.\,\vec{\xi}\right) - 2\left(\mu\,\underline{\underline{D}}(\vec{u}), \underline{\underline{D}}(\vec{\xi})\right) + \gamma\left\langle \varkappa\,\vec{\nu}, \vec{\xi}\right\rangle_{\Gamma(t)}\end{aligned} \tag{2.16}$$

for all $\vec{\xi} \in \mathbb{U}(\vec{0})$. Hence a standard weak formulation of (2.14a–g) is given as follows. Given $\Gamma(0) = \Gamma_0$ and $\vec{u}(\cdot,0) = \vec{u}_0$, for almost all $t \in (0,T)$ find $\Gamma(t)$ and $(\vec{u}, p, \varkappa)(t) \in$

$\mathbb{U}(\vec{g}) \times \widetilde{\mathbb{P}} \times H^1(\Gamma(t))$ such that

$$\left(\rho\,\vec{u}_t, \vec{\xi}\right) + \left(\rho\,(\vec{u}\,.\,\nabla)\,\vec{u}, \vec{\xi}\right) + 2\left(\mu\,\underline{\underline{D}}(\vec{u}), \underline{\underline{D}}(\vec{\xi})\right) - \left(p, \nabla\,.\,\vec{\xi}\right) - \gamma\,\left\langle \varkappa\,\vec{\nu}, \vec{\xi}\right\rangle_{\Gamma(t)} = \left(\vec{f}, \vec{\xi}\right) \quad \forall\ \vec{\xi} \in \mathbb{U}(\vec{0})\,, \tag{2.17a}$$

$$(\nabla\,.\,\vec{u}, \varphi) = 0 \quad \forall\ \varphi \in \widetilde{\mathbb{P}}\,, \tag{2.17b}$$

$$\left\langle \vec{\mathcal{V}} - \vec{u}, \chi\,\vec{\nu}\right\rangle_{\Gamma(t)} = 0 \quad \forall\ \chi \in H^1(\Gamma(t))\,, \tag{2.17c}$$

$$\langle \varkappa\,\vec{\nu}, \vec{\eta}\rangle_{\Gamma(t)} + \left\langle \nabla_s\,\vec{\mathrm{id}}, \nabla_s\,\vec{\eta}\right\rangle_{\Gamma(t)} = 0 \quad \forall\ \vec{\eta} \in [H^1(\Gamma(t))]^d \tag{2.17d}$$

holds for almost all times $t \in (0, T]$. We notice that if $p \in \mathbb{P}$ is part of a solution to (2.14a–g), then so is $p + c$ for an arbitrary $c \in \mathbb{R}$.

An alternative to the weak formulation (2.17a–d) can be obtained by following the paper [12]. These authors show that it is possible to derive an energy bound not only on the continuous level but also for a discretization based on this formulation. Due to the resulting structure of the equation, we will refer to this form as the antisymmetric weak formulation, and it is given as follows. Given $\Gamma(0) = \Gamma_0$ and $\vec{u}(\cdot, 0) = \vec{u}_0$, for almost all $t \in (0, T)$ find $\Gamma(t)$ and $(\vec{u}, p, \varkappa)(t) \in \mathbb{U}(\vec{g}) \times \widetilde{\mathbb{P}} \times H^1(\Gamma(t))$ such that

$$\begin{aligned} &\tfrac{1}{2}\left[\frac{\mathrm{d}}{\mathrm{d}t}(\rho\,\vec{u}, \vec{\xi}) + (\rho\,\vec{u}_t, \vec{\xi}) + (\rho, [(\vec{u}\,.\,\nabla)\,\vec{u}]\,.\,\vec{\xi} - [(\vec{u}\,.\,\nabla)\,\vec{\xi}]\,.\,\vec{u})\right] \\ &\qquad + 2\left(\mu\,\underline{\underline{D}}(\vec{u}), \underline{\underline{D}}(\vec{\xi})\right) - \left(p, \nabla\,.\,\vec{\xi}\right) - \gamma\,\left\langle \varkappa\,\vec{\nu}, \vec{\xi}\right\rangle_{\Gamma(t)} = \left(\vec{f}, \vec{\xi}\right) \quad \forall\ \vec{\xi} \in \mathbb{U}(\vec{0})\,, \end{aligned} \tag{2.18}$$

holds, together with (2.17b–d). For a derivation of (2.18) we refer to [12], see also [2].

2.2 ALE model and weak formulation

The Arbitrary Lagrangian Eulerian (ALE) technique consists in reformulating the time derivative $\vec{u}_t$ by expressing it with respect to a fixed reference configuration. A special homeomorphic map, called the ALE map, associates, at each time t, a point in the current computational domain $\Omega(t)$ to a point in the reference domain $\mathcal{D}$. Note that for convenience we consider the domain Ω to be time-dependent, even though in our situation it is fixed.

We now reformulate the two-phase Navier–Stokes system (2.14a–g), which is expressed in terms of Eulerian coordinates $\vec{z} \in \Omega(t)$, by rewriting the velocity time derivative $\vec{u}_t$ with respect to the so called ALE coordinate $\vec{q} \in \mathcal{D}$. Analogously to what is done in a front tracking approach to parametrize the interface $\Gamma(t)$, we can define the fixed reference manifold $\mathcal{D}$ and extend the map $\vec{x}(\cdot, t)$ to parametrize $\Omega(t)$ as $\Omega(t) = \vec{x}(\mathcal{D}, t)$. This extended map clearly still satisfies $\Gamma(t) = \vec{x}(\Upsilon, t)$, given that $\Upsilon \subset \mathcal{D}$. Moreover, we let $\partial\mathcal{D}$ be partitioned as $\partial\mathcal{D} = \partial_1\mathcal{D} \cup \partial_2\mathcal{D}$ with $\partial_1\mathcal{D} \cap \partial_2\mathcal{D} = \emptyset$.

Now, let $h : \Omega(t) \times [0, T] \to \mathbb{R}$ be a function defined on the Eulerian frame. The corresponding function on the ALE frame $\hat{h}$ is defined as

$$\hat{h} : \mathcal{D} \times [0, T] \to \mathbb{R}, \quad \hat{h}(\vec{q}, t) = h(\vec{x}(\vec{q}, t), t). \tag{2.19}$$

In order to compute the time derivative of (2.19) with respect to the ALE frame, using the chain rule, we have $\frac{\partial \hat{h}(\vec{q},t)}{\partial t} = \frac{\partial h(\vec{x}(\vec{q},t),t)}{\partial t} = \frac{\partial h(\vec{z},t)}{\partial t} + \vec{x}_t(\vec{q},t)\,.\,\nabla\, h(\vec{z},t)$. Therefore it holds that $\frac{\partial h(\vec{z},t)}{\partial t} = \frac{\partial \hat{h}(\vec{q},t)}{\partial t} - \vec{x}_t(\vec{q},t)\,.\,\nabla\, h(\vec{z},t)$. Finally, introducing the domain velocity

$$\vec{\mathcal{W}}(\vec{z},t) := \vec{x}_t(\vec{q},t) \quad \forall\ \vec{z} = \vec{x}(\vec{q},t) \in \Omega(t) \tag{2.20}$$

and the time derivative in the ALE frame

$$\left.\frac{\partial h(\vec{z},t)}{\partial t}\right|_{\mathcal{D}} := \frac{\partial \hat{h}(\vec{q},t)}{\partial t} \quad \forall\ \vec{z} = \vec{x}(\vec{q},t) \in \Omega(t)\,, \tag{2.21}$$

we obtain

$$h_t = h_t|_{\mathcal{D}} - \vec{\mathcal{W}}\,.\,\nabla\, h\,. \tag{2.22}$$

The identity (2.22) is naturally extended to vector valued functions. We stress the fact that the domain velocity $\vec{\mathcal{W}}$ for the interface points is consistent with the interface velocity $\vec{\mathcal{V}}$. Therefore it holds that $\vec{\mathcal{W}}\,|_{\Gamma(t)} = \vec{\mathcal{V}}$. We also point out that the ALE mapping is somehow arbitrary, apart from the requirement of conforming to the evolution of the domain boundary. Indeed the map of the boundary $\partial_i\mathcal{D}$ of the reference domain has then to provide, at all t, the boundary $\partial_i\Omega(t)$ of the current configuration, with $i = 1,\, 2$.

Using (2.22) in the momentum equation (2.14a), we can rewrite it as:

$$\rho\,(\,\vec{u}_t|_{\mathcal{D}} + ((\vec{u} - \vec{\mathcal{W}})\,.\,\nabla)\,\vec{u}) - 2\mu\,\nabla\,.\,\underline{\underline{D}}(\vec{u}) + \nabla\, p = \vec{f} \qquad \text{in } \Omega_{\pm}(t)\,. \tag{2.23}$$

We can notice that, with respect to the original formulation, there is a convective-type term due to the domain movement and the time derivative is computed in the fixed reference frame $\mathcal{D}$. Moreover, contrary to the original system, not only the regions $\Omega_{\pm}(t)$ are time dependent but also the whole domain $\Omega(t)$ is time dependent.

Obviously, if the domain is fixed, the additional convective term is zero and the time derivative in the ALE frame coincides with the usual time derivative in the Eulerian frame. This means that $\vec{\mathcal{W}} = \vec{0}$ corresponds to a pure Eulerian method, while $\vec{\mathcal{W}} = \vec{u}$ corresponds to a fully Lagrangian scheme. In our case, since we have a fixed domain Ω, the ALE formulation might not seem very useful. But, at the discrete level, we are concerned with the evolution of the discrete triangulated domains. Therefore the discrete ALE map describes the evolution of the grid during the domain movement. It is indeed at the discrete level that the advantage of the ALE formulation emerges, as in an ALE setting the time advancing scheme provides directly the evolution of the unknowns at mesh nodes and thus that of the degrees of freedom of the discrete solutions.

In order to define the ALE weak formulation, we need to use a different functional setting for the test functions with respect to the one used for the Eulerian weak formulations. The reason is that here it needs to be defined on the moving domain $\Omega(t)$. Therefore we introduce the admissible spaces of test functions on the reference domain $\mathcal{D}$

$$\begin{aligned} \mathcal{U}(\vec{b}) &:= \{\vec{\phi} \in [H^1(\mathcal{D})]^d : \vec{\phi} = \vec{b} \quad \text{on } \partial_1\mathcal{D}\,, \quad \vec{\phi}\,.\,\vec{\mathrm{n}} = 0 \quad \text{on } \partial_2\mathcal{D}\}\,, \\ \mathcal{P} &:= L^2(\mathcal{D})\,, \qquad \widetilde{\mathcal{P}} := \{\eta \in \mathcal{P} : \int_{\mathcal{D}} \eta\ \mathrm{d}\mathcal{L}^d = 0\}\,, \end{aligned}$$

for a given $\vec{b} \in [H^1(\mathcal{D})]^d$. Then, using the ALE mapping, we can define the admissible spaces of test functions on the moving domain $\Omega(t)$ by setting

$$\mathfrak{U}(\vec{b}) := \{\vec{\phi} : \bigcup_{t\in[0,T]} \Omega(t)\times\{t\} \to \mathbb{R}^d, \quad \vec{\phi} = \vec{\hat{\phi}} \circ \vec{x}^{-1}, \quad \vec{\hat{\phi}}(\cdot,t) \in \mathcal{U}(\vec{b}) \quad \forall\, t \in [0,T]\}\,,$$
$$\widetilde{\mathfrak{P}} := \{\eta : \bigcup_{t\in[0,T]} \Omega(t)\times\{t\} \to \mathbb{R}, \quad \eta = \hat{\eta} \circ \vec{x}^{-1}, \quad \hat{\eta}(\cdot,t) \in \widehat{\mathcal{P}} \quad \forall\, t \in [0,T]\}\,.$$

Hence, the ALE weak formulation of (2.23), (2.14b–g) is given as follows. Given $\Gamma(0) = \Gamma_0$ and $\vec{u}(\cdot,0) = \vec{u}_0$, find $(\Gamma(t))_{t\in[0,T]}$ and $(\vec{u}, p, \varkappa) \in \mathfrak{U}(\vec{g}) \times \widetilde{\mathfrak{P}} \times H^1(\bigcup_{t\in[0,T]} \Gamma(t)\times\{t\})$ such that

$$\begin{aligned}
&\left(\rho\ \vec{u}_t|_{\mathcal{D}}\,, \vec{\xi}\right)_{\Omega(t)} + \left(\rho\,((\vec{u}-\vec{\mathcal{W}})\,.\,\nabla)\,\vec{u}, \vec{\xi}\right)_{\Omega(t)} + 2\left(\mu\,\underline{\underline{D}}(\vec{u}), \underline{\underline{D}}(\vec{\xi})\right)_{\Omega(t)} \\
&\qquad - \left(p, \nabla\,.\,\vec{\xi}\right)_{\Omega(t)} - \gamma\,\left\langle \varkappa\,\vec{\nu}, \vec{\xi}\right\rangle_{\Gamma(t)} = \left(\vec{f}, \vec{\xi}\right)_{\Omega(t)} \quad \forall\ \vec{\xi} \in \mathfrak{U}(\vec{0})\,, \qquad (2.24a)\\
&(\nabla\,.\,\vec{u}, \varphi)_{\Omega(t)} = 0 \quad \forall\ \varphi \in \widetilde{\mathfrak{P}}\,, \qquad (2.24b)
\end{aligned}$$

and (2.17c,d) hold for almost all times $t \in (0,T]$. Here $(\cdot,\cdot)_{\Omega(t)}$ denotes the L^2–inner products on $\Omega(t)$.

3 Finite element approximations

3.1 Eulerian finite element approximations

We consider the partitioning $t_m = m\,\tau$, $m = 0,\ldots,M$, of $[0,T]$ into uniform time steps $\tau = \frac{T}{M}$, see [12]. Moreover, let $\mathcal{T}^m$, $\forall\ m \geq 0$, be a regular partitioning of the domain Ω into disjoint open simplices o^m_j, $j = 1,\ldots,J^m_\Omega$. From now on, the domain Ω which we consider is the polyhedral domain defined by the triangulation $\mathcal{T}^m$. On $\mathcal{T}^m$ we define the finite element spaces

$$S^m_k := \{\chi \in C(\overline{\Omega}) : \chi|_{o^m} \in \mathcal{P}_k(o^m)\ \forall\ o^m \in \mathcal{T}^m\}\,, \quad k \in \mathbb{N}\,,$$

where $\mathcal{P}_k(o^m)$ denotes the space of polynomials of degree k on o^m. Moreover, S^m_0 is the space of piecewise constant functions on $\mathcal{T}^m$ and let $\vec{I}^m_k$ be the standard interpolation operator onto $[S^m_k]^d$.

Let $\mathbb{U}^m(\vec{g}) \subset \mathbb{U}(\vec{I}^m_k\vec{g})$ and $\mathbb{P}^m \subset \mathbb{P}$ be the finite element spaces we use for the approximation of velocity and pressure, and let $\widetilde{\mathbb{P}}^m := \mathbb{P}^m \cap \widetilde{\mathbb{P}}$. In this paper, we restrict ourselves to the choice P2–(P1+P0), i.e. we set $\mathbb{U}^m(\vec{0}) = [S^m_2]^d \cap \mathbb{U}(\vec{0})$ and $\mathbb{P}^m = S^m_1 + S^m_0$, but generalizations to other spaces are easily possible.

We consider a fitted finite element approximation for the evolution of the interface $\Gamma(t)$. Let $\Gamma^m \subset \mathbb{R}^d$ be a $(d-1)$-dimensional polyhedral surface approximating the closed surface $\Gamma(t_m)$, $m = 0,\ldots,M$. Let Ω^m_+ denote the exterior of Γ^m and let Ω^m_- be the interior

of Γ^m, where we assume that Γ^m has no self-intersections. Then $\Omega = \Omega^m_- \cup \Gamma^m \cup \Omega^m_+$, and the fitted nature of our method implies that

$$\overline{\Omega^m_+} = \bigcup_{o \in \mathcal{T}^m_+} \overline{o} \quad \text{and} \quad \overline{\Omega^m_-} = \bigcup_{o \in \mathcal{T}^m_-} \overline{o}\,,$$

where $\mathcal{T}^m = \mathcal{T}^m_+ \cup \mathcal{T}^m_-$ and $\mathcal{T}^m_+ \cap \mathcal{T}^m_- = \emptyset$. Let $\vec{\nu}^m$ denote the piecewise constant unit normal to Γ^m such that $\vec{\nu}^m$ points into Ω^m_+. In addition, let $\mu^m = \mu_+ \, \mathcal{X}_{\Omega^m_+} + \mu_- \, \mathcal{X}_{\Omega^m_-} \in S^m_0$ and $\rho^m = \rho_+ \, \mathcal{X}_{\Omega^m_+} + \rho_- \, \mathcal{X}_{\Omega^m_-} \in S^m_0$.

In order to define the parametric finite element spaces on Γ^m, we assume that $\Gamma^m = \bigcup_{j=1}^{J_\Gamma} \overline{\sigma^m_j}$, where $\{\sigma^m_j\}_{j=1}^{J_\Gamma}$ is a family of mutually disjoint open $(d-1)$-simplices with vertices $\{\vec{q}^m_k\}_{k=1}^{K_\Gamma}$. Then we define $\underline{V}(\Gamma^m) := \{\vec{\chi} \in [C(\Gamma^m)]^d : \vec{\chi}\,|_{\sigma^m_j} \in \mathcal{P}_1(\sigma^m_j), j = 1, \ldots, J_\Gamma\} =: [W(\Gamma^m)]^d$, where $W(\Gamma^m) \subset H^1(\Gamma^m)$ is the space of scalar continuous piecewise linear functions on Γ^m, with $\{\chi^m_k\}_{k=1}^{K_\Gamma}$ denoting the standard basis of $W(\Gamma^m)$. As usual, we parametrize the new surface Γ^{m+1} over Γ^m using a parametrization $\vec{X}^{m+1} \in \underline{V}(\Gamma^m)$, so that $\Gamma^{m+1} = \vec{X}^{m+1}(\Gamma^m)$. Moreover, let $\langle \cdot, \cdot \rangle^h_{\Gamma^m}$ be the mass lumped inner product on Γ^m defined as

$$\langle v, w \rangle^h_{\Gamma^m} := \tfrac{1}{d} \sum_{j=1}^{J_\Gamma} \mathcal{H}^{d-1}(\sigma^m_j) \sum_{k=1}^{d} (v\,w)((\vec{q}^m_{j_k})^-), \tag{3.1}$$

where $\{\vec{q}^m_{j_k}\}_{k=1}^d$ are the vertices of σ^m_j, and where we define the limit $v((\vec{q}^m_{j_k})^-) := \lim_{\sigma^m_j \ni \vec{p} \to \vec{q}^m_{j_k}} v(\vec{p})$. We naturally extend (3.1) to vector valued functions. Finally, let $\langle \cdot, \cdot \rangle_{\Gamma^m}$ denote the standard L^2–inner product on Γ^m.

Our fully discrete finite element approximations based on the standard weak formulation (2.17a–d) are given as follows. Here we consider an explicit and an implicit treatment of the convective term in the Navier–Stokes equation.

($\mathcal{A}^h_{\text{ex}}$): Let Γ^0, an approximation to $\Gamma(0)$, and $\vec{U}^0 \in \mathbb{U}^0(\vec{g})$ be given. For $m = 0, \ldots, M-1$, find $(\vec{U}^{m+1}, P^{m+1}, \vec{X}^{m+1}, \kappa^{m+1}) \in \mathbb{U}^m(\vec{g}) \times \widetilde{\mathbb{P}}^m \times \underline{V}(\Gamma^m) \times W(\Gamma^m)$ such that

$$\left(\rho^m \, \frac{\vec{U}^{m+1} - \vec{I}^m_2 \, \vec{U}^m}{\tau}, \vec{\xi}\right) + \left(\rho^m (\vec{I}^m_2 \vec{U}^m \,.\, \nabla)\, \vec{U}^{m+1}\,, \vec{\xi}\right) + 2\left(\mu^m \, \underline{\underline{D}}(\vec{U}^{m+1}), \underline{\underline{D}}(\vec{\xi})\right)$$
$$- \left(P^{m+1}, \nabla \,.\, \vec{\xi}\right) - \gamma \left\langle \kappa^{m+1} \, \vec{\nu}^m, \vec{\xi} \right\rangle_{\Gamma^m} = \left(\rho^m \, \vec{f}^{m+1}_1 + \vec{f}^{m+1}_2, \vec{\xi}\right) \quad \forall\, \vec{\xi} \in \mathbb{U}^m(\vec{0})\,, \tag{3.2a}$$

$$\left(\nabla \,.\, \vec{U}^{m+1}, \varphi\right) = 0 \quad \forall\, \varphi \in \widetilde{\mathbb{P}}^m\,, \tag{3.2b}$$

$$\left\langle \frac{\vec{X}^{m+1} - \vec{\mathrm{id}}}{\tau}, \chi \, \vec{\nu}^m \right\rangle^h_{\Gamma^m} - \left\langle \vec{U}^{m+1}, \chi \, \vec{\nu}^m \right\rangle_{\Gamma^m} = 0 \quad \forall\, \chi \in W(\Gamma^m)\,, \tag{3.2c}$$

$$\left\langle \kappa^{m+1} \, \vec{\nu}^m, \vec{\eta} \right\rangle^h_{\Gamma^m} + \left\langle \nabla_s \vec{X}^{m+1}, \nabla_s \, \vec{\eta} \right\rangle_{\Gamma^m} = 0 \quad \forall\, \vec{\eta} \in \underline{V}(\Gamma^m) \tag{3.2d}$$

and set $\Gamma^{m+1} = \vec{X}^{m+1}(\Gamma^m)$. Here we have defined $\vec{f}^{m+1}_i(\cdot) := \vec{I}^m_2 \, \vec{f}_i(\cdot, t_{m+1})$, $i = 1, 2$.

($\mathcal{A}^h_{\text{im}}$): Let Γ^0, an approximation to $\Gamma(0)$, and $\vec{U}^0 \in \mathbb{U}^0(\vec{g})$ be given. For $m = 0, \ldots, M-1$, find $(\vec{U}^{m+1}, P^{m+1}, \vec{X}^{m+1}, \kappa^{m+1}) \in \mathbb{U}^m(\vec{g}) \times \widetilde{\mathbb{P}}^m \times \underline{V}(\Gamma^m) \times W(\Gamma^m)$ such

that

$$\left(\rho^m\,\frac{\vec{U}^{m+1}-\vec{I}_2^m\,\vec{U}^m}{\tau},\vec{\xi}\right)+\left(\rho^m(\vec{U}^{m+1}\,.\,\nabla)\,\vec{U}^{m+1}\,,\,\vec{\xi}\right)+2\left(\mu^m\,\underline{\underline{D}}(\vec{U}^{m+1}),\underline{\underline{D}}(\vec{\xi})\right)$$
$$-\left(P^{m+1},\nabla\,.\,\vec{\xi}\right)-\gamma\left\langle\kappa^{m+1}\,\vec{\nu}^m,\vec{\xi}\right\rangle_{\Gamma^m}=\left(\rho^m\,\vec{f}_1^{\,m+1}+\vec{f}_2^{\,m+1},\vec{\xi}\right)\quad\forall\,\vec{\xi}\in\mathbb{U}^m(\vec{0})\,,\quad(3.3)$$

and (3.2b–d), and set $\Gamma^{m+1}=\vec{X}^{m+1}(\Gamma^m)$. Since the convective term in (3.2a) is treated explicitly, the scheme $(\mathcal{A}^h_{\rm ex})$ is a linear scheme that leads to a coupled linear system of equations for the unknowns $(\vec{U}^{m+1},P^{m+1},\vec{X}^{m+1},\kappa^{m+1})$ at each time level. On the other hand, in (3.3) the convective term is treated implicitly, and so $(\mathcal{A}^h_{\rm im})$ is a nonlinear scheme that leads to a coupled nonlinear system of equations for the unknowns $(\vec{U}^{m+1},P^{m+1},\vec{X}^{m+1},\kappa^{m+1})$ at each time level.

An alternative fully discrete finite element approximation is based on the formulation (2.18), (2.17b–d). In particular, we consider the following scheme.

$(\mathcal{B}^h)$: Let Γ^0, an approximation to $\Gamma(0)$, and $\vec{U}^0\in\mathbb{U}^0(\vec{g})$ be given. For $m=0,\ldots,M-1$, find $(\vec{U}^{m+1},P^{m+1},\vec{X}^{m+1},\kappa^{m+1})\in\mathbb{U}^m(\vec{g})\times\widetilde{\mathbb{P}}^m\times\underline{V}(\Gamma^m)\times W(\Gamma^m)$ such that

$$\frac{1}{\tau}\left(\tfrac12(I_0^m\,\rho^{m-1}+\rho^m)\,\vec{U}^{m+1}-I_0^m\,\rho^{m-1}\vec{I}_2^m\,\vec{U}^m,\vec{\xi}\right)+\tfrac12\left((\rho^m\,\vec{I}_2^m\,\vec{U}^m\,.\,\nabla)\,\vec{U}^{m+1}\,,\,\vec{\xi}\right)$$
$$-\tfrac12\left((\rho^m\,\vec{I}_2^m\,\vec{U}^m\,.\,\nabla)\,\vec{\xi}\,,\,\vec{U}^{m+1}\right)+2\left(\mu^m\,\underline{\underline{D}}(\vec{U}^{m+1}),\underline{\underline{D}}(\vec{\xi})\right)-\left(P^{m+1},\nabla\,.\,\vec{\xi}\right)$$
$$-\gamma\left\langle\kappa^{m+1}\,\vec{\nu}^m,\vec{\xi}\right\rangle_{\Gamma^m}=\left(\rho^m\,\vec{f}_1^{\,m+1}+\vec{f}_2^{\,m+1},\vec{\xi}\right)\quad\forall\,\vec{\xi}\in\mathbb{U}^m(\vec{0})\,,\qquad(3.4)$$

and (3.2b–d), and set $\Gamma^{m+1}=\vec{X}^{m+1}(\Gamma^m)$. We note that the scheme $(\mathcal{B}^h)$ corresponds to [12, (4.6a–d)] in the setting of an unfitted finite element approximation. Similarly to [12, Theorem 4.1] it is then straightforward to show that there exists a unique solution to $(\mathcal{B}^h)$ that fulfils an energy inequality. In the unfitted case this energy inequality leads to an overall stability result if the bulk mesh does not change. Of course, in the fitted case considered here this will never be the case, unless the interface is stationary. Nevertheless, it is of interest to see how the scheme $(\mathcal{B}^h)$ performs in practice, compared to $(\mathcal{A}^h_{\rm ex})$ and $(\mathcal{A}^h_{\rm im})$.

Moreover, we remark that all three schemes, $(\mathcal{A}^h_{\rm ex})$, $(\mathcal{A}^h_{\rm im})$ and $(\mathcal{B}^h)$, share the equations (3.2d). Apart from defining the discrete curvature, the equations (3.2d) can also be viewed as a side constraint on the distribution of mesh points, see [8, 9] for further details. In particular, they lead to an implicit tangential motion of the vertices on the discrete interface, so that the vertices of Γ^m will in general be very well distributed in practice. For example, in the case $d=2$ it is possible to prove an asymptotic equidistribution property, see [2, Theorem 5].

Finally, we stress that the three schemes $(\mathcal{A}^h_{\rm ex})$, $(\mathcal{A}^h_{\rm im})$ and $(\mathcal{B}^h)$, all feature at least one term containing the interpolated velocity $\vec{I}_2^m\,\vec{U}^m$, and so an interpolation of the old velocity $\vec{U}^m$ onto the new mesh $\mathcal{T}^m$ is necessary, see also §4.1.4 below. In order to avoid such an interpolation, we consider an ALE formulation next.

3.2 ALE finite element approximation

In order to derive our fully discrete ALE finite element approximation, we first introduce a semidiscrete continuous-in-time fitted finite element approximation of (2.24a–d). Let $\mathfrak{T}^h$ be a regular partitioning of the reference domain $\mathcal{D}$ into disjoint open simplices o^h_j, $j = 1, \ldots, J^h_{\mathcal{D}}$. From now on, the reference domain $\mathcal{D}$, which we consider, is the polyhedral domain defined by the triangulation $\mathfrak{T}^h$. On $\mathfrak{T}^h$ we define the finite element spaces

$$S^h_k(\mathcal{D}) := \{\chi \in C(\overline{\mathcal{D}}) : \chi|_{o^h} \in \mathcal{P}_k(o^h) \;\forall\; o^h \in \mathfrak{T}^h\}\,, \quad k \in \mathbb{N}\,,$$

as well as $S^h_0(\mathcal{D})$, the space of piecewise constant functions on $\mathfrak{T}^h$.

Then, using the discrete ALE mapping $\vec{x}_h$, we can define the finite element spaces on the discretized moving domains $\Omega^h(t) = \vec{x}_h(\mathcal{D}, t)$ by

$$S^h_k(\Omega^h(t)) := \{\eta : \Omega^h(t) \to \mathbb{R}, \quad \eta = \hat{\eta} \circ \vec{x}_h^{-1}, \quad \hat{\eta} \in S^h_k(\mathcal{D})\}\,, \quad k \in \mathbb{N} \cup \{0\}\,.$$

Here $\vec{x}_h \in [S^h_1]^d$ maps each simplex with straight faces in $\mathfrak{T}^h$ to a simplex with straight faces belonging to the triangulation $\mathcal{T}^h(t)$ of the discretized moving domain $\Omega^h(t)$. For later use, we let $\mathfrak{U}^h(\vec{g}, t) \subset [S^h_2(\Omega^h(t))]^d$ and $\widetilde{\mathfrak{P}}^h(t) \subset S^h_1(\Omega^h(t)) + S^h_0(\Omega^h(t))$ denote the appropriate spaces for the discrete velocities and pressure. In addition, we also define the discrete domain velocity as

$$\vec{\mathcal{W}}^h(\vec{z}, t) := \sum_{k=1}^{K_\Omega} \left[\frac{\mathrm{d}}{\mathrm{d}t}\, \vec{p}^{\,h}_k(t)\right] \varphi^h_k(\vec{z}, t) \in [S^h_1(\Omega^h(t))]^d\,, \tag{3.5}$$

with $\{\varphi^h_k(\cdot, t)\}_{k=1}^{K_\Omega}$ denoting the standard basis of $S^h_1(\Omega^h(t))$, and $\{\vec{p}^{\,h}_k(t)\}_{k=1}^{K_\Omega}$ denoting the vertices of $\mathcal{T}^h(t)$.

Given $(\Gamma^h(t))_{t\in[0,T]}$, a family of $(d-1)$-dimensional polyhedral surfaces with unit normal $\vec{\nu}^h(t)$ pointing into $\Omega^h_+(t)$, the exterior of $\Gamma^h(t)$, analogously to the fully discrete case discussed in the previous section, we introduce the piecewise linear finite element spaces $W(\Gamma^h(t))$ and $\underline{V}(\Gamma^h(t))$, with $\{\chi^h_k(\cdot, t)\}_{k=1}^{K_\Gamma}$ denoting the standard basis of the former. Hence $\chi^h_k(\vec{q}^{\,h}_l(t), t) = \delta_{kl}$ for all $k, l \in \{1, \ldots, K_\Gamma\}$ and $t \in [0, T]$, where $\{\vec{q}^{\,h}_k(t)\}_{k=1}^{K_\Gamma}$ are the vertices of $\Gamma^h(t)$. We also notice that the discrete interface velocity $\vec{\mathcal{V}}^h$ defined as

$$\vec{\mathcal{V}}^h(\vec{z}, t) := \sum_{k=1}^{K_\Gamma} \left[\frac{\mathrm{d}}{\mathrm{d}t}\, \vec{q}^{\,h}_k(t)\right] \chi^h_k(\vec{z}, t) \in \underline{V}(\Gamma^h(t))\,, \tag{3.6}$$

see e.g. [11, (3.3)], is simply the trace of $\vec{\mathcal{W}}^h$ on Γ^h. We also introduce the interior $\Omega^h_-(t) = \Omega(t) \setminus \overline{\Omega^h_+(t)}$, and the L^2–inner products $(\cdot,\cdot)_{\Omega^h(t)}$ and $\langle\cdot,\cdot\rangle_{\Gamma^h(t)}$ on $\Omega^h(t)$ and $\Gamma^h(t)$, respectively. Moreover, let $\langle\cdot,\cdot\rangle^h_{\Gamma^h(t)}$ be the mass lumped inner product on $\Gamma^h(t)$. Finally, we let $\vec{f}^{\,h}_i(\cdot, t) := \vec{I}^h_2\, \vec{f}_i(\cdot, t)$, $i = 1, 2$, where $\vec{I}^h_2$ denotes the standard interpolation operator onto $[S^h_2(\Omega^h(t))]^d$, and set $\mu^h(t) = \mu_+\, \mathcal{X}_{\Omega^h_+(t)} + \mu_-\, \mathcal{X}_{\Omega^h_-(t)} \in S^h_0(\Omega^h(t))$ and $\rho^h(t) = \rho_+\, \mathcal{X}_{\Omega^h_+(t)} + \rho_-\, \mathcal{X}_{\Omega^h_-(t)} \in S^h_0(\Omega^h(t))$.

Then, given $\Gamma^h(0)$ and $\vec{U}^h(0) \in \mathfrak{U}^h(\vec{g},0)$, for $t \in (0,T]$ find $\Gamma^h(t)$ and $(\vec{U}^h, P^h, \vec{\mathcal{V}}^h, \kappa^h)(t) \in \mathfrak{U}^h(\vec{g},t) \times \widetilde{\mathfrak{P}}^h(t) \times \underline{V}(\Gamma^h(t)) \times W(\Gamma^h(t))$ such that

$$\begin{aligned}
&\Big(\rho^h\ \vec{U}^h_t\big|_{\mathcal{D}}, \vec{\xi}\Big)_{\Omega^h(t)} + \Big(\rho^h((\vec{U}^h - \vec{\mathcal{W}}^h)\,.\,\nabla)\,\vec{U}^h\,,\,\vec{\xi}\Big)_{\Omega^h(t)} + 2\,\Big(\mu^h\,\underline{\underline{D}}(\vec{U}^h), \underline{\underline{D}}(\vec{\xi})\Big)_{\Omega^h(t)} \\
&\qquad - \Big(P^h, \nabla\,.\,\vec{\xi}\Big)_{\Omega^h(t)} - \gamma\,\Big\langle \kappa^h\,\vec{\nu}^h, \vec{\xi}\Big\rangle_{\Gamma^h(t)} = \Big(\rho^h\,\vec{f}^h_1 + \vec{f}^h_2, \vec{\xi}\Big)_{\Omega^h(t)} \quad \forall\ \vec{\xi} \in \mathfrak{U}^h(\vec{0},t)\,, \quad (3.7a)\\
&\Big(\nabla\,.\,\vec{U}^h, \varphi\Big)_{\Omega^h(t)} = 0 \quad \forall\ \varphi \in \widetilde{\mathfrak{P}}^h(t)\,, \quad (3.7b)\\
&\Big\langle \vec{\mathcal{V}}^h, \chi\,\vec{\nu}^h\Big\rangle^h_{\Gamma^h(t)} - \Big\langle \vec{U}^h, \chi\,\vec{\nu}^h\Big\rangle_{\Gamma^h(t)} = 0 \quad \forall\ \chi \in W(\Gamma^h(t))\,, \quad (3.7c)\\
&\Big\langle \kappa^h\,\vec{\nu}^h, \vec{\eta}\Big\rangle^h_{\Gamma^h(t)} + \Big\langle \nabla_s\,\vec{\mathrm{id}}, \nabla_s\,\vec{\eta}\Big\rangle_{\Gamma^h(t)} = 0 \quad \forall\ \vec{\eta} \in \underline{V}(\Gamma^h(t))\,. \quad (3.7d)
\end{aligned}$$

The semidiscrete continuous-in-time fitted finite element scheme (3.7a–d) is a system of ODEs. Indeed, on defining the appropriate time-dependent matrices and vectors, we can rewrite (3.7a–d) in algebraic form as

$$\begin{aligned}
&\vec{M}^h_\Omega(t)\frac{\mathrm{d}}{\mathrm{d}t}\vec{U} + \vec{B}^h_\Omega(t)\vec{U} + \vec{D}^h_\Omega\Big(t, \vec{U}^h, \vec{\mathcal{W}}^h\Big)\vec{U} + \vec{C}^h_\Omega(t)P - \gamma\vec{N}^h_{\Gamma,\Omega}(t)\kappa = \vec{c}^{\,h}(t, \vec{f}^h_1, \vec{f}^h_2)\,, \quad (3.8a)\\
&[\vec{C}^h_\Omega(t)]^T\vec{U} = 0\,, \quad (3.8b)\\
&[\vec{N}^h_\Gamma(t)]^T\frac{\mathrm{d}}{\mathrm{d}t}\vec{X} - \vec{N}^h_{\Gamma,\Omega}(t)\vec{U} = 0\,, \quad (3.8c)\\
&\vec{N}^h_\Gamma(t)\kappa + \vec{A}^h_\Gamma(t)\vec{X} = 0\,, \quad (3.8d)
\end{aligned}$$

where $\vec{U}$, P, κ and $\vec{X}$ are the unknown vectors for the velocity, pressure, interface curvature and interface position, respectively. See [2, (6.15)] for details.

It is now straightforward to obtain a fully discrete finite element approximation of (3.8a–d), since the only missing part is the temporal discretization. On recalling the partitioning $t_m = m\,\tau$, $m = 0, \ldots, M$, of $[0,T]$, and on applying a semi-implicit Euler scheme to (3.8a–d), we obtain

$$\begin{aligned}
&\frac{1}{\tau}\vec{M}^h_\Omega(t^{m+1})\vec{U}^{m+1} + \vec{B}^h_\Omega(t^m)\vec{U}^{m+1} + \vec{D}^h_\Omega\Big(t^m, \vec{U}^{m+1}, \vec{\mathcal{W}}^{m+1}\Big)\vec{U}^{m+1} + \vec{C}^h_\Omega(t^m)P^{m+1}\\
&\qquad - \gamma\vec{N}^h_{\Gamma,\Omega}(t^m)\kappa^{m+1} = \frac{1}{\tau}\vec{M}^h_\Omega(t^{m+1})\vec{U}^m + \vec{c}^{\,h}(t^m, \vec{f}^{\,m+1}_1, \vec{f}^{\,m+1}_2)\,, \quad (3.9a)\\
&[\vec{C}^h_\Omega(t^m)]^T\vec{U}^{m+1} = 0\,, \quad (3.9b)\\
&\frac{1}{\tau}[\vec{N}^h_\Gamma(t^m)]^T\vec{X}^{m+1} - \vec{N}^h_{\Gamma,\Omega}(t^m)\vec{U}^{m+1} = \frac{1}{\tau}[\vec{N}^h_\Gamma(t^m)]^T\vec{X}^m\,, \quad (3.9c)\\
&\vec{N}^h_\Gamma(t^m)\kappa^{m+1} + \vec{A}^h_\Gamma(t^m)\vec{X}^{m+1} = 0\,, \quad (3.9d)
\end{aligned}$$

where $\vec{\mathcal{W}}^m$, $m = 1, \ldots, M$ are some fully discrete bulk mesh velocities which, starting from Ω^0, induce a sequence of bulk meshes $\Omega^m = \vec{x}^m(\mathcal{D})$, $m = 1, \ldots, M$, where $\vec{x}^m \in [S^h_1(\Omega^m)]^d$ is an approximation to $\vec{x}^h(\cdot, t_m)$.

Following [45], we now rewrite (3.9a–d) into a form that is common in the finite element literature. Firstly, let $\mathfrak{U}^m(\vec{g}) = \{\vec{\phi} : \Omega^m \to \mathbb{R}^d,\ \vec{\phi} = \vec{\hat{\phi}} \circ (\vec{x}^m)^{-1},\ \vec{\hat{\phi}} \in \mathcal{U}(\vec{g}) \cap [S^h_2(\mathcal{D})]^d\}$

and $\widetilde{\mathfrak{P}}^m = \{\eta : \Omega^m \to \mathbb{R},\ \eta = \hat{\eta} \circ (\vec{x}^m)^{-1},\ \hat{\eta} \in \widetilde{\mathcal{P}} \cap (S_1^h(\mathcal{D}) + S_0^h(\mathcal{D}))\}$. Now to ease the notation, we introduce the following convention. For a function h^k defined on Ω^k, we can immediately transport it to any other configuration Ω^m, with $k \neq m$, using the mapping $\vec{x}^k \circ (\vec{x}^m)^{-1}$. Hence, when we need to integrate h^k on Ω^m we will write simply $\int_{\Omega^m} h^k \,\mathrm{d}\mathcal{L}^d$ instead of $\int_{\Omega^m} h^k \circ \vec{x}^k \circ (\vec{x}^m)^{-1} \,\mathrm{d}\mathcal{L}^d$, and we will adopt the same notation used for the inner products. Then, our ALE finite element approximation (3.9a–d), which is based on the variational formulation (2.24a–d), can be written as follows.

($\mathcal{C}_{\mathrm{ALE}}^h$): Let Ω^0, Γ^0, and $\vec{U}^0 \in \mathfrak{U}^0(\vec{g})$ be given. For $m = 0, \ldots, M-1$, find $(\vec{U}^{m+1}, P^{m+1}, \vec{X}^{m+1}, \kappa^{m+1}, \vec{\mathcal{W}}^{m+1}) \in \mathfrak{U}^{m+1}(\vec{g}) \times \widetilde{\mathfrak{P}}^{m+1} \times \underline{V}(\Gamma^m) \times W(\Gamma^m) \times [S_1^h(\Omega^m)]^d$ such that $\vec{\mathcal{W}}^{m+1}|_{\Gamma^m} = \frac{1}{\tau}(\vec{X}^{m+1} - \vec{\mathrm{id}}|_{\Gamma^m})$ and such that

$$\begin{aligned}
&\Big(\frac{\rho^m}{\tau}\,\vec{U}^{m+1}, \vec{\xi}\Big)_{\Omega^{m+1}} + \Big(\rho^m((\vec{U}^{m+1} - \vec{\mathcal{W}}^{m+1})\,.\,\nabla)\,\vec{U}^{m+1}\,,\,\vec{\xi}\Big)_{\Omega^m} \\
&\qquad + 2\,\Big(\mu^m\,\underline{\underline{D}}(\vec{U}^{m+1}), \underline{\underline{D}}(\vec{\xi})\Big)_{\Omega^m} - \Big(P^{m+1}, \nabla\,.\,\vec{\xi}\Big)_{\Omega^m} - \gamma\,\Big\langle \kappa^{m+1}\,\vec{\nu}^m, \vec{\xi}\Big\rangle_{\Gamma^m} \\
&\qquad = \Big(\frac{\rho^m}{\tau}\,\vec{U}^m, \vec{\xi}\Big)_{\Omega^{m+1}} + \Big(\rho^m\,\vec{f}_1^{\,m+1} + \vec{f}_2^{\,m+1}, \vec{\xi}\Big)_{\Omega^m} \quad \forall\ \vec{\xi} \in \mathfrak{U}^m(\vec{0})\,, \qquad (3.10a) \\
&\Big(\nabla\,.\,\vec{U}^{m+1}, \varphi\Big)_{\Omega^m} = 0 \quad \forall\ \varphi \in \widetilde{\mathfrak{P}}^m\,, \qquad (3.10b)
\end{aligned}$$

and (3.2c,d), and set $\Gamma^{m+1} = \vec{X}^{m+1}(\Gamma^m)$, as well as $\Omega^{m+1} = \vec{x}^{m+1}(\mathcal{D})$, where $\vec{x}^{m+1} = \vec{x}^m + \tau\,\vec{\mathcal{W}}^{m+1}$. Here we note that in practice the bulk mesh velocity $\vec{\mathcal{W}}^{m+1}$ will be obtained with the help of an iterative solution and smoothing procedure, see the next section for more details. Moreover, we recall that the side constraint (3.2d) will lead to a uniform distribution of the vertices on Γ^m in the case $d = 2$, and to well distributed vertices in $d = 3$.

We stress that since the ALE formulation describes the evolution of the solution along trajectories, the velocity $\vec{U}^m$ is defined on all the domains, and so in particular on Ω^m and Ω^{m+1}. Hence no interpolation of the velocity is necessary in the ALE formulation, unless a full bulk remeshing is performed because of severe mesh deformations. Finally we remark that at the discrete level, the domain Ω^m often plays the role of the reference manifold, so that $\mathcal{D}$ is never used in practice.

4 Solution methods

4.1 Eulerian schemes

In this subsection we discuss the solution methods and other implementation issues for the three Eulerian schemes ($\mathcal{A}_{\mathrm{ex}}^h$), ($\mathcal{A}_{\mathrm{im}}^h$) and ($\mathcal{B}^h$).

4.1.1 Algebraic formulations

As is standard practice for the solution of linear systems arising from discretizations of Navier–Stokes equations, we avoid the complications of the constrained pressure space $\widetilde{\mathbb{P}}^m$ in practice by considering an overdetermined linear system with $\mathbb{P}^m$ instead. For example, for the scheme $(\mathcal{A}^h_{\mathrm{ex}})$ we consider the system (3.2a–d) with $\widetilde{\mathbb{P}}^m$ replaced by $\mathbb{P}^m$. For non-zero Dirichlet boundary data $\vec{g}$ this requires a correction term on the right hand side of (3.2b). In particular, we replace (3.2b) with

$$\left(\nabla \,.\, \vec{U}^{m+1}, \varphi\right) = \frac{(\varphi, 1)}{\mathcal{L}^d(\Omega)} \int_{\partial_1\Omega} (\vec{I}_2^m \, \vec{g}) \,.\, \vec{\mathrm{n}} \, \mathrm{d}\mathcal{H}^{d-1} \quad \forall\, \varphi \in \mathbb{P}^m \,. \tag{4.1}$$

The linear system of equations arising at each time step of the scheme $(\mathcal{A}^h_{\mathrm{ex}})$ then can be formulated as: Find $(\vec{U}^{m+1}, P^{m+1}, \kappa^{m+1}, \delta\vec{X}^{m+1})$, where $\vec{X}^{m+1} = \vec{X}^m + \delta\vec{X}^{m+1}$, such that

$$\begin{pmatrix} \vec{B}_\Omega & \vec{C}_\Omega & -\gamma\, \vec{N}_{\Gamma,\Omega} & 0 \\ \vec{C}_\Omega^T & 0 & 0 & 0 \\ \vec{N}_{\Gamma,\Omega}^T & 0 & 0 & -\frac{1}{\tau}\, \vec{N}_\Gamma^T \\ 0 & 0 & \vec{N}_\Gamma & \vec{A}_\Gamma \end{pmatrix} \begin{pmatrix} \vec{U}^{m+1} \\ P^{m+1} \\ \kappa^{m+1} \\ \delta\vec{X}^{m+1} \end{pmatrix} = \begin{pmatrix} \vec{c} \\ \vec{\beta} \\ 0 \\ -\vec{A}_\Gamma\, \vec{X}^m \end{pmatrix}, \tag{4.2}$$

where $(\vec{U}^{m+1}, P^{m+1}, \kappa^{m+1}, \delta\vec{X}^{m+1})$ here, in a slight abuse of notation, denote the coefficients of these finite element functions with respect to some chosen basis functions. Moreover, $\vec{X}^m$ denotes the coefficients of $\vec{\mathrm{id}}\,|_{\Gamma^m}$ with respect to the basis of $\underline{V}(\Gamma^m)$. The definitions of the matrices and vectors in (4.2) directly follow from (3.2a,c,d), (4.1), and their exact definitions can be found in e.g. [2, §5.5]. Note that for the submatrices we have used the convention that the subscripts refer to the test and trial domains, respectively. A single subscript is used where the two domains are the same. Of course, the algebraic formulation of the linear scheme $(\mathcal{B}^h)$ can also be stated in the form (4.2).

In order to solve the nonlinear scheme $(\mathcal{A}^h_{\mathrm{im}})$, we employ the following fixed point iteration. Let Γ^m and $\vec{U}^m \in \mathbb{U}^{m-1}(\vec{g})$ be given. Let $\vec{U}^{m+1,0} = \vec{I}_2^m\, \vec{U}^m$ and fix $\epsilon_f > 0$. Then, for $s \geq 0$, find $(\vec{U}^{m+1,s+1}, P^{m+1,s+1}, \vec{X}^{m+1,s+1}, \kappa^{m+1,s+1}) \in \mathbb{U}^m(\vec{g}) \times \mathbb{P}^m \times \underline{V}(\Gamma^m) \times W(\Gamma^m)$ such that

$$\begin{aligned} &\left(\rho^m\, \frac{\vec{U}^{m+1,s+1} - \vec{I}_2^m\, \vec{U}^m}{\tau}, \vec{\xi}\right) + \left((\rho^m\, \vec{U}^{m+1,s} \,.\, \nabla)\, \vec{U}^{m+1,s+1}\,,\, \vec{\xi}\right) + 2\left(\mu^m\, \underline{\underline{D}}(\vec{U}^{m+1,s+1}), \underline{\underline{D}}(\vec{\xi})\right) \\ &\quad - \left(P^{m+1,s+1}, \nabla \,.\, \vec{\xi}\right) - \gamma\, \left\langle \kappa^{m+1,s+1}\, \vec{\nu}^m, \vec{\xi} \right\rangle_{\Gamma^m} = \left(\rho^m\, \vec{f}_1^{m+1} + \vec{f}_2^{m+1}, \vec{\xi}\right) \quad \forall\, \vec{\xi} \in \mathbb{U}^m(\vec{0})\,, \end{aligned} \tag{4.3a}$$

$$\left(\nabla \,.\, \vec{U}^{m+1,s+1}, \varphi\right) = \frac{(\varphi, 1)}{\mathcal{L}^d(\Omega)} \int_{\partial_1\Omega} (\vec{I}_2^m\, \vec{g}) \,.\, \vec{\mathrm{n}}\, \mathrm{d}\mathcal{H}^{d-1} \quad \forall\, \varphi \in \mathbb{P}^m\,, \tag{4.3b}$$

$$\left\langle \frac{\vec{X}^{m+1,s+1} - \vec{\mathrm{id}}}{\tau}, \chi\, \vec{\nu}^m \right\rangle^h_{\Gamma^m} - \left\langle \vec{U}^{m+1,s+1}, \chi\, \vec{\nu}^m \right\rangle_{\Gamma^m} = 0 \quad \forall\, \chi \in W(\Gamma^m)\,, \tag{4.3c}$$

$$\left\langle \kappa^{m+1,s+1}\, \vec{\nu}^m, \vec{\eta} \right\rangle^h_{\Gamma^m} + \left\langle \nabla_s\, \vec{X}^{m+1,s+1}, \nabla_s\, \vec{\eta} \right\rangle_{\Gamma^m} = 0 \quad \forall\, \vec{\eta} \in \underline{V}(\Gamma^m)\,, \tag{4.3d}$$

and repeat the iteration until $\|\vec{U}^{m+1,s+1} - \vec{U}^{m+1,s}\|_{L^\infty} \leq \epsilon_f$. Finally, set $\Gamma^{m+1} = \vec{X}^{m+1,s+1}(\Gamma^m)$ and $(\vec{U}^{m+1}, P^{m+1}, \kappa^{m+1}) = (\vec{U}^{m+1,s+1}, P^{m+1,s+1}, \kappa^{m+1,s+1})$. Clearly, (4.3a–d) is a linear system for the unknowns $(\vec{U}^{m+1,s+1}, P^{m+1,s+1}, \vec{X}^{m+1,s+1}, \kappa^{m+1,s+1})$ that is of the form (4.2).

Remark 1. *At times we will be interested in numerical simulations for non divergence-free solutions, in particular when considering our convergence experiments in §5.1. Here the condition* (2.3b) *is replaced by* $\nabla \,.\, \vec{u} = f_{\text{div}}$ *in* $\Omega_\pm(t)$*. Clearly, this leads to an additional term on the right-hand side of the discrete continuity equation* (3.2b)*. In particular, we have to replace* (4.1) *with*

$$\Big(\nabla \,.\, \vec{U}^{m+1}, \varphi\Big) = \left(f_{\text{div}}, \varphi - \frac{(\varphi,1)}{(1,1)} \right) + \frac{(\varphi,1)}{\mathcal{L}^d(\Omega)} \int_{\partial_1\Omega} (\vec{I}_2^m\, \vec{g}) \,.\, \vec{n}\, \mathrm{d}\mathcal{H}^{d-1} \quad \forall\, \varphi \in \mathbb{P}^m\,. \tag{4.4}$$

The same adjustment is made on the right-hand side of (4.3b)*. In addition, we observe that the weak formulation* (2.18) *needs to be suitably adjusted, with the additional term* $\frac{1}{2}\,(\rho\, f_{\text{div}}\, \vec{u}, \vec{\xi})$ *on the right-hand side of* (2.18)*, see [2] for details. As a consequence, the term* $\frac{1}{2}\,(\rho^m\, f_{\text{div}}\, \vec{U}^m, \vec{\xi})$ *needs to be added to the right-hand side of* (3.4) *for the scheme* $(\mathcal{B}^h)$*.*

4.1.2 Solvers for the linear (sub)problems

Following the authors' approach in [3], see also [12, 10], for the solution of (4.2) we use a Schur complement approach that eliminates $(\kappa^{m+1}, \delta\vec{X}^{m+1})$ from (4.2), and then use an iterative solver for the remaining system in $(\vec{U}^{m+1}, P^{m+1})$. This approach has the advantage that for the reduced system well-known solution methods for finite element discretizations for standard Navier–Stokes discretizations may be employed. The desired Schur complement can be obtained as follows. Let

$$\Xi_\Gamma := \begin{pmatrix} 0 & -\frac{1}{\tau}\vec{N}_\Gamma^T \\ \vec{N}_\Gamma & \vec{A}_\Gamma \end{pmatrix}, \tag{4.5}$$

be the interface condensed operator. Then (4.2) can be reduced to

$$\begin{pmatrix} \vec{B}_\Omega + \gamma\,(\vec{N}_{\Gamma,\Omega}\ 0)\,\Xi_\Gamma^{-1} \begin{pmatrix} \vec{N}_{\Gamma,\Omega}^T \\ 0 \end{pmatrix} & \vec{C}_\Omega \\ \vec{C}_\Omega^T & 0 \end{pmatrix} \begin{pmatrix} \vec{U}^{m+1} \\ P^{m+1} \end{pmatrix} = \begin{pmatrix} \vec{c} + \gamma\,(\vec{N}_{\Gamma,\Omega}\ 0)\,\Xi_\Gamma^{-1} \begin{pmatrix} 0 \\ -\vec{A}_\Gamma\, \vec{X}^m \end{pmatrix} \\ \vec{\beta} \end{pmatrix} \tag{4.6a}$$

and

$$\begin{pmatrix} \kappa^{m+1} \\ \delta\vec{X}^{m+1} \end{pmatrix} = \Xi_\Gamma^{-1} \begin{pmatrix} -\vec{N}_{\Gamma,\Omega}^T\, \vec{U}^{m+1} \\ -\vec{A}_\Gamma\, \vec{X}^m \end{pmatrix}. \tag{4.6b}$$

We notice that when $\gamma = 0$ the linear system (4.6a) becomes

$$\begin{pmatrix} \vec{B}_\Omega & \vec{C}_\Omega \\ \vec{C}_\Omega^T & 0 \end{pmatrix} \begin{pmatrix} \vec{U}^{m+1} \\ P^{m+1} \end{pmatrix} = \begin{pmatrix} \vec{c} \\ \vec{\beta} \end{pmatrix}, \tag{4.7}$$

which for $\mu_+ = \mu_-$ and $\rho_+ = \rho_-$ corresponds to a discretization of the one-phase Navier–Stokes problem. This system is well known and a classical way to solve it is to use a preconditioned GMRES iteration. Therefore, we solve (4.6a), which is a slight modification of (4.7), employing a preconditioned GMRES iterative solver. For the inverse Ξ_Γ^{-1} we employ a sparse LU decomposition, which we obtain with the help of the sparse factorization package UMFPACK, see [18], in order to reuse the factorization. Having obtained $(\vec{U}^{m+1}, P^{m+1})$ from (4.6a), we solve (4.6b) for $(\kappa^{m+1}, \delta\vec{X}^{m+1})$.

As preconditioner for (4.6a) we employ the standard Navier–Stokes matrix

$$\begin{pmatrix} \vec{B}_\Omega & \vec{C}_\Omega \\ \vec{C}_\Omega^T & 0 \end{pmatrix}, \tag{4.8}$$

recall (4.7). Of course, as the pressure in the Navier–Stokes problem (4.7) is only unique up to a constant, the block matrix (4.8) is singular. Hence, in order to efficiently apply the preconditioner (4.8) as part of an iterative solution method for (4.6a), one can either use a sparse QR factorization method as provided by SPQR, see [19]. Alternatively, the matrix (4.8) can be suitably doctored to make it invertible, and then a sparse LU factorization can be obtained with the help of UMFPACK, see [18], as usual. In practice we pursued the latter approach, as it proved to be slightly more efficient. We refer to [2, §4.6] for more details.

4.1.3 Mesh generation and smoothing

Given the initial polyhedral surface Γ^0, we create a triangulation $\mathcal{T}^0$ of Ω that is fitted to Γ^0 with the help of the package `Gmsh`, see [33]. The mesh generation process is controlled by characteristic length parameter, which describes the fineness of the resulting mesh. Then, for $m \geq 0$, having computed the new interface Γ^{m+1}, we would like to obtain a bulk triangulation $\mathcal{T}^{m+1}$ that is fitted to Γ^{m+1}, and ideally is close to $\mathcal{T}^m$. This is to avoid unnecessary overhead from remeshing the domain Ω completely. To this end, we perform the following smoothing step on $\mathcal{T}^m$, which is inspired by the method proposed in [29], see also [31]. Having obtained $\delta\vec{X}^{m+1}$ from the solution of (4.2), we solve the linear elasticity problem: Find a displacement $\vec{\psi} \in [H^1(\Omega)]^d$ such that

$$\nabla \cdot (2\,\underline{\underline{D}}(\vec{\psi}) + (\nabla \cdot \vec{\psi})\,\underline{\underline{\mathrm{id}}}) = \vec{0} \quad \text{in } \Omega_\pm^m\,, \qquad \vec{\psi} = \delta\vec{X} \quad \text{on } \Gamma^m\,, \qquad \vec{\psi} \cdot \vec{\mathrm{n}} = 0 \quad \text{on } \partial\Omega\,. \tag{4.9}$$

In practice we approximate (4.9) with piecewise linear elements and solve the resulting system of linear equations with the UMPFACK package, see [18]. The obtained discrete variant of $\vec{\psi}$, at every vertex of the current bulk grid $\mathcal{T}^m$, then represents the variation in their position that we compute in order to obtain $\mathcal{T}^{m+1}$.

Occasionally the deformation of the mesh becomes too large, and so a complete remeshing of Ω becomes necessary. In order to detect the need for a complete remeshing, we define the angle criterion

$$\min_{o \in \mathcal{T}^{m+1}} \min_{\alpha \in \measuredangle(o)} \alpha \leq C_a\,, \tag{4.10}$$

where C_a is a fixed constant. Moreover, $\measuredangle(o)$ is the set of all the angles of the simplex o, which can be computed as $\alpha_{ij} = \cos^{-1}(-\vec{n}_i \,.\, \vec{n}_j)$, $\forall\ i, j \in \{0, \ldots, d\}$, with $\vec{n}_i$ the unitary normal to the simplex face i. In practice we perform a full remeshing if the criterion (4.10) holds with $C_a = 20°$. We stress that in all our numerical simulations we never need to remesh the interface Γ^m itself.

4.1.4 Velocity interpolation

We notice that in the three Eulerian schemes $(\mathcal{A}^h_{\text{ex}})$, $(\mathcal{A}^h_{\text{im}})$ and $(\mathcal{B}^h)$, all feature at least one term containing the interpolated velocity $\vec{I}^m_2 \vec{U}^m$. Here the interpolation operator is needed because the velocity $\vec{U}^m$, which has been computed at time level $m-1$, is defined on $\mathcal{T}^{m-1}$. Therefore, at the next time level m, the previous velocity $\vec{U}^m$ needs to be interpolated on $\mathcal{T}^m$, which will in general be different from $\mathcal{T}^{m-1}$, in order to be used.

The interpolation routine simply consists in evaluating the discrete function $\vec{U}^m$ in all the degrees of freedom of a piecewise quadratic function defined on $\mathcal{T}^m$. The efficiency of this computation hinges on quickly finding the element in the bulk triangulation $\mathcal{T}^{m-1}$ that contains the coordinates of the degree of freedom on $\mathcal{T}^m$ that is currently of interest. Of course, a naive linear looping over all elements in $\mathcal{T}^{m-1}$ is highly inefficient. A simple optimization to this naive search consists in starting a guided search from the last correctly found element, in the hope that the degrees of freedom on $\mathcal{T}^m$ are traversed in such a way, the successive points are close to each other. If the element does not contain the new point, then the guided search moves to the neighbouring element that is opposite the vertex with the largest distance to the new point. For a convex domain, this search is always well-defined. For nonconvex domains this guided search can be combined with a naive linear search whenever a boundary face is encountered in the local search for the next neighbour.

The last ingredient to our velocity interpolation algorithm is the traversal of the degrees of freedom on $\mathcal{T}^m$. Ideally we would like to have a continuous path of elements, which visits, only once, all the elements of $\mathcal{T}^m$. This ensures that successive degrees of freedom on the new mesh are spatially close, and so the local guided search for the next element of $\mathcal{T}^{m-1}$ to visit is likely to be very short. Unfortunately, since we use unstructured meshes, such a path, in general, is difficult to construct. Hence, instead of explicitly constructing such a path, we create a fictitious background lattice $\delta\,\mathbb{Z}^d \cap \overline{\Omega}$ that covers the domain $\overline{\Omega}$. Then each element of $\mathcal{T}^m$ is assigned to the point on the lattice that is closest to its barycentre. Depending on the size of the lattice, δ, multiple entities can be mapped to the same point on the lattice. Since the lattice is structured, it is straightforward to build a continuous path which visits all its points. For example, for a rectangular domain in 2d we can use a snaking path that traverses the lattice alternating left-to-right and right-to-left, see Figure 2. Now, in practice, the elements of $\mathcal{T}^m$ are visited following the given lattice path. This means that, for each lattice point traversed, the associated entities on $\mathcal{T}^m$ are visited. Clearly, for the algorithm to be efficient, the lattice size δ should be neither too small nor too large. In practice we use δ proportional to the characteristic

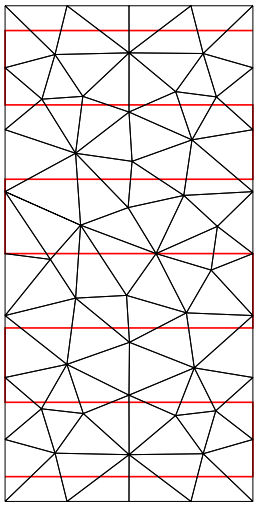


Figure 2: Continuous path (red) traversing a fictitious background lattice.

length of the bulk mesh.

4.2 ALE scheme

The solution technique used to solve the ALE scheme ($\mathcal{C}^h_{\mathrm{ALE}}$) is very similar to the one used to solve the nonlinear scheme ($\mathcal{A}^h_{\mathrm{im}}$), recall (4.3a–d), in that we use a fixed point iteration. However, the main difference is that at each time step for the ALE scheme we also find a bulk mesh velocity $\vec{\mathcal{W}}^{m+1}$, and hence a new bulk mesh Ω^{m+1}, as part of the solution process. More precisely, we find a solution for the scheme ($\mathcal{C}^h_{\mathrm{ALE}}$) as follows. Let Γ^m and $\vec{U}^m \in \mathfrak{U}^m(\vec{g})$ be given. Set $\vec{U}^{m+1,0} = \vec{U}^m$, $\vec{\psi}^{m+1,0} = \vec{0}$, $\Omega^{m+1,0} = \Omega^m$ and fix $\epsilon_f > 0$. Then, for $s \geq 0$, find $(\vec{U}^{m+1,s+1}, P^{m+1,s+1}, \vec{X}^{m+1,s+1}, \kappa^{m+1,s+1}) \in \mathfrak{U}^m(\vec{g}) \times \mathfrak{P}^m \times \underline{V}(\Gamma^m) \times W(\Gamma^m)$ such that

$$\begin{aligned}
&\left(\frac{\rho^m}{\tau}\,\vec{U}^{m+1,s+1}, \vec{\xi}\right)_{\Omega^{m+1,s}} + \left(\left(\rho^m \left(\vec{U}^{m+1,s} - \frac{\vec{\psi}^{m+1,s}}{\tau}\right) . \nabla\right) \vec{U}^{m+1,s+1}, \vec{\xi}\right)_{\Omega^m} \\
&\qquad + 2\left(\mu^m\, \underline{\underline{D}}(\vec{U}^{m+1,s+1}), \underline{\underline{D}}(\vec{\xi})\right)_{\Omega^m} - \left(P^{m+1,s+1}, \nabla . \vec{\xi}\right)_{\Omega^m} - \gamma \left\langle \kappa^{m+1,s+1}\, \vec{\nu}^m, \vec{\xi}\right\rangle_{\Gamma^m} \\
&\qquad = \left(\frac{\rho^m}{\tau}\,\vec{U}^m, \vec{\xi}\right)_{\Omega^{m+1,s}} + \left(\rho^m\, \vec{f}_1^{\,m+1} + \vec{f}_2^{\,m+1}, \vec{\xi}\right)_{\Omega^m} \quad \forall\ \vec{\xi} \in \mathfrak{U}^m(\vec{0})\,, \qquad (4.11a)
\end{aligned}$$

$$\left(\nabla . \vec{U}^{m+1,s+1}, \varphi\right)_{\Omega^m} = \frac{(\varphi, 1)_{\Omega^m}}{\mathcal{L}^d(\Omega)} \int_{\partial_1\Omega^m} (\vec{I}_2^{\,m}\, \vec{g}) . \vec{\mathrm{n}}\, \mathrm{d}\mathcal{H}^{d-1} \quad \forall\ \varphi \in \mathfrak{P}^m\,, \qquad (4.11b)$$

$$\left\langle \frac{\vec{X}^{m+1,s+1} - \vec{\mathrm{id}}}{\tau}, \chi\, \vec{\nu}^m \right\rangle^h_{\Gamma^m} - \left\langle \vec{U}^{m+1,s+1}, \chi\, \vec{\nu}^m \right\rangle_{\Gamma^m} = 0 \quad \forall\ \chi \in W(\Gamma^m)\,, \qquad (4.11c)$$

$$\left\langle \kappa^{m+1,s+1}\, \vec{\nu}^m, \vec{\eta} \right\rangle^h_{\Gamma^m} + \left\langle \nabla_s\, \vec{X}^{m+1,s+1}, \nabla_s\, \vec{\eta} \right\rangle_{\Gamma^m} = 0 \quad \forall\ \vec{\eta} \in \underline{V}(\Gamma^m)\,. \qquad (4.11d)$$

Having obtained $\vec{X}^{m+1,s+1}$, the new bulk displacement $\vec{\psi}^{m+1,s+1}$ is computed via the linear elasticity problem (4.9), see §4.1.3, and the new bulk mesh $\Omega^{m+1,s+1}$ is updated accordingly. This iteration is repeated until $\|U^{m+1,s+1} - U^{m+1,s}\|_{L^\infty} \leq \epsilon_f$ and $\|\vec{\psi}^{m+1,s+1} - \vec{\psi}^{m+1,s}\|_{L^\infty} \leq \epsilon_f$. Finally, set $\Gamma^{m+1} = \vec{X}^{m+1,s+1}(\Gamma^m)$, $\Omega^{m+1,s+1} = \Omega^{m+1}$ and $(\vec{U}^{m+1}, P^{m+1}, \kappa^{m+1}, \vec{\mathcal{W}}^{m+1}) = (\vec{U}^{m+1,s+1}, P^{m+1,s+1}, \kappa^{m+1,s+1}, \frac{1}{\tau}\,\psi^{m+1,s+1})$. Clearly, (4.11a–d) is once again a linear system of equations for the unknowns $(\vec{U}^{m+1,s+1}, P^{m+1,s+1}, \vec{X}^{m+1,s+1}, \kappa^{m+1,s+1})$ of the form (4.2).

Similarly to the Eulerian schemes, we monitor the mesh quality of the bulk meshes Ω^{m+1} with the help of the criterion (4.10). In practice, if (4.10) holds with $C_a = 20°$, then we perform a full remeshing of the domain Ω, keeping Γ^{m+1} fixed, and obtain the discrete velocity approximation on the new mesh via the procedure described in §4.1.4.

5 Numerical results

We implemented our numerical schemes within the `DUNE` framework, using the FEM toolbox `dune-fem`, see [14, 13, 21]. As grid manager we employed `ALBERTA`, [52], and all the meshes were created with the library `Gmsh`, [33]. All the schemes described in this paper are implemented as a `DUNE` module called `dune-navier-stokes-two-phase`, available at `https://github.com/magnese/dune-navier-stokes-two-phase`. The simulations were performed as single thread computations on a cluster of Xeon CPUs with frequency between 2.40GHz to 3.00GHz, and with at least 16GB of memory.

5.1 Convergence experiments

Here we consider convergence experiments for exact solutions to the incompressible two-phase Navier–Stokes flow problem. These exact solutions will be characterised by expanding spheres. In particular, let $\Gamma(t) = \{\vec{z} \in \mathbb{R}^d : |\vec{z}| = r(t)\}$ be a sphere of radius $r(t)$, with curvature $\varkappa(t) = -\frac{d-1}{r(t)}$. Then, for $\alpha \in \mathbb{R}_{\geq 0}$, it can be shown that the expanding sphere with radius

$$r(t) = \mathrm{e}^{\alpha\, t}\, r(0)\,, \tag{5.1a}$$

together with

$$\vec{u}(\vec{z},t) = \alpha\,\vec{z}\,, \quad p(\vec{z},t) = -\left[\gamma - 2\,\alpha\,\frac{\mu_+ - \mu_-}{d-1} r(t)\right] \varkappa(t) \left[\mathcal{X}_{\Omega_-(t)} - \frac{\mathcal{L}^d(\Omega_-(t))}{\mathcal{L}^d(\Omega)}\right], \tag{5.1b}$$

is an exact solution to the problem (2.14a–g), with (2.14b) replaced by

$$\nabla\,.\,\vec{u} = \alpha\, d \quad \text{in } \Omega_\pm(t)\,,$$

and with $\vec{f}(\cdot,t) = \rho\,\alpha^2\,\vec{\mathrm{id}}$ and $\vec{g} = \alpha\,\vec{\mathrm{id}}$ on e.g. $\Omega = (-1,1)^d$ with $\partial_1\Omega = \partial\Omega$.

		(5.1a,b)		(5.2a,b)	
J_Γ	τ	J_Ω^0	J_Ω^M	J_Ω^0	J_Ω^M
32	$6.4 \cdot 10^{-2}$	296	310	460	216
64	$1.6 \cdot 10^{-2}$	1 240	1 240	1 040	444
128	$4 \cdot 10^{-3}$	4 836	4 836	2 628	1 378
256	10^{-3}	18 476	18 476	7 460	4 476

Table 1: Discretization parameters for the convergence experiments. The values for J_Ω^M correspond to the scheme $(\mathcal{A}_{\text{im}}^h)$.

A nontrivial divergence free and radially symmetric solution $\vec{u}$ can be constructed on a domain that does not contain the origin. To this end, consider e.g. $\Omega = (-1,1)^d \setminus [-\frac{1}{3}, \frac{1}{3}]^d$. Then, for $\alpha \in \mathbb{R}_{\geq 0}$, the expanding sphere with radius

$$r(t) = ([r(0)]^d + \alpha\, t\, d)^{\frac{1}{d}}\,, \tag{5.2a}$$

together with

$$\vec{u}(\vec{z},t) = \alpha\, \frac{\vec{z}}{|\vec{z}\,|^d}\,, \quad p(\vec{z},t) = -\left(\gamma + 2\,\alpha\, \frac{\mu_+ - \mu_-}{r(t)^{d-1}}\right) \varkappa(t) \left[\mathcal{X}_{\Omega_-(t)} - \frac{\mathcal{L}^d(\Omega_-(t))}{\mathcal{L}^d(\Omega)}\right], \tag{5.2b}$$

is an exact solution to the problem (2.14a–g) with $\vec{f}(\vec{z},t) = \rho\,(1-d)\,\alpha^2\,|\vec{z}|^{-2d}\,\vec{z}$ and $\vec{g}(\vec{z}) = \alpha\,|\vec{z}|^{-d}\,\vec{z}$ on $\partial_1\Omega = \partial\Omega$.

For the following convergence tests, we choose the initial surface $\Gamma(0) = \{\vec{z} \in \mathbb{R}^d : |\vec{z}| = \frac{1}{2}\}$, $\partial_1\Omega = \partial\Omega$ and adaptive bulk meshes that use a finer resolution close to the interface. We compute the discrete solution over the time interval $[0,1]$. We begin with the exact solution (5.1a,b) for $\alpha = 0.15$ on $\Omega = (-1,1)^2$, with the physical parameters $\rho_+ = \rho_- = \mu_+ = \mu_- = \gamma = 1$. Details on the discretization parameters are given in Table 1, where we also state the final number of bulk elements, J_Ω^M, for the scheme $(\mathcal{A}_{\text{im}}^h)$. The values of J_Ω^M for the remaining three schemes were very similar. We report on the computed errors in Table 2, where we have defined the error quantities $\|\Gamma^h - \Gamma\|_{L^\infty} := \max_{m=1,\dots,M} \|\Gamma^m - \Gamma(t_m)\|_{L^\infty}$, where $\|\Gamma^m - \Gamma(t_m)\|_{L^\infty} := \max_{k=1,\dots,K_\Gamma} \operatorname{dist}(\vec{q}_k^{\,m}, \Gamma(t_m))$, as well as $\|\vec{U} - I_2^h\,\vec{u}\|_{L^2} := \left[\tau \sum_{m=1}^M \|\vec{U}^m - I_2^m\,\vec{u}(\cdot,t_m)\|_{L^2(\Omega)}^2\right]^{\frac{1}{2}}$, $\|\vec{U} - I_2^h\,\vec{u}\|_{L^2(H^1)} := \left[\tau \sum_{m=1}^M \|\vec{U}^m - I_2^m\,\vec{u}(\cdot,t_m)\|_{H^1(\Omega)}^2\right]^{\frac{1}{2}}$ and $\|P - p\|_{L^2} := \left[\tau \sum_{m=1}^M \|P^m - p(\cdot,t_m)\|_{L^2(\Omega)}^2\right]^{\frac{1}{2}}$. Moreover, we also define the estimated order of convergence EOC as $\text{EOC} = \ln \frac{\text{error}_1}{\text{error}_0} / \ln \frac{\text{h}_1}{\text{h}_0}$, where error_1 and error_0 are the errors for the computations with characteristic lengths $\text{h}_1 < \text{h}_0$. We observe from Table 2 that the schemes $(\mathcal{A}_{\text{ex}}^h)$, $(\mathcal{A}_{\text{im}}^h)$ and $(\mathcal{B}^h)$ capture the velocity solution exactly. This is possible because the true velocity is linear, see (5.1b). We also notice that all the schemes capture the interface position with the same accuracy and exhibit similar errors for the pressure. In terms of the overall CPU times, we notice that the two linear schemes $(\mathcal{A}_{\text{ex}}^h)$ and $(\mathcal{B}^h)$, as well as the two nonlinear schemes $(\mathcal{A}_{\text{im}}^h)$ and $(\mathcal{C}_{\text{ALE}}^h)$, have similar execution times, with the CPU times for the linear schemes lower than for the nonlinear schemes. Finally, we note that the number of full remeshings

J_Γ	$\|\Gamma^h - \Gamma\|_{L^\infty}$	$\|\vec{U} - I_2^h\,\vec{u}\|_{L^2}$	EOC	$\|\vec{U} - I_2^h\,\vec{u}\|_{L^2(H^1)}$	$\|P - p\|_{L^2}$	EOC	CPU[s]
				$(\mathcal{A}_{\mathrm{ex}}^h)$			
32	3.96456e-04	0	-	0	3.05157e-01	-	5
64	1.03429e-04	0	-	0	1.57053e-01	0.96	61
128	2.61302e-05	0	-	0	7.09596e-02	1.15	1 317
256	6.53463e-06	0	-	0	1.99794e-02	1.79	29 194
				$(\mathcal{A}_{\mathrm{im}}^h)$			
32	3.96456e-04	0	-	0	3.05157e-01	-	6
64	1.03429e-04	0	-	0	1.57053e-01	0.96	134
128	2.61302e-05	0	-	0	7.09596e-02	1.15	2 649
256	6.53463e-06	0	-	0	1.99794e-02	1.79	49 990
				$(\mathcal{B}^h)$			
32	3.96456e-04	0	-	0	3.05157e-01	-	5
64	1.03429e-04	0	-	0	1.57053e-01	0.96	45
128	2.61302e-05	0	-	0	7.09596e-02	1.15	1 567
256	6.53463e-06	0	-	0	1.99794e-02	1.79	30 997
				$(\mathcal{C}_{\mathrm{ALE}}^h)$			
32	3.96823e-04	1.94467e-06	-	1.74315e-05	3.01900e-01	-	10
64	1.03462e-04	1.27547e-07	3.93	1.33117e-06	1.57054e-01	0.94	160
128	2.61390e-05	4.26296e-08	1.58	3.78211e-07	7.09597e-02	1.15	2 420
256	6.53680e-06	1.09827e-08	1.91	9.46271e-08	1.99793e-02	1.79	47 181

Table 2: ($\rho_+ = \rho_- = \mu_+ = \mu_- = \gamma = 1, \alpha = 0.15$) Convergence experiments for (5.1a,b) on $(-1,1)^2$ over the time interval $[0,1]$, with the discretization parameters from Table 1.

performed when $J_\Gamma = 32$ was, depending on the scheme, between 1 and 4 while it was always 0 in all the other simulations for this test case.

In a second set of convergence experiments, we fix $\Omega = (-1,1)^2 \setminus [-\frac{1}{3}, \frac{1}{3}]^2$ for the exact solution (5.2a,b) with $\alpha = 0.15$ and the physical parameters

$$\rho_+ = 10^3\,, \quad \rho_- = 10^2\,, \quad \mu_+ = 10\,, \quad \mu_- = 1\,, \quad \gamma = 1\,.$$

We report on the errors of the exact solution (5.2a,b) in Table 3. Firstly, we observe that the velocity and the interface position is captured with similar accuracy by all four schemes. However, we notice that the pressure error does not appear to converge for the scheme $(\mathcal{B}^h)$, which may be caused by the discontinuous jumps in density and viscosity. In addition, the scheme $(\mathcal{C}_{\mathrm{ALE}}^h)$ exhibits slightly higher pressure errors compared to $(\mathcal{A}_{\mathrm{ex}}^h)$ and $(\mathcal{A}_{\mathrm{im}}^h)$. We also note the vastly superior CPU times for the scheme $(\mathcal{C}_{\mathrm{ALE}}^h)$ compared to the other schemes. This is caused by the fact that the domain is nonconvex, and so we employ a simple linear search in our velocity interpolation routine, recall §4.1.4. Of course, the ALE scheme only needs to interpolate the velocity when a complete bulk remeshing is performed, and this leads to a significant reduction in the total simulation time. Finally, we note that the number of full remeshings performed during any simulation for this test case was always between 3 and 9.

J_Γ	$\|\Gamma^h - \Gamma\|_{L^\infty}$	$\|\vec{U} - I_2^h\,\vec{u}\|_{L^2}$	EOC	$\|\vec{U} - I_2^h\,\vec{u}\|_{L^2(H^1)}$	$\|P - p\|_{L^2}$	EOC	CPU[s]
				$(\mathcal{A}_{\rm ex}^h)$			
32	4.13976e-03	1.24661e-03	-	2.59441e-02	2.28403e-00	-	8
64	1.07627e-03	4.80240e-04	1.38	1.35253e-02	1.20439e-00	0.92	102
128	2.55529e-04	3.70025e-04	0.38	1.20309e-02	5.89258e-01	1.03	2 810
256	6.66480e-05	1.42910e-04	1.34	6.48222e-03	2.69953e-01	1.10	88 056
				$(\mathcal{A}_{\rm im}^h)$			
32	3.99899e-03	1.23928e-03	-	2.61424e-02	2.30983e-00	-	11
64	1.07657e-03	4.80445e-04	1.37	1.35256e-02	1.20449e-00	0.94	126
128	2.55562e-04	3.70165e-04	0.38	1.20325e-02	5.89236e-01	1.03	3 223
256	6.66478e-05	1.42914e-04	1.34	6.48232e-03	2.69982e-01	1.10	95 315
				$(\mathcal{B}^h)$			
32	5.56564e-02	3.33673e-01	-	8.38557e-00	9.98404e+01	-	8
64	1.05282e-03	4.75502e-04	9.45	1.37127e-02	2.01792e+01	2.31	112
128	2.52742e-04	3.82170e-04	0.32	1.24067e-02	2.00690e+01	0.01	3 138
256	6.55951e-05	1.45885e-04	1.36	6.67192e-03	2.00294e+01	0	98 893
				$(\mathcal{C}_{\rm ALE}^h)$			
32	4.20717e-03	1.33379e-03	-	3.03240e-02	4.92318e-00	-	15
64	1.11280e-03	4.91358e-04	1.44	1.42754e-02	2.47229e-00	0.99	90
128	2.54196e-04	3.30194e-04	0.57	1.31294e-02	1.24058e-00	0.99	991
256	6.37496e-05	1.35898e-04	1.25	6.41819e-03	6.08051e-01	1.01	11 970

Table 3: ($\rho_+ = 10^3, \rho_- = 10^2, \mu_+ = 10, \mu_- = 1, \gamma = 1, \alpha = 0.15$) Convergence experiments for (5.2a,b) on $(-1,1)^2 \setminus [-\frac{1}{3},\frac{1}{3}]^2$ over the time interval $[0,1]$, with the discretization parameters from Table 1.

5.2 Benchmark computations

Here we will present benchmark computations for the two rising bubble test cases proposed in [41, Table I]. To this end, we use the setup described in [41, Figure 2], which is $\Omega = (0,1) \times (0,2)$ with $\partial_1\Omega = [0,1] \times \{0,2\}$ and $\partial_2\Omega = \{0,1\} \times (0,2)$. Moreover, the initial interface is $\Gamma_0 = \{\vec{z} \in \mathbb{R}^2 : |\vec{z} - (\frac{1}{2},\frac{1}{2})^T| = \frac{1}{4}\}$. We also let $\vec{f} = -0.98\,\rho\,\vec{e}_2$ and $\vec{g} = \vec{0}$. For the discretization parameters, we choose $\tau = 10^{-3}$, $T = 3$ and nearly uniform bulk meshes.

We recall that the paper [41] used the proposed benchmarks to compare two different codes based on the level-set method with a direct front tracking ALE method, while the same benchmarks were used in [4, 32] to investigate different phase-field methods. Finally, in [12] an unfitted front tracking method was used for these benchmark problems.

J_Γ	J_Ω^0	J_Ω^M			
		$(\mathcal{A}^h_{\text{ex}})$	$(\mathcal{A}^h_{\text{im}})$	$(\mathcal{B}^h)$	$(\mathcal{C}^h_{\text{ALE}})$
32	2 210	1 816	1 816	1 794	1 820
64	8 822	7 544	7 156	7 490	7 566
128	35 092	29 276	29 014	28 968	28 296

Table 4: Discretization parameters for the benchmark problem I.

	$J_\Gamma = 32$	$J_\Gamma = 64$	$J_\Gamma = 128$	$J_\Gamma = 32$	$J_\Gamma = 64$	$J_\Gamma = 128$
	$(\mathcal{A}^h_{\text{ex}})$			$(\mathcal{A}^h_{\text{im}})$		
$\not{s}_{\min}$	0.8929	0.8975	0.9001	0.8925	0.8973	0.9004
$t_{\not{s}=\not{s}_{\min}}$	1.9040	1.9040	1.9170	2.0410	1.9120	1.9160
$V_{c,\max}$	0.2439	0.2424	0.2411	0.2439	0.2423	0.2410
$t_{V_c=V_{c,\max}}$	0.9350	0.9300	0.9180	0.9340	0.9300	0.9190
$z_c(t=3)$	1.0829	1.0852	1.0822	1.0820	1.0841	1.0819
CPU[s]	10 236	29 610	234 396	17 771	72 837	403 904
	$(\mathcal{B}^h)$			$(\mathcal{C}^h_{\text{ALE}})$		
$\not{s}_{\min}$	0.8788	0.8851	0.8883	0.9013	0.9010	0.9030
$t_{\not{s}=\not{s}_{\min}}$	1.7920	1.7190	1.6790	1.9460	1.9050	1.8460
$V_{c,\max}$	0.2359	0.2344	0.2334	0.2435	0.2422	0.2410
$t_{V_c=V_{c,\max}}$	0.8860	0.8820	0.8750	0.9370	0.9330	0.9200
$z_c(t=3)$	1.0648	1.0622	1.0607	1.0877	1.0865	1.0824
CPU[s]	5 887	19 703	146 538	13 628	63 136	351 867

Table 5: ($\rho_+ = 10^3, \rho_- = 10^2, \mu_+ = 10, \mu_- = 1, \gamma = 24.5$) Benchmark problem I, with the discretization parameters from Table 4.

5.2.1 Benchmark problem I

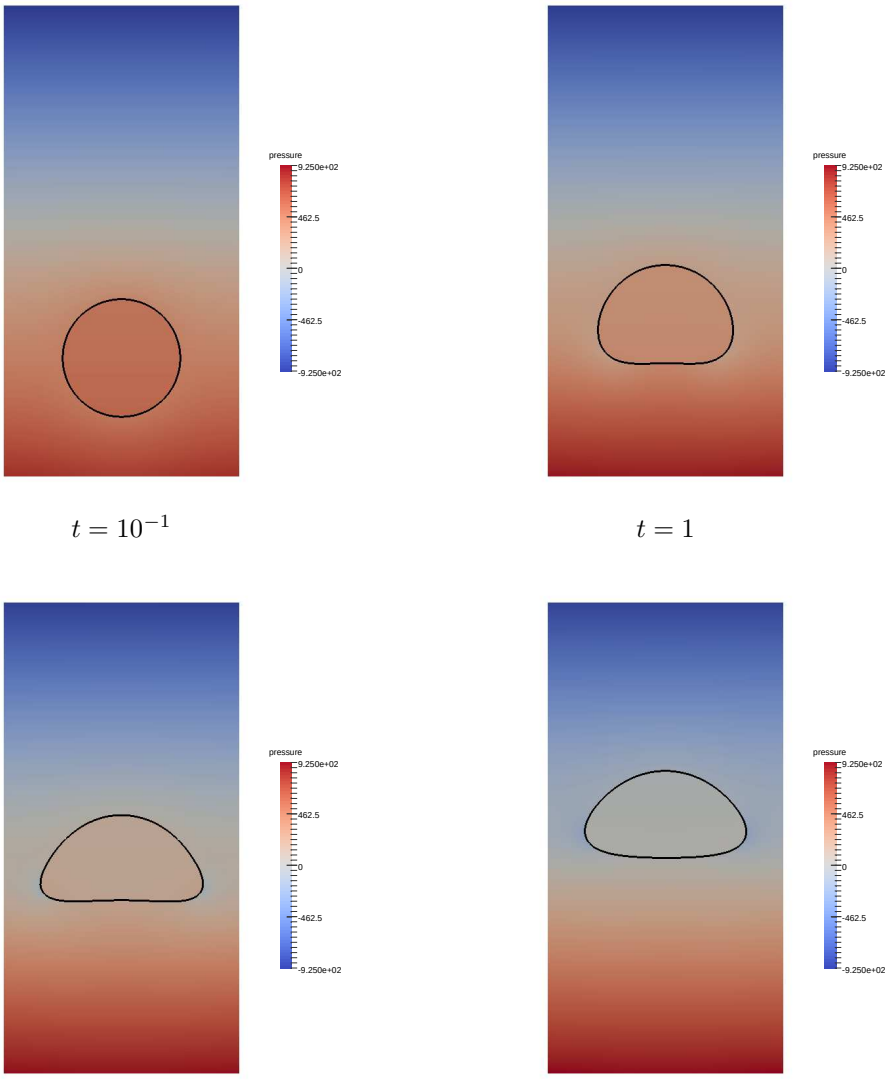
The physical parameters for the test case 1 in [41, Table I] are given by

$$\rho_+ = 10^3\,, \quad \rho_- = 10^2\,, \quad \mu_+ = 10\,, \quad \mu_- = 1\,, \quad \gamma = 24.5\,. \tag{5.3}$$

We list our discretization parameters for this benchmark computation in Table 4, and report on the quantitative results for our four schemes in Table 5. Here we have defined the $\vec{\mathrm{e}}_d$-component of the bubble's centre of mass as $z_c^m = \int_{\Omega_-^m} \vec{\mathrm{id}}\,.\,\vec{\mathrm{e}}_d \;\mathrm{d}\mathcal{L}^d/\mathcal{L}^d(\Omega_-^m)$, the degree of sphericity as $\not{s}^m = \pi^{\frac{1}{d}}[2\,d\,\mathcal{L}^d(\Omega_-^m)]^{\frac{d-1}{d}}/\mathcal{H}^{d-1}(\Gamma^m)$, and the bubble's rise velocity as $V_c^m = \int_{\Omega_-^m} \vec{U}^m\,.\,\vec{\mathrm{e}}_d \;\mathrm{d}\mathcal{L}^d/\mathcal{L}^d(\Omega_-^m)$. We observe that the results in Table 5 are in very good agreement, except for the scheme $(\mathcal{B}^h)$. For this reason, we do not consider the scheme $(\mathcal{B}^h)$ further in this section. The results for the remaining three schemes agree very well with the corresponding numbers from the finest discretization run of group 3 in [41], which are given by 0.9013, 1.9000, 0.2417, 0.9239 and 1.0817, and which are generally accepted as the most accurate in that paper. In fact, very similar results are also obtained in [12,

Tables 2 and 3]. Moreover, we note that the number of full remeshings performed during any simulation for this test case was always between 2 and 4.

In Figure 3 we show the evolution of the discrete pressures for a simulation with $J_\Gamma = 128$ interface elements using the scheme $(\mathcal{A}^h_{\text{im}})$, while the velocities are visualized in Figure 4. Finally, in Figures 5, 6, and 7 are reported, respectively, the evolution of the

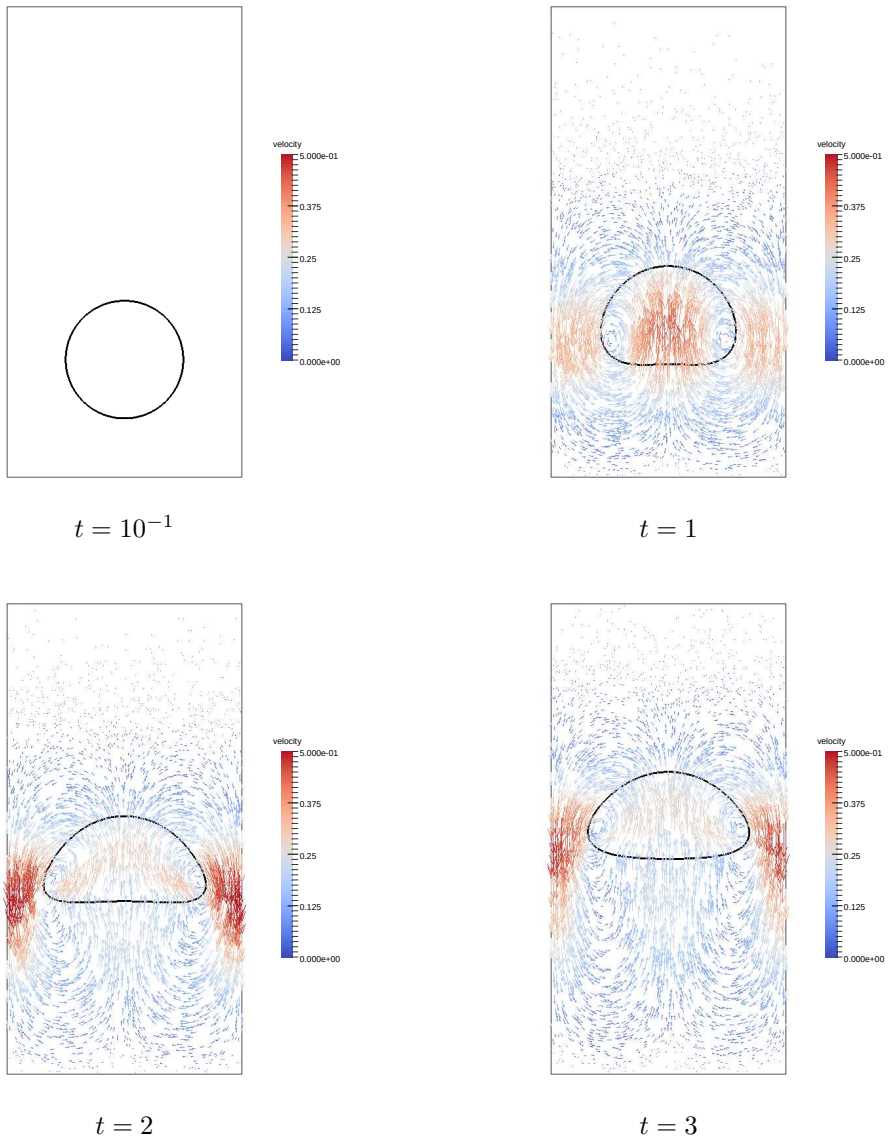


$t = 10^{-1}$ $t = 1$

$t = 2$ $t = 3$

Figure 3: $(\rho_+ = 10^3, \rho_- = 10^2, \mu_+ = 10, \mu_- = 1, \gamma = 24.5)$ Pressure evolution for the benchmark problem I for the scheme $(\mathcal{A}^h_{\text{im}})$ with $J_\Gamma = 128$ interface elements.

sphericity $\not\Phi$, rising velocity V_c and barycentre z_c over time. We notice almost identical results for the three schemes $(\mathcal{A}^h_{\text{ex}})$, $(\mathcal{A}^h_{\text{im}})$ and $(\mathcal{C}^h_{\text{ALE}})$, which are in very good agreement with the plots shown in [41]. We also note the excellent area conservation shown in 8, where we visualize the evolution of the relative inner area $\frac{\mathcal{L}^2(\Omega^m_-)}{\mathcal{L}^2(\Omega^0_-)}$ over time.



$t = 10^{-1}$ $t = 1$

$t = 2$ $t = 3$

Figure 4: Velocity vector field for the simulation in Figure 3.

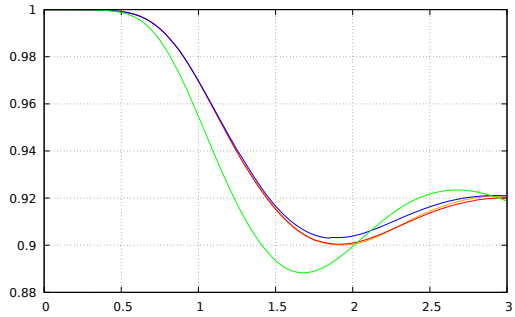


Figure 5: A plot of the sphericity $\not\Phi$ over time for the simulation in Figure 3 for the schemes $(\mathcal{A}^h_{\text{ex}})$ (orange), $(\mathcal{A}^h_{\text{im}})$ (red), $(\mathcal{B}^h)$ (green) and $(\mathcal{C}^h_{\text{ALE}})$ (blue).

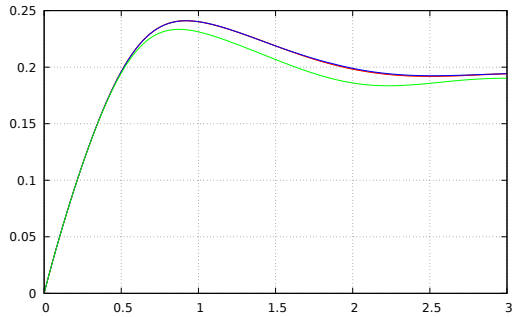


Figure 6: A plot of the rising velocity V_c over time for the simulation in Figure 3 for the schemes $(\mathcal{A}^h_{\text{ex}})$ (orange), $(\mathcal{A}^h_{\text{im}})$ (red), $(\mathcal{B}^h)$ (green) and $(\mathcal{C}^h_{\text{ALE}})$ (blue).

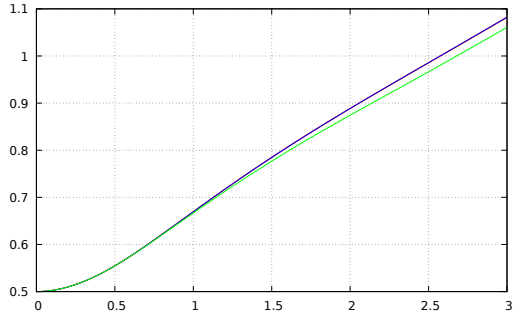


Figure 7: A plot of the barycentre z_c over time for the simulation in Figure 3 for the schemes $(\mathcal{A}^h_{\text{ex}})$ (orange), $(\mathcal{A}^h_{\text{im}})$ (red), $(\mathcal{B}^h)$ (green) and $(\mathcal{C}^h_{\text{ALE}})$ (blue).

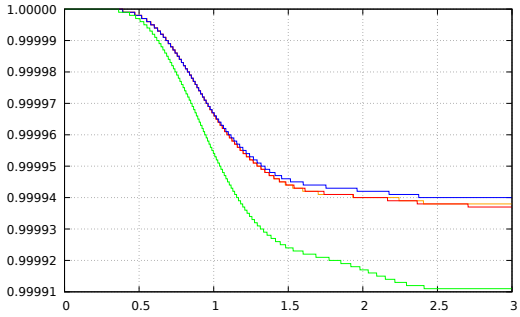


Figure 8: A plot of the relative inner area $\frac{\mathcal{L}^2(\Omega^m_-)}{\mathcal{L}^2(\Omega^0_-)}$ over time for the simulation in Figure 3 for the schemes $(\mathcal{A}^h_{\text{ex}})$ (orange), $(\mathcal{A}^h_{\text{im}})$ (red), $(\mathcal{B}^h)$ (green) and $(\mathcal{C}^h_{\text{ALE}})$ (blue).

	$(\mathcal{A}^h_{\rm ex})$	$(\mathcal{A}^h_{\rm im})$	$(\mathcal{C}^h_{\rm ALE})$
$\not{s}_{\min}$	0.5435	0.5473	0.5266
$t_{\not{s}=\not{s}_{\min}}$	3.0000	3.0000	3.0000
$V_{c,\max 1}$	0.2503	0.2502	0.2502
$t_{V_c=V_{c,\max 1}}$	0.7290	0.7290	0.7300
$V_{c,\max 2}$	0.2403	0.2403	0.2400
$t_{V_c=V_{c,\max 2}}$	2.1070	2.1540	2.0670
$z_c(t=3)$	1.1383	1.1387	1.1385
CPU[s]	132 498	236 635	236 253

Table 6: $(\rho_+ = 10^3, \rho_- = 1, \mu_+ = 10, \mu_- = 0.1, \gamma = 1.96)$ Benchmark problem II on $\Omega = (0,1) \times (0,2)$ over the time interval $[0,3]$.

5.2.2 Benchmark problem II

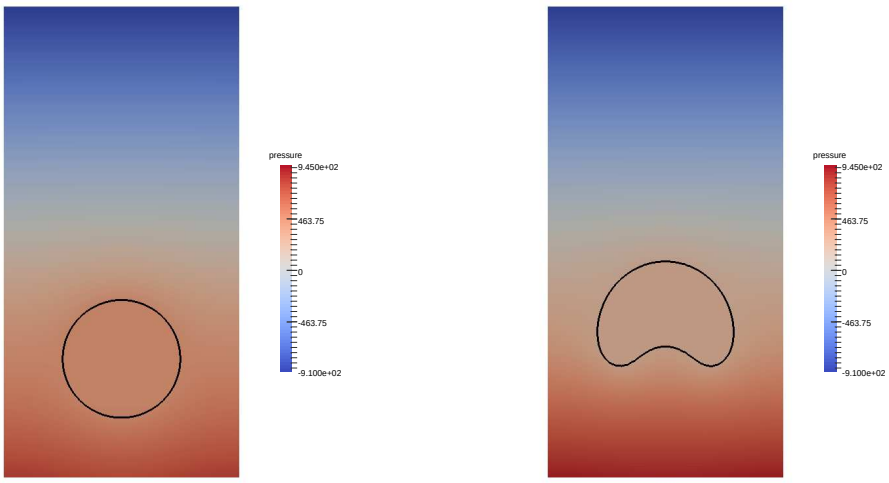
The physical parameters for the test case 2 in [41, Table I] are given by

$$\rho_+ = 10^3\,, \quad \rho_- = 1\,, \quad \mu_+ = 10\,, \quad \mu_- = 0.1\,, \quad \gamma = 1.96\,. \tag{5.4}$$

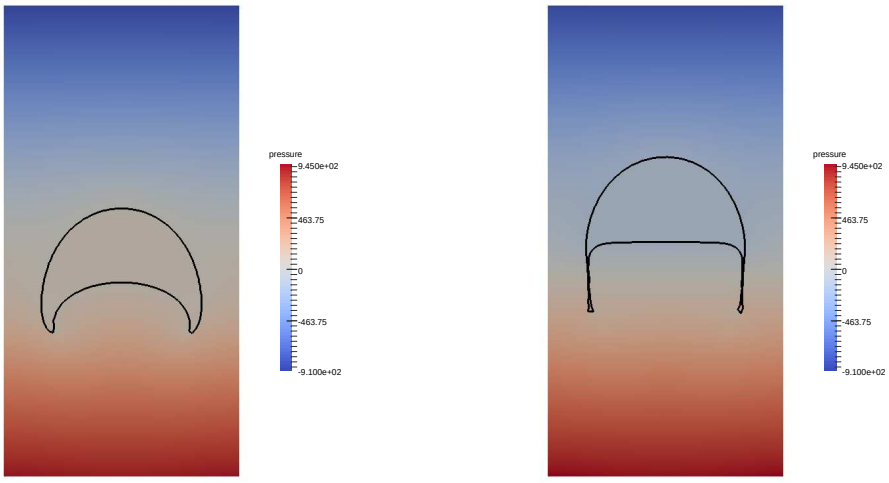
We report on the quantitative results for this benchmark computation in Table 6, where we have used $J_\Gamma = 128$ interface elements and $J^0_\Omega = 35\,092$ initial bulk elements. The final number of bulk elements, J^M_Ω, for the schemes $(\mathcal{A}^h_{\rm ex})$, $(\mathcal{A}^h_{\rm im})$ and $(\mathcal{C}^h_{\rm ALE})$ was 10 638, 10 716 and 9 972, respectively. We observe that the results in Table 6 are in good agreement with the corresponding numbers from the finest discretization run of group 3 in [41], which are given by 0.5144, 3.0000, 0.2502, 0.7317, 0.2393, 2.0600 and 1.1376. Here we note that there is little agreement on these results between the three groups in [41], but we believe the numbers of group 3 to be the most reliable ones. In fact, very similar results are also obtained in [12, Tables 5 and 6]. Moreover, we note that the number of full remeshings performed for this test case using the schemes $(\mathcal{A}^h_{\rm ex})$, $(\mathcal{A}^h_{\rm im})$ and $(\mathcal{C}^h_{\rm ALE})$ was 787, 788 and 808, respectively. These large numbers are caused by the strong deformation of the rising bubble, with the narrow tail structure leading to very elongated bulk elements. In fact, until time $t = 2$ only 6 remeshings are needed for each of the three schemes.

It is interesting to note that for this benchmark, the methods in [41] do not agree on whether the two tails of the rising bubble experience pinch-off of some droplets. But we note that the most accurate method there indicates that no pinch-off occurs, in agreement with our results and the results in [4, 12]. Of course, in a front tracking method like ours topological changes need to be performed heuristically when they occur, and this can be done, for example, as described in [17, 51].

In Figure 9 we show the evolution of the discrete pressures for a simulation with $J_\Gamma = 128$ interface elements using the scheme $(\mathcal{A}^h_{\rm ex})$, while the velocities are visualized in Figure 10. As before we show plots of the sphericity $\not{s}$, the rising velocity V_c, the barycentre z_c and the relative inner area $\frac{\mathcal{L}^2(\Omega^m_-)}{\mathcal{L}^2(\Omega^0_-)}$ over time in Figures 11, 12, 13 and 14, respectively. In the former three figures we note the good agreement with the plots shown

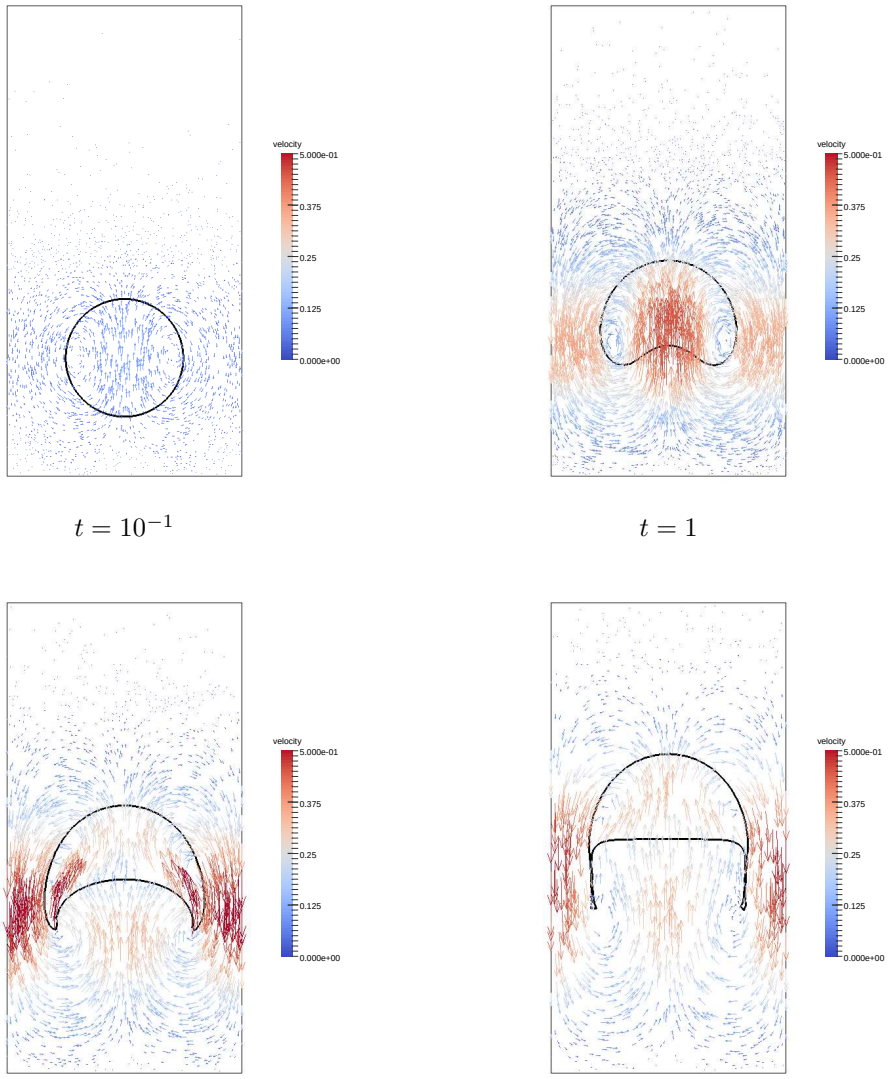


$t = 10^{-1}$ $t = 1$



$t = 2$ $t = 3$

Figure 9: $(\rho_+ = 10^3, \rho_- = 1, \mu_+ = 10, \mu_- = 0.1, \gamma = 1.96)$ Pressure evolution for the benchmark problem II for the scheme $(\mathcal{A}^h_{\text{ex}})$, with $J_\Gamma = 128$ interface elements.



$t = 10^{-1}$ $t = 1$

$t = 2$ $t = 3$

Figure 10: Velocity vector field for the simulation in Figure 9.

in [41]. As regards the plot of the relative inner area in Figure 14, we observe oscillations in the relative inner area between $t = 2.1$ and $t = 2.3$ which are caused by the very thin filaments originating at the bottom of the rising bubble.

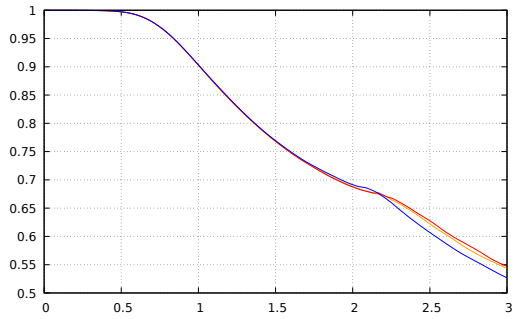


Figure 11: A plot of the sphericity $\not\Phi$ over time for the simulation in Figure 9 for the schemes $(\mathcal{A}^h_{\rm ex})$ (orange), $(\mathcal{A}^h_{\rm im})$ (red) and $(\mathcal{C}^h_{\rm ALE})$ (blue).

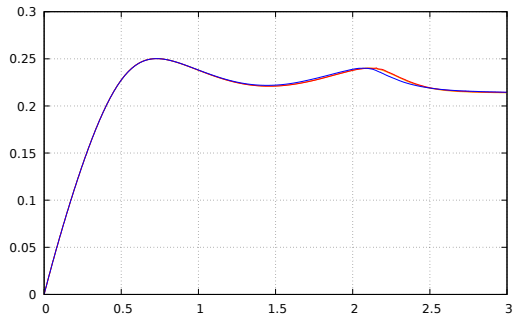


Figure 12: A plot of the rising velocity V_c over time for the simulation in Figure 9 for the schemes $(\mathcal{A}^h_{\rm ex})$ (orange), $(\mathcal{A}^h_{\rm im})$ (red) and $(\mathcal{C}^h_{\rm ALE})$ (blue).

Conclusions

We have introduced three Eulerian and one ALE scheme for the fitted finite element approximation of two-phase Navier–Stokes flow. Computationally the main difference between the Eulerian schemes and the ALE scheme is that for the former a bulk mesh adjustment needs to be performed after every time step, since the interface mesh in general moves, and so the fitted bulk mesh needs to be adjusted. Here we employ a bulk mesh smoothing procedure based on a linear elasticity problem. The mesh smoothing necessitates the interpolation of the discrete velocity approximation onto the new bulk mesh. It is often thought that the interpolation errors introduced by this procedure, together with the additional computational effort for the calculation of the interpolation, makes ALE schemes vastly superior to standard Eulerian approximations. However, our numerical results do not bear this out. In particular, for the physical and discretization

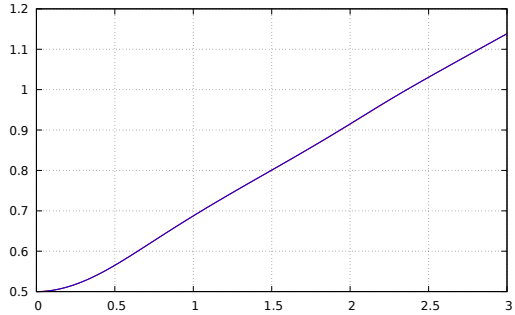


Figure 13: A plot of the barycentre z_c over time for the simulation in Figure 9 for the schemes $(\mathcal{A}^h_{\mathrm{ex}})$ (orange), $(\mathcal{A}^h_{\mathrm{im}})$ (red) and $(\mathcal{C}^h_{\mathrm{ALE}})$ (blue).

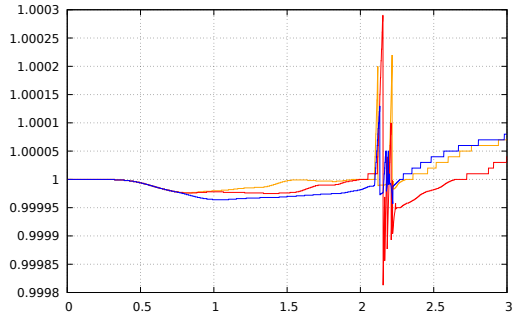


Figure 14: A plot of the relative inner area $\frac{\mathcal{L}^2(\Omega^m_-)}{\mathcal{L}^2(\Omega^0_-)}$ over time for the simulation in Figure 9 for the schemes $(\mathcal{A}^h_{\mathrm{ex}})$ (orange), $(\mathcal{A}^h_{\mathrm{im}})$ (red) and $(\mathcal{C}^h_{\mathrm{ALE}})$ (blue).

parameters that we consider, the Eulerian schemes $(\mathcal{A}^h_{\mathrm{ex}})$ and $(\mathcal{A}^h_{\mathrm{im}})$ perform as well as the ALE scheme $(\mathcal{C}^h_{\mathrm{ALE}})$ in most of our convergence tests and benchmark computations. We conjecture that this good performance of the Eulerian schemes is connected to the fact that all our schemes maintain the interface mesh quality throughout, and so the interface mesh, and the finite element information stored on it, is never heuristically adjusted, not even during a full bulk remeshing. Moreover, thanks to an efficient element-traversal algorithm, the additional computational effort needed for the velocity interpolation was only noticeable on nonconvex domains.

References

[1] H. Abels, H. Garcke, and G. Grün, *Thermodynamically consistent, frame indifferent diffuse interface models for incompressible two-phase flows with different densities*, Math. Models Methods Appl. Sci., 22 (2012), p. 1150013.

[2] M. Agnese, *Front tracking finite element methods for two-phase Navier–Stokes flow*, PhD thesis, Imperial College London, London, 2017.

[3] M. Agnese and R. Nürnberg, *Fitted finite element discretization of two-phase Stokes flow*, Internat. J. Numer. Methods Fluids, 82 (2016), pp. 709–729.

[4] S. Aland and A. Voigt, *Benchmark computations of diffuse interface models for two-dimensional bubble dynamics*, Internat. J. Numer. Methods Fluids, 69 (2012), pp. 747–761.

[5] D. M. Anderson, G. B. McFadden, and A. A. Wheeler, *Diffuse-interface methods in fluid mechanics*, in Annual review of fluid mechanics, Vol. 30, Annual Reviews, Palo Alto, CA, 1998, pp. 139–165.

[6] Ľ. Baňas and R. Nürnberg, *Numerical approximation of a non-smooth phase-field model for multicomponent incompressible flow*, M2AN Math. Model. Numer. Anal., 51 (2017), pp. 1089–1117.

[7] E. Bänsch, *Finite element discretization of the Navier–Stokes equations with a free capillary surface*, Numer. Math., 88 (2001), pp. 203–235.

[8] J. W. Barrett, H. Garcke, and R. Nürnberg, *A parametric finite element method for fourth order geometric evolution equations*, J. Comput. Phys., 222 (2007), pp. 441–462.

[9] ——, *On the parametric finite element approximation of evolving hypersurfaces in* $\mathbb{R}^3$, J. Comput. Phys., 227 (2008), pp. 4281–4307.

[10] ——, *Eliminating spurious velocities with a stable approximation of viscous incompressible two-phase Stokes flow*, Comput. Methods Appl. Mech. Engrg., 267 (2013), pp. 511–530.

[11] ——, *On the stable numerical approximation of two-phase flow with insoluble surfactant*, M2AN Math. Model. Numer. Anal., 49 (2015), pp. 421–458.

[12] ——, *A stable parametric finite element discretization of two-phase Navier–Stokes flow*, J. Sci. Comp., 63 (2015), pp. 78–117.

[13] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhuber, M. Ohlberger, and O. Sander, *A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part II: Implementation and Tests in DUNE*, Computing, 82 (2008), pp. 121–138.

[14] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, and O. Sander, *A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part I: Abstract Framework*, Computing, 82 (2008), pp. 103–119.

[15] F. Boyer, *A theoretical and numerical model for the study of incompressible mixture flows*, Comput. & Fluids, 31 (2002), pp. 41–68.

[16] F. Boyer and S. Minjeaud, *Hierarchy of consistent n-component Cahn–Hilliard systems*, Math. Models Methods Appl. Sci., 24 (2014), pp. 2885–2928.

[17] T. Brochu and R. Bridson, *Robust topological operations for dynamic explicit surfaces*, SIAM J. Sci. Comput., 31 (2009), pp. 2472–2493.

[18] T. A. Davis, *Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 196–199.

[19] ——, *Algorithm 915, SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization*, ACM Trans. Math. Software, 38 (2011), pp. 1–22.

[20] K. Deckelnick, G. Dziuk, and C. M. Elliott, *Computation of geometric partial differential equations and mean curvature flow*, Acta Numer., 14 (2005), pp. 139–232.

[21] A. Dedner, R. Klöfkorn, M. Nolte, and M. Ohlberger, *A Generic Interface for Parallel and Adaptive Scientific Computing: Abstraction Principles and the DUNE-FEM Module*, Computing, 90 (2010), pp. 165–196.

[22] H. Ding, P. D. Spelt, and C. Shu, *Diffuse interface model for incompressible two-phase flows with large density ratios*, J. Comput. Phys., 226 (2007), pp. 2078–2095.

[23] J. Donea, *Arbitrary Lagrangian Eulerian methods*, Comput. Meth. Trans. Anal., 1 (1983).

[24] S. Dong, *An efficient algorithm for incompressible N-phase flows*, J. Comput. Phys., 276 (2014), pp. 691–728.

[25] ——, *Physical formulation and numerical algorithm for simulating N immiscible incompressible fluids involving general order parameters*, J. Comput. Phys., 283 (2015), pp. 98–128.

[26] S. Elgeti and H. Sauerland, *Deforming fluid domains within the finite element method: five mesh-based tracking methods in comparison*, Arch. Comput. Methods Eng., 23 (2016), pp. 323–361.

[27] X. Feng, *Fully discrete finite element approximations of the Navier–Stokes–Cahn–Hilliard diffuse interface model for two-phase fluid flows*, SIAM J. Numer. Anal., 44 (2006), pp. 1049–1072.

[28] L. Formaggia and F. Nobile, *Stability analysis of second-order time accurate schemes for ALE–FEM*, Comput. Methods Appl. Mech. Engrg., 193 (2004), pp. 4097–4116.

[29] S. Ganesan, *Finite element methods on moving meshes for free surface and interface flows*, PhD thesis, University Magdeburg, Magdeburg, Germany, 2006.

[30] S. Ganesan, A. Hahn, K. Simon, and L. Tobiska, *ALE-FEM for two-phase and free surface flows with surfactants*, in Transport processes at fluidic interfaces, Adv. Math. Fluid Mech., Birkhäuser/Springer, Cham, 2017, pp. 5–31.

[31] S. Ganesan and L. Tobiska, *An accurate finite element scheme with moving meshes for computing 3D-axisymmetric interface flows*, Internat. J. Numer. Methods Fluids, 57 (2008), pp. 119–138.

[32] H. Garcke, M. Hinze, and C. Kahle, *A stable and linear time discretization for a thermodynamically consistent model for two-phase incompressible flow*, Appl. Numer. Math., 99 (2016), pp. 151–171.

[33] C. Geuzaine and J.-F. Remacle, *Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities*, Internat. J. Numer. Methods Engrg., 79 (2009), pp. 1309–1331.

[34] S. Gross and A. Reusken, *An extended pressure finite element space for two-phase incompressible flows with surface tension*, J. Comput. Phys., 224 (2007), pp. 40–58.

[35] ——, *Numerical methods for two-phase incompressible flows*, vol. 40 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2011.

[36] G. Grün and F. Klingbeil, *Two-phase flow with mass density contrast: Stable schemes for a thermodynamic consistent and frame-indifferent diffuse-interface model*, J. Comput. Phys., 257 (2014), pp. 708–725.

[37] A. Hahn, K. Held, and L. Tobiska, *ALE-FEM for two-phase flows*, PAMM. Proc. Appl. Math. Mech., 13 (2013), pp. 319–320.

[38] C. W. Hirt and B. D. Nichols, *Volume of fluid (VOF) method for the dynamics of free boundaries*, J. Comput. Phys., 39 (1981), pp. 201–225.

[39] P. C. Hohenberg and B. I. Halperin, *Theory of dynamic critical phenomena*, Rev. Mod. Phys., 49 (1977), pp. 435–479.

[40] T. Hughes, W. Liu, and T. Zimmermann, *Lagrangian-Eulerian finite element formulation for incompressible viscous flows*, Comput. Methods Appl. Mech. Engrg., 29 (1981), pp. 329–349.

[41] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska, *Quantitative benchmark computations of two-dimensional bubble dynamics*, Internat. J. Numer. Methods Fluids, 60 (2009), pp. 1259–1288.

[42] D. Kay, V. Styles, and R. Welford, *Finite element approximation of a Cahn–Hilliard–Navier–Stokes system*, Interfaces Free Bound., 10 (2008), pp. 15–43.

[43] R. J. LeVeque and Z. Li, *Immersed interface methods for Stokes flow with elastic boundaries or surface tension*, SIAM J. Sci. Comput., 18 (1997), pp. 709–735.

[44] J. Lowengrub and L. Truskinovsky, *Quasi-incompressible Cahn–Hilliard fluids and topological transitions*, R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci., 454 (1998), pp. 2617–2654.

[45] F. Nobile, *Numerical Approximation Of Fluid-Structure Interaction Problems With Application To Haemodynamics*, PhD thesis, EPFL, Lausanne, 2001.

[46] F. Nobile and L. Formaggia, *A stability analysis for the arbitrary lagrangian: Eulerian formulation with finite elements*, East-West J. Numer. Math., 7 (1999), pp. 105–132.

[47] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, vol. 153 of Applied Mathematical Sciences, Springer-Verlag, New York, 2003.

[48] C. S. Peskin, *The immersed boundary method*, Acta Numer., 11 (2002), pp. 479–517.

[49] S. Popinet, *An accurate adaptive solver for surface-tension-driven interfacial flows*, J. Comput. Phys., 228 (2009), pp. 5838–5866.

[50] Y. Renardy and M. Renardy, *PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method*, J. Comput. Phys., 183 (2002), pp. 400–421.

[51] A. Sacconi, *Front-tracking finite element methods for a void electro-stress migration problem*, PhD thesis, Imperial College London, London, UK, 2015.

[52] A. Schmidt and K. G. Siebert, *Design of Adaptive Finite Element Software: The Finite Element Toolbox ALBERTA*, vol. 42 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, 2005.

[53] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge, 1999.

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

[55] P. Sváček, *On numerical approximation of incompresible fluid flow with free surface influenced by surface tension*, EPJ Web Conf., 143 (2017), p. 02122.

[56] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan, *A front-tracking method for the computations of multiphase flow*, J. Comput. Phys., 169 (2001), pp. 708–759.

[57] S. O. Unverdi and G. Tryggvason, *A front-tracking method for viscous, incompressible multi-fluid flows*, J. Comput. Phys., 100 (1992), pp. 25–37.