INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng (2016)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.5493

Fully implicit finite element method for the modeling of free surface flows with surface tension effect

Aymen Laadhari*,† and Gábor Székely

Institut für Bildverarbeitung, Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology Zürich (Eidgenössische Technische Hochschule Zürich), Zürich, CH-8092, Switzerland

SUMMARY

A new approach based on the use of the Newton and level set methods allows to follow the motion of interfaces with surface tension immersed in an incompressible Newtonian fluid. Our method features the use of a high-order fully implicit time integration scheme that circumvents the stability issues related to the explicit discretization of the capillary force when capillary effects dominate. A strategy based on a consistent Newton–Raphson linearization is introduced, and performances are enhanced by using an exact Newton variant that guarantees a third-order convergence behavior without requiring second-order derivatives. The problem is approximated by mixed finite elements, while the anisotropic adaptive mesh refinements enable us to increase the computational accuracy. Numerical investigations of the convergence properties and comparisons with benchmark results provide evidence regarding the efficacy of the methodology. The robustness of the method is tested with respect to the standard explicit method, and stability is maintained for significantly larger time steps compared with those allowed by the stability condition. Copyright © 2016 John Wiley & Sons, Ltd.

Received 28 March 2016; Revised 4 December 2016; Accepted 8 December 2016

KEY WORDS: free surface flow; surface tension; cubically convergent Newton method; finite element method; anisotropic mesh adaptation; adaptive time stepping

1. INTRODUCTION

This framework is concerned with the numerical modeling of the dynamics of interfacial flows with surface tension in a surrounding incompressible Newtonian flow. The influence of the surface tension of the interface is significant in several scientific, engineering, and geophysical applications such as inkjet printing [1], atomization processes [2], and gases transfer between the atmosphere and seas [3].

Over the past decade, a multitude of computational strategies have been dedicated to this problem [4–9]. Typically, they can be roughly sorted according to the way they track the interface motion. We can distinguish between the interface tracking approaches and the interface capturing approaches.

Regarding the first group, a moving mesh explicitly fits the fluid interface and follows its movement. This Lagrangian representation results in separate conservation equations for the fluid inside and outside the interface, while the interfacial condition allows to set appropriate boundary conditions; for a detailed description of such an arbitrary Lagrangian Eulerian (ALE) formulation, see, for example, [10,11]. The boundary integral method, which expresses the flow in terms of distributions of singularities over the interfaces, can also be classified in this category [12].

*Correspondence to: Aymen Laadhari, Institut für Bildverarbeitung, Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology Zürich (Eidgenössische Technische Hochschule Zürich), CH-8092 Zürich, Switzerland.

†Email: laadhari@vision.ee.ethz.ch

‡Additional supporting information may be found online in the supporting information tab for this article.

In the second group, the flow in each fluid domain and on the interface is considered as a single continuum model, inherently taking into account the interfacial condition. In such an Eulerian description, the movement of the interface is followed in an implicit manner. An additional equation describing the transport of the interface is then considered. Several methods are used in the published literature such as the level set method [13, 14], the volume-of-fluid (VOF) method [15–17], and the phase field method [18]. Remark that there exist other hybrid methods between the interface tracking and interface capturing methods (e.g., [19, 20]).

While the Lagrangian approaches feature the sharpness of the interface representation, they may exhibit problems related for instance to the large displacement of the interface, mesh distortion, and changes of topology such as interface merging or breaking. The Eulerian approaches feature ease of implementation. They avoid problems related to the mesh distortion and naturally handle the topological changes of the interface. Although an additional transport equation of the interface needs to be solved, the corresponding computational cost is in essentially all cases insignificant compared with the cost of solving the Navier–Stokes equations.

In an Eulerian framework, a fixed mesh is usually used along the simulation period. Recently, a few mesh adaptation techniques have been employed together with Eulerian methods. In [4, 21], a VOF method combined with an adaptive and structured quad/octree mesh refinement technique is used for the numerical simulations of droplets, showing that the mesh refinement enables more computational accuracy. In [22], adapted unstructured meshes are employed together with the immersed boundary method and the level set method, resulting in a more accurate flow description especially near the wall. In [23], an anisotropic mesh adaptation technique is presented and allows to adapt a general unstructured mesh in the vicinity of the interface. Numerical results in the cases of LeVeque's test and the mean curvature motion depict that the rate of convergence of the method is improved by the mesh adaptation.

The surface tension is a major component in free surface flows, and numerical coupling of hydrodynamical and capillary effects remains challenging. Indeed, the capillary force lies at the interface and is consequently singular therein. A crucial difficulty is due to the numerical discretization of the surface tension term. For such a problem, the capillary number Ca is used to compare the relative effect of viscous forces with the surface tension acting across the interface between immiscible fluids. Most existing approaches have used explicit decoupling strategies that are conditionally stable, yielding a severe capillary stability condition for the temporal resolution of type $\Delta t < C\sqrt{\mathrm{Ca}}h^{3/2}$ especially for small capillary numbers (i.e., high surface tension) [24], where h stands for the mesh size and C is a constant value. For small velocities, the time step limit is rather given by the viscous time step limit, which corresponds to the characteristic diffusion time. For such a slow flow, this time step limitation of type $\Delta t < C_\nu h^2$, where C_ν is a constant depending on the kinematic viscosities (ratio of the dynamic viscosity to the density) of both inner and outer fluids, becomes more stringent than the previous convective time step restriction (e.g., [7, 25]). Therefore, exceeding this time step restriction may lead to an unphysical evolution of the interface or divergence of the numerical algorithm. Regarding the time discretization, almost exclusively first-order schemes have been used for problems of interfaces with capillary force, whereas only few works have used second-order schemes such as in [26, 27] or third-order schemes for differential-algebraic formulation of the ALE approach [28]. However, no works that used an adaptive time scheme of second order is known to us.

In [29], it was suggested that an implicit treatment of the surface tension would mitigate the time step restriction significantly. Although limited works have been devoted to the time integration strategies, some works introduced the semi-implicit schemes in order to lift the time step restrictions. Following the seminal work of Dziuk [30], Hysing described a semi-implicit method for the treatment of the surface tension term by introducing an additional implicit term that represents a diffusion of velocities tangential to the interface and induced by the surface tension [31]. The method was implemented in a finite element framework, and results show that the method remains stable up to almost 80 times the capillary time step constraint. In [32], the previous method was implemented in a VOF-based finite volume framework, showing that this time step restriction can be exceeded by at least a factor of five without destabilizing the numerical solution. In [33], a fully coupled numerical framework on arbitrary meshes in a finite volume framework is presented, including a novel method to evaluate the curvature from volume fractions. More stability is observed for larger density ratios.

DOI: 10.1002/nme

In the present paper, we intend to give a new computational way to solve the time-dependent coupled Navier–Stokes and level set equations involving interfacial flows with surface tension. The proposed method can be placed in the interface capturing group, as we use a fully Eulerian description of the interface in which the interface condition is incorporated by the level set method. A second-order scheme with adaptive time step sizes is introduced. We describe two fully implicit and monolithic approaches where we attempt to solve the entire coupled system using Newton-type strategies. To the knowledge of the authors, the Newton–Raphson linearization of this kind of problems is a novel concept. Our approach avoids the usual stability issues and time step limitations characterizing explicit strategies. Indeed, we achieve that stability is maintained for considerably larger time steps than those predicted by the explicit surface tension time step constraint [24]. Two variants of the exact Newton method featuring second-order and third-order convergence behaviors, respectively, are described. The quantitative comparison of their performances is presented subsequently. Our method is entirely based on a finite element discretization on arbitrary unstructured meshes, whereas an anisotropic metric-based mesh adaptation procedure allows to refine and enhance the computational accuracy on the interface and in zones with complex flow patterns. The present fully coupled implicit strategy is used to study and compare the maximum temporal resolution with respect to an explicit method, showing that the capillary time step constraint can be significantly exceeded. We numerically validate our framework in the context of the rising bubble benchmark presented in [34]. The method is further tested in the case of the oscillating two-dimensional bubble and in a simple three-dimensional case.

We have arranged the remainder of this paper as follows. Section 2 outlines the mathematical model, including the level set formulation and the Navier–Stokes equations governing the fluid dynamics. The derivation of the tangent problem and the linearization procedure are presented in Section 3, where we also present two alternative Newton strategies. We also provide details about the numerical discretization and the mesh adaptation procedure. In Section 4, various numerical simulations are performed to illustrate the efficiency and robustness of the proposed method. We close with some remarks and current extensions in Section 5.

2. GOVERNING EQUATIONS

Let $T > 0$ represent the period of the experiment. For any time $t \in (0, T)$, let $\Omega(t) \subset \mathbb{R}^d, d = 2, 3$ denote the internal domain having a Lipschitz continuous boundary $\Gamma(t) = \partial\Omega(t)$. The interface $\Gamma(t)$ is embedded in a larger computational domain Λ such that $\Gamma(t) \cap \partial\Lambda = \emptyset$, for all $t \in (0, T)$. Let $\boldsymbol{n}$ and $\boldsymbol{\nu}$ denote the unit outer normal vector on the interface $\Gamma(t)$ and the external boundary $\partial\Lambda$, respectively. Let $\boldsymbol{t}$ be a unit tangent vector on the boundary $\partial\Lambda$. We introduce the mean curvature $H = div\, \boldsymbol{n}$ as the sum of the principle curvatures on $\Gamma(t)$.

From now, the explicit dependence of Ω and Γ from t will be understood.

2.1. Level set formulation

An Eulerian framework is adopted to follow the interfacial motion in such a way that Γ represents an isosurface $\varphi = 0$ of a level set function φ. A time-dependent partial differential equation, initialized by a signed distance function $\varphi(t = 0)$ to $\Gamma(0)$, describes its motion

$$\partial_t \varphi + \boldsymbol{u} \cdot \boldsymbol{\nabla} \varphi = 0 \qquad \text{in } (0, T) \times \Lambda.$$

Geometrical quantities, for example, $\boldsymbol{n}$ and H, are encoded in terms of φ and are consequently extended to the entire domain Λ. It is well known that the signed distance property is lost after the advection of φ, giving sometimes rise to difficult scenarios where $|\boldsymbol{\nabla}\varphi|$ becomes very large or very small near Γ and deteriorates the accuracy of computations over Γ. To reestablish the signed distance property, a redistancing problem is commonly solved to initialize the level set function as a signed distance function [35]. However, it has been revealed in the literature that the zero level set may

DOI: 10.1002/nme

be unphysically shifted during the redistancing process, leading to substantial errors [36]. To alleviate this difficulty, several approaches were developed based on some mesh refinement techniques [37, 38] or by introducing an explicit forcing term to fix the isosurface 0 during the redistancing process [39–41]. In the present work, we use the redistancing approach presented in [42, 43] where a Lagrange multiplier allows to enforce a zero displacement on the interface Γ. Let τ be a pseudo-time variable. At every time $t \in (0, T)$, we initialize $\phi(\tau = 0, \cdot; t) = \varphi(t, \cdot)$, and we solve until convergence the problem:

$$\partial_\tau \phi(\tau, \cdot; t) + s_\varepsilon(\varphi(t, \cdot)) \, |\boldsymbol{\nabla} \, \phi(\tau, \cdot; t)| = s_\varepsilon \, (\varphi(t, \cdot)) + \lambda(\tau, \cdot; t), \qquad \text{in } (0, +\infty) \times \Lambda,$$

where $s_\varepsilon(\cdot)$ is a regularized sign function that helps us impose a zero displacement on Γ, while λ is a forcing term that prevents the unphysical shifting of Γ during the redistancing process. A full description of the approach is provided in [43]. We thereafter update $\varphi(t, \cdot)$ with $\phi(\infty, \cdot; t)$.

Let ε be a regularization parameter proportional to the local mesh size h. The sharp Heaviside function $\mathcal{H}$, the Dirac measure δ_Γ, and the sign function s are regularized within a banded strip of width 2ε such that

$$\mathcal{H}_\varepsilon(\varphi) = \frac{1}{2}\left(1 + \frac{\min(\max(\varphi, 0), \varepsilon)}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi \min(\max(\varphi, 0), \varepsilon)}{\varepsilon}\right)\right),$$
$$\delta_\varepsilon(\varphi) = \frac{d\mathcal{H}_\varepsilon}{d\varphi}(\varphi) \qquad \text{and} \qquad s_\varepsilon(\varphi) = 2\mathcal{H}_\varepsilon(\varphi) - 1.$$

In such an Eulerian-based framework, all integrals over Γ are transformed into integrals over Λ. For any given function $\eta(\cdot)$ defined on Γ, let $\tilde{\eta}(\cdot)$ be an extension over the domain Λ. We have

$$\int_\Gamma \eta(\boldsymbol{x}) \;\; d\boldsymbol{s} = \int_\Lambda |\nabla \varphi| \quad \delta_\Gamma \; \tilde{\eta}(\boldsymbol{x}) \;\; d\boldsymbol{x} \approx \int_\Lambda |\nabla \varphi| \; \delta_\varepsilon \, (\varphi) \; \tilde{\eta}(\boldsymbol{x}) \;\; d\boldsymbol{x}. \tag{2.1}$$

2.2. *Statement of the nonlinear capillary problem*

To model the surface tension effects in the case of immiscible and incompressible fluids, we consider the instationary Navier–Stokes equations with a free capillary boundary. We assume constant fluid density $\rho_{i/o}$ and viscosity $\mu_{i/o}$ in both sides of the interface Γ, where the subscripts 'i' and 'o' stand for the internal and external domains, respectively. The regularized global viscosity and density functions are given by

$$\mu_\varepsilon(\varphi) = \mu_i + (\mu_o - \mu_i)\mathcal{H}_\varepsilon(\varphi) \qquad \text{and} \qquad \rho_\varepsilon(\varphi) = \rho_i + (\rho_o - \rho_i)\mathcal{H}_\varepsilon(\varphi).$$

Let $\boldsymbol{\sigma}(\boldsymbol{u},p,\varphi) = 2\mu(\varphi)\mathbf{D}(\boldsymbol{u}) - p\mathbf{Id}$ and $\mathbf{D}(\boldsymbol{u}) = (\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)/2$ be the fluid Cauchy stress tensor and the strain tensor, respectively. Let $\left[.\right]_-^+$ denote the jump across Γ and γ be the surface tension coefficient. The capillary problem is characterized by the continuity of the velocity across the interface, while the stress discontinuity across the interface is calibrated according to the membrane tension. We also introduce an external body force $\boldsymbol{g}$.

We first present the non-dimensionalized problem. Let us introduce the Reynolds number $\mathrm{Re} = \rho_o U_g D/\mu_o$, where D is the diameter of the interface at $t = 0$ and $U_g = \sqrt{D|\boldsymbol{g}|}$ represents the velocity induced by the external force. We also consider the Eötvös number $\mathrm{Eo} = \rho_o U_g^2 D/\gamma$, which compares the gravitational forces with the surface tension effects. The capillary number is given by the ratio $\mathrm{Ca} = \mathrm{Eo}/\mathrm{Re}$. Let us denote by $\mu^\star = \mu_i/\mu_o$ and $\rho^\star = \rho_i/\rho_o$ the viscosity ratio and the density ratio between both sides of Γ, respectively. From now, the non-dimensionalized counterpart of the external force is also denoted $\boldsymbol{g}$ and all quantities and domains are non-dimensionalized. In addition, the same notations will be used for ease of exposition.

Let $\mathbf{div}(\cdot)$ and $div(\cdot)$ denote the divergence of tensor fields and vector fields, respectively. The non-dimensionalized problem reads:

DOI: 10.1002/nme

$\mathcal{P}$: *find the velocity* $\boldsymbol{u} = \boldsymbol{u}(t,\boldsymbol{x})$ *and pressure* $p = p(t,\boldsymbol{x})$ *such that*

$$\text{Re } \rho(\varphi)\big(\partial_t \boldsymbol{u} + \boldsymbol{u}.\nabla \boldsymbol{u}\big) - \mathbf{div}\big(2\mu(\varphi)\mathbf{D}(\boldsymbol{u})\big) + \boldsymbol{\nabla} p = \text{Re } \rho(\varphi)\boldsymbol{g}, \qquad \text{in } (0,T)\times\Lambda, \tag{2.2}$$

$$div\ \boldsymbol{u} = 0, \qquad \text{in } (0,T)\times\Lambda, \tag{2.3}$$

$$\big[\boldsymbol{u}\big]_-^+ = \boldsymbol{0} \qquad \text{and} \qquad \big[\boldsymbol{\sigma n}\big]_-^+ = \frac{1}{\text{Ca}} H\boldsymbol{n}, \qquad \text{on } (0,T)\times\Gamma. \tag{2.4}$$

3. NUMERICAL APPROXIMATION AND IMPLEMENTATION DETAILS

3.1. Semi-discretization in time

Let us divide $[0,\ T]$ into N subintervals $[t_n,\ t_{n+1})$, $n = 0, \dots, N-1$ of variable step size Δt_n. We consider an adaptive time stepping scheme to properly ensure the convergence of the Newton subiterations. Indeed, the problem is solved again with a decreased Δt_n by a factor 1/2 if the Newton loop does not converge after six iterations, whereas we increase Δt_{n+1} for the next time step with an amplification factor 1.2 if the convergence holds. For any $n \geqslant 1$, the unknowns $\boldsymbol{u}_n$, p_n, and φ_n at time step n are computed by induction. We use the adapted backward differentiation scheme of second order, referred to as BDF2, to approximate the time derivative terms [44]. The scheme is bootstrapped by the initial conditions $\boldsymbol{u}_{-1} = \boldsymbol{u}_0 = \boldsymbol{u}(0)$ and $\varphi_{-1} = \varphi_0 = \varphi(0)$, where $\boldsymbol{u}_{-1}$ and φ_{-1} only represent suitable notations. Let us introduce the ratio $\vartheta_n = \Delta t_n/\Delta t_{n-1}$.

Concerning the boundary conditions for the velocity, let $\Sigma_D \in \partial\Lambda$ be the subset on which essential boundary conditions are assigned, while free stresses are imposed on the remaining boundary $\partial\Lambda\backslash\Sigma_D$. We introduce the functional spaces:

$$\mathbb{V}(\boldsymbol{u}_b) = \left\{ \boldsymbol{v} \in \left(H^1(\Lambda)\right)^d : \boldsymbol{v} = \boldsymbol{u}_b \text{ on } \Sigma_D \right\} \quad \text{and} \quad \mathbb{Q} = \left\{ q \in L^2(\Lambda) : \int_\Omega q\, d\boldsymbol{x} = 0 \right\}.$$

Let $\Sigma_- = \{\boldsymbol{x} \in \partial\Lambda : \boldsymbol{u}\cdot\boldsymbol{\nu}(\boldsymbol{x}) < 0\}$ represent the upstream boundary. We introduce the space of admissible level set functions:

$$\mathbb{X}(\varphi_b) = \left\{ \psi \in W^{1,\infty}(\Lambda) \cap H^1(\Lambda) : \psi = \varphi_b \text{ on } \Sigma_- \right\}.$$

At every t_n, $\varphi_b = \varphi_{n-1}$ is the steady-state solution of the redistancing problem and approximates a signed distance function; it allows to set the boundary condition for φ_n on Σ_-. Considering the adaptive advancing formula of the BDF2 [44], the semi-discrete approximation in time of the capillary problem $\mathcal{P}$ reads:

$\mathcal{P}_n$: *find* $(\boldsymbol{u}_n, p_n, \varphi_n) \in \mathbb{V}(\boldsymbol{u}_b)\times\mathbb{Q}\times\mathbb{X}(\varphi_{n-1})$ *such that*

$$\begin{aligned} &\text{Re}\int_\Lambda \rho(\varphi_n)\frac{(1+2\vartheta_n)\,\boldsymbol{u}_n - (1+\vartheta_n)^2\boldsymbol{u}_{n-1} + \vartheta_n^2\boldsymbol{u}_{n-2}}{(1+\vartheta_n)\,\Delta t_n}\cdot\boldsymbol{v} + \text{Re}\int_\Lambda \rho(\varphi_n)\big(\boldsymbol{u}_n\cdot\boldsymbol{\nabla}\boldsymbol{u}_n\big)\cdot\boldsymbol{v} \\ &+ \int_\Lambda 2\mu(\varphi_n)\ \mathbf{D}(\boldsymbol{u}_n) : \mathbf{D}(\boldsymbol{v}) - \int_\Lambda p_n div\ \boldsymbol{v} - \frac{1}{\text{Ca}}\int_{\Gamma_n} H_n\frac{\boldsymbol{\nabla}\varphi_n}{|\boldsymbol{\nabla}\varphi_n|}\cdot\boldsymbol{v} - \text{Re}\int_\Lambda \rho(\varphi_n)\boldsymbol{g}\cdot\boldsymbol{v} = 0, \end{aligned} \tag{3.1}$$

$$\int_\Lambda q \quad div\ \boldsymbol{u}_n = 0, \tag{3.2}$$

$$\int_\Lambda \frac{(1+2\vartheta_n)\,\varphi_n - (1+\vartheta_n)^2\varphi_{n-1} + \vartheta_n^2\varphi_{n-2}}{(1+\vartheta_n)\,\Delta t_n}\psi + \int_\Lambda (\boldsymbol{u}_n\cdot\nabla\varphi_n)\,\psi = 0, \tag{3.3}$$

for all test functions $\boldsymbol{v} \in \mathbb{V}(\boldsymbol{0})$, $q \in \mathbb{Q}$, and $\psi \in \mathbb{X}(0)$.

DOI: 10.1002/nme

Following the approach described in [43, Section 3.3.2], we additionally consider an a posteriori mass correction term that enforces at every time step the condition $|\Omega_n| = |\Omega|_{\text{exact}}$.

Regarding the redistancing problem, we solve it at every time step after solving the system (3.1–3.3). At the numerical level, we use a first-order combined characteristics and finite difference discretization method to approximate the advection term. In addition, we use the Gauss–Lobatto quadrature formula that guarantees further stability for the characteristics method [45]. Further details about the numerical scheme and the method of characteristics are available in [43, 46–48].

3.2. Consistent linearization and Newton–Raphson method

For a given functional $\mathcal{F}(\varphi(\boldsymbol{x}))$, let $D\mathcal{F}(\varphi)[\delta\varphi]$ denote the Gâteaux derivative of $\mathcal{F}$ at φ along the direction $\delta\varphi$. Let $\boldsymbol{\nabla}_s = (\mathbf{Id} - \boldsymbol{n} \otimes \boldsymbol{n})\boldsymbol{\nabla} = \boldsymbol{\nabla} - \boldsymbol{n}\cdot(\boldsymbol{n}.\boldsymbol{\nabla})$ denote the surface gradient operator, and $\otimes$ represent the tensor product between vectors.

We first provide some useful directional derivatives in the direction of the level set increment $\delta\varphi$:

$$D\,\boldsymbol{\nabla}\varphi[\delta\varphi]=\boldsymbol{\nabla}\,\delta\varphi,\quad D|\,\boldsymbol{\nabla}\varphi|[\delta\varphi]=D\sqrt{\boldsymbol{\nabla}\,\varphi\cdot\boldsymbol{\nabla}\varphi}[\delta\varphi]=\boldsymbol{\nabla}\,\delta\varphi\cdot\boldsymbol{n},\quad D\frac{1}{|\,\boldsymbol{\nabla}\,\varphi|}[\delta\varphi]=-\boldsymbol{\nabla}\,\delta\varphi\cdot\frac{\boldsymbol{\nabla}\,\varphi}{|\,\boldsymbol{\nabla}\,\varphi|^3},$$

$$D\boldsymbol{n}[\delta\varphi]=\frac{\boldsymbol{\nabla}\,\delta\varphi}{|\,\boldsymbol{\nabla}\,\varphi|}-\frac{(\boldsymbol{\nabla}\,\varphi\cdot\boldsymbol{\nabla}\,\delta\varphi)\,\boldsymbol{\nabla}\,\varphi}{|\,\boldsymbol{\nabla}\,\varphi|^3}=\frac{\boldsymbol{\nabla}\,\delta\varphi}{|\,\boldsymbol{\nabla}\,\varphi|}-\frac{(\boldsymbol{n}\cdot\boldsymbol{\nabla}\,\delta\varphi)\boldsymbol{n}}{|\,\boldsymbol{\nabla}\,\varphi|}=\frac{\boldsymbol{\nabla}_s\delta\varphi}{|\,\boldsymbol{\nabla}\,\varphi|},\quad DH[\delta\varphi]=div\left(\frac{\boldsymbol{\nabla}_s\delta\varphi}{|\,\boldsymbol{\nabla}\,\varphi|}\right),$$

$$D\mu_\varepsilon(\varphi)[\delta\varphi]=(1-\mu^\star)\delta_\varepsilon(\varphi)\delta\varphi\qquad\text{and}\qquad D\rho_\varepsilon(\varphi)[\delta\varphi]=(1-\rho^\star)\delta_\varepsilon(\varphi)\delta\varphi.$$

Let us introduce the mixed variable $\Psi \equiv H$ and proceed with the computation of Ψ in the weak sense to decrease the derivation order. We assume that Γ never touches the external boundary $\partial\Lambda$. Remark that the evaluation of Ψ is only needed in a surrounding of the interface Γ. Consequently, an accurate computation of Ψ is only requested near Γ. By using the Green transformation, the boundary term vanishes, and we introduce the residual $\mathcal{R}_\Psi$ as follows:

$$\langle\mathcal{R}_\Psi\left(\Psi,\varphi\right),\xi\rangle_{H^{-1},H_0^1}\equiv\int_\Lambda\Psi\xi+\int_\Lambda\frac{\boldsymbol{\nabla}\,\varphi}{|\,\boldsymbol{\nabla}\,\varphi|}\cdot\boldsymbol{\nabla}\,\xi=0,\qquad\forall\xi\in H_0^1(\Lambda),\tag{3.4}$$

where the space $H^{-1}(\Lambda)$ represents the dual space of $H_0^1(\Lambda)$. After the linearization with respect to the variables φ and Ψ, we obtain the following equation coupling the increments $\delta\Psi$ and $\delta\varphi$:

$$\int_\Lambda\delta\Psi\;\xi+\int_\Lambda\frac{\boldsymbol{\nabla}_s\delta\varphi}{|\,\boldsymbol{\nabla}\,\varphi|}\cdot\boldsymbol{\nabla}\,\xi=-\langle\mathcal{R}_\Psi\left(\Psi,\varphi\right),\xi\rangle_{H^{-1},H_0^1},\qquad\forall\xi\in H_0^1(\Lambda).$$

Let us introduce the following weighted multilinear forms:

$$a(\boldsymbol{u},\boldsymbol{v};w)=\int_\Lambda 2w\;\mathbf{D}(\boldsymbol{u})\;:\;\mathbf{D}(\boldsymbol{v}),\qquad b(\boldsymbol{u},q)=-\int_\Lambda q\;\ddot{\mathrm{o}}\;\boldsymbol{u},\qquad d(\varphi,\boldsymbol{v};w)=\int_\Lambda w\,\boldsymbol{\nabla}\,\varphi\cdot\boldsymbol{v},$$

$$c(\boldsymbol{u},\boldsymbol{v};w,\boldsymbol{w})=\int_\Lambda w\;\Big(\left(\boldsymbol{u}\cdot\boldsymbol{\nabla}\right)\boldsymbol{w}+\left(\boldsymbol{w}\cdot\boldsymbol{\nabla}\right)\boldsymbol{u}\Big)\cdot\boldsymbol{v},\qquad e(\varphi,\psi;w)=\int_\Lambda w\;\varphi\psi,$$

$$f(\varphi,\psi;\mathbf{T})=\int_\Lambda(\mathbf{T}\,\boldsymbol{\nabla}\,\varphi)\cdot\boldsymbol{\nabla}\,\psi,\qquad g(\varphi,\boldsymbol{v};\boldsymbol{w})=\int_\Lambda\varphi\boldsymbol{v}\cdot\boldsymbol{w},\qquad m(\boldsymbol{u},\boldsymbol{v};w)=\int_\Lambda w\;\boldsymbol{u}\cdot\boldsymbol{v},$$

$$i(\varphi,\psi;\boldsymbol{w})=\int_\Lambda\psi\boldsymbol{w}\cdot\boldsymbol{\nabla}\,\varphi\quad\text{and}\quad h(\varphi,\boldsymbol{v};w,\boldsymbol{w})=\int_\Lambda 2\varphi w\;\mathbf{D}(\boldsymbol{v})\;:\;\mathbf{D}(\boldsymbol{w}),$$

defined for all $\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}\in\left(H^1(\Lambda)\right)^d$; $q\in L^2(\Lambda)$; $w\in L^\infty(\Lambda)$; $\varphi,\psi\in H^1(\Lambda)$; and $\mathbf{T}\in(L^\infty(\Lambda))^{d\times d}$.

The mixed problem is given by the system (3.1–3.4). Let $\Xi_n\equiv(\boldsymbol{u}_n,\,p_n,\,\Psi_n,\,\varphi_n)^T$ and $\mathcal{R}(\Xi_n)$ be the corresponding vector of unknowns and the global residual, respectively. The Newton–Raphson method reduces the nonlinear problem $\mathcal{P}_n$ into a sequence of linear subproblems [49]. Thereafter, we drop the subscript n whenever it is contextually clear. After linearization with respect to Ξ_n, the tangent system reads:

DOI: 10.1002/nme

Given Ξ^k, *find* $\delta\Xi^k = \left(\delta\boldsymbol{u}^k, \delta p^k, \delta\Psi^k, \delta\varphi^k\right) \in \mathbb{V}(\boldsymbol{u}_b) \times \mathbb{Q} \times H^1(\Lambda) \times \mathbb{X}(\varphi_b)$ *such that*

$$
\begin{aligned}
&\frac{\mathrm{Re}(1+2\vartheta_n)}{(1+\vartheta_n)\Delta t_n} m\Big(\delta\boldsymbol{u}^k, \boldsymbol{v}; \rho_\varepsilon(\varphi^k)\Big) + \mathrm{Re}\; c\Big(\delta\boldsymbol{u}^k, \boldsymbol{v}; \rho_\varepsilon(\varphi^k), \boldsymbol{u}^k\Big) + a\Big(\delta\boldsymbol{u}^k, \boldsymbol{v}; \mu_\varepsilon\left(\varphi^k\right)\Big) \\
&+ \mathrm{Re}\left(1-\rho^\star\right) g\left(\delta\varphi^k, \boldsymbol{v}; \delta_\varepsilon\left(\varphi^k\right)\left(\frac{(1+2\vartheta_n)\boldsymbol{u}^k - (1+\vartheta_n)^2\boldsymbol{u}_{n-1} + \vartheta_n^2\boldsymbol{u}_{n-2}}{(1+\vartheta_n)\Delta t_n} + \boldsymbol{u}^k\cdot\boldsymbol{\nabla}\,\boldsymbol{u}^k\right)\right) \\
&- \frac{1}{\mathrm{Ca}} g\Big(\delta\Psi^k, \boldsymbol{v}; \delta_\varepsilon\left(\varphi^k\right)\boldsymbol{\nabla}\,\varphi^k\Big) - \frac{1}{\mathrm{Ca}} g\Big(\delta\varphi^k, \boldsymbol{v}; \Psi^k\delta'_\varepsilon\left(\varphi^k\right)\boldsymbol{\nabla}\,\varphi^k\Big) - \frac{1}{\mathrm{Ca}} d\Big(\delta\varphi^k, \boldsymbol{v}; \Psi^k\delta_\varepsilon\left(\varphi^k\right)\Big) \\
&+ \left(1-\mu^\star\right) h\Big(\delta\varphi^k, \boldsymbol{v}; \delta_\varepsilon\left(\varphi^k\right), \boldsymbol{u}^k\Big) - \mathrm{Re}\left(1-\rho^\star\right)\; g\Big(\delta\varphi^k, \boldsymbol{v}; \delta_\varepsilon\left(\varphi^k\right)\boldsymbol{g}\Big) \\
&+ b\Big(\boldsymbol{v}, \delta p^k\Big) = -\langle\mathcal{R}_\Xi\left(\Xi^{T,k}\right), \boldsymbol{v}\rangle_{\mathbb{V}(\boldsymbol{0})',\mathbb{V}(\boldsymbol{0})},
\end{aligned}
\tag{3.5}
$$

$$
b\left(\delta\boldsymbol{u}^k, q\right) = -\langle\mathcal{R}_p\left(\boldsymbol{u}^k\right), q\rangle_{\mathbb{Q}',\mathbb{Q}}, \tag{3.6}
$$

$$
e\Big(\delta\Psi^k, \xi; 1\Big) + f\left(\delta\varphi^k, \xi; \left|\boldsymbol{\nabla}\,\varphi^k\right|^{-1}\left(\mathbf{Id} - \boldsymbol{n}^k\otimes\boldsymbol{n}^k\right)\right) = -\langle\mathcal{R}_\Psi\left(\Psi^k, \varphi^k\right), \xi\rangle_{H^{-1},H_0^1}, \tag{3.7}
$$

$$
e\left(\delta\varphi^k, \psi; \frac{1+2\vartheta_n}{(1+\vartheta_n)\Delta t_n}\right) + i\Big(\delta\varphi^k, \psi; \boldsymbol{u}^k\Big) + g\Big(\psi, \delta\boldsymbol{u}^k; \boldsymbol{\nabla}\,\varphi^k\Big) = -\langle\mathcal{R}_\varphi\left(\varphi^k, \boldsymbol{u}^k\right), \psi\rangle_{\mathbb{X}(0)',\mathbb{X}(0)}, \tag{3.8}
$$

for all $(\boldsymbol{v}, q, \xi, \psi) \in \mathbb{V}(\boldsymbol{0}) \times \mathbb{Q} \times H_0^1(\Lambda) \times \mathbb{X}(0)$, where the corresponding residuals express as

$$
\begin{aligned}
\langle\mathcal{R}_\Xi\left(\Xi^{T,k}\right), \boldsymbol{v}\rangle_{\mathbb{V}(\boldsymbol{0})',\mathbb{V}(\boldsymbol{0})} &= a\Big(\boldsymbol{u}^k, \boldsymbol{v}; \mu_\varepsilon\left(\varphi^k\right)\Big) + b\Big(\boldsymbol{v}, p^k\Big) - \frac{1}{\mathrm{Ca}} d\Big(\varphi^k, \boldsymbol{v}; \Psi^k\delta_\varepsilon\left(\varphi^k\right)\Big) \\
&\quad + \frac{\mathrm{Re}}{(1+\vartheta_n)\Delta t_n}\, m\left((1+2\vartheta_n)\boldsymbol{u}^k - (1+\vartheta_n)^2\boldsymbol{u}_{n-1}\right. \\
&\qquad \left. + \vartheta_n^2\boldsymbol{u}_{n-2}, \boldsymbol{v}; \rho_\varepsilon\left(\varphi^k\right)\right) \\
&\quad + \frac{\mathrm{Re}}{2} c\Big(\boldsymbol{u}^k, \boldsymbol{v}; \rho_\varepsilon\left(\varphi^k\right), \boldsymbol{u}^k\Big) - \mathrm{Re}\; m\Big(\boldsymbol{g}, \boldsymbol{v}; \rho_\varepsilon\left(\varphi^k\right)\Big), \\
\langle\mathcal{R}_p\left(\boldsymbol{u}^k\right), q\rangle_{\mathbb{Q}',\mathbb{Q}} &= b\left(\boldsymbol{u}^k, q\right), \\
\langle\mathcal{R}_\varphi\left(\varphi^k, \boldsymbol{u}^k\right), \psi\rangle_{\mathbb{X}(0)',\mathbb{X}(0)} &= e\left(\frac{(1+2\vartheta_n)\varphi^k - (1+\vartheta_n)^2\varphi_{n-1} + \vartheta_n^2\varphi_{n-2}}{(1+\vartheta_n)\Delta t_n}, \psi; 1\right) \\
&\quad + d\Big(\varphi^k, \boldsymbol{u}^k; \psi\Big), \\
\langle\mathcal{R}_\Psi\left(\Psi^k, \varphi^k\right), \xi\rangle_{H^{-1},H_0^1} &= e\Big(\Psi^k, \xi; 1\Big) + f\Big(\varphi^k, \xi; \left|\boldsymbol{\nabla}\,\varphi^k\right|^{-1}\mathbf{Id}\Big).
\end{aligned}
$$

Given the solutions at $t < t_n$, we iteratively compute Ξ_n^k at t_n by using two strategies that enable us to compute the increments $\delta\Xi_n^k = \left(\delta\boldsymbol{u}_n^k, \delta p_n^k, \delta\Psi_n^k, \delta\varphi_n^k\right)$. For any subiteration $k \geqslant 1$, we consider the following:

(i) Strategy I is the classical quadratically convergent Newton method, and it consists in solving

$$
\langle D\mathcal{R}\left(\Xi_n^k\right)\left[\delta\Xi_n^k\right], \boldsymbol{\xi}\rangle = -\langle\mathcal{R}\left(\Xi_n^k\right), \boldsymbol{\xi}\rangle, \forall\boldsymbol{\xi}.
$$

We explicitly update afterwards the solution as follows: $\Xi_n^{k+1} = \Xi_n^k + \delta\Xi_n^k$.

(ii) Strategy II is a formulation in the multidimensional case of the method introduced by Kou *et al.* [50]. It features a third-order convergence behavior in some neighborhood of the solution

DOI: 10.1002/nme

without requiring the evaluation of the second derivatives of the residual. The increment is computed at each Newton iteration in two steps as follows:

$$(1)\ \Xi_n^{k+0.5} = \Xi_n^k + \delta\Xi_n^k \qquad \text{with } \langle D\mathcal{R}\left(\Xi_n^k\right)\left[\delta\Xi_n^k\right], \boldsymbol{\xi}\rangle \quad = \langle \mathcal{R}\left(\Xi_n^k\right), \boldsymbol{\xi}\rangle, \forall \boldsymbol{\xi}.$$

$$(2)\ \Xi_n^{k+1} = \Xi_n^{k+0.5} + \delta\Xi_n^{k+0.5} \text{ with } \langle D\mathcal{R}\left(\Xi_n^k\right)\left[\delta\Xi_n^{k+0.5}\right], \boldsymbol{\xi}\rangle = -\langle \mathcal{R}\left(\Xi_n^{k+0.5}\right), \boldsymbol{\xi}\rangle, \forall \boldsymbol{\xi}.$$

Observe that, for Strategy II, the minus sign appears in front of the residual only in the second step. The efficiency index is an indicator commonly used to compare the Newton variants; it measures the corresponding order of convergence per residual or Jacobian evaluation. However, we notice that this index does not represent always the best way of comparison in the multidimensional case, as it does not account for the differences in the computational cost of the Jacobian assembly, Jacobian factorization, and residual evaluation. Strategy II requires two evaluations of the residual and one assembly of the Jacobian, yielding an efficiency index of $\sqrt[3]{3}$. This efficiency index is better than the one of Strategy I given by $\sqrt{2}$ [50]. The Kou method is based on a modification of the Werrakoon–Fernando method [51], and its efficiency compared with the main existing third-order methods is demonstrated in [50] for several basic monodimensional test cases. A similar cubically convergent method was subsequently developed in [52]. The latter strategy does not require the evaluation of the second derivative and needs one evaluation of the residual and two evaluations of the Jacobian per iteration.

To compare the computational cost of both residuals and Jacobian's assembly, we proceed with the assembly of the linear system using a structured mesh having $75'600$ triangles and $29'600$ vertices. The computational cost of the residual's assembly is 20.7138 s, whereas the computational cost of the Jacobian's assembly is equal to 39.1561 s, and the cost of the Jacobian factorization is 35.7054 s. Hence, the evaluation of the residual is significantly cheaper than the assembly and factorization of the Jacobian matrix. It is more beneficial to consider a method that only needs to assemble and factorize the Jacobian once at each Newton iteration. Consequently, we opt to use the Kou method developed in [50] for the capillary problem among other cubically convergent method, and we consider the aforementioned generalization of this method to the multidimensional case.

Furthermore, we use a second-order extrapolation of the solution at previous times to assign the starting values of the velocity and level set at each Newton loop. For practical computations, the stopping criterion is based on the evaluation of the global residual, and we set the Newton tolerance to 10^{-9} in our computations.

3.3. Space discretization and banded level set approach

We consider a partition $\mathcal{T}_h$ of Λ consisting of geometrically conforming open simplicial elements K, that is, triangles for $d = 2$ and tetrahedra for $d = 3$, such that $\overline{\Lambda} = \underset{K\in\mathcal{T}_h}{\cup}$. Regarding the finite dimensional spaces, we consider a Taylor–Hood finite element approximation for the discretization of velocity and pressure (e.g., [53]), while we use the P2 Lagrange finite elements to approximate the level set function φ and the mixed variable Ψ.

We now focus on the resolution of the Jacobian system (3.5–3.8) arising from the linearization of (3.1–3.4) after time and space discretizations. The pressure is only computed up to an arbitrary constant, and the global matrix of the linear system is then singular. To solve the discretized problem using a direct method, we can use, for instance, the procedure that consists in adding a real Lagrange multiplier that imposes a constant average pressure value [46], which corresponds to adding one line in the global linear system. Otherwise, we can fix the pressure value on one degree of freedom before solving the linear system.

The linear system corresponding to the discretized problem (3.5–3.8) has a sparse and block structure. We use the package Mumps[§] for the factorization and as direct solver on distributed memory architectures. This package provides a direct method using the multifrontal method that represents a version of the Gaussian elimination for large sparse systems of equations with either symmetric or

[§]MUMPS – http://graal.ens&LWx02010;lyon.fr/MUMPS.

DOI: 10.1002/nme

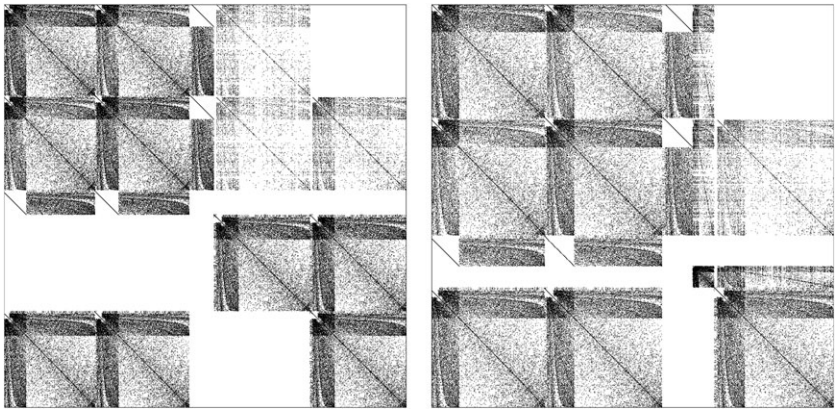

Figure 1. Two-dimensional case. Sparsity pattern of the global matrix exported in the standard Matrix Market format. Left: Equation (3.7) assembled in the entire Λ yields a matrix size $= 288'489^2$. Right: Equation (3.7) assembled in the banded domain $B_{\varepsilon'}$ yields a matrix size $= 223'338^2$.

unsymmetric matrices [54]. We refer to [54–57] for detailed descriptions of the methods and algorithms used. For the tridimensional problem, iterative methods using for instance the preconditioned conjugate gradient algorithm can also be used [45].

We thereafter give further insight into the structure of the matrix of the linear system corresponding to (3.5–3.8), and we show how the size of the linear system can be reduced. Indeed, the capillary term in the momentum equation is only localized in the vicinity of the interface Γ, and we only need to evaluate the mixed variable Ψ in a small surrounding of Γ. Consequently, we only assemble the equation (3.7) in a banded strip of width ε' around the interface, with ε' is a parameter proportional to the mesh size. We commonly choose $\varepsilon' = 4h$ in our computations. That helps to avoid unnecessary computational effort. The banded strip is given by

$$B_{\varepsilon'}(t) = \{K \in \mathcal{T}_h \,:\, \delta_{\varepsilon'}(\varphi) \neq 0\}.$$

Only the coefficients that correspond to these elements are then considered in the global matrix. To graphically show the size reduction of the linear system, we use the Matrix Market format[¶] (ASCII-based) to display the sparse and block-structured Jacobian matrix. This format is elaborated in such a way that only the non-zero entries are encoded and the corresponding coordinates are explicitly stored. We thereafter convert into the Matlab sparse format[‖] to graphically visualize the sparsity pattern. Let us consider a two-dimensional case and a structured mesh of $33'836$ triangular elements. We first consider Equation (3.7) in the entire Λ, and we assemble the matrix of the corresponding linear system. The corresponding matrix displayed in Figure 1 (left) shows a sparse block structure, and it has the size $288'489^2$. By restricting (3.7) in the banded domain $B_{\varepsilon'}$, the matrix size is reduced to $223'338^2$ (Figure 1, right). A significant reduction of the size of the linear system is accordingly obtained and corresponds to almost 22.6% in the number of rows and columns.

3.4. Anisotropic mesh adaptation

We use the bidimensional anisotropic mesh generator BAMG based a Delaunay-type triangulation [58, 59]. We use a metric-based approach, and we follow the mesh adaptation procedure described in [23, 43]. However, we consider a different meshing criterion more appropriate to the present capillary problem. Compared with our previous work [23], the mesh adaptation procedure also aims to refine the mesh in areas of less regular velocity, whereas it must coarsen the mesh in zones of regular velocity, even coarser than the initial mesh.

[¶]Matrix Market format – http://math.nist.gov/MatrixMarket/index.html.
[‖]Matlab – http://ch.mathworks.com/.

DOI: 10.1002/nme

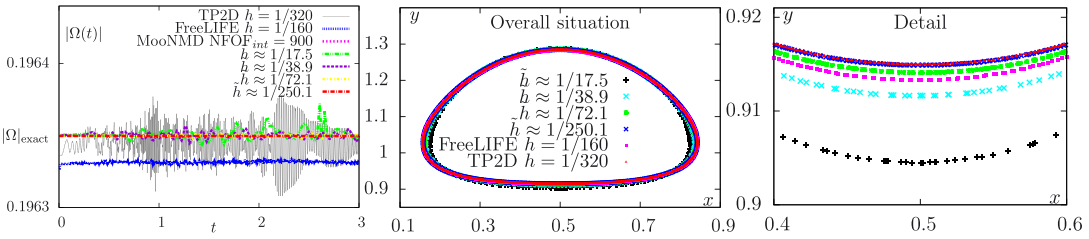

Figure 2. Example 1: Left: Temporal evolution of the area. Middle: Comparison of the final shapes with the reference solutions in [34]. Right: A close view in zones of maximal discrepancies between the different shapes. The same color code as for the global view is used.

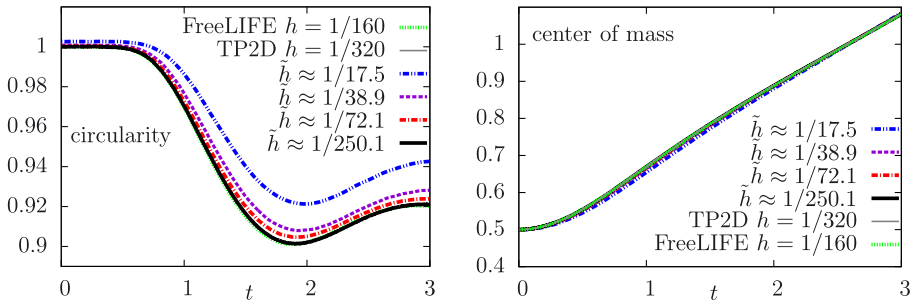

Figure 3. Example 1: Temporal evolution of the circularity /c and the center of mass Y_c and comparisons with reference solutions in [34].

Given the solution Ξ_n at time step t^n, we shall consider a meshing criterion $\chi(\Xi_n)$. We introduce the metric tensor given by the Hessian matrix of the previous scalar field: $\boldsymbol{\nabla\nabla}\chi(\Xi_n)$. For each triangle K in the mesh $\mathcal{T}_h$, a singular value decomposition of the Jacobian of the affine transformation that maps a reference equilateral triangle into K is needed. An adapted mesh is then generated by shrinking all triangles in both eigenvectors' directions of $\boldsymbol{\nabla\nabla}\chi(\Xi_n)$, while we adjust the triangle's sizes in these directions such that the interpolation error becomes equidistributed. To increase the computational accuracy near the interface and in zones where more complex flow patterns (such as recirculation zones / vortices) occur, we introduce the following criterion:

$$\chi(\Xi) = \Big(\big(\rho^\star + (1-\rho^\star)\mathcal{H}_\varepsilon(\varphi)\big)\ \boldsymbol{u}^2 + 4\big(\mu^\star + (1-\mu^\star)\mathcal{H}_\varepsilon(\varphi)\big)\ \mathbf{D}(\boldsymbol{u})^2 \Big)^{1/2}.$$

We tune the parameters of the mesh adaptation, such as the maximum anisotropy ratio, the smallest edge's length, and the maximal aspect ratio of all mesh elements $K \in \mathcal{T}_h$, to generate either almost isotropic or anisotropic meshes (see snapshots in Figures 5 and 9 and Movies 1–5 in the supporting information). In the numerical examples, we use moderately anisotropic meshes unless otherwise stated.

4. NUMERICAL EXAMPLES

In what follows, we provide a set of numerical examples in both two-dimensional and three-dimensional cases to demonstrate the main features of the proposed method.

Software implementation.The presented method has been implemented using the Rheolef [45] environment. It is a general purpose C++ library for scientific computing, with special emphasis on finite elements and parallel computation. Distributed memory parallelism is performed via MPI.[**]

[**]Message Passing Interface – http://www.mpi&LWx02010;forum.org.

DOI: 10.1002/nme

The Scotch library is used for distributed mesh partitioning.[††] The library relies upon the Boost,[‡‡] Blas,[§§] and UMFPACK[¶¶] libraries for much of its functionalities. The different meshes have been generated using GMSH[‖‖] and BAMG.[***] The computational results are displayed graphically using the software Paraview,[†††] while the plots are generated using Gnuplot.[‡‡‡]

4.1. Example 1: Numerical validation using the benchmark of the rising bubble dynamics

To proceed with the validation of our numerical method, we consider the quantitative benchmark of the rising bubble dynamics in a heavier fluid introduced by Hysing *et al.* [34]. For a comparative study, we only consider the setup of the ellipsoidal bubble where full agreement is obtained by the different codes in [34], while we only show a sample result for the setup of the skirted bubble and the corresponding adapted mesh in thin filamentary regions. Unless otherwise stated, the numerical results are obtained using Strategy II, which behaves generally computationally cheaper than Strategy I. More details about the computational costs will be provided afterwards.

The bubble is initially circular having a radius of $r_0 = 0.25$, yielding an area $|\Omega|_{\text{exact}} = \pi^2/16$ and placed at $\boldsymbol{x} = (0.5,0.5)$ in the rectangular computational domain $\Lambda = [0,1] \times [0,2]$. Simulations are performed over the time period $(0,T) = (0,3)$. An external gravity force $\boldsymbol{g} \approx (0, -0.98)^T$ is considered and points toward the bottom of the domain. The physical parameters are given by $\rho_o = 10^3$, $\rho_i = 10^2$, $\mu_o = 10$, $\mu_i = 1$, and $\gamma = 24.5$, which leads to the dimensionless numbers $\rho^\star = 0.1$, $\mu^\star = 0.1$, $\text{Re} = 35$, $\text{Eo} = 10$, and $\text{Ca} = 0.286$. The no-slip wall conditions are imposed on the horizontal boundaries (i.e., representing Σ_D), while the free slip conditions $\boldsymbol{u}\cdot\boldsymbol{\nu} = 0$ and $\boldsymbol{t}\cdot\mathbf{D}(\boldsymbol{u})\cdot\boldsymbol{\nu} = 0$ are prescribed on the remaining boundaries.

Let u_x and u_y be the velocity components such that $\boldsymbol{u} = (u_x, u_y)$. Following the benchmark, some quantities of interest are considered and represent the center of mass $Y_{\boldsymbol{c}}(t)$, the degree of circularity $\phi(t)$, the minimum circularity $\phi_{\min}$, and the corresponding incidence time $t|_{\phi=\phi_{\min}}$, the rise velocity $V_{\boldsymbol{c}}(t)$, the maximal velocity $V_{\boldsymbol{c},\max}$ and the corresponding incidence time $t|_{V_{\boldsymbol{c}}=V_{\boldsymbol{c},\max}}$. We have

$$Y_{\boldsymbol{c}}(t) = \frac{1}{|\Omega|}\int_\Omega y \, \mathrm{d}\boldsymbol{x}, \qquad \phi(t) = \frac{2\sqrt{\pi|\Omega|}}{|\Gamma|} \qquad \text{and} \qquad V_{\boldsymbol{c}}(t) = \frac{1}{|\Omega|}\int_\Omega u_y \, bx.$$

Let NTS and NV be the numbers of time steps and mesh vertices, respectively. We evaluate the mean mesh size $\tilde{h}$ during the simulation period as the mesh size of a structured triangular mesh having the same NV at every time step:

$$\tilde{h} = \frac{1}{\text{NTS}}\sum^{\text{NTS}} \frac{4}{\sqrt{1+8\text{NV}}-3}.$$

Because that characterizes a structured mesh of similar size, it leads to a problem with the same number of degrees of freedom as the problem solved on the adapted mesh and consequently having the same size of the linear system.

In Figure 2, the bubble shapes at the final time $T = 3$ with different mesh resolutions are compared with the shapes obtained at $T = 3$ in [34] using the finest meshes. A close-up view in the zone of maximal discrepancy is also provided in Figure 2 (right), showing overall a good agreement. The bubble's area is clearly preserved equal to $|\Omega|_{\text{exact}}$ during the simulation period, as depicted in Figure 2 (left). Figure 3 shows that no significant differences can be seen in the temporal evolution of the center of mass $Y_{\boldsymbol{c}}$. The curves of the circularity ϕ agree well for the finest mesh, while only an enlarged section around $t|_{V_{\boldsymbol{c}}=V_{\boldsymbol{c},\max}}$ depicts some differences that are minimized for finer meshes.

[††] Scotch – http://www.labri.fr/perso/pelegrin/scotch.
[‡‡] Boost libraries – http://www.boost.org.
[§§] Basic Linear Algebra Subprograms library – http://www.netlib.org/blas.
[¶¶] Umfpack routines – http://www.cise.ufl.edu/research/sparse/umfpack/.
[‖‖] GMSH – http://www.geuz.org/gmsh.
[***] BAMG – http://people.sc.fsu.edu/~jburkardt/data/bamg/bamg.html.
[†††] Paraview – http://www.paraview.org.
[‡‡‡] Gnuplot – http://www.gnuplot.info.

DOI: 10.1002/nme

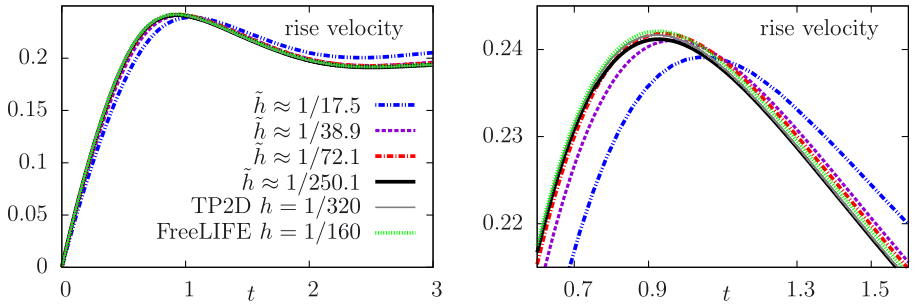

Figure 4. Example 1: Temporal evolution of the rise velocity V_c and detailed view around the peak of velocity and comparisons with reference solutions in [34]. The same labels and color code are used.

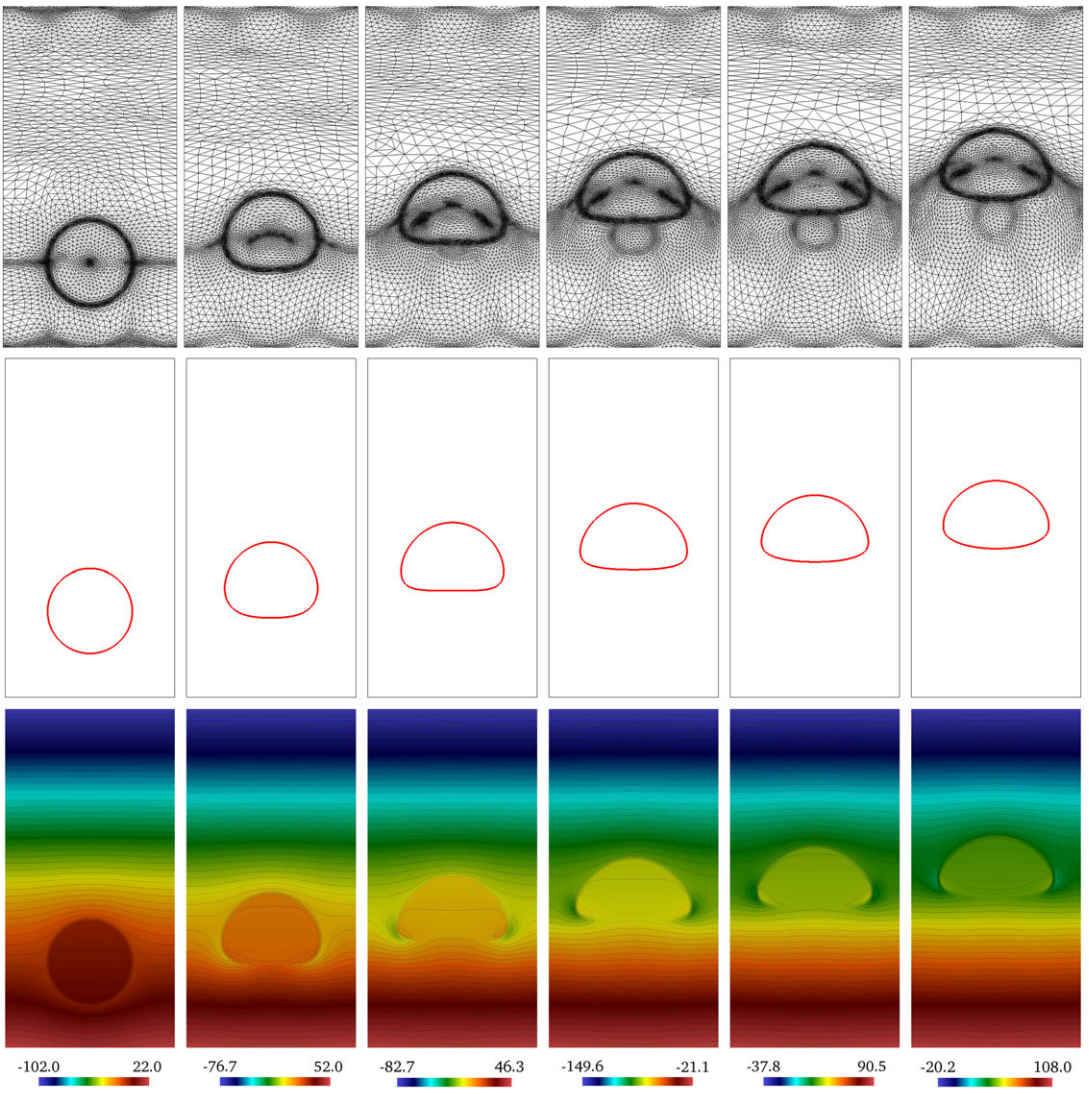

Figure 5. Example 1: Snapshots showing the moderately anisotropic adapted meshes, the pressure isocontour lines, and the corresponding shapes Γ of the rising bubble at times $t =$ 0.214, 1.178, 1.75, 2.321, and 2.571,3, respectively. During the simulation period, the meshes are characterized by $1/\tilde{h} \approx 37.1$, $1/h_{\min} \approx 58.8$, and $1/h_{\max} \approx 10$.

DOI: 10.1002/nme

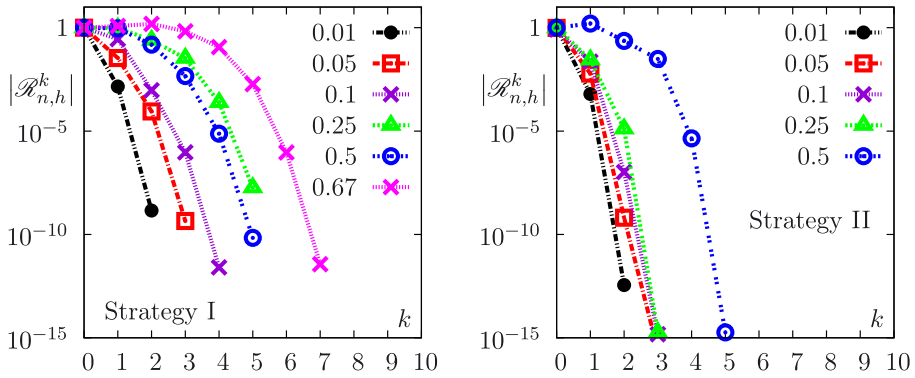

Figure 6. Example 1: Convergence curves of the residuals using Strategies I and II in the semi-logarithmic scale.

Similarly, the evolution of the rise velocity in our simulations is agreeing well with the reference results (Figure 4, left). The close-up view in Figure 4 (right) clearly shows the agreement, while complete convergence seems to be achieved. In Figure 5, we provide some snapshots showing the temporal evolution of the interface during the simulation period and the corresponding adapted meshes. Observe that the initial bubble deforms while staying compact for all $t \in (0,3)$. During the simulation period, the meshing procedure clearly allows to generate high-density meshes around the free interface and in particular areas of high norm of the strain tensor. However, big mesh elements are generated in front of the rising bubble where the velocity field has a more regular pattern; see also the movies in the supporting information.

We now proceed with a quantitative comparison of the aforementioned benchmark quantities against some computational results available in the published literature using different computational frameworks, as detailed in Hysing *et al.* [34], Štrubelj *et al.* [7], Klostermann *et al.* [9], and Doyeux *et al.* [8]. Several mesh refinement levels are used, and computational results are provided in Table I showing overall very consistent results. The mean mesh size and the smallest edge of triangles $K \in \mathcal{T}_h$ are also provided.

Similar to [34], we measure the temporal evolution of the different benchmark quantities q_t against a reference solution, referred to as q_t^r, obtained with the finest meshes and the smallest time step size. We compute the following relative errors:

$$\|e\|_1 = \frac{\sum_{t=1}^{\mathrm{NTS}} |q_t^r - q_t|}{\sum_{t=1}^{\mathrm{NTS}} |q_t^r|}, \qquad \|e\|_2 = \sqrt{\frac{\sum_{t=1}^{\mathrm{NTS}} |q_t^r - q_t|^2}{\sum_{t=1}^{\mathrm{NTS}} |q_t^r|^2}} \qquad \text{and} \qquad \|e\|_\infty = \frac{\max_t |q_t^r - q_t|}{\max_t |q_t^r|}.$$

In addition, we compute the convergence rate ROC for the desired quantities that evaluates the convergence of the method toward an approximate discrete solution:

$$\mathrm{ROC}_i = \frac{\log_{10}\left(\|e^{l-1}\|_i / \|e^l\|_i\right)}{\log_{10}\left(h^{l-1}/h^l\right)}, \qquad i \in \{1, 2, \infty\},$$

where e^l and e^{l-1} stand for the errors generated on consecutive meshes of sizes h^l and h^{l-1}, respectively. We consider the standard linear interpolation to account for the differences in the time step sizes.

Let us first consider structured triangular meshes. The relative errors are provided in Table II, showing a convergence rate of about 2 for the circularity in the l_1 and l_2 norms and a convergence rate of about 1.5 in the l_∞ norm. The convergence rate of the center of mass approaches 1.5 in all norms. When using the adaptive mesh technique aforementioned, results in Table III suggest that the

DOI: 10.1002/nme

Table I. Example 1: Comparisons with benchmark quantities and available published results.

$1/\tilde{h}$	17.5	35.9	72.1	143.4	**250.1**	Hysing *et al.* [34]	Štrubelj *et al.* [7]	Klostermann *et al.* [9]	Doyeux *et al.* [8]
$1/h_{\min}$	25.1	57.9	90.5	195.3	**334.4**				
$/\phi_{\min}$	0.9213	0.9079	0.9046	0.9013	**0.9013**	0.9012 ± 0.0001	0.8876	0.9044	0.9001
$t\|_{\ni=\ni_{\min}}$	2.0000	1.9328	1.8964	1.8851	**1.8835**	1.8895 ± 0.0145	1.8915	1.9625	1.9000
$V_{c,\max}$	0.2391	0.2410	0.2418	0.2419	**0.2419**	0.2419 ± 0.0002	0.2457	0.2348	0.2412
$t\|_{V_c=V_{c,\max}}$	1.0357	0.9743	0.9428	0.9290	**0.9281**	0.9263 ± 0.0050	0.9235	0.9516	0.9248
$Y_c(\mathrm{t}=3)$	1.0918	1.0812	1.0810	1.0809	**1.0809**	1.0808 ± 0.0009	1.0679	1.0696	1.0815

The results are obtained with the mesh adaptation technique.

DOI: 10.1002/nme

Table II. Example 1: Convergence results for the circularity ϕ and the center of mass Y_c.

	$1/h$	$\|e\|_1$	ROC_1	$\|e\|_2$	ROC_2	$\|e\|_\infty$	ROC_∞
	17	1.4875E-2		1.1948E-2		2.5962E-2	
ϕ	39	3.9292E-3	1.603	4.4786E-3	1.812	7.4143E-3	1.509
	72	1.1574E-3	1.993	1.3324E-3	1.977	2.6123E-3	1.701
	143	3.2724E-4	1.841	3.7146E-4	1.861	8.4124E-4	1.651
	17	2.0930E-2		2.1738E-2		3.0102E-2	
Y_c	39	6.8396E-3	1.347	7.1977E-3	1.331	6.7916E-3	1.793
	72	2.5769E-3	1.592	2.9968E-3	1.429	2.9706E-3	1.387
	143	9.2256E-4	1.497	1.1577E-3	1.386	1.2075E-3	1.312

The results are obtained with structured meshes.

Table III. Example 1: Convergence results for the circularity ϕ and the center of mass Y_c.

	$1/\tilde{h}$	$\|e\|_1$	ROC_1	$\|e\|_2$	ROC_2	$\|e\|_\infty$	ROC_∞
	17.5	2.5921E-3		2.6717E-3		3.0156E-3	
ϕ	38.9	7.5355E-4	1.547	7.7084E-4	1.556	8.5744E-4	1.574
	72.1	1.6915E-4	2.421	1.9821E-4	2.201	2.9010E-4	1.756
	143.4	3.1350E-5	2.451	4.5012E-5	2.402	6.5744E-5	2.159
	17.5	9.9752E-3		1.0110E-2		3.9854E-2	
Y_c	38.9	1.9744E-3	2.028	2.2811E-3	1.864	4.8755E-3	2.630
	72.1	3.9801E-4	2.595	5.2915E-4	2.368	8.9808E-4	2.741
	143.4	7.0714E-5	2.512	8.7084E-5	2.624	1.7741E-4	2.359

The results are obtained with structured meshes.

circularity approaches a convergence order of 2.5 in the l_1 and l_2 norms, while a rate of 2 is observed in the l_∞ norm. The convergence order of the center of mass is of about 2.5 in all norms.

In the following, we proceed with a comparative study between Strategies I and II.

The first test concerns the convergence properties of both strategies, and we particularly highlight the convergence rates of each Newton variant. Let us consider an initial mesh having $25'600$ elements, and we perform simulations using fixed time steps. The residuals $\mathcal{R}(\Xi_n^k)$ at iteration k are computed in the L^∞ norm.

Given the same initial conditions, we perform computations for various time step sizes Δt, and we report the residuals' convergence curves in Figure 6. The corresponding convergence rates are expressed as

$$\frac{\ln\left(|\mathcal{R}(\Xi_n^k)|/|\mathcal{R}(\Xi_n^{k-1})|\right)}{\ln\left(|\mathcal{R}(\Xi_n^{k-1})|/|\mathcal{R}(\Xi_n^{k-2})|\right)}, \qquad \text{for } k \geqslant 2. \tag{4.1}$$

Observe that, overall, the convergence is very fast for both strategies. The convergence rates in Table IV show that the quadratic convergence is obtained with Strategy I, while the third-order convergence rate is obtained with Strategy II. These correspond to the expected behavior of the two strategies and confirm that the tangent problem is correctly derived and implemented. Notice that any small error would deteriorate the quadratic or cubic behavior, and rather lead to a linear convergence slope. Gradually increasing the time step impacts the performance of the Newton method. Indeed, for large values of Δt, the starting value for the Newton loop becomes far from the expected solution, yielding a plateau where the residual first decreases slowly; it converges subsequently with

DOI: 10.1002/nme

Table IV. Example 1: Convergence rates of the residuals (4.1) using Strategies I and II for various time step sizes.

k	$\Delta t = 0.01$	0.05	0.1	0.25	0.5	0.67
			Strategy I			
2	2.109	1.766	4.455	45.209	27.229	1.220
3		2.050	1.218	1.482	1.917	3.512
4			1.851	2.453	1.835	2.163
5				1.926	1.807	2.329
6						1.847
7						1.637
			Strategy II			
k	$\Delta t = 0.01$	0.025	0.05	0.1	0.25	0.5
2	2.883	3.106	3.154	3.435	2.148	4.207
3				1.470	3.248	1.036
4						4.418
5						2.440

Table V. Example 1: Comparison between the computing times when using Strategies I and II for several values of the time step size.

Δt	CPU (Strategy I)	CPU (Strategy II)
0.001	8.9481	12.9463
0.01	15.9305	15.8132
0.1	27.4434	20.6427
0.15	33.7927	22.9934
0.2	42.9935	26.9260
0.275	59.6415	34.8457

the expected quadratic or cubic behavior. Above a threshold value of Δt, a complete breakdown of the algorithm holds. Remark that there exist numerical strategies that improve the Newton convergence when the initial guess is far from the solution such as the damped Newton algorithm, but that is definitely beyond the goal of this work.

We now focus on the computational cost for both strategies. To have a quantitative comparison, we evaluate the wall clock time for serial simulations that indicates, together with the errors evaluation, how much effort is needed to establish a certain accuracy. Further insights into the parallel performances and scalability will be only provided subsequently in the three-dimensional case.

We perform computations with the same structured mesh having 25 600 triangles. In Table V, we plot the timings of the computations for both strategies with respect to the time step size. We can observe that Strategy I is slightly cheaper than Strategy II when the time step is very small (almost for $\Delta t \leqslant 0.01$). In such a situation, both algorithms converge in less than two iterations. Strategy II is then more expensive than Strategy I because Strategy II requires one more residual's assembly and resolution of the linear system in each Newton subiteration. However, Strategy II becomes more beneficial for larger time steps, thanks to the third-order convergence of this method.

Thereafter, we compare the time step size used in the present method and the time step limit from the stability criterion. Only structured meshes are used, and the time step adaptation is disabled. A set of simulations for successively refined meshes is performed, and we display the maximum time steps enabled for both strategies. Using an explicit decoupling scheme where the capillary term is

DOI: 10.1002/nme

considered as a right-hand side in the momentum equation leads to a stability condition given by the convective time step limit. It imposes restrictions on the time step size: $\Delta t_{\text{CFL}} < \sqrt{\frac{\langle\rho\rangle h^3}{2\pi\gamma}}$, where $\langle\rho\rangle$ is the average fluid density at the interface [29, 31]. We perform numerical simulations using both strategies, and we report in Table VI the maximum time step sizes. The results illustrate the gain obtained by using the fully implicit schemes, for which we can increase Δt up to almost 300 times Δt_{CFL}. Moreover, we observe that, although Strategy II converges faster than Strategy I, Strategy I allows to use larger values of Δt than Strategy II.

Let us focus now on the adaptive time scheme, and perform simulations using the same structured mesh. In Figure 7, we compare the adapted values of Δt for both strategies against the capillary time step restriction for explicit schemes, and we additionally report the corresponding numbers of Newton subiterations. We mainly observe that Δt has almost similar values when using both strategies. However, while Strategy II generally converges after two or three iterations, Strategy I usually needs five iterations.

Consequently, although the quadratically convergent Strategy I allows higher time steps, Strategy II with the third-order convergence performs globally better. Indeed, Strategy II is computationally cheaper, thanks to its cubic convergence, and we do not usually use too large time steps as those allowed by Strategy I. In what follows, Strategy II will be used in our computations.

Table VI. Example 1: Comparison between the maximal time step $\Delta t_{\text{max}}$for Strategies I and II and the time step limit from the stability criterion for several values of the mesh size h.

Mesh size	Δt_{max}(Strategy I)	Δt_{max}(Strategy II)	Δt_{CFL}
1/15	0.89	0.51	0.032
1/30	0.91	0.50	0.011
1/60	0.86	0.44	0.0041
1/120	0.435	0.325	0.0014
1/240	0.172	0.114	0.0005

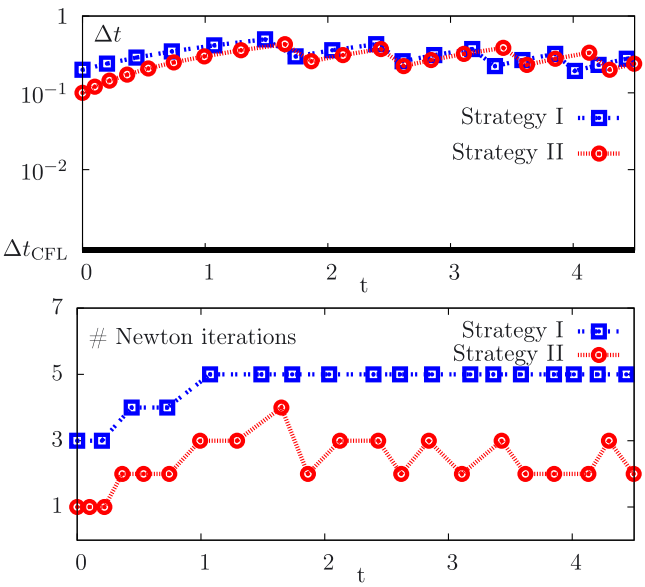

Figure 7. Example 1: The adaptive time step sizes (top) and the number of Newton iterations (bottom) during the simulation period for both strategies. The semi-logarithmic scale is used in the first figure.

DOI: 10.1002/nme

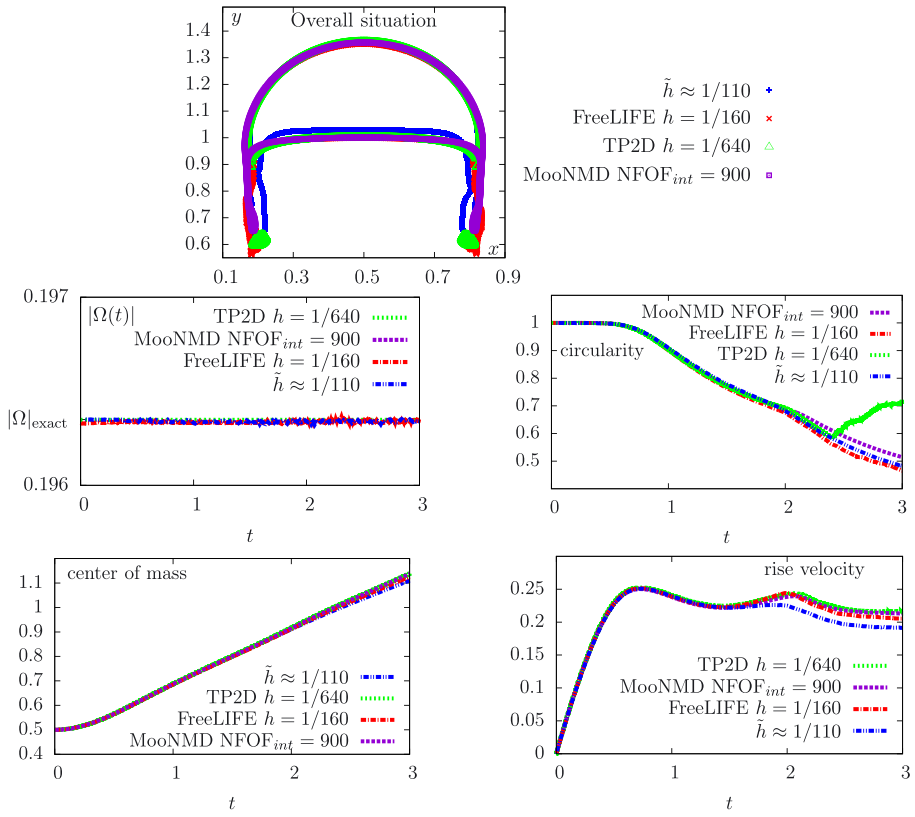

Figure 8. Example 1: Shrinked bubble and comparison with available computational results in [34]. Top: Final shapes at $t = 3$. Middle: Temporal evolution of the circularity and the bubble's area. Bottom: Temporal evolution of the center of mass and rise velocity.

In this last paragraph, we consider the second setup of the skirted bubble described in Hysing *et al.* [31]. We only present a sample result obtained using the adaptive time steps scheme and the mesh adaptation procedure, without making any grid convergence study. We consider the same computational domain and initial conditions as adopted in the previous test case. The physical parameters are $\rho_o = 10^3$, $\rho_i = 1$, $\mu_o = 10$, $\mu_i = 0.1$, and $\gamma = 1.96$, while the dimensionless parameters are $\rho^\star = 10^{-3}$, $\mu^\star = 10^{-2}$, Re = 35, Eo = 125 and Ca ≈ 3.571.

Figure 9 provides successive snapshots showing the adapted meshes and the corresponding shapes of the interfaces; see also Movie 6 in the supporting information. The adapted mesh provides high mesh density and then more accurate computations in the zones of small filaments. In Figure 8, the bubble's shape at $t = 3$ is compared with the shapes obtained by the different groups using the finest meshes in [31]. Moreover, Figure 8 plots the temporal development of the bubble's area $|\Omega(t)|$, the circularity $\phi(t)$, the center of mass $Y_c(t)$, and the rise velocity V_c. In Figure 8, an acceptable agreement is generally observed (Figure 9).

4.2. *Example 2: Oscillating bubble dynamics*

The second example concerns the relaxing viscous bubble subject to surface tension [7, 60–63]. We consider the setup described in [64, 65] of the ellipsoidal interface having initially the semi-major axis $a = 0.75$ and the semi-minor axis $b = 0.5$, respectively, in the x-direction and y-direction and immersed in a fluid having the density $\rho = 1$ and the viscosity $\mu = 1$. We set the tension coefficient to $\gamma = 10$. The interface is initially centered in a square computational domain $\Lambda = [-1.5,1.5] \times [-1.5,1.5]$. Homogeneous Dirichlet boundary conditions are considered for the velocity. Under the effect of the surface tension, the bubble starts oscillating, and the capillary force

DOI: 10.1002/nme

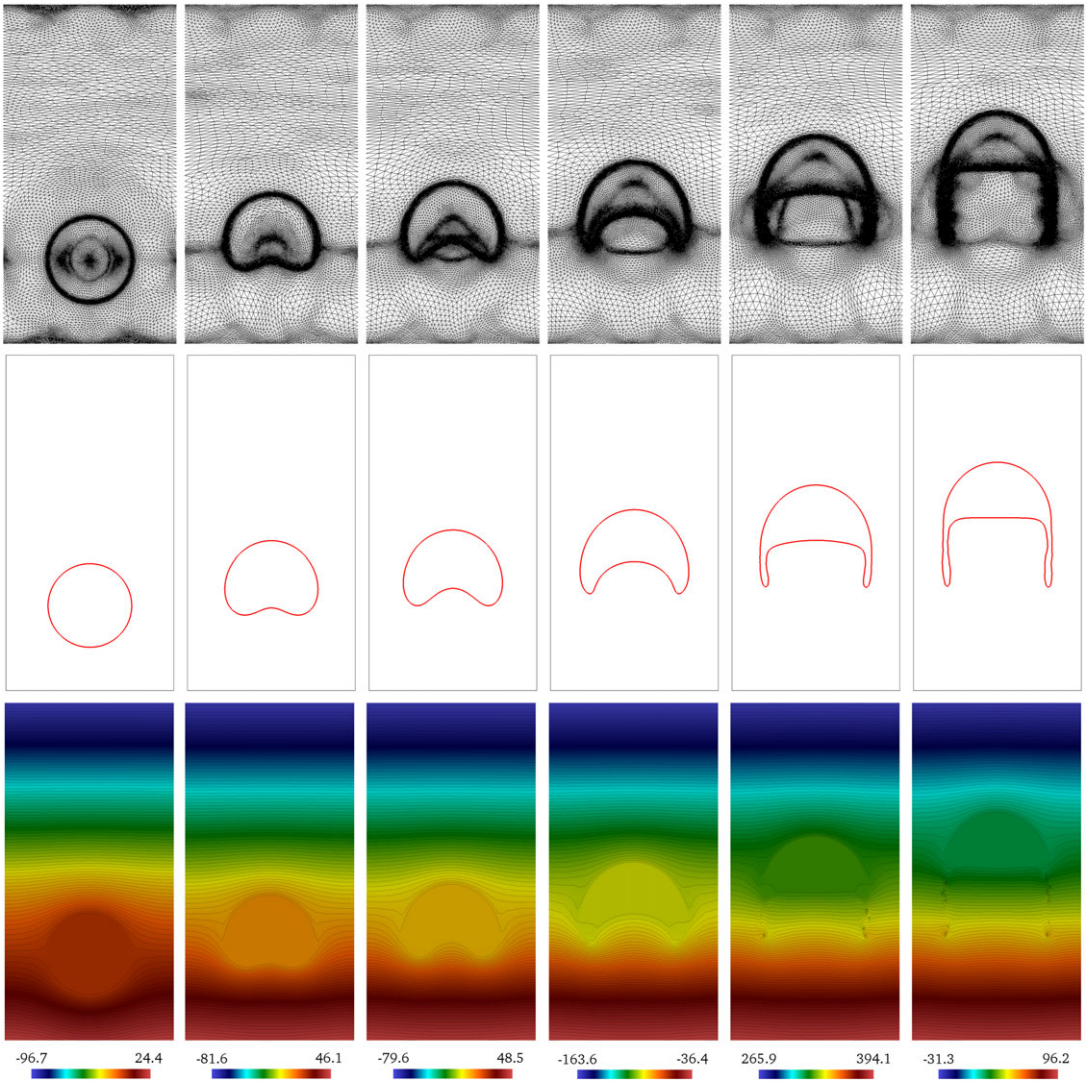

Figure 9. Example 1: Shrinked bubble. Snapshots showing the almost isotropic meshes, the pressure isocontour lines, and the corresponding shapes Γ of the rising bubble at times $t = 0.135, 0.765, 1.065, 1.635$, and 2.340,3, respectively. The meshes are characterized by $1/\tilde{h} \approx 110$, $1/h_{\min} \approx 138.1$, and $1/h_{\max} \approx 67.5$.

shall bring the ellipse back to an equilibrium circular steady state having the same area and the radius $R_\infty = \sqrt{ab} \approx 0.61237$, in which the capillary force vanishes. We observe the relaxation of the bubble in the time interval (0, 6.5).

The purpose of this test case is to perform a comparison of the stability properties between the present fully implicit and a fully explicit scheme. For such an explicit scheme, the time step is first bounded by the convective time step limitation Δt_{CFL}, whereas the most limiting time step at the end is the viscous time step limit corresponding to small velocities.

We consider the explicit scheme detailed in Appendix A, in which the surface tension appears as a right-hand side in the momentum equation. For all simulations in both explicit and implicit cases, we consider the same time step size $\Delta t = 0.012$ without applying the adaptive time scheme. The computational domain is discretized on a family of regular meshes without adapting the mesh. The successively refined meshes have the mesh size $h < 1/20$. We notice that the stability criterion is violated for all meshes, and the simulations will obviously fail depending on the mesh size.

Figure 10 shows the relaxation of the bubble and the temporal evolution of the bubble's axes (i.e., the vertical and horizontal radii) for several mesh sizes. The numerical results clearly depict that the

DOI: 10.1002/nme

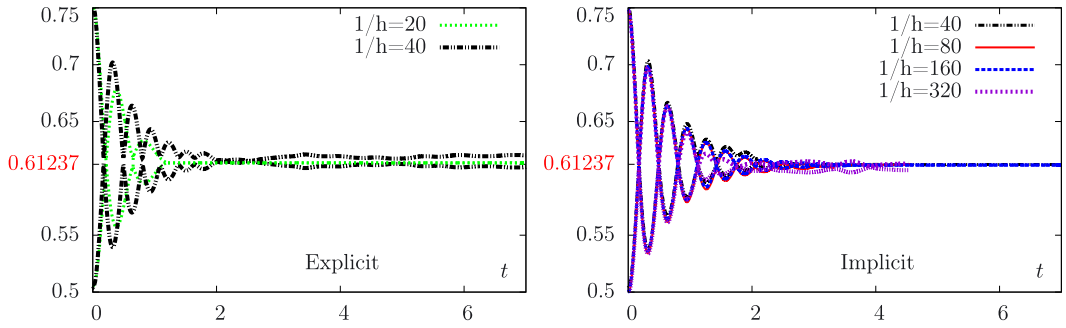

Figure 10. Example 2: Time evolution of the horizontal and vertical radii of the relaxing bubble for different values of the time step size. Left: Fully explicit scheme. Right: Fully implicit scheme.

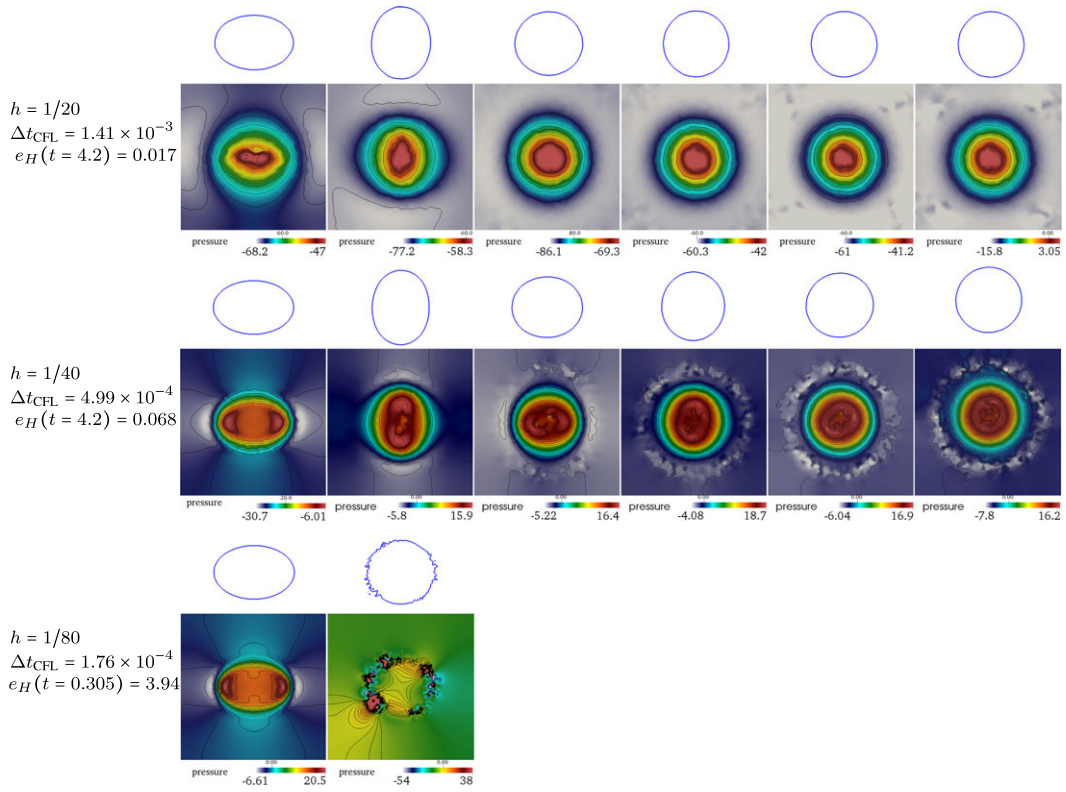

Figure 11. Example 2: Temporal evolution of the shape of the oscillating bubble with the fully explicit scheme for $t = 0$, 0.305, 0.617, 0.948, 1.259, and 4.2 (from left to right).

explicit treatment of the capillary force leads to severe time step restrictions. However, the stability is significantly improved with the fully implicit scheme when using large time steps and relatively fine meshes.

In Figures 11 and 12, we provide some snapshots showing the shapes of the bubble during the relaxation. To better infer the limit of stability, we also plot the pressure profile and some isocontour lines, and we compute some errors when the equilibrium state is reached. Let $\Delta p|_\Gamma$ represent the pressure jump across the interface and $H_\infty = 1/R_\infty$ be the exact mean curvature on Γ in the steady state. Let $\Delta\hat{p}\big|_\Gamma$ denote the jump of pressure across Γ obtained at the equilibrium state with $h = 1/350$ and $\Delta t = 2 \times 10^{-3}$. We introduce the following errors:

DOI: 10.1002/nme

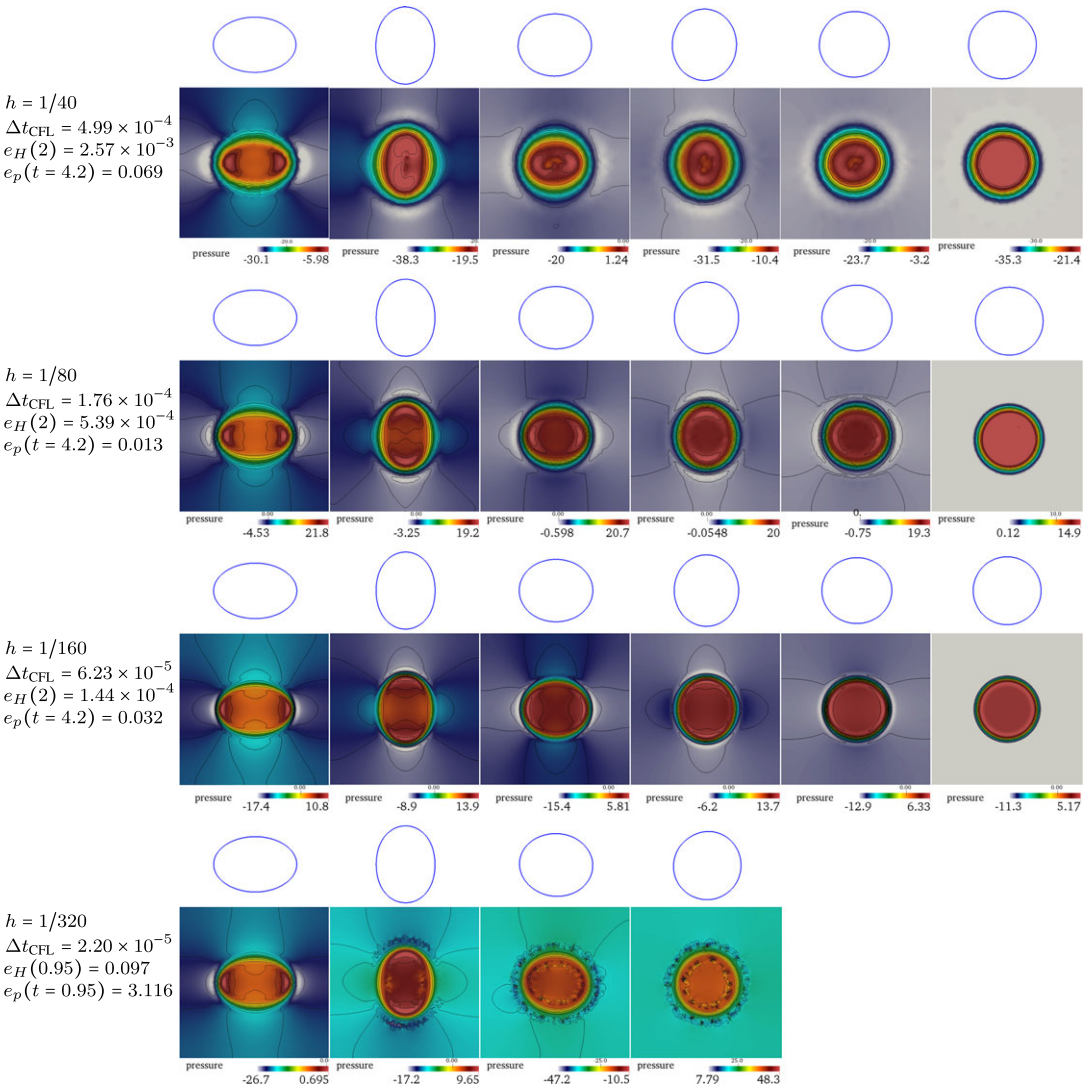

Figure 12. Example 2: Temporal evolution of the shape of the oscillating bubble with the fully implicit scheme for $t = 0$, 0.305, 0.617, 0.948, 1.259, and 4.2 (from left to right).

$$e_H = \sqrt{\frac{\int_\Lambda \left| div\boldsymbol{n} - H_\infty \right|^2 \delta_\varepsilon(\varphi) |\boldsymbol{\nabla}\varphi|}{2\pi R_\infty^{-1}}} \qquad \text{and}$$

$$e_p(t) = \frac{\left|\Delta p|_\Gamma(t) - \Delta \hat{p}|_\Gamma\right|}{\left|\Delta \hat{p}|_\Gamma\right|} \qquad \text{with} \qquad \Delta p\big|_\Gamma = \left| \frac{\int_\Lambda \mathcal{H}_\varepsilon(\varphi) p}{\int_\Lambda \mathcal{H}_\varepsilon(\varphi)} - \frac{\int_\Lambda (1 - \mathcal{H}_\varepsilon(\varphi))\, p}{\int_\Lambda (1 - \mathcal{H}_\varepsilon(\varphi))} \right|$$

When using the fully explicit scheme, the solution remains stable when using a mesh size $h = 1/20$. Indeed, that corresponds to a time step size that is almost eight times larger than Δt_{CFL}. Further mesh refinement results in non-physical numerical oscillations that pollute the solution and lead finally to a complete breakdown of the algorithm.

However, Figure 12 shows the stability of the solution is maintained for significantly finer meshes, and we observe that it is still possible to obtain a stable solution when using time step sizes that exceed the capillary time step size by almost a factor of 160. For a mesh resolution $h = 1/320$, we observe that oscillations start to appear for $t \geqslant 0.3$. The pressure isocontour lines and the error e_p clearly show that the solution is no longer stable.

DOI: 10.1002/nme

Table VII. Example 2: Temporal error history associated with the time approximation using BDF2.

Δt	$e_{\Delta t,1}(\boldsymbol{u})$	$\mathrm{ROC}_{\Delta t,1}(\boldsymbol{u})$	$e_{\Delta t,0}(p)$	$\mathrm{ROC}_{\Delta t,0}(p)$	$e_{\Delta t,0}\left(\mathcal{H}_\epsilon(\varphi)\right)$	$\mathrm{ROC}_{\Delta t,0}\left(\mathcal{H}_\epsilon(\varphi)\right)$
5×10^{-3}	0.50086		0.20515047		0.014140	
2.5×10^{-3}	0.13235	1.920	0.05470264	1.907	0.004035	1.809
1.25×10^{-3}	0.03509	1.915	0.01418740	1.947	0.001153	1.807
6.25×10^{-4}	0.00959	1.871	0.00374819	1.920	0.000336	1.775

Errors are computed on a fine mesh of size $h = 1/100$.

Notice that although the implicit method allows us to maintain stability for significantly larger time steps, it is not always useful to use too large values in practice. In fact, using too large time steps may not help to capture the physics of the problem.

Thereafter, we verify the temporal convergence of the proposed numerical strategy. We consider the time horizon $t \in (0, T = 1)$. Let q_h^N represent the discrete approximation of a generic field q evaluated at the final time $t^N = T$. We compute the errors corresponding to the temporal discretization of the field q with respect to a reference solution $\hat{q}_h^N$ obtained with the finest time step $\Delta t = 3.125\times10^{-4}$ as follows:

$$e_{\Delta t,i}(q) = q_h^N - \hat{q}_{h\ i,\Omega}^N \quad \text{and} \quad \mathrm{ROC}_{\Delta t,i}(q) = \frac{\log_{10}\left(e_{\Delta t,i}(q)/e_{\overline{\Delta t},i}(q)\right)}{\log_{10}\left(\Delta t/\overline{\Delta t}\right)},$$

where $i \in \{0,1\}$, and Δt and $\overline{\Delta t}$ represent two consecutive time steps. In Table VII, we provide the errors corresponding to the temporal discretization for successively refined time steps. Accordingly, a second-order convergence behavior is observed.

4.3. Example 3: Three-dimensional test case

In this example, we present numerical simulations showing the applicability of the present methodology in the three-dimensional case ($d = 3$). We emphasize that the aim of this test case is not to perform grid convergence studies. To that end, we consider a setup of the rising bubble analogous to that described in Example 4.1 with a larger initial bubble. Moreover, we will present a study of the parallel performances.

The computational domain is $\Lambda = (0,1)\times(0,1)\times(0,2)$, and the time interval for the computations is (0,1.2). The bubble is initially circular, centered in (0.5, 0.5, 0.5) and has a radius equal to 0.3. The fluid is assumed to be initially at rest. The dimensionless parameters are given by $\rho^\star = 0.1$, $\mu^\star = 0.1$, $\mathrm{Re} = 35$, $\mathrm{Eo} = 52.5$, and $\mathrm{Ca} = 1.5$. In Figure 13, we provide snapshots of the numerical results, showing the interface Γ (colored in red) and the velocity field. The bubble at final time $T = 1.2$ has a nonconvex shape.

We thereafter study the scalability and parallel performance of the numerical solver. However, we do not aim to improve the parallel properties of the solver with respect to the state-of-the-art solvers. The scalability represents an important concept, especially in the three-dimensional case, and measures the ability of the parallel implementation to maintain a constant efficiency. We first introduce the speedup, which compares the computational time required on a parallel architecture with the time required by a sequential implementation. We characterize the parallel performances through the strong scalability and the weak scalability.

At each Newton iteration, the discretized problem is solved in three steps: We first assemble the linear system, then the Jacobian matrix is factorized, and we finally solve the linear system. Thereafter, we consider the aforementioned test case, and we study the scaling efficiency for each step. We generate several unstructured meshes using the software GMSH. The nomenclature, number of tetrahedra, and the corresponding degrees of freedom are provided in Table VIII. Figure 14 plots on the logarithmic scale the computational timings, that is, CPU, with respect to the degrees of freedom per processor for each step and for the different meshes.

DOI: 10.1002/nme

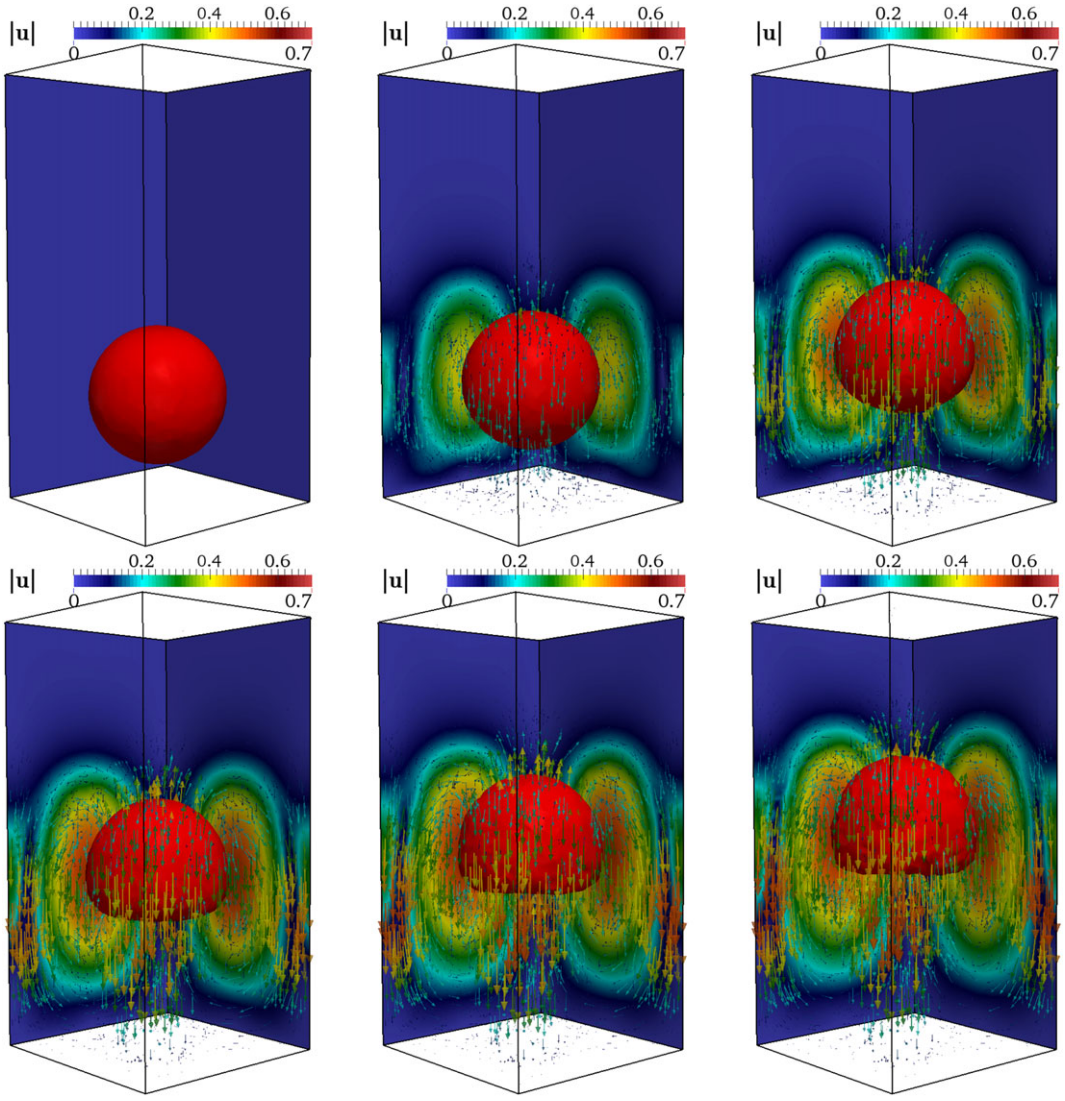

Figure 13. Example 3: Snapshots showing the shapes Γ of the rising bubble and the velocity field in the vertical mid-planes of the bubble at times $t \in \{0, 0.3, 0.6, 0.85, 1.05, 1.2\}$, respectively.

Regarding the strong scalability, it measures, for a fixed problem size, how much the computing time changes if the number of processors increases. A perfect strong scalability corresponds to a slope equal to one in the logarithmic scale. Indeed, a two-time faster execution holds if the number of processors is doubled. Figure 14 shows that the assembly and the factorization steps are strongly scalable, whereas strong scalability is deteriorated when solving the linear system. That does not affect significantly the strong scalability of the entire computation, as the time required to solve the linear system is always very small compared with the previous steps.

Regarding the weak scalability, it studies the difference in the computational time with respect to the number of processors for a fixed problem size per CPU. A perfect weak scalability corresponds to an exact overlap of the curves corresponding to the use of different meshes. Figure 14 shows very close curves during the assembly step when using the different meshes. That results in almost a perfect weak scalability. However, the weak scalability of the Jacobian factorization and resolution of the linear system are not as good as the assembly step because the different timing curves are not superimposed.

DOI: 10.1002/nme

Table VIII. Example 3: Nomenclature and characteristics of the different meshes used for the scalability study.

Mesh code	Number of tetrahedra	Number of DOFs
S	20'893	171'729
M	49'524	389'278
F	160'894	1'218'199

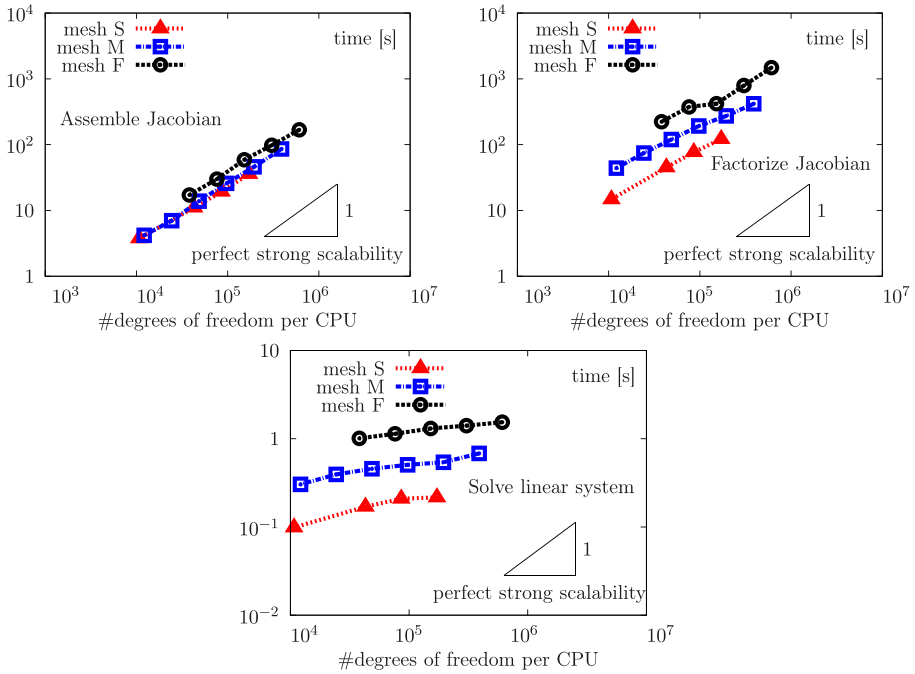

Figure 14. Example 3: Parallel performances of the numerical method. Timings for the assembly, the factorization, and the resolution of the linear system using the meshes listed in Table VIII and number of CPUs.

Without claiming to be exhaustive, we provide qualitative comparison of the parallel performances of the present solver with respect to other similar solvers using the level set method. Compared with the level set solver implemented within the C++ library LifeV in a finite element framework and tested on similar mesh resolutions [66], we observe similar good scaling properties for the assembly of the linear system. Both the strong and weak scalabilities are similarly deteriorated when solving the linear system. In [67], a different method is presented, and the fluid interface solver uses a hybrid front tracking/level set method: the level contour reconstruction method, while the code is written in Fortran 2003. The parallel performances are tested on highly refined grids, showing better performances in weak scalability.

5. CONCLUSION

In this paper, we have presented new fully implicit and monolithic approaches to solve free surface problems with surface tension. For the first time, two numerical strategies based on the use of two variants of the Newton method in a finite element framework have been introduced, and we have reported computational results in both two-dimensional and three-dimensional cases. The two strategies require one assembly of the Jacobian system, while Strategy II requires one more evaluation of the global residual. Efficient derivation and implementation of the tangent problem in both cases

DOI: 10.1002/nme

have been proved through the study of the convergence properties. Numerical investigations have been performed, showing second-order and third-order convergences for both strategies, as expected from a theoretical viewpoint. Fully implicit methods feature to be unconditionally stable, and we have shown numerically that stability is ensured for considerably larger time steps compared to the explicit decoupling strategy most commonly used in the published literature [24, 29, 31]. In addition, we have observed that, although Strategy I allows the use of higher time steps, Strategy II is in general computationally cheaper and represents our preferrable approach.

The improvements and extensions of the present method should be further explored. In particular, we are focusing on the development of inexact variants of the Newton method that can be efficient and computationally cheaper compared with the present strategies. Because the Newton method presents only local convergence properties, that is, the starting values must be close enough to the solution, we are currently investigating globalized Newton variants based on damping techniques to allow the use of larger time steps. Moreover, future work should investigate the robustness of the present method in the case of high Reynolds numbers. Last but not least, we also foresee the applicability of the proposed framework to the simulation of interfacial flows with topology changes.

The present approach is general and can be useful for a broad set of engineering problems with multiphase flows. In particular, the current developments are part of an ongoing work to study the hydrodynamics of lipid bilayer vesicles mimicking red blood cells under bending forces in small capillaries where the Reynolds number is small enough for the Stokes limit to be valid [68–70]. The vesicle problem is highly nonlinear, and the bending force includes fourth-order derivatives with respect to the cell shape [71]. Fully explicit decoupling strategies are always used in the published literature, yielding a more severe stability condition than the one characterizing the capillary problem. The derivation and study of fully implicit strategies remain a challenging topic, and the present approach can provide an efficient way to implicitly solve the vesicle problem.

APPENDIX A: EXPLICIT SCHEME FOR A COMPARATIVE STUDY

To compare the present fully implicit method with respect to the explicit methods, we introduce the following scheme in which an explicit treatment of surface tension holds, while the backward differentiation scheme of second order approximates the time derivative terms, and an explicit time discretization of the inertia term is considered.

The explicit algorithm consists in decoupling and solving the fluid problem and the level equation in a segregated fashion. The velocity and pressure are first computed using a monolithic approach for the Navier–Stokes system, in which the surface tension force is added as a source term in the right-hand side of the momentum equation. The level set function is subsequently advected using the computed velocity. Finally, the mean curvature is computed and projected in the appropriate finite element space. The semi-discrete problem reads:

(i) Given φ_{n-1}, find the velocity $\boldsymbol{u}_n \in \mathbb{V}(\boldsymbol{u}_b)$ and pressure $p_n \in \mathbb{Q}$ such that

$$\begin{aligned} \mathrm{Re}\rho_\varepsilon(\varphi_{n-1})\left(\frac{3\boldsymbol{u}_n - 4\boldsymbol{u}_{n-1} + \boldsymbol{u}_{n-2}}{2\Delta t} + (\boldsymbol{u}_{n-1}\cdot\boldsymbol{\nabla})\,\boldsymbol{u}_n\right) - \mathbf{div}\big(2\mu_\varepsilon(\varphi_{n-1})\mathbf{D}\,(\boldsymbol{u}_n)\big) \\ + \boldsymbol{\nabla}\, p_n = \frac{1}{\mathrm{Ca}} H_{n-1}\frac{\boldsymbol{\nabla}\,\varphi_{n-1}}{|\,\boldsymbol{\nabla}\,\varphi_{n-1}|}\delta_\varepsilon\,(\varphi_{n-1}) + \mathrm{Re}\rho_\varepsilon(\varphi_{n-1})\boldsymbol{g}, \end{aligned} \tag{A.1}$$

$$div\,\boldsymbol{u}_n = 0. \tag{A.2}$$

(ii) *Given* $\boldsymbol{u}_n$, *find the level set function* $\varphi_n \in \mathbb{X}(\varphi_{n-1})$ *such that*

$$\frac{3\varphi_n - 4\varphi_{n-1} + \varphi_{n-2}}{2\Delta t} + \boldsymbol{u}_n \cdot \nabla\varphi_n = 0. \tag{A.3}$$

DOI: 10.1002/nme

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support by the Swiss National Science Foundation through the grant 320030-149567. The authors would like to thank the three anonymous reviewers for their insightful comments that help improve the quality of the paper.

REFERENCES

1. Josserand C, Zaleski S. Droplet splashing on a thin liquid film. *Physics of Fluids* 2003; **15**(6):1650–1657.
2. Eggers J, Villermaux E. Physics of liquid jets. *Reports on Progress in Physics* 2008; **71**(3):036601. (Available from: https://doi.org/10.1088/0034-4885/71/3/036601).
3. Coantic M. A model of gas transfer across air–water interfaces with capillary waves. *Journal of Geophysical Research: Oceans* 1986; **91**(C3):3925–3943.
4. Fuster D, Agbaglah G, Josserand C, Popinet S, Zaleski S. Numerical simulation of droplets, bubbles and waves: state of the art. *Fluid Dynamics Research* 2009; **41**(6):065001. (Available from: https://doi.org/10.1088/0169-5983/41/6/065001).
5. Tryggvason G, Bunner B, Esmaeeli A, Juric D, Al-Rawahi N, Tauber W, Han J, Nas S, Jan Y-J. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics* 2001; **169**(2):708–759.
6. Gross S, Reusken A. *Numerical Methods for Two-Phase Incompressible Flows*, vol. 40. Springer Berlin Heidelberg, 2011.
7. Štrubelj L, Tiselj I, Mavko B. Simulations of free surface flows with implementation of surface tension and interface sharpening in the two-fluid model. *International Journal of Heat and Fluid Flow* 2009; **30**(4):741–750.
8. Doyeux V, Guyot Y, Chabannes V, Prud'homme C, Ismail M. Simulation of two-fluid flows using a finite element/level set method. Application to bubbles and vesicle dynamics. *Journal of Computational and Applied Mathematics* 2013; **246**:251–59.
9. Klostermann J, Schaake K, Schwarze R. Numerical simulation of a single rising bubble by VOF with surface compression. *International Journal for Numerical Methods in Fluids* 2013; **71**(8):960–982.
10. Demirdzic I, Perić M. Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries. *International Journal for Numerical Methods in Fluids* 1990; **10**(7):771–790.
11. Muzaferija S, Peric M. Computation of free-surface flows using the finite-volume method and moving grids. *Numerical Heat Transfer Part B-Fundamentals* 1997; **32**(4):369–384.
12. Pozrikidis C. Interfacial dynamics for Stokes flow. *Journal of Computational Physics* 2001; **169**(2):250–301.
13. Osher S, Fedkiw R. *Level Set Methods and Dynamic Implicit Surfaces*, vol. 153. Springer New York, 2003.
14. Losasso F, Fedkiw R, Osher S. Spatially adaptive techniques for level set methods and incompressible flow. *Computers & Fluids* 2006; **35**(10):995–1010.
15. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**(1):201–225.
16. Rudman M. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**(7):671–691.
17. Scardovelli R, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics* 1999; **31**(1):567–603.
18. Liu C, Shen J. A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method. *Physica D: Nonlinear Phenomena* 2003; **179**(3–4):211–228.
19. Shin S, Juric D. A hybrid interface method for three-dimensional multiphase flows based on front tracking and level set techniques. *International Journal for Numerical Methods in Fluids* 2009; **60**(7):753–778.
20. Marić T, Marschall H, Bothe D. lentfoam – a hybrid level set/front tracking method on unstructured meshes. *Computers & Fluids* 2015; **113**:20–31.
21. Popinet S. An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics* 2009; **228**(16):5838–5866.
22. Abgrall R, Beaugendre H, Dobrzynski C. An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques. *Journal of Computational Physics* 2014; **257, Part A**:83–101.
23. Laadhari A, Saramito P, Misbah C. An adaptive finite element method for the modeling of the equilibrium of red blood cells. *International Journal for Numerical Methods in Fluids* 2016; **80**(7):397–428.
24. Sussman M, Ohta M. A stable and efficient method for treating surface tension in incompressible two-phase flow. *SIAM: SIAM Journal on Scientific Computing* 2009; **31**(4):2447–2471.
25. Prosperetti A, Tryggvason G. *Computational Methods for Multiphase Flow*. Cambridge University Press: New York, 2007.
26. Bänsch E, Weller S. Linearly implicit time discretization for free surface problems. In *PAMM Proceedings in Applied Mathematics and Mechanics*, vol. 12, Darmstadt, Germany, 2012; 525–526.
27. Bänsch E, Weller S. A comparison of several time discretization methods for free surface flows. In *Proceedings of ALGORITMY*, Vysoké Tatry - Podbanské, Slovakia, 2012; 331–341.
28. Montefuscolo F, Sousa FS, Buscaglia GC. High-order ALE schemes for incompressible capillary flows. *Journal of Computational Physics* 2014; **278**:133–147.

DOI: 10.1002/nme

29. Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. *Journal of Computational Physics* 1992; **100**(2):335–354.
30. Dziuk G. Computational parametric Willmore flow. *Numerische Mathematik* 2008; **111**(1):55–80.
31. Hysing S. A new implicit surface tension implementation for interfacial flows. *International Journal for Numerical Methods in Fluids* 2006; **51**:659–672.
32. Raessi M, Bussmann M, Mostaghimi J. A semi-implicit finite volume implementation of the CSF method for treating surface tension in interfacial flows. *International Journal for Numerical Methods in Fluids* 2009; **59**(10):1093–1110.
33. Denner F, van Wachem BGM. Fully-coupled balanced-force VOF framework for arbitrary meshes with least-squares curvature evaluation from volume fractions. *Numerical Heat Transfer, Part B: Fundamentals* 2014; **65**(3):218–255.
34. Hysing S, Turek S, Kuzmin D, Parolini N, Burman E, Ganesan S, Tobiska L. Quantitative benchmark computations of two-dimensional bubble dynamics. *International Journal for Numerical Methods in Fluids* 2009; **60**(11):1259–1288.
35. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**(1):12–49.
36. Sussman M, Fatemi E. An efficient, interface preserving level set re-distancing algorithm and its application to interfacial incompressible fluid flow. *SIAM: SIAM Journal on Scientific Computing* 1998; **20**(4):1165–1191.
37. Gomez P, Hernandez J, Lopez J. On the reinitialization procedure in a narrow-band locally refined level set method for interfacial flows. *International Journal for Numerical Methods in Engineering* 2005; **63**(10):1478–1512.
38. Herrmann M. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *Journal of Computational Physics* 2008; **227**(4):2674–2706.
39. Merriman B, Bence JK, Osher SJ. Motion of multiple junctions: a level set approach. *Journal of Computational Physics* 1994; **112**(2):334–363.
40. Hartmann D, Meinke M, Schröder W. Differential equation based constrained reinitialization for level set methods. *Journal of Computational Physics* 2008; **227**(14):6821–6845.
41. Hartmann D, Meinke M, Schröder W. The constrained reinitialization equation for level set methods. *Journal of Computational Physics* 2010; **229**(5):1514–1535.
42. Laadhari A, Saramito P, Misbah C. Improving the mass conservation of the level set method in a finite element context. *Comptes Rendus Mathematique* 2010; **348**(9):535–540.
43. Laadhari A, Saramito P, Misbah C. Computing the dynamics of biomembranes by combining conservative level set and adaptive finite element methods. *Journal of Computational Physics* 2014; **263**(0):328–352.
44. Hairer E, Wanner G. *Solving Ordinary Differential Equations II, Stiff and Differential-algebraic Problems*, Springer Series in Computational Mathematics, vol. 14. Springer Berlin Heidelberg, 1996.
45. Saramito P. *Efficient C++ finite element computing with rheolef, version 6.7* (CNRS-CCSD), 2016. cel.archives& LWx02010;ouvertes.fr/cel&LWx02010;00573970/file/rheolef.pdf [Accessed: 2016-11-23].
46. Guermond J-L, Ern A. *Theory and Practice of Finite Elements*, vol. 159. Springer New York, 2004.
47. Morton KW, Priestley A, Suli E. Stability of the Lagrange–Galerkin method with non-exact integration. *ESAIM: Mathematical Modelling and Numerical Analysis–Modélisation Mathématique et Analyse Numérique* 1988; **22**(4):625–653.
48. Boukir K, Maday Y, Métivet B, Razafindrakoto E. A high-order characteristics/finite element method for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1997; **25**(12):1421–1454.
49. Laadhari A, Ruiz-Baier R, Quarteroni A. Fully Eulerian finite element approximation of a fluid–structure interaction problem in cardiac cells. *International Journal for Numerical Methods in Engineering* 2013; **96**(11):712–738.
50. Kou J, Li Y, Wang X. A modification of Newton method with third-order convergence. *Applied Mathematics and Computation* 2006; **181**(2):1106–1111.
51. Weerakoon S, Fernando TGI. A variant of Newton's method with accelerated third-order convergence. *Applied Mathematics Letters* 2000; **13**(8):87–93.
52. Kou J, Li Y, Wang X. Third-order modification of Newton's method. *Journal of Computational and Applied Mathematics* 2007; **205**(1):1–5.
53. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*, vol. 15. Springer New York, 1991.
54. Amestoy PR, Duff IS, l'Excellent J-Y. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering* 2000; **184**(2–4):501–520.
55. MUMPS. *Multifrontal massively parallel solver, version 5.0.1*. CERFACS, CNRS, ENS Lyon, 2015.
56. Amestoy PR, Guermouche A, l'Excellent J-Y, Pralet S. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing* 2006; **32**(2):136–156.
57. Amestoy PR, Duff IS, Koster J, l'Excellent J-Y. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM: Journal of Matrix Analysis and Applications* 2001; **23**(1):15–41.
58. Huang W. Metric tensors for anisotropic mesh generation. *Journal of Computational Physics* 2005; **204**:633–665.
59. Hecht F. Bidimensional anisotropic mesh generator, 2006. people.sc.fsu.edu/~jburkardt/data/bamg/bamg.html.
60. Stückrad B, Hiller WJ, Kowalewski TA. Measurement of dynamic surface tension by the oscillating droplet method. *Experiments in Fluids* 1993; **15**(4–5):332–340.
61. Vincent S, Caltagirone J-P. Test-case no 10: parasitic currents induced by surface tension (pc). *Multiphase Science and Technology* 2004; **16**(1–3):69–74.
62. Torres DJ, Brackbill JU. The point-set method: front-tracking without connectivity. *Journal of Computational Physics* 2000; **165**(2):620–644.
63. Fyfe DE, Oran ES, Fritts MJ. Surface tension and viscosity with Lagrangian hydrodynamics on a triangular mesh. *Journal of Computational Physics* 1988; **76**(2):349–384.

DOI: 10.1002/nme

64. Le DV. An immersed interface method for solving viscous incompressible flows involving rigid and flexible boundaries. *PhD Thesis*, Singapore-MIT Alliance, 2005.
65. Tan Z, Le DV, Li Zhilin, Lim KM, Khoo BC. An immersed interface method for solving incompressible viscous flows with piecewise constant viscosity across a moving elastic membrane. *Journal of Computational Physics* 2008; **227**(23):9955–9983.
66. Quinodoz S. Numerical simulation of orbitally shaken reactors. *Ph.D. Thesis*, EPFL, Lausanne, 2012. PUBLISHED.
67. Shin S, Chergui J, Juric D. A solver for massively parallel direct numerical simulation of three-dimensional multiphase flows. *ArXiv e-prints: 1410.8568* 2014, available at 1410.8568. http://adsabs.harvard.\ignorespacesedu/abs/2014arXiv1410.8568S.
68. Helfrich W. Elastic properties of lipid bilayers: theory and possible experiments. *Zeitschrift für Naturforschung C* 1973; **28**:693–703.
69. Pozrikidis C. Numerical simulation of the flow-induced deformation of red blood cells. *Annals of Biomedical Engineering* 2003; **31**:1194–1205.
70. Salac D, Miksis M. A level set projection model of lipid vesicles in general flows. *Journal of Computational Physics* 2011; **230**(22):8192–8215.
71. Laadhari A, Misbah C, Saramito P. On the equilibrium equation for a generalized biological membrane energy by using a shape optimization approach. *Physica D* 2010; **239**:1567–1572.

DOI: 10.1002/nme