Fluids-membrane interaction with a full Eulerian approach based on the level set method

Alexis Dalmon, Kalyani Kentheswaran, Guillaume Mialhe, Benjamin Lalanne, Sébastien Tanguy

Fluids-membrane interaction with a full Eulerian approach based on the level set method

Alexis Dalmon [a,b], Kalyani Kentheswaran [c,d], Guillaume Mialhe [a], Benjamin Lalanne [c], Sébastien Tanguy [a,*]

[a] Institut de Mécanique des Fluides de Toulouse, Université de Toulouse, CNRS, INPT, UPS, Toulouse, France
[b] Centre National d'Études Spatiales, 18 Avenue Edouard Belin, 31401 Toulouse Cedex 9, France
[c] Laboratoire de Génie Chimique, Université de Toulouse, CNRS, INPT, UPS, Toulouse, France
[d] TBI, Université de Toulouse, CNRS, INRA, INSA, Toulouse, France

ARTICLE INFO

Keywords:
Membrane
Level set
Ghost fluid

ABSTRACT

A fully Eulerian approach to predict fluids-membrane behaviours is presented in this paper. Based on the numerical model proposed by li et al. (2012), we present a sharp methodology to account for the jump conditions due to hyperelastic membranes. The membrane is considered infinitely thin and is represented by the level set method. Its deformations are obtained from the transport of the components of the left Cauchy-Green tensor throughout time. Considering the linear or a hyperelastic material law, the surface stress tensor is computed and gives the force exerted by the membrane on the surrounding fluids. The membrane force is taken into account in the Navier-Stokes equations as jump conditions on the pressure and on the velocity derivatives by imposing suitable singular source terms in cells crossed by the interface. To prevent stability issues, an extension algorithm has been developed to remove the normal derivatives of the scalar fields specific to the membrane. In particular, a subcell resolution at the interface of the extrapolated variable is proposed for increasing the accuracy of the extension algorithm. These improvements are validated by comparing our numerical results with benchmarks from the literature. Moreover, a new benchmark is proposed for fluids with both different viscosities and different densities to target applications where a gas and a liquid phase are separated by a membrane.

1. Introduction

The fluids-membrane interaction study of this paper is part of a global project on propellant sloshing in satellite tanks. The latter phenomenon happens during a satellite manoeuvre and can be a major disturbance of the stability. The tanks contain liquid propellant and gas to maintain a sufficient pressure within the tank. During a manoeuvre, inertial forces lead to a motion of the fluids and thus of the centre of mass. This generates disturbing forces and torques on the whole structure which may deteriorate the quality of satellite imaging. Considering simple tanks, numerical methods have been developed in our home-made code DIVA (Dynamics of Interface for Vaporisation and Atomisation) to model propellant sloshing in

* Corresponding author.
E-mail address: tanguy@imft.fr (S. Tanguy).

https://doi.org/10.1016/j.jcp.2019.109171

micro-gravity conditions [1,2]. A parametric study has been done on typical rotational manoeuvres exerted by satellites in space and has been validated by comparisons with data from the FLUIDICS (FLUId DynamICs in Space) experiment, performed in the International Space Station (ISS) [1,3,4]. The work presented in this paper is a first step towards the extension of the sloshing study to diaphragm tanks for which a hyperelastic membrane separates the liquid propellant and the gas. The modelling of the interaction between the membrane and the fluids within the tank is crucial. It must be predicted accurately and thus numerical developments are required to do so.

The fluid-membrane interaction is a challenging problem to solve numerically. Peskin [5] developed the Immersed Boundary Method (IBM) to predict incompressible flows with moving elastic boundaries. These first applications were the modelling of blood flows in the heart [6] and has been extended to many biological problems cited in [7]. With this method, the Navier-Stokes equations are solved on a Cartesian mesh and the membrane is described by Lagrangian markers. The force exerted by the membrane is deduced from the position of the markers and is interpolated onto the Cartesian mesh using Dirac delta functions. The Navier-Stokes equations are then solved with the forcing term corresponding to the elastic contribution from the membrane. Finally, the predicted fluid velocity is used to update the location of the marker points defining the membrane. This methodology is repeated for each time step. The smoothed Dirac functions induce a numerical smearing of the membrane in the fluid grid which may affect the accuracy of the method.

The Immersed Interface Method (IIM) of Leveque and Li [8] replaces the interpolated forcing term due to the elastic membrane with sharp jump conditions. This method has been introduced for elliptic equations and extended to Stokes flows [9] and incompressible viscous flows [10]. The artificial smearing of the elastic force in the fluid grid generates spurious velocities at the membrane and a smoothing of the pressure field where a discontinuous jump must appear. With the IIM, the normal component of the elastic force is enforced through a jump condition on the pressure when solving the Poisson equation. The tangential component of the force induces jumps in the derivatives of the velocity across the membrane.

The Material Point Method (MPM) describes both the solid and the fluid phases using Lagrangian markers in the whole computational domain [11]. This method has been used in the case of fluid-membrane interaction [12] and allows large deformations of the linear elastic membrane.

Full Eulerian approaches do not use Lagrangian particles to follow the membrane motion and to compute the elastic forces. Cottet and Maitre [13] use the level set method to follow the motion of an elastic membrane immersed in a fluid. Without the reinitialisation algorithm of the level set function [14], the variation of the level set gradient can be directly related to the membrane stretching. The authors use this property to enforce the elastic material law on the zero level set and to integrate the resulting force in the Cartesian mesh using delta functions. However, this method only gives the membrane stretching and limits the complexity of the membrane model. Moreover, the membrane is still artificially thickened by the delta functions.

Another full Eulerian method consists in defining the components of the deformation tensor of the membrane as scalar variables in the whole computational domain [15]. This approach is based on the work of Sugiyama [16] which developed a full Eulerian fluid-structure interaction model working with hyperelastic solid bodies. The position of each phase, fluid and solid, is described by the volume-of-fluid function (VOF) [17] and only one set of governing equations is solved in the whole domain. In the fluid region, the stress tensor contains the pressure term and the viscous stress tensor, in the solid region, the stress tensor is deduced from the solid deformation and the hyperelastic material law. The left Cauchy-Green deformation tensor is updated throughout time thanks to a transport equation in the whole computational domain. This allows to follow the deformation of the hyperelastic solid in a Eulerian manner. Then, the solid stress tensor can be computed following the hyperelastic material law and enforced in the Cartesian mesh. Ii et al. [15] extend this method to fluid-membrane interactions, which means that the solid region is reduced to a codimension-one subspace. Following the methodology of Barthes-Biesel and Rallison [18], the deformation tensor of the membrane is defined as a solid deformation tensor projected onto the tangent plane of the membrane. The normal projection of the membrane deformation tensor is not considered because the membrane material is supposed to be incompressible. The membrane stresses are computed thanks to the deformation tensor and the hyperelastic material law and the resulting force is integrated in the Navier-Stokes equations with delta functions. Nevertheless, this method presents losses of accuracy due to the numerical dissipation of the smoothed Dirac function. Moreover, instability issues may appear over long time periods because the values of the deformation tensor far from the membrane may evolve chaotically with the fluid velocity and may influence the computation at the membrane.

In this paper, we propose to improve the full Eulerian method of Ii et al. [15] by adding some aspects of the sharp methodology of the IIM. Moreover, we extend this model to handle different fluids on each side of the membrane. In section 2, the modelling of two-phase flows and the membrane model are described. The numerical methods implemented in the code are explained in section 3. More particularly, the discretisation of the jump conditions from the Ghost Fluid Method [19,20] is detailed. Furthermore, an extension algorithm based on a subcell resolution has been developed to reduce the numerical instability by removing the spurious normal derivatives. The validation of the methodology is done in section 4 with several benchmarks of growing complexity from the literature: from the simple case of a bubble rising due to a surface tension gradient until the much more complex cases of a stretched membrane separating different fluids and a capsule immersed in a shear flow. Finally, the paper is concluded by some remarks and perspectives in section 5.

2. Eulerian fluid membrane model

2.1. Two-phase flow model

We consider a domain Ω with a boundary $\partial\Omega$ which contains two different fluids defined by Ω^+ and Ω^- such as $\Omega = \Omega^+ \cup \Omega^-$. The interface between the two fluids regions is denoted Γ and its outward normal vector is $\mathbf{n}$. Each fluid is incompressible and Newtonian and follows the Navier-Stokes equations.

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u}\right) = -\nabla p + \nabla\cdot\left(2\mu\bar{\bar{\boldsymbol{D}}}\right), \tag{2}$$

with $\mathbf{u} = (u, v, w)$ the velocity field, t the time, ρ the fluid density, μ the fluid viscosity, p the pressure and $\bar{\bar{\boldsymbol{D}}}$ the rate of deformation tensor defined as

$$\bar{\bar{\boldsymbol{D}}} = \frac{\nabla\mathbf{u} + \nabla\mathbf{u}^T}{2}. \tag{3}$$

Considering the entire domain Ω, special care must be taken at the interface Γ between the two fluids regions. The following jump conditions must be accounted for,

$$[\rho]_\Gamma = \rho^+ - \rho^-, \tag{4}$$

$$[\mu]_\Gamma = \mu^+ - \mu^-, \tag{5}$$

$$[\mathbf{n}\cdot\bar{\bar{\boldsymbol{\sigma}}}]_\Gamma = \mathbf{f}_\Gamma, \tag{6}$$

with $\bar{\bar{\boldsymbol{\sigma}}}$ the stress tensor and $\mathbf{f}_\Gamma$ the local force density at the interface. In the case of two-phase flows with surface tension, the local force density becomes $\mathbf{f}_\Gamma = \gamma\kappa\mathbf{n}$ with γ the surface tension and $\kappa = \nabla\cdot\mathbf{n}$ the mean curvature of the interface. When a membrane separates the two fluids, the local force is defined in section 2.2.

Considering the entire domain Ω, we define the density and viscosity field as

$$\rho(\mathbf{x}) = \rho^- + (\rho^+ - \rho^-)H(\mathbf{x}), \tag{7}$$

$$\mu(\mathbf{x}) = \mu^- + (\mu^+ - \mu^-)H(\mathbf{x}), \tag{8}$$

with H the Heaviside function defined as $H(\mathbf{x}) = 0$ in Ω^- and $H(\mathbf{x}) = 1$ in Ω^+.

The Navier-Stokes equations in the entire domain Ω can therefore be written as

$$\nabla \cdot \mathbf{u} = 0, \tag{9}$$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u}\right) = -\nabla p + \nabla\cdot\left(2\mu\bar{\bar{\boldsymbol{D}}}\right) + \delta_\Gamma\mathbf{f}_\Gamma, \tag{10}$$

with δ_Γ the Dirac function located at the interface.

2.2. Membrane model

In this section, the large deformable membrane model of Barthes-Biesel and Rallison [18] is described. The local force density of the membrane on the fluids is computed with a full Eulerian formulation. This formulation is inspired by the work of Sugiyama et al. [16] for 3D solid problems and adapted to the membranes by Ii et al. [15].

In a stress-free state, each particle of a solid is defined by its coordinate vector $\mathbf{X}$. We denote by $\mathbf{x}(\mathbf{X}, t)$ the position, at time t, of a particle located in $\mathbf{X}$ in the stress-free state. The link between the current state and the stress-free state is the deformation gradient tensor $\bar{\bar{\boldsymbol{F}}}$ defined as

$$\bar{\bar{\boldsymbol{F}}} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}. \tag{11}$$

The material time derivative of the deformation gradient tensor $\bar{\bar{\boldsymbol{F}}}$ follows

$$\frac{d\bar{\bar{\boldsymbol{F}}}}{dt} = (\nabla\mathbf{u}_\mathbf{s})\bar{\bar{\boldsymbol{F}}}, \tag{12}$$

with $\mathbf{u}_\mathbf{s}$ the velocity of the membrane particles. This velocity vector is obtained by extending $\mathbf{u}$ from its values at the membrane toward the normal direction. The extension method is defined in section 3.6.

Now, we consider the special case of a membrane which thickness is very small compared to its other dimensions. Therefore, we will neglect the thickness and represent the membrane as a surface in a 3D space. Each fibre $\mathbf{dX}$ of the membrane in the reference state belongs to the tangent plane of the membrane. Similarly, each deformed fibre $\mathbf{dx}$ belongs to the tangent plane of the membrane in the current state. This means that only the components perpendicular to the normal direction of the membrane must be considered by the deformation gradient tensor. Let $\mathbf{n_R}$ be the outward normal of the membrane in the reference state and $\mathbf{n}$ the outward normal in the current state, the previous condition can be expressed respectively as $\bar{\bar{\boldsymbol{F}}}_{\boldsymbol{s}}.\mathbf{n_R} = \mathbf{0}$ and $\mathbf{n}.\bar{\bar{\boldsymbol{F}}}_{\boldsymbol{s}} = \mathbf{0}$ with $\bar{\bar{\boldsymbol{F}}}_{\boldsymbol{s}}$ the surface deformation gradient tensor of the membrane. This tensor can then be written

$$\bar{\bar{\boldsymbol{F}}}_{\boldsymbol{s}} = \bar{\bar{\boldsymbol{P}}}\bar{\bar{\boldsymbol{F}}}\bar{\bar{\boldsymbol{P}}}_{\boldsymbol{R}}, \tag{13}$$

with $\bar{\bar{\boldsymbol{P}}} = \bar{\bar{\boldsymbol{I}}} - \mathbf{n}\otimes\mathbf{n}$ the surface projection tensor in the current state and $\bar{\bar{\boldsymbol{P}}}_{\boldsymbol{R}} = \bar{\bar{\boldsymbol{I}}} - \mathbf{n_R}\otimes\mathbf{n_R}$ in the reference state. The local deformation of the membrane can be obtained with the surface left Cauchy-Green tensor,

$$\bar{\bar{\boldsymbol{B}}}_{\boldsymbol{s}} = \bar{\bar{\boldsymbol{F}}}_{\boldsymbol{s}}\bar{\bar{\boldsymbol{F}}}_{\boldsymbol{s}}^{T}. \tag{14}$$

The Cauchy-Green tensor $\bar{\bar{\boldsymbol{B}}}_{\boldsymbol{s}}$ is symmetric and its eigenvalues λ_1^2 and λ_2^2 correspond to the square of the two principal strains of the membrane in its tangent plane and the third eigenvalue zero corresponds to the deformation in the normal direction which has no existence because of the projections performed previously. The scalar invariants of the surface left Cauchy-Green tensor are

$$I_1 = \mathrm{tr}(\bar{\bar{\boldsymbol{B}}}_{\boldsymbol{s}}) = \lambda_1^2 + \lambda_2^2, \tag{15}$$

$$I_2 = \frac{1}{2}\left(\mathrm{tr}(\bar{\bar{\boldsymbol{B}}}_{\boldsymbol{s}})^2 - \mathrm{tr}(\bar{\bar{\boldsymbol{B}}}_{\boldsymbol{s}}^{\boldsymbol{2}})\right) = \lambda_1^2\lambda_2^2, \tag{16}$$

$$I_3 = \det(\bar{\bar{\boldsymbol{B}}}_{\boldsymbol{s}}) = 0. \tag{17}$$

This strain tensor, considering (13) and (14), can be written thanks to an intermediate tensor $\bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}}$,

$$\bar{\bar{\boldsymbol{B}}}_{\boldsymbol{s}} = \bar{\bar{\boldsymbol{P}}}\bar{\bar{\boldsymbol{G}}}_s\bar{\bar{\boldsymbol{P}}} \quad \text{with} \quad \bar{\bar{\boldsymbol{G}}}_s = \bar{\bar{\boldsymbol{F}}}\bar{\bar{\boldsymbol{P}}}_{\boldsymbol{R}}\bar{\bar{\boldsymbol{F}}}^{T}. \tag{18}$$

Considering (12) and (18), the material time derivative of $\bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}}$ is given by

$$\frac{d\bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}}}{dt} = (\nabla\mathbf{u_s})\bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}} + \bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}}(\nabla\mathbf{u_s})^{T}. \tag{19}$$

In the case of hyperelastic materials, the stress tensor $\bar{\bar{\boldsymbol{\sigma}}}$ can be written according to the surface strain energy function $W(I_1, I_2, I_3)$ which depends on the left Cauchy-Green tensor and its scalar invariants. In the case of a membrane, Barthes-Biesel and Rallison [18] show that the surface stress tensor $\bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}}$ is,

$$\bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}} = \frac{2}{\sqrt{I_2}}\left(\frac{\partial W}{\partial I_1}\bar{\bar{\boldsymbol{B}}}_{\boldsymbol{s}} + I_2\frac{\partial W}{\partial I_2}\bar{\bar{\boldsymbol{P}}}\right). \tag{20}$$

Several strain energy functions [21–24] exist to describe different hyperelastic behaviours. All functions depending on the scalar invariants of the Cauchy-Green tensor can be considered with this methodology. We will consider in this study the strain energy function of the neo-Hookean solid [25], adapted to the membrane case by considering the incompressibility of the membrane as

$$W = \frac{E_s}{6}(I_1 + \frac{1}{I_2} - 3), \tag{21}$$

with E_s the surface elastic modulus defined as $E_s = Eh_R$ with E the Young's modulus of the membrane and h_R the thickness of the membrane in the reference state. The surface stress tensor becomes for this material model,

$$\bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}} = \frac{E_s}{3\sqrt{I_2}}\left(\bar{\bar{\boldsymbol{B}}}_{\boldsymbol{s}} - \frac{\bar{\bar{\boldsymbol{P}}}}{I_2}\right). \tag{22}$$

Finally, the local force density exerted by the membrane is given by the surface divergence of the membrane stress tensor,

$$\mathbf{f_\Gamma} = \nabla_s\cdot\bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}}, \tag{23}$$

with ∇_s the surface gradient operator defined by $\nabla_s = \bar{\bar{\boldsymbol{P}}}.\nabla$. Considering the distance property of the level set function, which implies $(\mathbf{n}\cdot\nabla)\mathbf{n} = \mathbf{0}$, and the geometric properties of the membrane, some simplifications can be made,

$$\bar{\bar{\boldsymbol{P}}}\bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}} = \bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}}, \quad \mathbf{n} \cdot \bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}} = \mathbf{0}, \tag{24}$$

and the local force becomes

$$\mathbf{f}_{\boldsymbol{\Gamma}} = \nabla \cdot \bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}}. \tag{25}$$

The local force density can be divided into two components, respectively toward the normal and the tangent directions of the membrane such as

$$f_n = \mathbf{f}_{\boldsymbol{\Gamma}} \cdot \mathbf{n} = -\bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}} : \nabla \mathbf{n} = -\left(\bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}} \nabla\right) \cdot \mathbf{n}, \tag{26}$$

because the stress tensor is symmetric, and

$$\mathbf{f}_{\tau} = \bar{\bar{\boldsymbol{P}}}\mathbf{f}_{\boldsymbol{\Gamma}} = \bar{\bar{\boldsymbol{P}}}(\nabla \cdot \bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}}). \tag{27}$$

3. Numerical methods

This section describes specific numerical methods implemented to model the membrane behaviour and its interaction with the surrounding fluids. These developments have been integrated in the home-made code DIVA. This solver is based on several numerical methods dedicated to the computation of two-phase flows. The DIVA code can also consider liquid-vapour phase change [26–30] and compressible flows [31]. Complex geometry can be accounted for by using the irregular domain method proposed in [32] for single phase flows and extended to two-phase flows in irregular domains in [2]. In this study, the interface between the two fluids corresponds to the membrane and is represented by the level set method [33,14]. The Ghost-Fluid method [19,34–36,20,37] is used to consider the sharp jump conditions at the membrane. It should be pointed out here that Ghost Fluid Method is first order accurate for imposing jump conditions, but some recent works have proposed extensions to second order accuracy as in [38–40].

3.1. Interface tracking method

The level set method [33,14] is used to track the infinitely thin membrane throughout time. We define the scalar field ϕ which corresponds to the signed distance from the membrane Γ. Each fluid region corresponds to the sign of the level set function as $\Omega^+ = \{\mathbf{x} : \phi(\mathbf{x}) > 0\}$, $\Omega^- = \{\mathbf{x} : \phi(\mathbf{x}) < 0\}$ and the membrane corresponds to the zero level set $\Gamma = \{\mathbf{x} : \phi(\mathbf{x}) = 0\}$. The motion of the membrane is updated by solving the following transport equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \tag{28}$$

To keep the signed distance property of the level set function throughout time, we consider the reinitialisation algorithm [14]. The following equation is solved iteratively to correct the distance between each level set

$$\begin{cases} \dfrac{\partial d}{\partial \tau} + \text{sign}(\phi)(1 - ||\nabla d||) = 0, \\ d(\tau = 0) = \phi, \end{cases} \tag{29}$$

with τ a fictitious time along which the reinitialised distance function d is corrected to maintain the distance property $||\nabla d|| = 1$. The smoothed sign function $\text{sign}(\phi)$ is defined in [19,14].

The level set method allows us to compute the geometric properties of the membrane such as its outward normal using simple differencing,

$$\mathbf{n} = \frac{\nabla \phi}{||\nabla \phi||}, \tag{30}$$

and thus, the projection tensor $\bar{\bar{\boldsymbol{P}}} = \bar{\bar{\boldsymbol{I}}} - \mathbf{n} \otimes \mathbf{n}$ can be computed in the whole domain.

The spatial derivatives of the transport equation and the reinitialisation algorithm are computed using the WENO-Z scheme [41]. The temporal scheme of the transport equation is the second order TVD Runge-Kutta scheme. The reinitialisation algorithm is solved at the end of each time step.

3.2. Two-phase flow solvers

In this section, we describe the methodology to solve the Navier-Stokes equations with two different ways to consider the membrane contribution. The first one is the “delta” formulation for which the elastic force is a source term in the right-hand side. The second method integrates the elastic force through sharp jump conditions on the pressure and on the velocity derivatives.

3.2.1. The "delta" formulation

The "delta" formulation [42,14] solves the Navier-Stokes equations for two-phase flows by smoothing the interface on an artificial thickness. The method is based on a projection method inspired by Chorin [43] in the case of single phase flows. For two-phase flows, the method we use is derived from the Ghost-Fluid viscous Conservative Method with an Implicit scheme of Lepilliez et al. [2], inspired by the work of Sussman et al. [37]. First, given a velocity field $\mathbf{u}^n$ at the time $t^n = n\Delta t$, an intermediate velocity $\mathbf{u}^*$ is computed without considering the pressure term,

$$\rho^{n+1}\mathbf{u}^* - \Delta t\nabla\cdot\left(2\mu^{n+1}\bar{\bar{\boldsymbol{D}}}^*\right) = \rho^{n+1}\left(\mathbf{u}^n - \Delta t\left(\mathbf{u}^n\cdot\nabla\right)\mathbf{u}^n\right) + \Delta t\delta_\varepsilon{\mathbf{f}_\Gamma}^n. \tag{31}$$

The viscous term is considered implicitly to avoid its time step restriction. This leads to the resolution of a large linear system where the three components of the velocity are coupled. The advection term is computed with WENO-Z schemes [41] and the viscous term with central differencing schemes [2]. The local force ${\mathbf{f}_\Gamma}^n$ is multiplied by the smoothed Dirac distribution δ_ε defined below and added to the right-hand side. As the resulting matrix is diagonally dominant, the system is solved using a few steps of the Gauss-Seidel algorithm.

Then, the pressure field p^{n+1} serves as the scalar potential function of the Hodge decomposition which satisfies the following Poisson equation

$$\nabla\cdot\left(\frac{\nabla p^{n+1}}{\rho^{n+1}}\right) = \frac{\nabla\cdot\mathbf{u}^*}{\Delta t}, \tag{32}$$

with homogeneous Neumann boundary conditions on $\partial\Omega$. The resolution of this equation is done with the Black Box Multigrid method [44,45] to reduce the computation time.

Finally, the velocity field $\mathbf{u}^{n+1}$ is defined as the projection of the intermediate velocity $\mathbf{u}^*$ onto the divergence-free space

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho^{n+1}}\nabla p^{n+1}. \tag{33}$$

In the framework of the Whole Domain Formulation, the density and viscosity fields are updated with the level set function using a smoothed Heaviside distribution

$$\rho(\mathbf{x}) = \rho^- + (\rho^+ - \rho^-)H_\varepsilon(\mathbf{x}), \tag{34}$$

$$\mu(\mathbf{x}) = \mu^- + (\mu^+ - \mu^-)H_\varepsilon(\mathbf{x}), \tag{35}$$

with $H_\varepsilon(\mathbf{x})$ the smoothed Heaviside distribution defined as

$$H_\varepsilon(\mathbf{x}) = \begin{cases} 0 \text{ if } \phi(\mathbf{x}) < -\varepsilon, \\ \dfrac{1}{2}\left(1 + \dfrac{\phi(\mathbf{x})}{\varepsilon} + \dfrac{\sin(\pi\phi(\mathbf{x})/\varepsilon)}{\pi}\right) \text{ if } |\phi(\mathbf{x})| < \varepsilon, \\ 1 \text{ if } \phi(\mathbf{x}) > \varepsilon, \end{cases} \tag{36}$$

with ε the fictitious thickness of the membrane, which is equal to two or three times the size of a cell [46]. Similarly, the smoothed Dirac distribution $\delta_\varepsilon(\mathbf{x})$ is defined as the derivative of the smoothed Heaviside function,

$$\delta_\varepsilon(\mathbf{x}) = \begin{cases} 0 \text{ if } |\phi(\mathbf{x})| > \varepsilon, \\ \dfrac{1}{2\varepsilon}\left(1 + \cos\left(\dfrac{\pi\phi(\mathbf{x})}{\varepsilon}\right)\right) \text{ if } |\phi(\mathbf{x})| < \varepsilon. \end{cases} \tag{37}$$

3.2.2. The sharp formulation

Let us consider that $\mathbf{n}$ is the normal vector to the membrane and $\left(\mathbf{e}_{\tau_1}, \mathbf{e}_{\tau_2}\right)$ are two orthogonal vectors belonging to the tangent plane of the membrane. The no-slip condition on both sides of the membrane enforces $[\mathbf{u}]_\Gamma = 0$ and gives

$$\left[\frac{\partial\mathbf{u}}{\partial\mathbf{e}_{\tau_1}}\right]_\Gamma = 0 \text{ and } \left[\frac{\partial\mathbf{u}}{\partial\mathbf{e}_{\tau_2}}\right]_\Gamma = 0. \tag{38}$$

The incompressibility condition $\nabla\cdot\mathbf{u} = 0$ in both fluids ensures that there is no jump at the membrane $[\nabla\cdot\mathbf{u}]_\Gamma = 0$ and gives the following relation from [47]

$$\left[\frac{\partial\mathbf{u}}{\partial\mathbf{n}}\right]_\Gamma\cdot\mathbf{n} = 0. \tag{39}$$

The density and viscosity fields are piecewise constant and their jumps at the membrane correspond to equations (4) and (5). With the sharp formulation, the local density force corresponding to the membrane contribution is expressed as three primary jump conditions:

$$[p]_\Gamma = 2\left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]_\Gamma \cdot \mathbf{n} + \mathbf{f}_\Gamma \cdot \mathbf{n} = 2[\mu]_\Gamma \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \cdot \mathbf{n} + f_n, \tag{40}$$

$$\left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]_\Gamma \cdot \mathbf{e}_{\boldsymbol{\tau_1}} + \left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{e}_{\boldsymbol{\tau_1}}}\right]_\Gamma \cdot \mathbf{n} = -\left(\bar{\bar{\boldsymbol{P}}}\mathbf{f}_\Gamma\right) \cdot \mathbf{e}_{\boldsymbol{\tau_1}} = -\mathbf{f}_\tau \cdot \mathbf{e}_{\boldsymbol{\tau_1}}, \tag{41}$$

$$\left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]_\Gamma \cdot \mathbf{e}_{\boldsymbol{\tau_2}} + \left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{e}_{\boldsymbol{\tau_2}}}\right]_\Gamma \cdot \mathbf{n} = -\left(\bar{\bar{\boldsymbol{P}}}\mathbf{f}_\Gamma\right) \cdot \mathbf{e}_{\boldsymbol{\tau_2}} = -\mathbf{f}_\tau \cdot \mathbf{e}_{\boldsymbol{\tau_2}}. \tag{42}$$

These jump conditions have been proven by Xu and Wang [48] in single phase flows and extended to piecewise constant viscosity by Tan et al. [49,47]. In this study, we only consider these primary jump conditions which account for all the physical phenomena of interest for this study. It is noticeable that higher accuracy numerical methods have been achieved in [49,47,48] in the simpler case of linear elasticity, by imposing further secondary jump conditions on the second order velocity derivatives and first order pressure derivatives. However, these numerical methods have not been generalised to more complex configurations such as hyperelastic membranes.

Following the Ghost-Fluid Conservative viscous Method (GFCM), Lalanne et al. [46] show that only the normal component of the force must be considered in the projection and the correction steps because the viscous component is already taken into account in the predictor step of the projection method. The remaining jump condition on the pressure is then

$$[p]_\Gamma = \mathbf{f}_\Gamma \cdot \mathbf{n} = f_n. \tag{43}$$

From equations (38), the jump conditions (41) and (42) can be written as

$$\left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]_\Gamma \cdot \mathbf{e}_{\boldsymbol{\tau_1}} + [\mu]_\Gamma \frac{\partial \mathbf{u}}{\partial \mathbf{e}_{\boldsymbol{\tau_1}}} \cdot \mathbf{n} = -\mathbf{f}_\tau \cdot \mathbf{e}_{\boldsymbol{\tau_1}}, \tag{44}$$

$$\left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]_\Gamma \cdot \mathbf{e}_{\boldsymbol{\tau_2}} + [\mu]_\Gamma \frac{\partial \mathbf{u}}{\partial \mathbf{e}_{\boldsymbol{\tau_2}}} \cdot \mathbf{n} = -\mathbf{f}_\tau \cdot \mathbf{e}_{\boldsymbol{\tau_2}}. \tag{45}$$

Following a similar approach to that of Lalanne et al. [46], the source terms due to the tangential components of the elastic force are considered as jumps that are only enforced on the first term: $\left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]_\Gamma \cdot \mathbf{e}_{\boldsymbol{\tau_k}}$ with $k = [1, 2]$. The second terms of equations (44) and (45) will be zero if there is no viscosity jump, even in the case where the tangential component of the local force is not zero. This approach still stands with a viscosity jump and has been used in [50] where several simulations with surfactants were performed. Consequently, the splitting of the jump conditions allows a simple numerical implementation of the source terms in the cells cut by the membrane. Indeed, the jump conditions on the tangent components of the viscosity-scaled velocity gradient in the first step of the GFCM can be taken into account by imposing

$$\left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]_\Gamma \cdot \mathbf{e}_{\boldsymbol{\tau_k}} = -\mathbf{f}_\tau \cdot \mathbf{e}_{\boldsymbol{\tau_k}}, \tag{46}$$

with $k = [1, 2]$. Because the jump of the normal component of the viscosity-scaled velocity gradient is only due to the jump on the viscosity, the following source term is considered in the first step of the algorithm

$$\left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]_\Gamma = -\bar{\bar{\boldsymbol{P}}}\mathbf{f}_\Gamma = -\mathbf{f}_\tau. \tag{47}$$

In section 4.1, a validation of this methodology to deal with jumps of both tangential and normal stresses is presented by comparing simulations and the theoretical solution of a droplet rising in a surface tension gradient in the Stokes regime. The second terms of equations (44) and (45) are zero in this case, with or without viscosity jump. As the simulations match to the theoretical solution, this comparison demonstrates the relevance of the presented approach. For a general case with nonzero terms $[\mu]_\Gamma \frac{\partial \mathbf{u}}{\partial \mathbf{e}_{\boldsymbol{\tau_1}}} \cdot \mathbf{n}$ and $[\mu]_\Gamma \frac{\partial \mathbf{u}}{\partial \mathbf{e}_{\boldsymbol{\tau_2}}} \cdot \mathbf{n}$ and viscosity jump, we assume that all the terms of equations (44) and (45) are well taken into account. To our knowledge, a demonstration for this general case has never been provided and further investigation would be appreciated. It appears from this discussion that the key-point is to introduce a consistent numerical approximation of the source term resulting from the jump condition. Finally, the jump condition will be automatically satisfied if a consistent approximation of the source term is added at the right place. Other general considerations on this specific point can be found in Appendix B. From this, some analogies can be found between jump condition formulations and δ formulations for two-phase flows Navier-Stokes equations.

To solve the Navier-Stokes equations for two-phase flows with the Ghost Fluid viscous Conservative Method, the same projection method inspired by [43] is considered with jump conditions to describe the membrane force. First, the intermediate velocity is computed using the same implicit scheme for the viscous term,

$$\begin{cases} \rho^{n+1}\mathbf{u}^* - \Delta t \nabla \cdot \left(2\mu^{n+1}\bar{\bar{\boldsymbol{D}}}^*\right) = \rho^{n+1}\left(\mathbf{u}^n - \Delta t\left(\mathbf{u}^n \cdot \nabla\right)\mathbf{u}^n\right), \\ \left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]_\Gamma = -\mathbf{f}_\tau, \end{cases} \tag{48}$$

with the source term due to the tangent component of the singular force imposed as a jump on the viscous stress. It is noteworthy that, unlike the delta formulation, the prediction step of the projection method only contains this tangent component, since the normal component is accounted for in the projection and correction steps. The analogy between sharp methods using jump conditions and the "delta" formulation is thoroughly explained in [46]. The linear system resulting from (48) is a single linear system where all the velocity components are coupled. This allows a fully implicit discretization of the viscous term without additional time step constraints due to viscosity jump and it can be solved easily with a Gauss-Seidel algorithm as pointed out by Lepilliez et al. in [2]. The numerical implementation of the jump conditions on the viscous stress tensor is detailed in section 3.5.

Then the Poisson equation is solved with

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}, \tag{49}$$

with the following pressure jump condition accounting for the normal component of the local force

$$[p]_\Gamma = f_n \, , \tag{50}$$

and the following jump condition on the pressure normal gradient

$$\left[\frac{\mathbf{n} \cdot \nabla p^{n+1}}{\rho^{n+1}} \right] = \left[\frac{\mu \mathbf{n} \cdot \Delta \mathbf{u}^n}{\rho^{n+1}} \right] \, , \tag{51}$$

details of which can be found in Appendix B. The numerical discretization of this singular source term is identical to the one described by Liu et al. in [20] and is detailed in section 3.4.

These equations based on a jump condition formulation can also be rewritten in a similar way with singular source terms instead of jump condition. It will give a mathematically identical system that will express as

$$\rho^{n+1}\mathbf{u}^* - \Delta t \nabla \cdot \left(2\mu^{n+1} \bar{\bar{\boldsymbol{D}}}^* \right) = \rho^{n+1} \left(\mathbf{u}^n - \Delta t \left(\mathbf{u}^n \cdot \nabla \right) \mathbf{u}^n \right) + \Delta t \mathbf{f}_\tau \delta_\Gamma, \tag{52}$$

to compute the velocity field in the predictor step. Next, the pressure field can be computed with the appropriate jump condition with the following equation

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} + \nabla \cdot \left(\frac{f_n \mathbf{n} \delta_\Gamma}{\rho^{n+1}} \right), \tag{53}$$

which is identical to equations (49) and (50). Indeed, the term $2[\mu]_\Gamma \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \cdot \mathbf{n}$ is included in $\nabla \cdot \mathbf{u}^*$, for more details, see Appendix B. Finally, the velocity is corrected,

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho^{n+1}} (\nabla p^{n+1} - f_n \mathbf{n} \delta_\Gamma). \tag{54}$$

By considering that modifications of the numerical schemes accounting for jump conditions act as sharp approximations of singular source terms, it can be understood that the two presented formulations are identical. See for instance [46] for more details on this specific point.

3.3. Discretisation of the membrane model

Considering the Cartesian mesh, the standard MAC grid is used: the scalar variables are located in the centre of the mesh cells and the components of the velocity field are staggered at the cell faces in each direction. Unlike in [15], all the components of the different solid tensors are located at the centre of the mesh cells in this study. The location of the different variables in a 2D mesh cell is illustrated in Fig. 1.

At each time step, the position of the interface is updated using (28). The six components of the intermediate strain tensor $\bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}}$ in 3D are then computed using equation (19)

$$\bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}}^{n+1} = \bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}}^{n} - \Delta t \left((\mathbf{u_s}^n \cdot \nabla) \bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}}^{n} - (\nabla \mathbf{u_s}^n) \bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}}^{n} - \bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}}^{n} (\nabla \mathbf{u_s}^n)^T \right). \tag{55}$$

The advection term is computed with the WENO-Z scheme [41] and the velocity derivatives are obtained with central differencing schemes. For example, in 2D, the advection term corresponding to the first component of the intermediate strain tensor is computed as

$$(\mathbf{u_s}^n \cdot \nabla) G_{11}^n = u_s\big|_{i,j} \frac{\partial G_{11}}{\partial x}\Big|_{i,j} + v_s\big|_{i,j} \frac{\partial G_{11}}{\partial y}\Big|_{i,j} \tag{56}$$

with

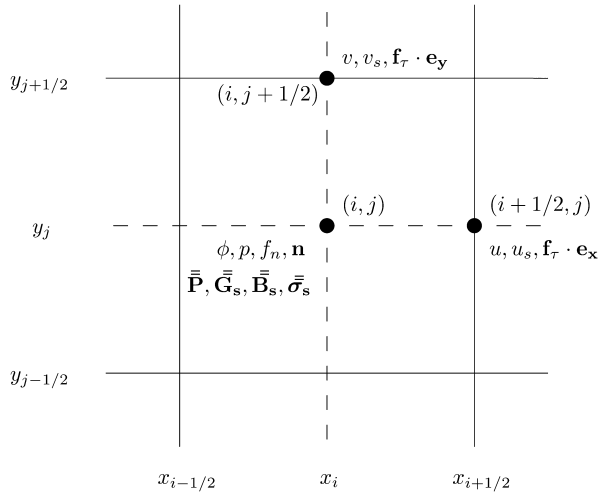

Fig. 1. Location of the variables on a 2D mesh cell.

$$u_s|_{i,j} = \frac{1}{2}\left(u_s|_{i-1/2,j} + u_s|_{i+1/2,j}\right) \text{ and } v_s|_{i,j} = \frac{1}{2}\left(v_s|_{i,j-1/2} + v_s|_{i,j+1/2}\right). \tag{57}$$

Considering now the derivatives of the x-component of the velocity,

$$\frac{\partial u_s}{\partial x}\Big|_{i,j} = \frac{u_s|_{i+1/2,j} - u_s|_{i-1/2,j}}{\Delta x} \text{ and } \frac{\partial u_s}{\partial y}\Big|_{i,j} = \frac{u_s|_{i,j+1/2} - u_s|_{i,j-1/2}}{\Delta y}, \tag{58}$$

with

$$u_s|_{i,j\pm1/2} = \frac{1}{4}\left(u_s|_{i-1/2,j} + u_s|_{i+1/2,j} + u_s|_{i-1/2,j\pm1} + u_s|_{i+1/2,j\pm1}\right). \tag{59}$$

The computation of the components of the surface left Cauchy-Green tensor and the stress tensor at the cell centres follows equations (14) and (20).

The components of the surface density force $\mathbf{f_\Gamma}$ are computed at the centre of the faces of the cells following (25). Considering the sharp methodology and the force decomposition (26, 27), the normal component of the force is computed at the centre of the cell and the tangent components are staggered at the cell faces. Considering a 2D example, the discretisation of the normal component of the force, following (26), is

$$\begin{aligned} f_n|_{i,j} &= \sigma_{11}|_{i,j}\frac{\partial n_x}{\partial x}\Big|_{i,j} + \sigma_{12}|_{i,j}\frac{\partial n_x}{\partial y}\Big|_{i,j} + \sigma_{21}|_{i,j}\frac{\partial n_y}{\partial x}\Big|_{i,j} + \sigma_{22}|_{i,j}\frac{\partial n_y}{\partial y}\Big|_{i,j} \\ &= \sigma_{11}|_{i,j}\frac{n_x|_{i+1,j} - n_x|_{i-1,j}}{2\Delta x} + \sigma_{12}|_{i,j}\frac{n_x|_{i,j+1} - n_x|_{i,j-1}}{2\Delta y} + \sigma_{21}|_{i,j}\frac{n_y|_{i+1,j} - n_y|_{i-1,j}}{2\Delta x} \\ &+ \sigma_{22}|_{i,j}\frac{n_y|_{i,j+1} - n_y|_{i,j-1}}{2\Delta y}. \end{aligned} \tag{60}$$

The tangent component of the force in the x-direction from (27) gives

$$\begin{aligned} \mathbf{f}_\tau \cdot \mathbf{e_x}|_{i+1/2,j} &= P_{11}|_{i+1/2,j}\left(\frac{\partial \sigma_{11}}{\partial x}\Big|_{i+1/2,j} + \frac{\partial \sigma_{12}}{\partial y}\Big|_{i+1/2,j}\right) + P_{12}|_{i+1/2,j}\left(\frac{\partial \sigma_{12}}{\partial x}\Big|_{i+1/2,j} + \frac{\partial \sigma_{22}}{\partial y}\Big|_{i+1/2,j}\right) \\ &= P_{11}|_{i+1/2,j}\left(\frac{\sigma_{11}|_{i+1,j} - \sigma_{11}|_{i,j}}{\Delta x} + \frac{\sigma_{12}|_{i+1/2,j+1/2} - \sigma_{12}|_{i+1/2,j-1/2}}{\Delta y}\right) \\ &+ P_{12}|_{i+1/2,j}\left(\frac{\sigma_{12}|_{i+1,j} - \sigma_{12}|_{i,j}}{\Delta x} + \frac{\sigma_{22}|_{i+1/2,j+1/2} - \sigma_{22}|_{i+1/2,j-1/2}}{\Delta y}\right), \end{aligned} \tag{61}$$

with

$$P_{kl}|_{i+1/2,j} = \frac{1}{2}\left(P_{kl}|_{i+1,j} + P_{kl}|_{i,j}\right) \text{ and } \sigma_{kl}|_{i+1/2,j+1/2} = \frac{1}{4}\left(\sigma_{kl}|_{i,j} + \sigma_{kl}|_{i+1,j} + \sigma_{kl}|_{i,j+1} + \sigma_{kl}|_{i+1,j+1}\right). \tag{62}$$

With the "delta" formulation, the discretisation of the elastic force corresponds to equation (61) with $\bar{\bar{\boldsymbol{I}}}$ instead of $\bar{\bar{\boldsymbol{P}}}$ to consider all the components of the force. This method is identical in the other directions and can easily be extrapolated in 3D.

3.4. Jump condition on the pressure

The pressure jump is taken into account following the Ghost Fluid Method [19,20]. The Poisson equation on the pressure (53) can be written as

$$\nabla \cdot (\beta \nabla p) = RHS, \tag{63}$$

with $\beta = 1/\rho$ the diffusion coefficient. The equation contains jump conditions on the pressure denoted $a = f_n$ and on the diffusion coefficient $[\beta]_\Gamma$.

The discretisation of the 2D Poisson equation at point (i, j) gives

$$\frac{\beta_{i+1/2,j}\left(\frac{p_{i+1,j}-p_{i,j}}{\Delta x}\right) - \beta_{i-1/2,j}\left(\frac{p_{i,j}-p_{i-1,j}}{\Delta x}\right)}{\Delta x} + \frac{\beta_{i,j+1/2}\left(\frac{p_{i,j+1}-p_{i,j}}{\Delta y}\right) - \beta_{i,j-1/2}\left(\frac{p_{i,j}-p_{i,j-1}}{\Delta y}\right)}{\Delta y} = RHS_{i,j} + g_{i,j}, \tag{64}$$

with $(\beta_{i+1/2,j}, \beta_{i-1/2,j}, \beta_{i,j+1/2}, \beta_{i,j-1/2})$ the harmonic averages of the diffusion coefficient at the centre of the cell borders. If the interface crosses the right mesh segment $[x_{i,j}, x_{i+1,j}]$,

$$\beta_{i+1/2,j} = \begin{cases} \dfrac{\beta^+\beta^-}{\beta^-\theta^R + \beta^+(1-\theta^R)} \text{ if } \phi_{i,j} < 0 \text{ and } \phi_{i+1,j} > 0, \\ \dfrac{\beta^+\beta^-}{\beta^+\theta^R + \beta^-(1-\theta^R)} \text{ if } \phi_{i,j} > 0 \text{ and } \phi_{i+1,j} < 0, \end{cases} \tag{65}$$

with

$$\theta^R = \frac{|\phi_{i+1,j}|}{|\phi_{i+1,j}| + |\phi_{i,j}|}. \tag{66}$$

Similarly, if the interface crosses the left mesh segment $[x_{i-1,j}, x_{i,j}]$,

$$\beta_{i-1/2,j} = \begin{cases} \dfrac{\beta^+\beta^-}{\beta^-\theta^L + \beta^+(1-\theta^L)} \text{ if } \phi_{i,j} < 0 \text{ and } \phi_{i-1,j} > 0, \\ \dfrac{\beta^+\beta^-}{\beta^+\theta^L + \beta^-(1-\theta^L)} \text{ if } \phi_{i,j} > 0 \text{ and } \phi_{i-1,j} < 0, \end{cases} \tag{67}$$

with

$$\theta^L = \frac{|\phi_{i-1,j}|}{|\phi_{i-1,j}| + |\phi_{i,j}|}. \tag{68}$$

The $g_{i,j}$ term in (64) corresponds to the jumps enforced when the interface crosses at least one of the four neighbouring mesh segments. The latter are denoted by the superscripts R, L, T and B which correspond respectively to the right, left, top and bottom borders. As a general rule,

$$g_{i,j} = g^R_{i,j} + g^L_{i,j} + g^T_{i,j} + g^B_{i,j}. \tag{69}$$

Each one of these values exists only if the membrane crosses the mesh segments and then equals

$$g^R_{i,j} = \pm\frac{\beta_{i+1/2,j}a^R_\Gamma}{\Delta x^2},\ g^L_{i,j} = \pm\frac{\beta_{i-1/2,j}a^L_\Gamma}{\Delta x^2},\ g^T_{i,j} = \pm\frac{\beta_{i,j+1/2}a^T_\Gamma}{\Delta y^2},\ g^B_{i,j} = \pm\frac{\beta_{i,j-1/2}a^B_\Gamma}{\Delta y^2}, \tag{70}$$

with $\pm$ corresponding to the opposite sign of $\phi_{i,j}$ and

$$a^R_\Gamma = a_{i,j}\theta^R + a_{i+1,j}(1-\theta^R), \quad a^L_\Gamma = a_{i,j}\theta^L + a_{i-1,j}(1-\theta^L), \tag{71}$$

$$a^T_\Gamma = a_{i,j}\theta^T + a_{i,j+1}(1-\theta^T), \quad a^B_\Gamma = a_{i,j}\theta^B + a_{i,j-1}(1-\theta^B). \tag{72}$$

The extension of these schemes to 3D problems is straightforward.

3.5. Jump condition on the velocity derivatives

The discretisation of each component of the viscous-stress tensor is described in this section with a specific emphasis on how to enforce suitable jump conditions. Considering a 2D example, the divergence of the viscous-stress tensor gives

$$\nabla \cdot (2\mu\bar{\bar{\boldsymbol{D}}}) = \begin{pmatrix} \frac{\partial}{\partial x}\left(2\mu\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right) \\ \frac{\partial}{\partial x}\left(\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right) + \frac{\partial}{\partial y}\left(2\mu\frac{\partial v}{\partial y}\right) \end{pmatrix}. \tag{73}$$

With (47), the jump condition of each velocity derivative can be expressed as

$$\left[\mu\frac{\partial u}{\partial x}\right]_\Gamma = n_x\left[\mu\frac{\partial u}{\partial \mathbf{n}}\right]_\Gamma = -n_x\left(\mathbf{f}_\tau\cdot\mathbf{e_x}\right), \quad \left[\mu\frac{\partial u}{\partial y}\right]_\Gamma = n_y\left[\mu\frac{\partial u}{\partial \mathbf{n}}\right]_\Gamma = -n_y\left(\mathbf{f}_\tau\cdot\mathbf{e_x}\right), \tag{74}$$

$$\left[\mu\frac{\partial v}{\partial x}\right]_\Gamma = n_x\left[\mu\frac{\partial v}{\partial \mathbf{n}}\right]_\Gamma = -n_x\left(\mathbf{f}_\tau\cdot\mathbf{e_y}\right), \quad \left[\mu\frac{\partial v}{\partial y}\right]_\Gamma = n_y\left[\mu\frac{\partial v}{\partial \mathbf{n}}\right]_\Gamma = -n_y\left(\mathbf{f}_\tau\cdot\mathbf{e_y}\right). \tag{75}$$

Considering the first term of the viscous-stress tensor in (73), let $\beta = 2\mu$ be the diffusion coefficient and $b = -n_x\left(\mathbf{f}_\tau\cdot\mathbf{e_x}\right)$ be the jump condition on the velocity derivative computed at the right and left cell borders. The discretisation of this term at point $(i+1/2,\, j)$ gives

$$\frac{\partial}{\partial x}\left(\beta\frac{\partial u}{\partial x}\right)\Big|_{i+1/2,\,j} = \frac{\beta_{i+1,\,j}\left(\frac{u_{i+3/2,\,j}-u_{i+1/2,\,j}}{\Delta x}\right) - \beta_{i,\,j}\left(\frac{u_{i+1/2,\,j}-u_{i-1/2,\,j}}{\Delta x}\right)}{\Delta x} - h_{i+1/2,\,j} \tag{76}$$

with $\beta_{i,\,j}$ and $\beta_{i+1,\,j}$ the harmonic averages of the diffusion coefficient at the centre of the cells which are computed with the same methodology as in equations (65) to (68).

The jump conditions to be enforced are $h_{i+1/2,\,j} = h^R_{i+1/2,\,j} + h^L_{i+1/2,\,j}$ if the membrane crosses respectively the interval $[x_{i+1/2,\,j}, x_{i+3/2,\,j}]$ or $[x_{i-1/2,\,j}, x_{i+1/2,\,j}]$ defined as

$$h^R_{i+1/2,\,j} = \pm\frac{\beta_{i+1,\,j}\,b^R_\Gamma\theta^R}{\beta^\pm\Delta x} \;\text{ and }\; h^L_{i+1/2,\,j} = \mp\frac{\beta_{i,\,j}\,b^L_\Gamma\theta^L}{\beta^\pm\Delta x} \tag{77}$$

with $\pm$ the opposite sign of $\phi_{i+1/2,\,j}$, $\mp$ the sign of $\phi_{i+1/2,\,j}$,

$$b^R_\Gamma = b_{i+1/2,\,j}\theta^R + b_{i+3/2,\,j}(1-\theta^R) \;\text{ and }\; b^L_\Gamma = b_{i+1/2,\,j}\theta^L + b_{i-1/2,\,j}(1-\theta^L), \tag{78}$$

with

$$\theta^R = \frac{|\phi_{i+3/2,\,j}|}{|\phi_{i+3/2,\,j}| + |\phi_{i+1/2,\,j}|} \quad\text{and}\quad \theta^L = \frac{|\phi_{i-1/2,\,j}|}{|\phi_{i-1/2,\,j}| + |\phi_{i+1/2,\,j}|}. \tag{79}$$

The same method is applied for all the second derivatives of the same variable. Special care must be taken in the case of mixed derivatives. Considering the last term of the divergence of the viscous stress tensor in the x-direction, the discretisation at point $(i+1/2,\, j)$ gives

$$\frac{\partial}{\partial y}\left(\beta\frac{\partial v}{\partial x}\right)\Big|_{i+1/2,\,j} = \frac{\beta_{i+1/2,\,j+1/2}\left(\frac{v_{i+1,\,j+1/2}-v_{i,\,j+1/2}}{\Delta x}\right) - \beta_{i+1/2,\,j-1/2}\left(\frac{v_{i+1,\,j-1/2}-v_{i,\,j-1/2}}{\Delta x}\right)}{\Delta y} - h_{i+1/2,\,j}. \tag{80}$$

The diffusion coefficient is now $\beta = \mu$ and the jump condition on the mixed derivative is $b = -n_x\left(\mathbf{f}_\tau\cdot\mathbf{e_y}\right)$. The harmonic averages of the diffusion coefficient are similarly computed following equations (65) to (68). The jump conditions $h_{i+1/2,\,j} = h^T_{i+1/2,\,j} + h^B_{i+1/2,\,j}$ exist only if the membrane crosses respectively the interval $[x_{i+1,\,j+1/2}, x_{i,\,j+1/2}]$ or $[x_{i+1,\,j-1/2}, x_{i,\,j-1/2}]$ and are computed similarly as in equation (77),

$$h^T_{i+1/2,\,j} = \pm\frac{\beta_{i+1/2,\,j+1/2}\,b^T_\Gamma\theta^T}{\beta^\pm\Delta y} \;\text{ and }\; h^B_{i+1/2,\,j} = \mp\frac{\beta_{i+1/2,\,j-1/2}\,b^B_\Gamma\theta^B}{\beta^\pm\Delta y}, \tag{81}$$

with $\pm$ corresponding to the opposite sign of $\phi_{i+1/2,\,j}$ and $\mp$ the same sign than $\phi_{i+1/2,\,j}$.

Considering that the membrane crosses the top mesh segment, the sign of the level set function at the calculation point $(i+1/2,\, j)$ gives the expression of b^T_Γ and θ^T. If $\phi_{i+1/2,\,j}\phi_{i,\,j+1/2} < 0$, the computation of the jump condition becomes

$$b^T_\Gamma = b_{i+1,\,j+1/2}\theta^T + b_{i,\,j+1/2}(1-\theta^T) \;\text{ with }\; \theta^T = \frac{|\phi_{i,\,j+1/2}|}{|\phi_{i,\,j+1/2}| + |\phi_{i+1,\,j+1/2}|}. \tag{82}$$

Conversely, if $\phi_{i+1/2,\,j}\phi_{i,\,j+1/2} > 0$, the jump condition becomes

$$b^T_\Gamma = b_{i+1,\,j+1/2}(1-\theta^T) + b_{i,\,j+1/2}\theta^T \;\text{ with }\; \theta^T = \frac{|\phi_{i+1,\,j+1/2}|}{|\phi_{i,\,j+1/2}| + |\phi_{i+1,\,j+1/2}|}. \tag{83}$$

Considering that the membrane crosses the bottom mesh segment, the same methodology is applied: if $\phi_{i+1/2,\,j}\phi_{i,\,j-1/2} < 0$, the computation of the jump condition becomes

$$b^B_\Gamma = b_{i+1,\,j-1/2}\theta^B + b_{i,\,j-1/2}(1-\theta^B) \;\text{ with }\; \theta^B = \frac{|\phi_{i,\,j-1/2}|}{|\phi_{i,\,j-1/2}| + |\phi_{i+1,\,j-1/2}|}. \tag{84}$$

Conversely, if $\phi_{i+1/2,j}\phi_{i,j-1/2} > 0$, the jump condition becomes

$$b_\Gamma^T = b_{i+1,j-1/2}(1-\theta^B) + b_{i,j-1/2}\theta^B \text{ with } \theta^B = \frac{|\phi_{i+1,j-1/2}|}{|\phi_{i,j-1/2}| + |\phi_{i+1,j-1/2}|}. \tag{85}$$

The jump conditions of each term of the viscous stress tensor are added to the right-hand side of the first step of the projection method without changing the coefficients of the coupled linear system. More details about the linear system to solve the velocity are presented in [2]. The extension of these schemes to 3D problems is straightforward.

3.6. Extension algorithm

The components of the solid tensors and the solid velocity $\mathbf{u_s}$ are scalar fields defined in the whole computational domain. Nevertheless, they are only physically meaningful at the interface (where $\phi = 0$). The values of these scalar fields in the fluids domain are not physical and must not influence the computation at the membrane position. Through equation (19), shear flows or vortices, even far from the interface, may affect the solid tensors by generating parasitic strains. The tensors components may grow exponentially and lead to numerical instability [15]. To avoid this problem, the authors replace the surface left Cauchy-Green tensor by the current projection tensor in the regions remote from the membrane. This method improves the computation near the membrane but stability problems still occur in the long run [15].

In this paper, an extension algorithm has been developed to avoid these stability issues by extending the components of the velocity field and the solid tensor from the membrane toward its normal direction. Therefore, only the physical values are extended in the whole domain and the gradient of the scalar fields in the normal direction of the membrane equal zero. The methodology is inspired by the reinitialisation algorithm of level set functions [51–53] where a subcell resolution has been proposed. The extension of a scalar function h is described by the following equation

$$\begin{cases} \dfrac{\partial h}{\partial \tau} + (\text{sign}(\phi)\mathbf{n}) \cdot \nabla h = 0, \\ h(\mathbf{x}, \tau) = h(\mathbf{x}, 0) \text{ for } \tau > 0 \text{ and } \mathbf{x} \in \Gamma, \end{cases} \tag{86}$$

with τ a fictitious time. After the convergence, the normal derivative of the function tends toward zero in the whole domain and the scalar field h tends toward its value at the closer point of the membrane.

The spatial derivatives are discretised using second order ENO finite differences in each direction. In the x-direction, two one-sided ENO finite differences exist,

$$\frac{\partial h}{\partial x}\Big|_r = \frac{h_{i+1} - h_i}{\Delta x} - \frac{\Delta x}{2}\text{minmod}\left(\frac{h_{i+1} - 2h_i + h_{i-1}}{\Delta x^2}, \frac{h_{i+2} - 2h_{i+1} + h_i}{\Delta x^2}\right), \tag{87}$$

$$\frac{\partial h}{\partial x}\Big|_l = \frac{h_i - h_{i-1}}{\Delta x} + \frac{\Delta x}{2}\text{minmod}\left(\frac{h_{i+1} - 2h_i + h_{i-1}}{\Delta x^2}, \frac{h_i - 2h_{i-1} + h_{i-2}}{\Delta x^2}\right), \tag{88}$$

with the minmod function defined as $\text{minmod}(a,b) = \begin{cases} \min(|a|, |b|) & \text{if } ab > 0, \\ 0 & \text{else.} \end{cases}$

The choice of the one-sided finite difference depends on the sign of $(\text{sign}(\phi)\mathbf{n}) \cdot \mathbf{e_x}$ to propagate the function values from the membrane toward the fluid domain,

$$\frac{\partial h}{\partial x} = \begin{cases} \dfrac{\partial h}{\partial x}\Big|_l & \text{if } (\text{sign}(\phi)\mathbf{n}) \cdot \mathbf{e_x} > 0, \\ \dfrac{\partial h}{\partial x}\Big|_r & \text{else.} \end{cases} \tag{89}$$

The method is identical in the other directions. The value of the function at the interface is of utmost importance because it will be extended to the whole domain. The interface does not conform with the mesh grid and special care is needed to compute the interface values within mesh cells. We propose in this paper a subcell method to improve the accuracy on the cells cut by the membrane.

First, considering that the membrane cuts the interval $[x_i, x_{i+1}]$, we define $\theta_q \Delta x$ the distance between x_i and the membrane (*cf.* Fig. 2). This distance can be computed using the level set function and second order Taylor expansions

$$\phi(x_m) = 0 = \phi_i + \theta_q \Delta x \frac{\partial \phi}{\partial x}\Big|_i + \frac{(\theta_q \Delta x)^2}{2}\frac{\partial^2 \phi}{\partial x^2}\Big|_i + O(\Delta x^2). \tag{90}$$

We construct the derivatives of ϕ using central finite differences and solve the quadratic polynomial to find θ_q,

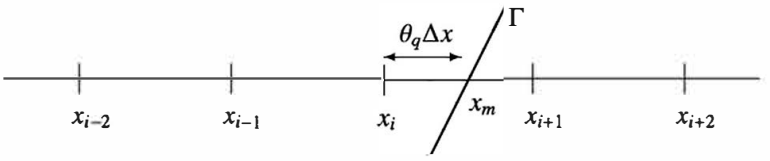

Fig. 2. The membrane Γ crossing the x-grid.

$$\theta_q = \begin{cases} \dfrac{1}{2}\left(\dfrac{-(\phi_{i+1}-\phi_{i-1}) - \text{sign}(\phi_i)\sqrt{(\phi_{i+1}-\phi_{i-1})^2 - 8\phi_i(\phi_{i+1}-2\phi_i+\phi_{i-1})}}{\phi_{i+1}-2\phi_i+\phi_{i-1}}\right) & \text{if } \left|\dfrac{\partial^2\phi}{\partial x^2}\Big|_i\right| > \epsilon, \\ -\dfrac{2\phi_i}{\phi_{i+1}-\phi_{i-1}} \text{ else,} \end{cases} \tag{91}$$

with $\epsilon = 10^{-8}$, the limiting value below which we consider the quadratic polynomial linear.

Similarly, if the membrane cuts the interval $[x_{i-1}, x_i]$, θ_q becomes

$$\theta_q = \begin{cases} \dfrac{1}{2}\left(\dfrac{(\phi_{i+1}-\phi_{i-1}) - \text{sign}(\phi_i)\sqrt{(\phi_{i+1}-\phi_{i-1})^2 - 8\phi_i(\phi_{i+1}-2\phi_i+\phi_{i-1})}}{\phi_{i+1}-2\phi_i+\phi_{i-1}}\right) & \text{if } \left|\dfrac{\partial^2\phi}{\partial x^2}\Big|_i\right| > \epsilon, \\ \dfrac{2\phi_i}{\phi_{i+1}-\phi_{i-1}} \text{ else.} \end{cases} \tag{92}$$

The value of the scalar field h_m at the membrane location can then be computed. If the membrane cuts the interval $[x_i, x_{i+1}]$,

$$h_m = \frac{1}{2}(-\theta_q + \theta_q^2)h_{i-1} + (1-\theta_q^2)h_i + \frac{1}{2}(\theta_q + \theta_q^2)h_{i+1}, \tag{93}$$

and, if the membrane cuts the interval $[x_{i-1}, x_i]$,

$$h_m = \frac{1}{2}(\theta_q + \theta_q^2)h_{i-1} + (1-\theta_q^2)h_i + \frac{1}{2}(-\theta_q + \theta_q^2)h_{i+1}. \tag{94}$$

The choice of the data points allows us to not favour values of points from one region instead of the other. Even if there might have a difference in the computation of h at the membrane, the computation is consistent because the error decreases with the mesh size.

Finally, the gradient computation of the extension algorithm is modified so that the point located at the membrane is considered in the numerical scheme. For example, if the membrane cut the interval $[x_i, x_{i+1}]$,

$$\frac{\partial h}{\partial x}\Big|_{\text{Ir}} = \frac{h_m - h_i}{\theta_q \Delta x} - \frac{\theta_q \Delta x}{2}\,\text{minmod}\left(\frac{h_{i+1}-2h_i+h_{i-1}}{\Delta x^2}, \frac{h_{i+2}-2h_{i+1}+h_i}{\Delta x^2}\right), \tag{95}$$

and, if the membrane cut the interval $[x_{i-1}, x_i]$,

$$\frac{\partial h}{\partial x}\Big|_{\text{Il}} = \frac{h_i - h_m}{\theta_q \Delta x} + \frac{\theta_q \Delta x}{2}\,\text{minmod}\left(\frac{h_{i+1}-2h_i+h_{i-1}}{\Delta x^2}, \frac{h_i-2h_{i-1}+h_{i-2}}{\Delta x^2}\right). \tag{96}$$

This method is identical in the other directions. The example in section 4.2 justifies the use of the subcell method by comparing results from the extension algorithm with or without the subcell resolution.

To avoid the time step restriction due to the subcell method, the extension algorithm is solved implicitly with the Gauss-Seidel method. The extension algorithm can then be written as

$$h_{i,j}^{n+1} + \Delta\tau\ \text{sign}(\phi)\left(n_x\frac{\partial h}{\partial x}^{n+1} + n_y\frac{\partial h}{\partial y}^{n+1} + n_z\frac{\partial h}{\partial z}^{n+1}\right)\Big|_{i,j} = h_{i,j}^n \tag{97}$$

The fictitious time step $\Delta\tau$ is defined as

$$\Delta\tau = \alpha \min(\Delta x, \Delta y, \Delta z), \tag{98}$$

with $\alpha = 0.3$ which minimises the computation time to reach the fully extended field. Few iterations of the Gauss-Seidel algorithm are needed to reach the extended state because the scalar field at the membrane faintly evolves at each time step.

In our fluid-membrane solver, the extension algorithm is applied at each time step on the solid velocity field $\mathbf{u_s}$ which allows to compute the components of the Cauchy-Green tensor (see section 3.3). The solid velocity field corresponds to the fluid velocity at the membrane extended in the whole domain. This algorithm is also applied directly on the components of $\bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}}$ every 10 temporal iteration. The execution of the extension algorithm on the tensor components at each time step would increase the computation time and decreasing the extension frequency would generate fictitious normal derivatives at the membrane location.

3.7. Time step constraints and stability condition

The effects of the convection and the membrane elasticity lead to time step restrictions to ensure the numerical stability:

$$\Delta t_{conv} = \frac{1}{\dfrac{\max(|u|)}{\Delta x} + \dfrac{\max(|v|)}{\Delta y} + \dfrac{\max(|w|)}{\Delta z}} \quad \text{and} \quad \Delta t_{elas} = \frac{1}{2}\sqrt{\frac{\max(\rho^+, \rho^-)}{E_s}}\min(\Delta x, \Delta y, \Delta z)^{3/2}. \tag{99}$$

The restriction due to the elasticity is similar to the restriction due to the surface tension effects in two-phase flow cases [36]. There is no restriction due to viscosity because of the implicit computation of the viscous term. The global condition on the time step becomes

$$\frac{1}{\Delta t} > \frac{1}{\Delta t_{conv}} + \frac{1}{\Delta t_{elas}}. \tag{100}$$

All the numerical algorithms, from the transport of the level set function to the solving of the Navier-Stokes equations with the elastic contribution, are integrated in a second order TVD Runge-Kutta scheme.

To fully resolve the dynamics near the membrane, the mesh grid must be fine enough to compute accurately the elastic contribution on the flow field. This results in a stability condition depending on a mesh Reynolds number. This mesh Reynolds number is based upon the characteristic velocity due to the membrane elasticity defined as $U_{elas} = \sqrt{E_s/(\max(\rho^+, \rho^-)L)}$ and on the mesh size Δx such as

$$Re_{\Delta x} = \frac{U_{elas}\Delta x}{\min(\nu^+, \nu^-)} = \frac{\Delta x}{\min(\nu^+, \nu^-)}\sqrt{\frac{E_s}{\max(\rho^+, \rho^-)L}}, \tag{101}$$

with $\nu = \mu/\rho$ the kinematic viscosity. The stability condition requires a mesh Reynolds number around one or less.

4. Numerical results

In this section, we compute some well-known benchmarks and compare our numerical results with those from the literature. In a first time, we present a detailed comparison between the proposed numerical solver against a theoretical solution involving the droplet migration induced by a prescribed surface tension gradient. This preliminary test-case allows us demonstrating the correct behaviour of the proposed numerical discretization to impose the jump condition on the normal derivative of the tangential velocity. Next, the improved extension algorithm is implemented on the benchmark from [54] to highlight the efficiency of the method and its convergence rate. Then, the stretched and pressurized membrane immersed in a fluid, first introduced in [55], is considered with these new numerical methods. After the grid sensitivity study, the results on the radii evolution, the pressure jump and the velocity fields are compared to the results from [49]. The test case is extended to different fluids in the computational domain. First, different viscosities, and then, different densities, are enforced on each side of the membrane. Finally, the capsule immersed in a shear flow from [56] is implemented and its deformation throughout time is compared to the results from the literature.

4.1. Test-case: rising of a droplet in a quiescent liquid due to Marangoni stresses

In this section, we provide a numerical example to test the accuracy of the jump condition computation on the normal derivatives of the tangential velocity, by performing a simulation of an axisymmetric droplet with surface tension gradient. This configuration is based on the Marangoni effects test-case presented in [57]. Along a droplet interface, non uniform local temperature or surfactants concentration can trigger gradients of surface tension γ, which generates a tangential stress jump. Then, a flow, called Marangoni convection, occurs to balance the gradients of γ, leading to a rising motion of the droplet in the quiescent liquid. This viscous tangential stress jump is assimilable to the jump condition of equation (46) where

$$\mathbf{f}_\tau = \nabla_s \gamma \ . \tag{102}$$

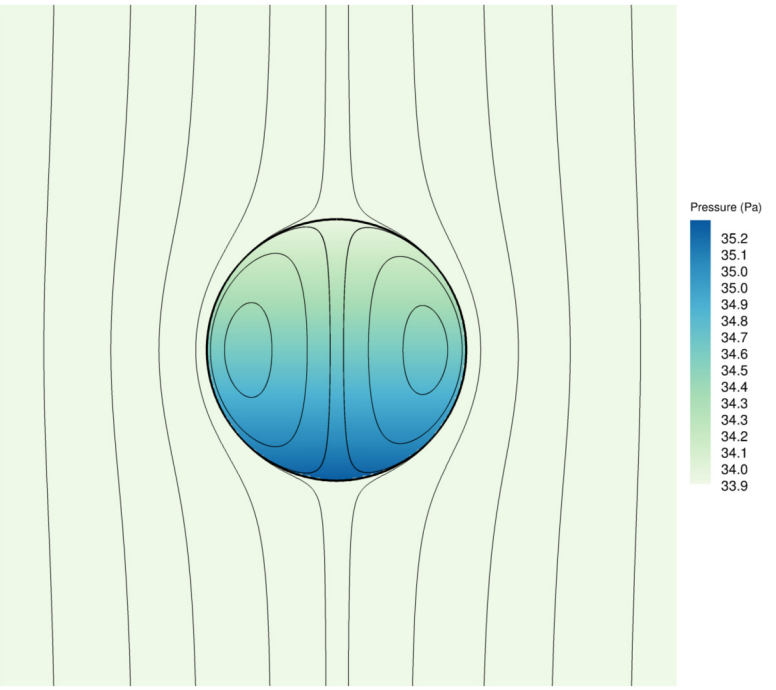

Fig. 3. Pressure field and streamlines of the rising droplet. (For interpretation of the colours in the figures, the reader is referred to the web version of this article.)

We performed axisymmetric simulations in a domain of size $[l_r, l_z]$ in respectively radial and longitudinal directions, in a moving frame until a steady state. We consider a droplet of radius R, viscosity μ^+ and density ρ^+, in a quiescent liquid (μ^-, ρ^-), with a gradient of surface tension imposed by

$$\gamma(z) = \gamma_0\left(1 - \beta\frac{z}{l_z}\right), \tag{103}$$

where $\gamma_0 = 10$, $\beta = 0.27$, $l_z = 16R$, and $l_r = l_z/2$. To ensure that the motion is solely due to Marangoni stresses, we impose $\mu^+ = \mu^- = 0.1$, and $\rho^+ = \rho^- = 0.02$. From the theoretical studies of [58] and [59], where the Stokes equations are solved around and within a spherical droplet, an analytical expression of the droplet rising velocity is proposed in [60] accounting for the jump condition due to Marangoni stresses. In [57], the authors have adapted this analytical velocity to propose a benchmark case without any gravity:

$$u_\infty = \frac{2\gamma_0\beta R}{l_z(6\mu^- + 9\mu^+)}\,. \tag{104}$$

This expression is valid for a spherical droplet, i.e. at Capillary number $Ca = \mu^- u_d/\gamma_0 \ll 1$, and for the creeping flow limit at Reynolds number $Re = \rho^- u_d D/\mu^- \ll 1$, where u_d is the droplet velocity and D its diameter. The parameters were chosen in a such a way that $Re = 0.045$ and $Ca = 0.0023$, based on the terminal velocity of equation (104). From the studies of [58], [59] and [60], and considering the tangential stress jump as a boundary condition, we have calculated the analytical expression of tangential velocity around and within the droplet (more details are provided in the Appendix C),

$$u_\theta^- = -u_\infty\left(1 + \frac{1}{2}R^3 r^{-3}\right)\sin(\theta) \tag{105}$$

$$u_\theta^+ = -\frac{3u_\infty}{2R^2}\left(2r^2 - R^2\right)\sin(\theta) \tag{106}$$

with u_θ^- and u_θ^+ the tangential velocities respectively outside and inside the droplet. From this solution, the tangential velocity at the interface is given by the following expression,

$$u_\theta^-|_{r=R} = u_\theta^+|_{r=R} = \frac{3}{2}u_\infty\sin(\theta)\,. \tag{107}$$

In Fig. 3, are plotted the droplet shape and the pressure field inside the droplet which generates its migration. In Figs. 4 and 5, the dimensionless rising velocity of the bubble is plotted until steady state for four meshes with respectively implicit

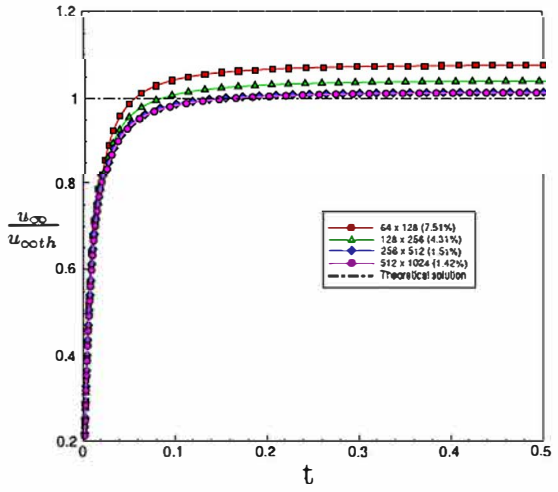

Fig. 4. Rising velocity of the droplet as a function of time, for the implicit resolution, with the error rate when comparing to the theoretical solution.

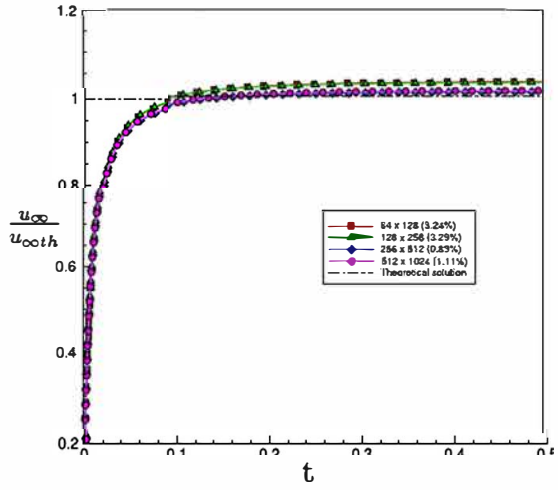

Fig. 5. Rising velocity of the droplet as a function of time, for the explicit resolution, with the error rate when compating to the theoretical solution.

and explicit resolutions for the temporal discretization of the viscous terms. The error rates were calculated by comparison with the analytical solution. The numerical results are in good agreement with the theoretical expression of the rising velocity provided by eq. (104), for both resolutions.

In Figs. 6 and 7, we plot the dimensionless tangential velocity profile along the droplet surface, compared to the analytical expression eq. (107) that we present in the appendix. The more the mesh is refined, the better the agreement is with the theoretical solution.

In Figs. 8 and 9, the tangential stress profile $\mu \partial u_\theta / \partial n$ is plotted along the axis $z = 0$ for the most refined mesh, at the droplet equator, and compared to the tangential stress calculated from the solutions eq. (105) and eq. (106). Then, the numerical accuracy of this jump condition is verified in Table 1 by comparing the tangential stress jump from simulations to the calculated $\nabla_s \gamma$ value from eq. (103). At each mesh refinement, the error rate is divided by two, typical of a first order accuracy as it was expected for the numerical discretization of the singular source terms.

4.2. Extension algorithm

In this section, we present numerical results of the extension algorithm coupled with the subcell resolution on an example inspired by [54]. Let us consider a 2D computational domain $[-\pi, \pi] \times [-\pi, \pi]$. Initially, a level set function is defined as $\phi(x, y) = \sqrt{x^2 + y^2} - R_0$, the zero level set of which corresponds to a circle of radius $R_0 = 2$. A scalar field denoted by h is initialised as

$$h(x, y) = \cos(x) \sin(y), \tag{108}$$

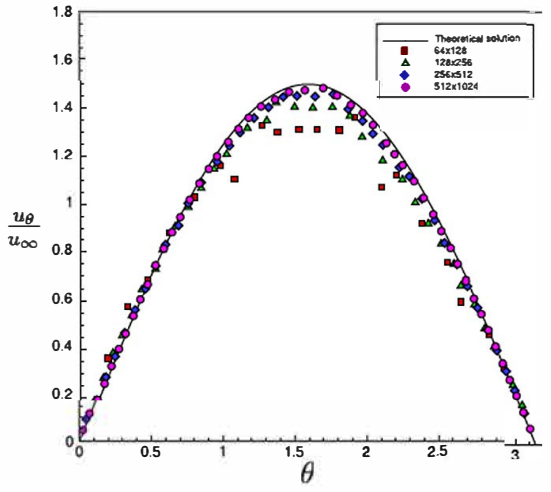

Fig. 6. Tangential velocity profile along the droplet surface, as a function of the angle θ, for the implicit resolution.

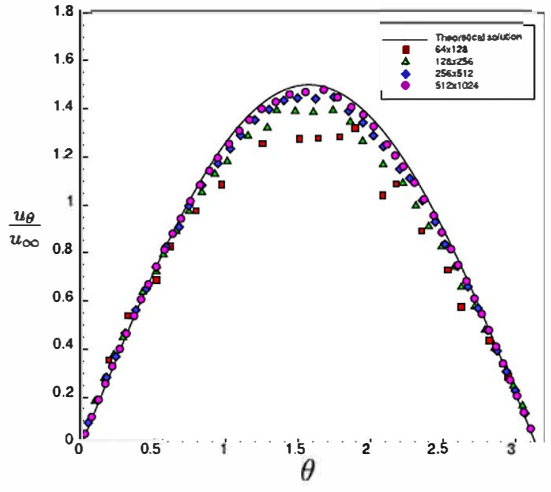

Fig. 7. Tangential velocity profile along the droplet surface, as a function of the angle θ, for the explicit resolution.

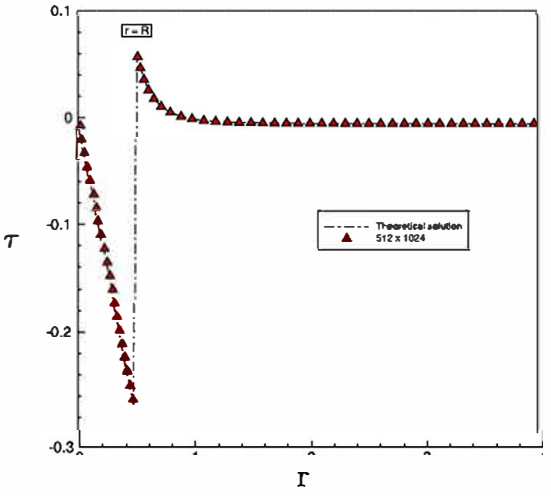

Fig. 8. Tangential viscous stress profile along the radius, at $z = 0$ for the implicit resolution.

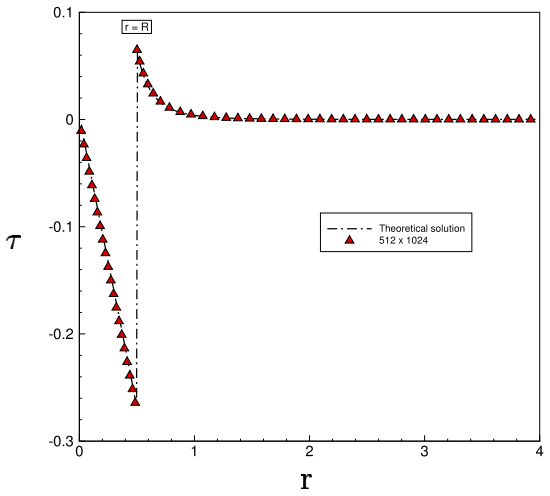

Fig. 9. Tangential viscous stress profile along the radius, at $z = 0$ for the explicit resolution.

Table 1
Numerical accuracy of tangential stress jump compared to the calculated value of $\nabla_s \gamma$ from eq. (103).

Grid	Implicit	Explicit
64×128	15.98%	17.21%
128×256	8.68%	9.22%
256×512	4.49%	4.68%
512×1024	2.28%	2.37%

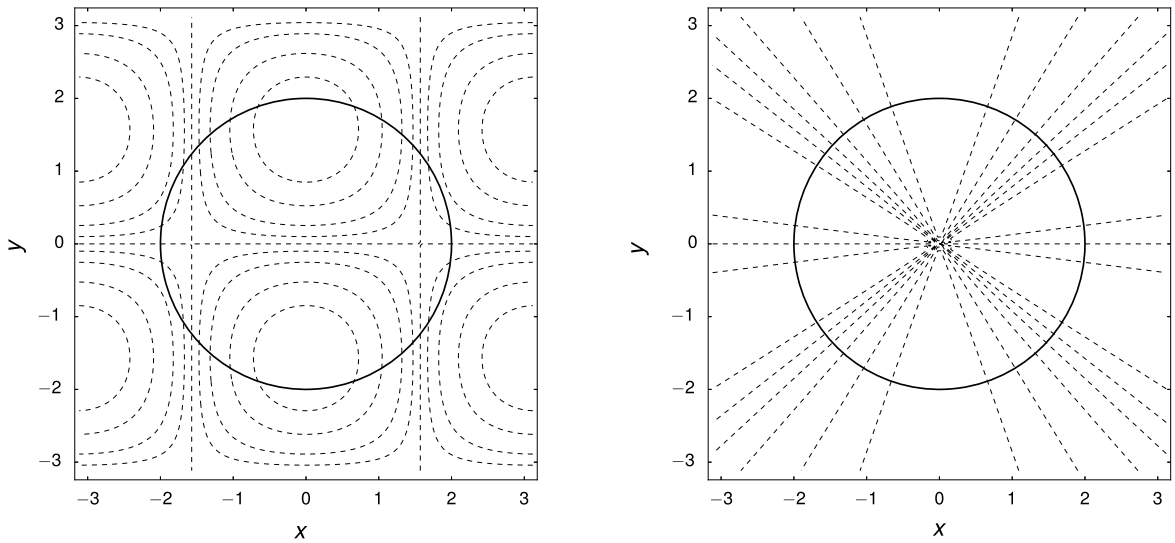

Fig. 10. Initial and final scalar fields $h(x, y)$ with the circular zero level set represented by the bold solid black line.

and some contours are shown in the left plot of Fig. 10 with the zero level set line depicted in solid black.

The extension algorithm described in section 3.6 is enforced on the scalar field h and the result is plotted on the right plot of Fig. 10. We observe that the contours of the obtained scalar field respect well the condition of no normal derivative at the membrane. The exact solution of the extended field h is easily computable and equals

$$h(x, y) = \cos\left(R_0 \frac{x}{\sqrt{x^2 + y^2}}\right) \sin\left(R_0 \frac{y}{\sqrt{x^2 + y^2}}\right). \tag{109}$$

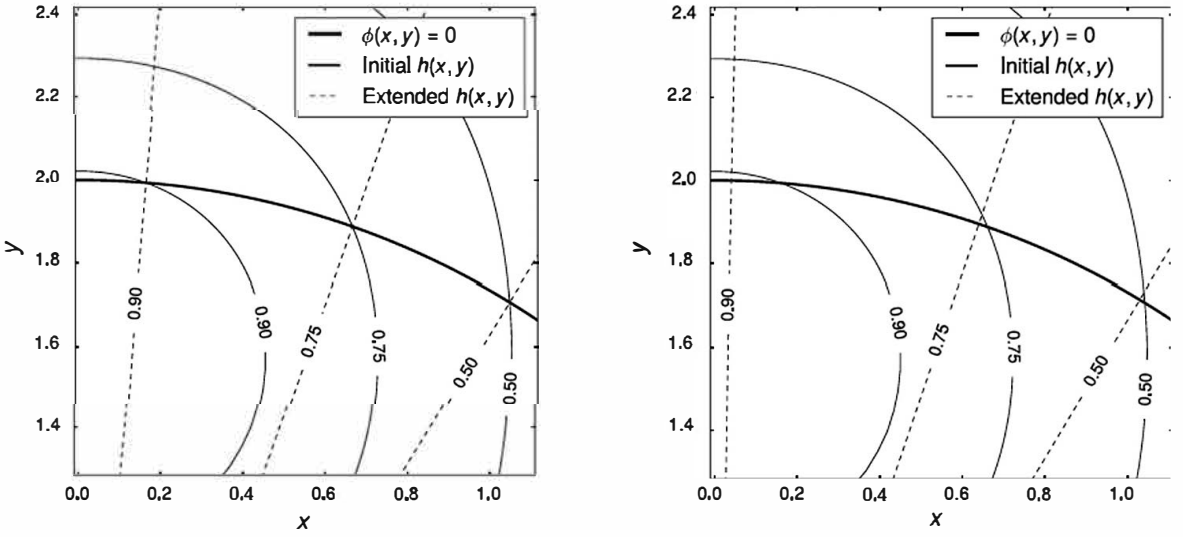

Fig. 11. Zoom on the scalar field $h(x, y)$ before and after the extension with the subcell resolution on the left and without on the right.

Table 2
Numerical accuracy of the extension method with the subcell resolution in the circular case.

Grid	Whole domain $\phi(x,y) > -1.6$				Near the membrane $\lvert\phi(x,y)\rvert < 1.2\Delta x$			
	E_1	R_1	E_∞	R_∞	E_1	R_1	E_∞	R_∞
64^2	1.19×10^{-3}		4.17×10^{-2}		5.37×10^{-5}		3.18×10^{-4}	
128^2	3.38×10^{-4}	1.82	1.76×10^{-2}	1.25	6.56×10^{-6}	3.03	4.37×10^{-5}	2.87
256^2	9.01×10^{-5}	1.91	5.71×10^{-3}	1.62	8.24×10^{-7}	2.99	6.20×10^{-6}	2.82
512^2	2.30×10^{-5}	1.97	1.65×10^{-3}	1.79	1.01×10^{-7}	3.03	7.27×10^{-7}	3.09

Table 3
Numerical accuracy of the extension method without the subcell resolution in the circular case.

Grid	Whole domain $\phi(x,y) > -1.6$				Near the interface $\lvert\phi(x,y)\rvert < 1.2\Delta x$			
	E_1	R_1	E_∞	R_∞	E_1	R_1	E_∞	R_∞
64^2	1.11×10^{-2}		6.12×10^{-2}		1.06×10^{-2}		2.82×10^{-2}	
128^2	3.47×10^{-3}	1.68	2.38×10^{-2}	1.36	3.50×10^{-3}	1.59	8.72×10^{-3}	1.69
256^2	1.66×10^{-3}	1.06	9.30×10^{-3}	1.35	1.68×10^{-3}	1.06	5.77×10^{-3}	0.60
512^2	6.93×10^{-4}	1.26	3.68×10^{-3}	1.34	6.91×10^{-4}	1.28	2.74×10^{-3}	1.07

Fig. 11 focuses on the scalar field h near the membrane, represented by a bold solid black line, in the specific region $[0, 1.1] \times [1.3, 2.4]$. The initial contours of h are plotted with solid black lines and the extended scalar field with dashed black lines. In the left plot of Fig. 11, we observe that the values of the scalar field h at the membrane do not change after the extension. The solid lines correspond exactly to the dashed lines at the membrane location. The same computation is done without the subcell method and the result is illustrated in the right plot of Fig. 11. We observe that the contours do not correspond as well. The zero normal derivative at the membrane is well respected but the values of the scalar field at the membrane have been modified. The scalar field variation is more important where the initial scalar field is tangent to the membrane. Indeed, the contour $h(x, y) = 0.9$ after the extension without the subcell method exhibits an important variation at the membrane position.

Table 2 summarises the errors of the extended scalar field with the subcell resolution compared to the exact solution of equation (109). Two errors are computed: in the whole domain, except from the centre of the circle where the normal directions to the membrane meet up, and in the region near the membrane. The extension algorithm with the subcell method is third order accurate near the zero level set and almost second order in the whole region.

Considering the extension algorithm without the subcell method, the accuracy decreases notably, as shown in Table 3. Near the membrane, the order of magnitude of the error is at least two times higher than with the subcell resolution. The convergence rate of the method is around one in the whole domain and near the zero level set. The study of the accuracy and the contour plots of the scalar field h explains that the subcell method must be considered to extend the right value of the scalar field at the membrane toward its normal direction.

Now, the level set function is modified to obtain an ellipse of major axis $a = 2.5$ and minor axis $b = 1.25$ as the zero level set. The initialisation of the scalar field $h(x, y)$ stays identical and the exact solution of the extended field is obtained

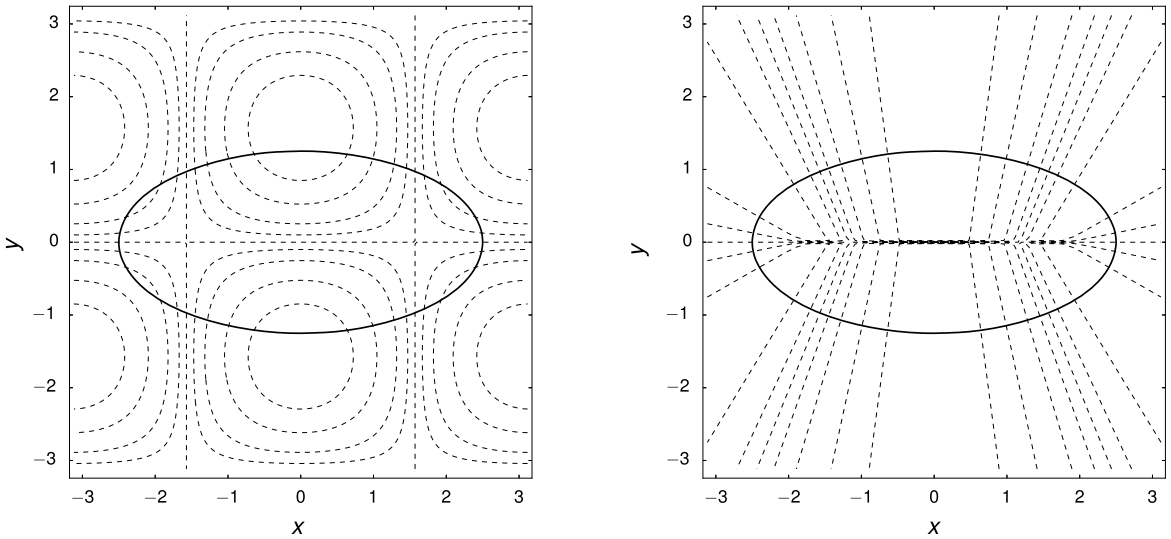

Fig. 12. Initial and final scalar fields $h(x, y)$ with the zero level set represented by the bold solid black line.

Table 4
Numerical accuracy of the extension method with the subcell resolution in the case of the ellipse.

Grid	Whole domain $\|y\| > 0.15$				Near the membrane $\|\phi(x, y)\| < 1.2\Delta x$			
	E_1	R_1	E_∞	R_∞	E_1	R_1	E_∞	R_∞
64^2	6.49×10^{-4}		7.93×10^{-3}		8.83×10^{-5}		8.60×10^{-4}	
128^2	1.74×10^{-4}	1.90	4.25×10^{-3}	0.90	9.74×10^{-6}	3.18	1.23×10^{-4}	2.80
256^2	4.37×10^{-5}	1.99	1.36×10^{-3}	1.64	1.31×10^{-6}	2.90	1.65×10^{-5}	2.90
512^2	1.10×10^{-5}	1.99	3.89×10^{-4}	1.81	1.68×10^{-7}	2.96	2.28×10^{-6}	2.85
1024^2	2.75×10^{-6}	2.00	1.04×10^{-4}	1.90	2.24×10^{-8}	2.91	2.95×10^{-7}	2.95

Table 5
Numerical accuracy of the extension method without the subcell resolution in the case of the ellipse.

Grid	Whole domain $\|y\| > 0.15$				Near the interface $\|\phi(x, y)\| < 1.2\Delta x$			
	E_1	R_1	E_∞	R_∞	E_1	R_1	E_∞	R_∞
64^2	4.95×10^{-2}		1.60×10^{-1}		5.62×10^{-2}		1.58×10^{-1}	
128^2	9.15×10^{-3}	2.44	4.46×10^{-2}	1.84	1.05×10^{-2}	2.42	4.24×10^{-2}	1.89
256^2	1.75×10^{-3}	2.38	1.07×10^{-2}	2.06	2.05×10^{-3}	2.36	9.88×10^{-3}	2.10
512^2	5.08×10^{-4}	1.79	2.95×10^{-3}	1.86	5.69×10^{-4}	1.85	2.78×10^{-3}	1.83
1024^2	2.19×10^{-4}	1.22	9.58×10^{-4}	1.62	2.38×10^{-4}	1.26	9.32×10^{-4}	1.58

by an iterative process. Fig. 12 shows the initial and final scalar field $h(x, y)$ with the ellipse corresponding to the zero level set represented in bold solid black. The same study on the extension algorithm is done on the ellipse case with and without the subcell resolution and are detailed in Tables 4 and 5 respectively.

The errors with the exact solution are computed in the whole domain, except from the region $|y| < 0.15$ where singularities due to the computation of the normal direction to the membrane appear, and in the region near the membrane. With the subcell resolution, the errors in the whole domain and near the membrane are always at least one order of magnitude lower than without it. The extension algorithm is still third order accurate near the membrane with the subcell resolution and second order accurate in the whole domain. Without the subcell resolution, the computed order of convergence is important, around 2, for the coarser meshes but decreases significantly with finer meshes.

The study on the ellipse case shows that the third order accuracy obtained near the interface is due to the subcell resolution and is not an artefact due to the simple circular membrane. Without the subcell resolution, the computed order of convergence oscillates between 1 and 2 depending on the membrane shape and the mesh considered.

4.3. Stretched and pressurized membrane immersed in a fluid

The stretched and pressurized membrane immersed in a viscous fluid is implemented in this section. It has been used by several authors to test the immersed boundary method and the immersed interface method [61,9,55]. In this example, the elastic contribution will be implemented using the Ghost-Fluid viscous Conservative Method of section 3.2.2. It consists in a 2D membrane, the shape of which is an ellipse of major axis $a = 0.75$ and minor axis $b = 0.5$, relaxing in a fluid at rest.

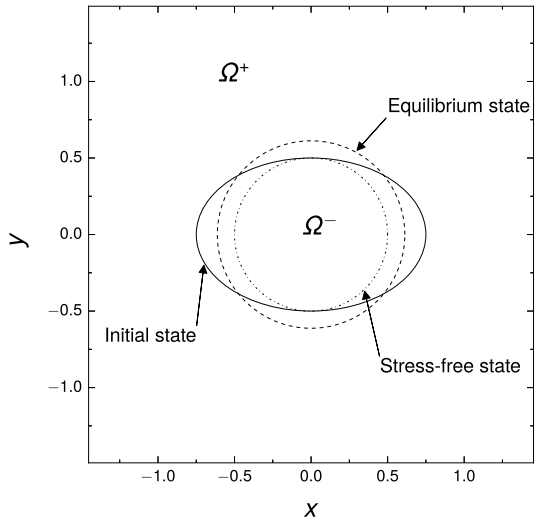

Fig. 13. The membrane in different states.

In its stress-free state, we consider that the membrane is a circle of radius $r_0 = 0.5$. The equilibrium shape of the initially stretched membrane is a circle with the same amount of liquid within. The equilibrium radius equals $r_e = \sqrt{ab} \approx 0.61237$. The domain is a closed square of length $L = 3$ and wall boundary conditions are considered. The ellipse is initially centred in the domain. The three different states of the membrane are depicted in Fig. 13. Different densities and viscosities of the fluid are considered: $\rho \in [1, 10, 100]$ and $\mu \in [0.01, 0.1, 1]$. Depending on the fluid properties, the membrane will relax toward its circular shape differently.

In this benchmark, the membrane follows the linear Hooke law. However, the numerical methods presented previously in this paper deal with hyperelastic material laws. Fortunately, in 2D cases, the linear Hooke law can be considered using the following methodology because the membrane is only stretched along its tangent direction denoted $\mathbf{e}_\tau$. The first invariant of the Cauchy-Green tensor gives directly the square of the principal strain λ_τ. The linear Hooke law corresponds to the following surface stress tensor

$$\bar{\bar{\boldsymbol{\sigma}}}_{\boldsymbol{s}} = E_s(\sqrt{I_1} - 1)\bar{\bar{\boldsymbol{P}}}, \tag{110}$$

with $I_1 = \lambda_\tau^2$, $\bar{\bar{\boldsymbol{P}}} = \mathbf{e}_\tau \otimes \mathbf{e}_\tau$ and the value of the surface elastic modulus $E_s = 10$.

Therefore, the surface stress tensor can directly be computed with the projection tensor and the first invariant of the surface Cauchy-Green tensor. The latter tensor is initialised by the projection tensor multiplied by the initial stretching of the membrane. Considering a uniform stretching, it corresponds to the square of the ratio of the membrane perimeter between its initial state P_0 and its stress-free state P_R,

$$\bar{\bar{\boldsymbol{B}}}_{\boldsymbol{s}}(t=0) = \lambda_\tau^2(t=0)\bar{\bar{\boldsymbol{P}}}(t=0) = \left(\frac{P_0}{P_R}\right)^2 \bar{\bar{\boldsymbol{P}}}(t=0) \tag{111}$$

with $P_0/P_R \approx 1.262$.

The evolution of the radii is investigated and compared to the results from [49]. First, to ensure the validity of our computation, a grid sensitivity study has been performed for $\rho = 1$ and two different viscosities $\mu = 0.1$ and $\mu = 0.01$. The computations have been performed on three different meshes for each viscosity. Fig. 14 shows that for $\mu = 0.1$, the three meshes, 128^2, 256^2 and 512^2, give the same radii evolution. In this case, the 128^2 mesh is fine enough to do the comparison. With $\mu = 0.01$, finer grids are needed to reach a steady radii evolution. The temporal evolutions are close to each other at early flow time but deflect with time. This deviation decreases when refining the mesh. For meshes containing more than 512 cells in each direction, the same solution is obtained and gives consistent results with [49]. In what follows, the 512^2 mesh will be considered in case of a viscosity of $\mu = 0.01$. For lower values of the viscosity, the 128^2 mesh is considered.

We observe in Fig. 14 that in both cases, the membrane oscillates around its equilibrium value. The magnitude of the oscillations decreases with time until the membrane reaches the circular shape. The theoretical radius r_e is obtained for the horizontal r_x and vertical r_y radii in the steady state. The damping of the oscillations increases with the viscosity but its frequency remains stable.

Fig. 15 shows the pressure distribution in the computational domain at $t = 0.5$ and $t = 2$. The sharp jump on the pressure is clearly visible at the membrane location. At early flow time, the pressure field exhibits important variations in each fluid

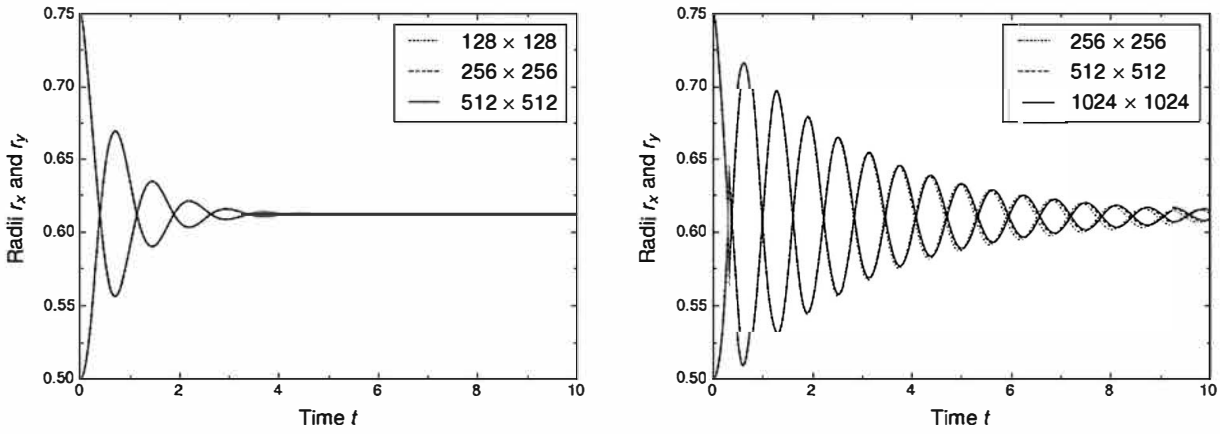

Fig. 14. Radii evolution with $\mu = 0.1$ on the left and $\mu = 0.01$ on the right for three different meshes.

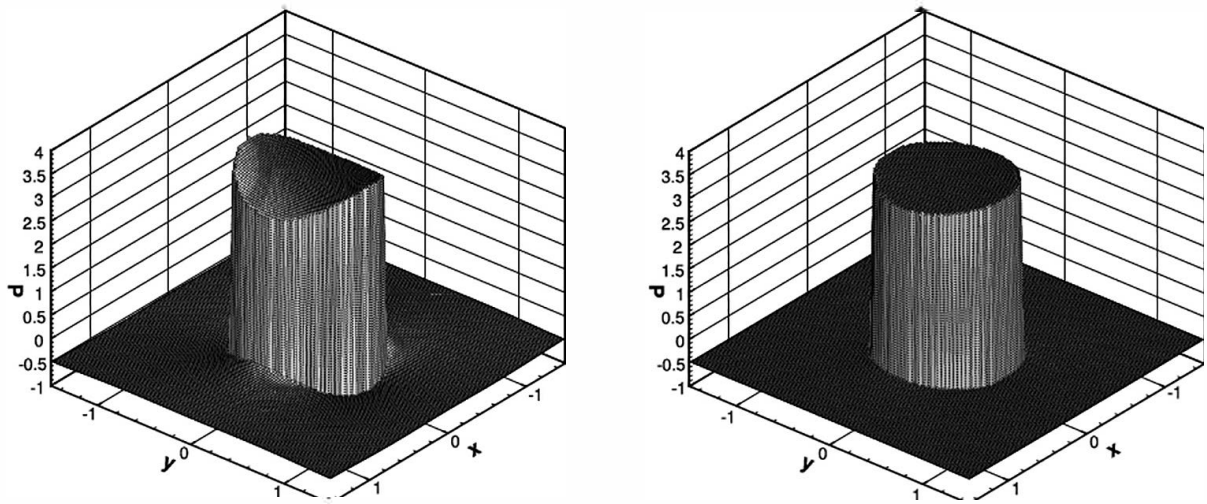

Fig. 15. Pressure distribution with $\mu = 0.1$ at $t = 0.5$ on the left and $t = 2$ on the right.

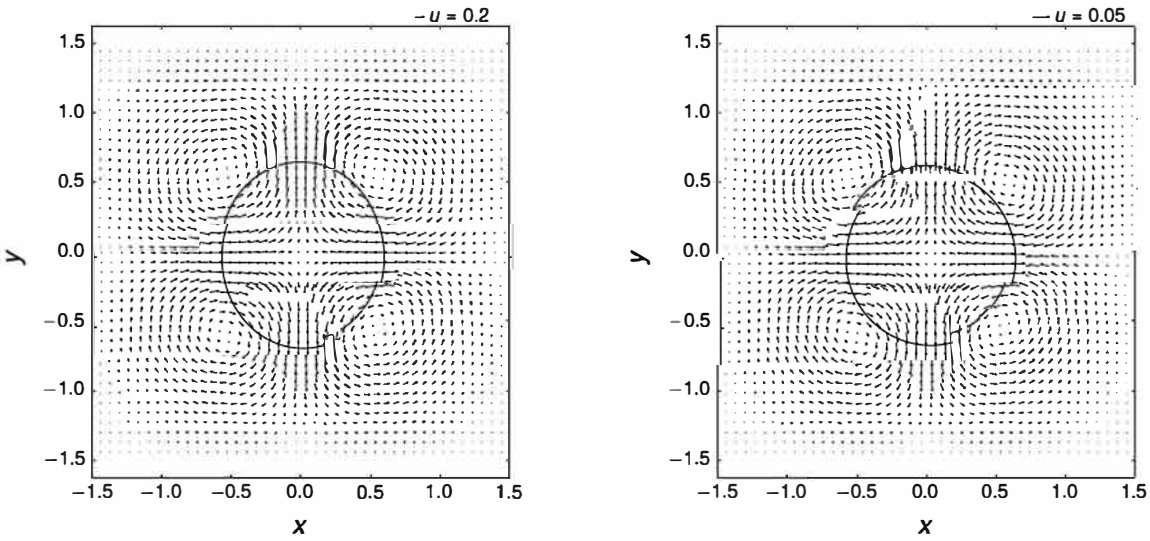

Fig. 16. Velocity field with $\mu = 0.1$ at $t = 0.5$ on the left and $t = 2$ on the right.

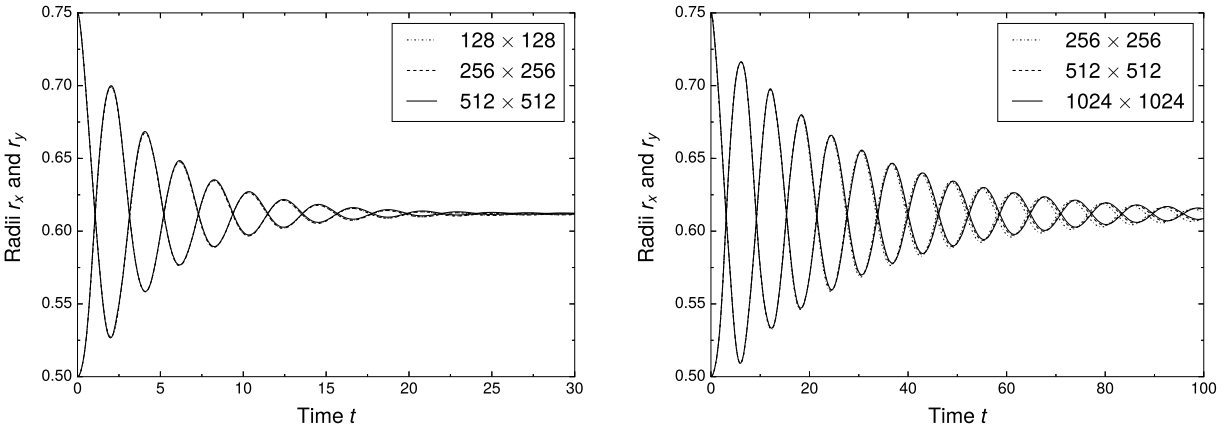

Fig. 17. Radii evolution with $\rho = 10$ on the left and $\rho = 100$ on the right for three different meshes.

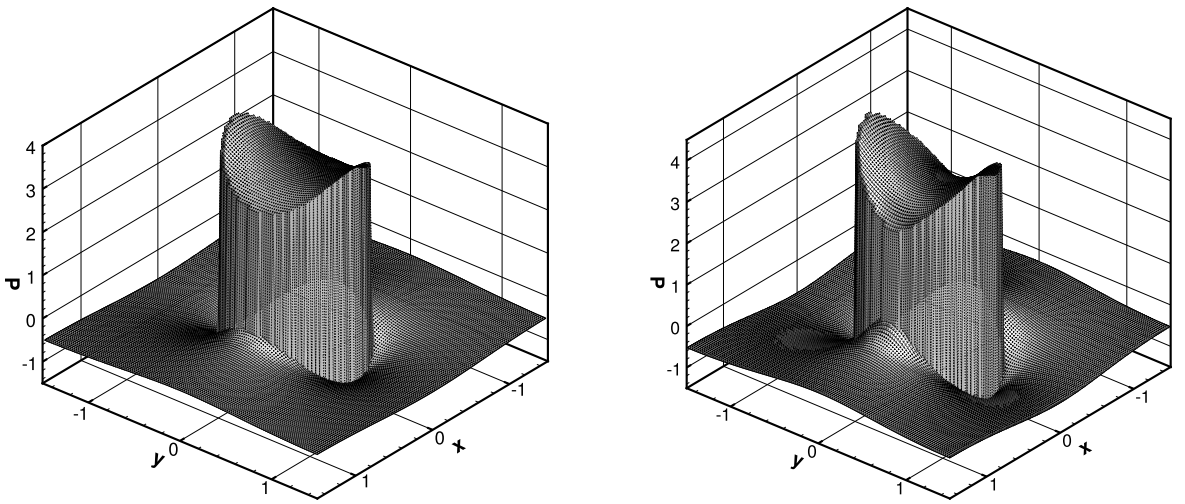

Fig. 18. Pressure distribution with $\rho = 10$ at $t = 1.5$ on the left and with $\rho = 100$ at $t = 5$ on the right.

region because of the fluid motion. At $t = 2$, the pressure field is almost constant inside and outside the membrane. In the steady state, only the pressure jump remains with constant values of the pressure in each phase.

Fig. 16 shows the velocity field at two different times with $\mu = 0.1$ in the whole domain. At $t = 0.5$ and $t = 2$, four main vortices are located in the fluid outside the membrane leading the membrane to deform in the vertical direction. The pressure and velocity fields of Figs. 15 and 16 are in good agreement with the results from [49].

These simulations have also been implemented without the extension algorithm and the results are presented in Appendix A. This study shows that the extension algorithm is necessary to keep the computation stable throughout time by removing the normal derivatives of the solid scalar fields.

The membrane behaviour is investigated for higher values of the density. The viscosity is kept at $\mu = 0.1$ and the density is increased to $\rho = 10$ and $\rho = 100$. A grid sensitivity study is done on several meshes. We observe in Fig. 17 that the radii evolution reaches the same solution from 128 cells onwards in each direction with $\rho = 10$ and from 512 cells onwards in each direction with $\rho = 100$. In what follows, the 512^2 mesh will be considered when the density reaches $\rho = 100$, the 128^2 mesh will be used otherwise.

We observe that the time to relax the membrane increases with the density. The frequency of the oscillations seems to decrease with the square root of the density. Fig. 18 exhibits two examples of the pressure field for the two higher densities. The sharp jump condition on the pressure is well predicted by the numerical simulations even in the right plot of Fig. 18 where the pressure variation is important in each region.

4.4. Stretched and pressurized membrane immersed in fluids with different viscosities

Now, we distinguish the fluids inside and outside the membrane. In this section, the two fluids have different viscosities μ^+ and μ^-. As depicted in Fig. 13, the superscript $+$ corresponds to the fluid outside the membrane and the superscript $-$

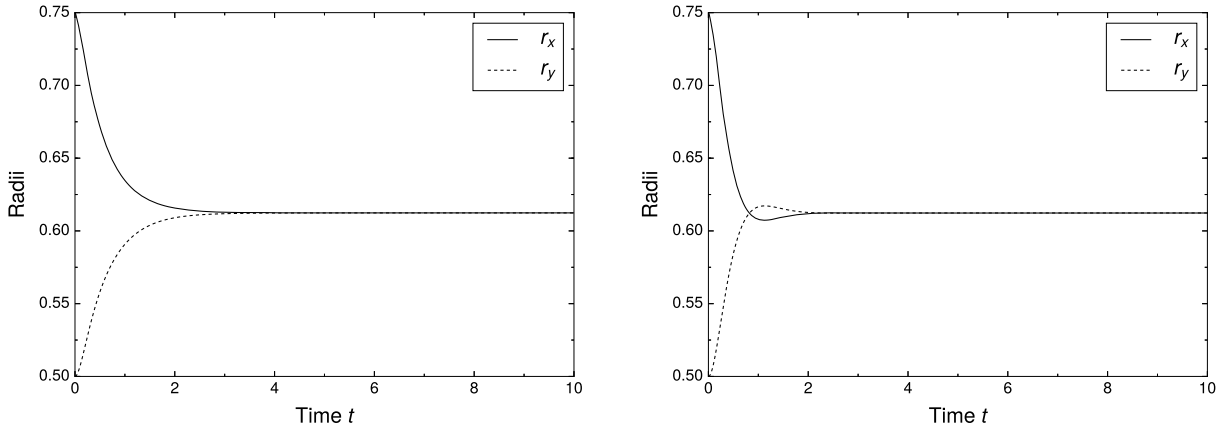

Fig. 19. Radii evolution with $\mu^+ = 1$ and $\mu^- = 0.1$ on the left and $\mu^+ = 0.1$ and $\mu^- = 1$ on the right.

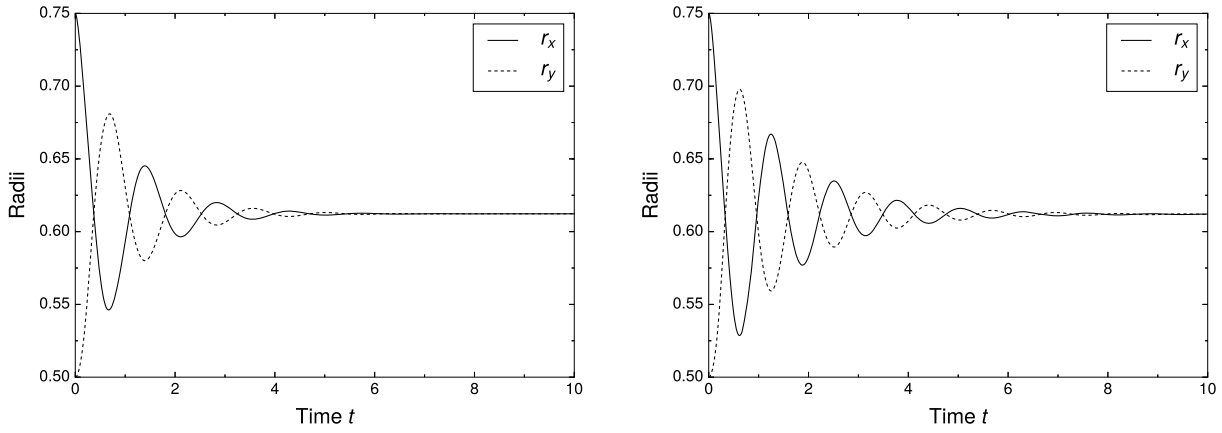

Fig. 20. Radii evolution with $\mu^+ = 0.1$ and $\mu^- = 0.01$ on the left and $\mu^+ = 0.01$ and $\mu^- = 0.1$ on the right.

to the fluid inside the membrane. The same density $\rho^+ = \rho^- = 1$ is enforced and following the work of [49], the viscosity ratio takes the values of 0.1 and 10.

First, we consider two cases with large viscosities: $\mu^+ = 1$, $\mu^- = 0.1$ and $\mu^+ = 0.1$, $\mu^- = 1$. Fig. 19 shows the radii evolution throughout time in both cases. The membrane does not oscillate in the first case, it relaxes gradually to its final position. When the fluid with the higher viscosity is inside the membrane, the latter performs only one small oscillation. The vertical and horizontal radii tend toward the theoretical value r_e in both cases.

Then, the same viscosity ratios are maintained but the fluid viscosities are divided by 10. Fig. 20 shows the radii evolution for these two cases. The vertical and horizontal radii oscillate around the equilibrium value with frequencies of the same order of magnitude. Similarly to Fig. 19, the damping is more important when the fluid with the higher viscosity is outside the membrane. The two plots of Fig. 19 and 20 are really close to the results from [49].

An example of the jump on the velocity derivatives is shown in Fig. 21. The evolution of both components of the velocity at $y = 0.3$ is plotted at the same time $t = 1$ with $\mu^+ = 0.1$ and $\mu^- = 1$. We observe that in both cases, the velocity is continuous but its derivatives are discontinuous at the membrane location (represented by the vertical dashed lines). This validates the sharp methodology of the jump condition on the viscosity-scaled velocity gradient.

4.5. Stretched and pressurized membrane immersed in fluids with different densities

In this section, we consider that the fluids inside and outside the membrane have the same viscosity $\mu^+ = \mu^- = 0.1$ but different densities. First, we enforce densities of 1 and 10 on both sides of the membrane. The radii evolutions of these two cases are represented in Fig. 22. As observed in the left plot of Fig. 17, the relaxing time of the membrane increases with the density. The membrane oscillates around its circular shape, the radius of which corresponds to the theoretical value r_e at the end of the simulation. It takes longer to reach the equilibrium state when the heaviest fluid is inside the membrane. The frequency of the oscillations has the same order of magnitude in both cases.

Then, the fluids densities are multiplied by 10. Fig. 23 exhibits the radii evolutions throughout time. We observe that the radii evolution with $\rho^+ = 10$ and $\rho^- = 100$ is close to the radii evolution with $\rho = 100$ in the whole domain (*cf.* Fig. 17).

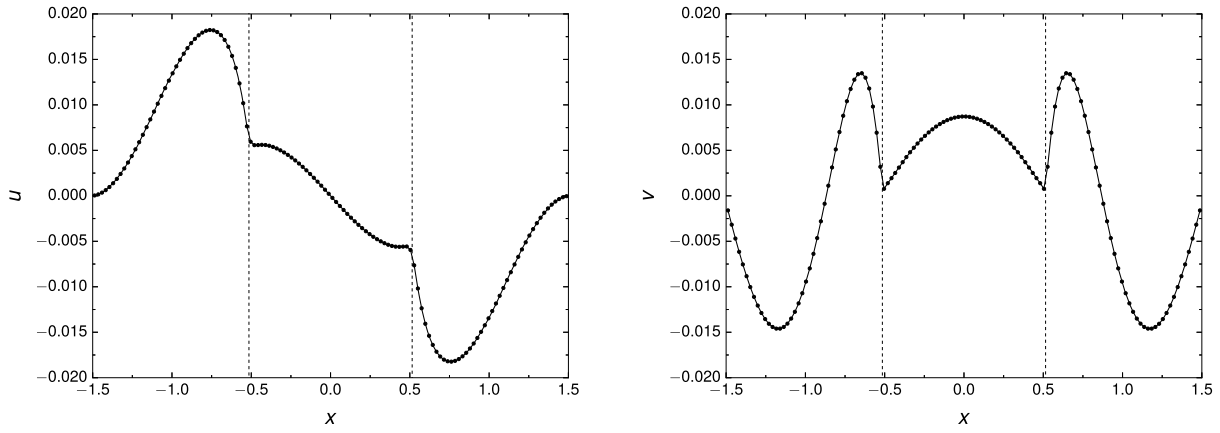

Fig. 21. Components of the velocity $\mathbf{u} = (u, v)$ at $t = 1$ and $y = 0.3$ with $\mu^+ = 0.1$ and $\mu^- = 1$.

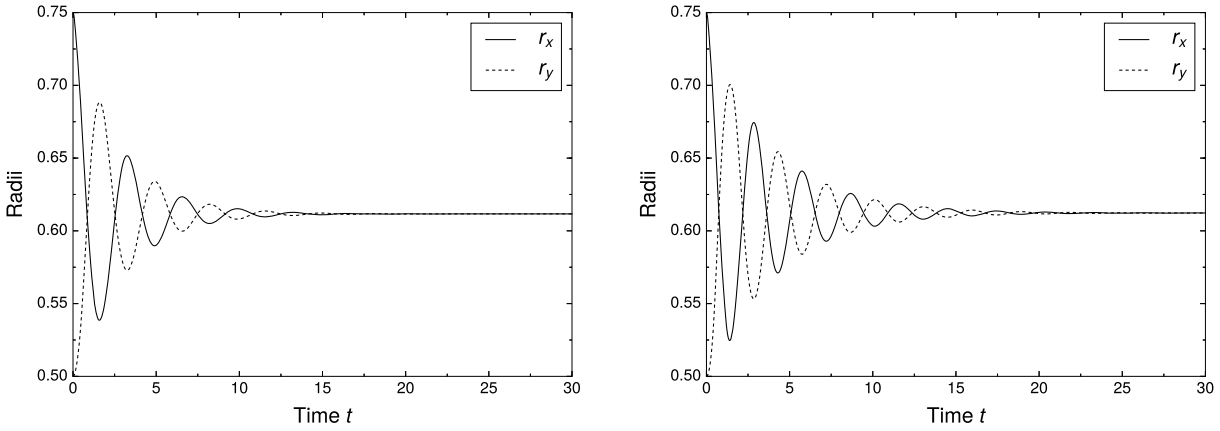

Fig. 22. Radii evolution with $\rho^+ = 10$ and $\rho^- = 1$ on the left and $\rho^+ = 1$ and $\rho^- = 10$ on the right.

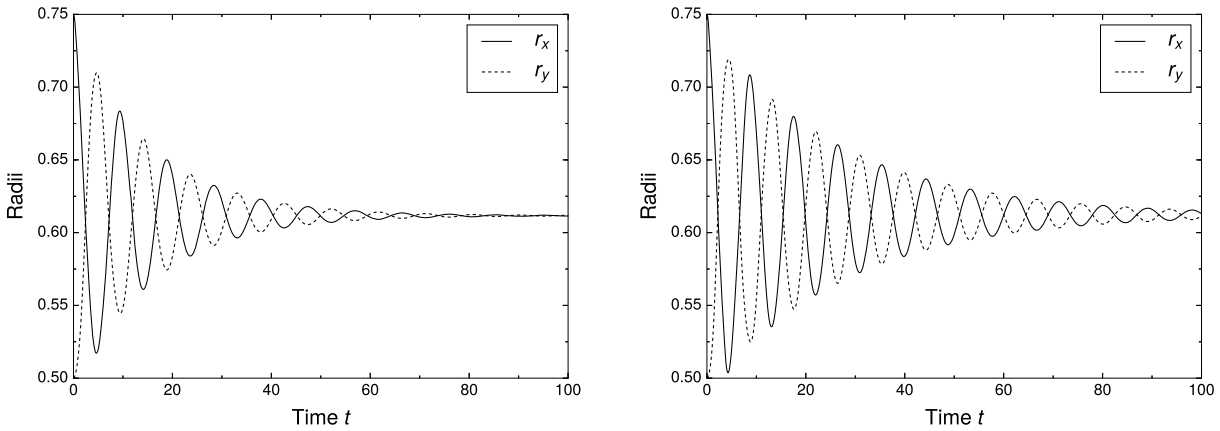

Fig. 23. Radii evolution with $\rho^+ = 100$ and $\rho^- = 10$ on the left and $\rho^+ = 10$ and $\rho^- = 100$ on the right.

Similarly, the damping rate of the oscillations is less efficient when the heaviest fluid is inside the membrane. The oscillation frequency has the same order of magnitude in these two cases.

Because the viscosity stays constant in the whole domain, the jumps on the velocity derivatives are directly linked to the tangent component of the elastic force. Fig. 24 shows the profiles of the two components of the velocity at $y = 0.3$ with $\rho^+ = 10$ and $\rho^- = 1$. The jump on the velocity derivatives is clearly visible at the membrane location.

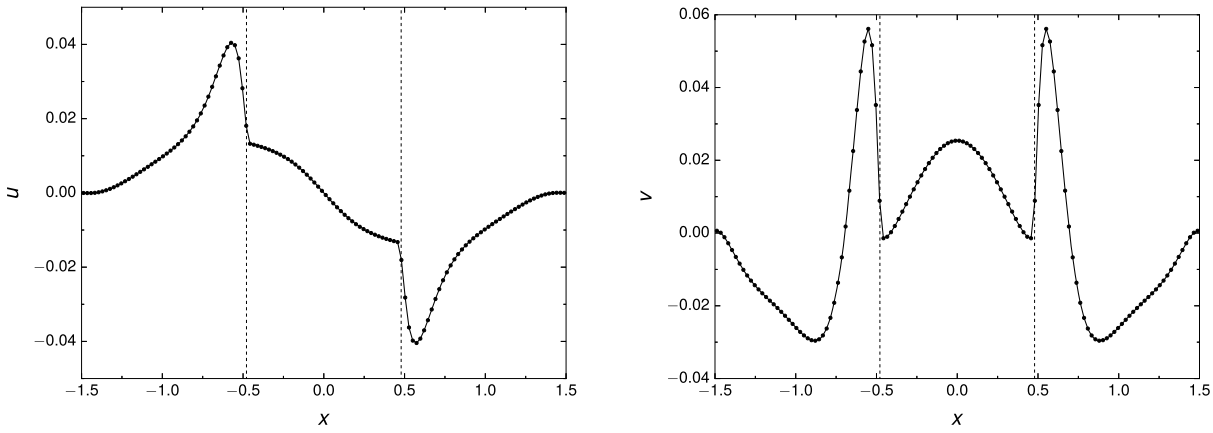

Fig. 24. Components of the velocity at $y = 0.3$ with $\rho^+ = 10$ and $\rho^- = 1$ at $t = 1.5$.

4.6. Immersed capsule in a shear flow

In this section, we consider a 3D spherical membrane immersed in a shear flow. The membrane deforms until the equilibrium between the viscous and the elastic forces is reached. This study case has been first introduced by Pozrikidis [56] and has been used to validate several immersed boundary methods [62,63]. This example aims at validating the use of hyperelastic laws with our model.

Initially, the membrane is a sphere of radius a. It is located in the centre of the computational domain, a cube with a side of $4a$. The shear flow enforced in the domain is characterised by the shear rate k such as $\mathbf{u} = (ky, 0, 0)$ at the top and bottom sides. On the other sides of the domain, periodic boundary conditions are imposed. We consider that the same fluid is inside and outside the membrane. Its physical properties are its density ρ and viscosity μ. The membrane is made of a hyperelastic material following the neo-Hookean model. The surface elastic modulus of the membrane is denoted E_s. The membrane is initially at rest, leading to $\bar{\bar{\boldsymbol{B}}}_{\boldsymbol{s}} = \bar{\bar{\boldsymbol{G}}}_{\boldsymbol{s}} = \bar{\bar{\boldsymbol{P}}}$ at $t = 0$. We define the dimensionless parameter G which compares the effect of the viscosity of the fluid over the elasticity of the membrane

$$G = \frac{\mu k a}{E_s}. \tag{112}$$

Five different values of G are considered: $G \in [0.0125, 0.025, 0.05, 0.1, 0.2]$. The deformation of the membrane is expressed through the Taylor deformation parameter

$$D_{xy} = \frac{L - l}{L + l}, \tag{113}$$

with L and l the maximum and minimum radii of the contour of the membrane with the (x, y)-plane computed with the non-linear least square method.

In this example, we use the "delta" formulation described in section 3.2.1. Sharp methods cannot be used in this example because of the compression zones on the membrane [64,65,63]. Indeed, where at least one of the two principal membrane stresses becomes negative, a buckling instability appears [65]. Because no bending stiffness is considered, a zone of the membrane which is not under tension may buckle and break down the computation. In this case, folds appear at the membrane surface which may overlap and lead to a stability issue. Fig. 25 shows an example of folds which appear in the equatorial area in our computation with $G = 0.0125$ after $kt = 0.8$. Sharp methods, based on the local equilibrium at the nodes crossed by the membrane, become unstable when compression zones appear [64]. The use of the smoothed method introduces a numerical dissipation which stabilises the computation and allows the solver to tolerate negative stress locally. Nevertheless, in specific cases such as $G = 0.0125$, stability issues appear in the long run whatever the numerical methods used.

Fig. 26 shows the evolution of the deformation parameter D_{xy} throughout time for 5 different values of G and three different meshes. We observe that the membrane deformation increases with G. At early flow time, the membrane loses its spherical shape and deforms with different magnitudes until it reaches a steady shape. The computations have been done on a maximum time of $kt = 2$ to be able to compare with the results from the literature. For the two lowest G, the buckling instability described below appears before $kt = 2$ and prevents us to compute the deformation parameter in the whole time period.

With $G = 0.0125$ and $G = 0.025$, the membrane deformation stays low and can be approached by the linear elasticity theory for which theoretical values of the membrane deformation have been computed in [18]. The evolution of the Taylor parameter with the linear theory is depicted by triangle markers in Fig. 26. The Taylor deformation parameter obtained

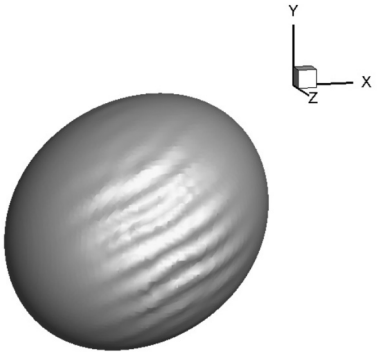

Fig. 25. Folds due to the buckling instability at the membrane surface with $G = 0.0125$ at $kt > 0.8$.

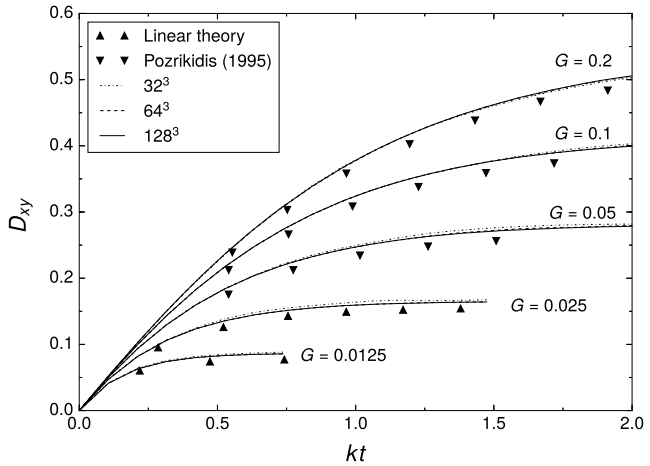

Fig. 26. Temporal evolution of the Taylor deformation parameter for the five values of G and three different meshes: 32^3, 64^3 and 128^3.

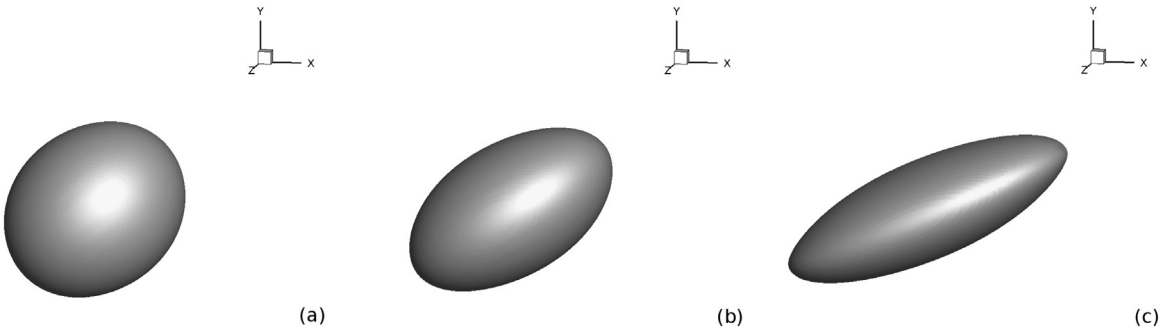

Fig. 27. Steady state or final state of the membrane for a) $G = 0.0125$, b) $G = 0.05$ and c) $G = 0.2$.

at $kt = 0.8$ with $G = 0.0125$ is $D_{xy} = 0.086$ and the theoretical value is about $D_{xy} = 0.078$. With $G = 0.025$, the steady deformation parameter computed is $D_{xy} = 0.164$ and the theoretical value from [18] is $D_{xy} = 0.156$.

Considering higher values of G, the linear theory cannot be used because the non-linearity of the material cannot be neglected. Our numerical results are compared to simulations done with the Immersed Boundary method and the Boundary Element method [62,56,63]. The Taylor parameter obtained by Pozrikidis [56] for the highest values of G are plotted in Fig. 26 with filled triangles. Pictures of the steady state of the membrane for $G \in [0.0125, 0.05, 0.2]$ are depicted in Fig. 27 and show different deformations computed with our numerical methods.

We observe in Fig. 26 the good convergence of the proposed numerical methods since the solution depends weakly on the grid size. In particular, the Taylor deformation parameter is the same for both finest grids for all values of G. Our numerical results are in close agreement with the reference results presented in Fig. 26. However, we report a slight overestimation on the Taylor deformation parameter whatever the considered case.

5. Conclusion

In this paper, we propose a full Eulerian method to deal with fluids-membrane interaction inspired by the work of Ito et al. in [15]. The main advantage of the Eulerian approach is to replace the use of Lagrangian markers to follow the membrane motion and deformations by scalar fields transported in the whole domain. Significant improvements are proposed in this paper to extend the method firstly introduced in [15]. A sharp methodology has been implemented to consider the elastic forces as jump conditions in the fluid equations. Specific developments have been done to be able to predict the behaviour of a membrane separating different fluids in the Eulerian framework. Because of important stability issues, an extension algorithm has been developed to remove the parasitic normal derivative of the scalar fields specific to the membrane. These achievements have been tested on benchmarks from the literature and ensure the suitable prediction of the proposed numerical methods. The jump conditions on the pressure and the velocity derivatives are well predicted, even in cases where the viscosity or the density are piecewise constant across the membrane.

Declaration of competing interest

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

Acknowledgements

The authors thank Airbus Defence & Space and the CNES (the French national space agency) for the funding of the PhD thesis of Alexis Dalmon. The authors gratefully acknowledge the CNRS (Centre National pour la Recherche Scientifique) for funding compensation of teaching hours to S. Tanguy during the year 2017-2018. This work was granted access to the HPC resources of CALMIP supercomputing center under the allocation P17035.

Appendix A. Stretched and pressurized membrane without the extension algorithm

The objective of this appendix is to show through an example that the extension algorithm is necessary. We consider the benchmark of section 4.3, the stretched and pressurized membrane immersed in a fluid with $\rho = 1$ and $\mu = 0.1$ in the whole computational domain. The computation breaks down after a few hundreds of temporal iterations without the extension algorithm, whatever the mesh. The comparison is done at early flow time, at $t = 0.1$.

Fig. A.28 exhibits the scalar field I_1 obtained with the 128^2 mesh without the extension algorithm on the left and with it on the right at $t = 0.1$. The first invariant $I_1 = \lambda_\tau^2$ is the trace of the surface Cauchy-Green tensor, it represents the square of the principal strain of the membrane and has no physical meaning in the fluid regions.

We observe in the left plot that the first scalar invariant of the strain tensor exhibits important variations at the membrane. Where the membrane crosses the axis $y = 0$, the first invariant increases sharply between the outside and inside fluid regions. Moreover, strong tangential variations of the scalar invariant are visible at the membrane. In the right plot, the scalar field does not present variations in the normal direction and the tangential variation along the membrane is smooth and does not present singularities. Without the extension algorithm, nothing prevents the appearance of jumps in the solid variables.

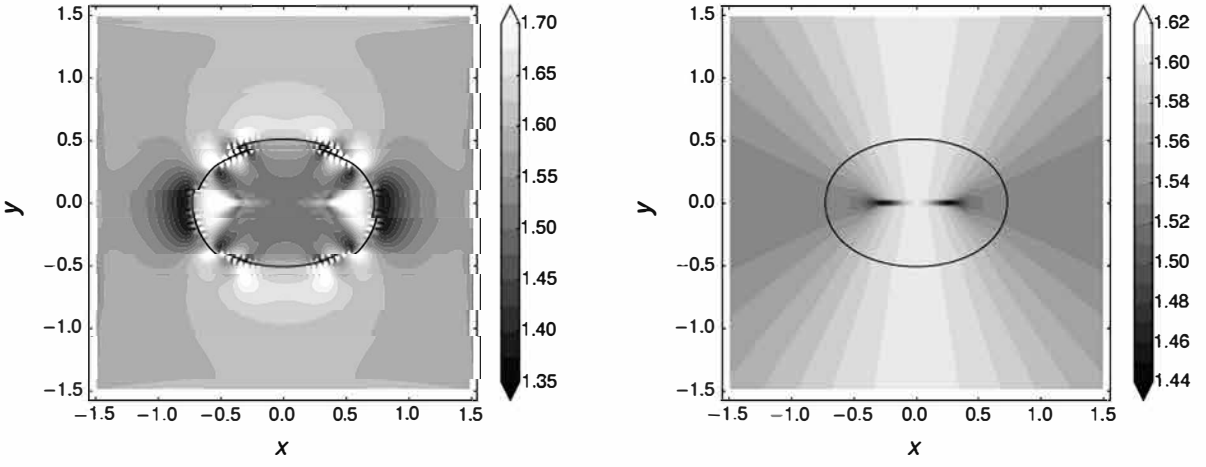

Fig. A.28. Contour fields of I_1 without the extension algorithm on the left and with it on the right, with $\rho = 1$ and $\mu = 0.1$ at $t = 0.1$.

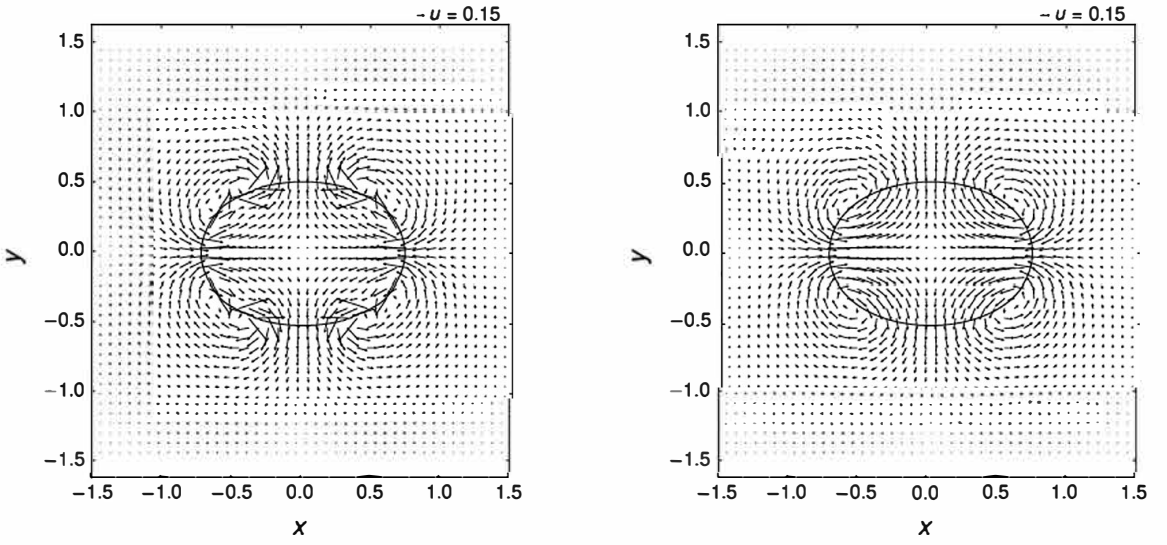

Fig. A.29. Velocity fields without the extension algorithm on the left and with it on the right, with $\rho = 1$ and $\mu = 0.1$ at $t = 0.1$.

Fig. A.29 depicts the velocity fields for the same configurations as in Fig. A.28. We observe that the velocity field is disturbed at the membrane without the extension algorithm. Far from the zero level set, the two plots of Fig. A.29 are quite similar but some peaks of velocity appear in the left plot near the membrane.

This instability phenomenon appears for all the benchmarks tested in section 4 without the extension algorithm. To conclude, the extension algorithm stabilises the computation by cancelling the normal derivatives of the solid variables which enables improving the computations stability.

Appendix B. About primary and secondary jump conditions

We present in this section some developments to show how secondary jump conditions can be taken into account by imposing primary jump conditions. Indeed, it is well-known that primary jump conditions, especially the one on viscosity, will involve several secondary jump conditions on the pressure field or on the pressure gradient for instance. However, if one solves the Navier-Stokes equations for two-phase flows, these secondary jump conditions can be accounted for when imposing the primary jump conditions. For the sake of simplicity we will consider here a Stokes flow without elastic constraints and an explicit solver for viscous terms, but such a development can be generalized to more complex configurations. From [46], one can write the Poisson equation for incompressible flows as

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} + \nabla \cdot \left(\frac{\gamma \kappa \mathbf{n} \delta}{\rho^{n+1}} \right), \tag{B.1}$$

where $\mathbf{u}^*$ is expressed as

$$\mathbf{u}^* = \left(\mathbf{u}^n - \Delta t \left(\mathbf{u}^n \cdot \nabla \right) \mathbf{u}^n \right) + \Delta t \frac{\nabla \cdot \left(2\mu \bar{\bar{D}} \right)^n}{\rho^{n+1}}. \tag{B.2}$$

Remarking that the divergence of the viscous stress tensor can be split in two parts, as previously stated in [46],

$$\nabla \cdot \left(2\mu \bar{\bar{D}} \right) = \mu \Delta \mathbf{u} + 2[\mu] \bar{\bar{D}} \cdot \mathbf{n} \delta, \tag{B.3}$$

by injecting (B.2) and (B.3) in (B.1), one can write the Poisson equation as

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \nabla \cdot \left(\frac{\mathbf{u}^n}{\Delta t} - \mathbf{u}^n \cdot \nabla \mathbf{u}^n + \frac{\mu \Delta \mathbf{u}^n}{\rho^{n+1}} + \frac{2[\mu] \bar{\bar{D}} \cdot \mathbf{n} \delta}{\rho^{n+1}} + \frac{\gamma \kappa \mathbf{n} \delta}{\rho^{n+1}} \right) = f, \tag{B.4}$$

where the right hand side f can be decomposed in a continuous part f_c and a discontinuous part f_d with

$$f_c = \nabla \cdot \left(\frac{\mathbf{u}^*}{\Delta t} - \mathbf{u}^n \cdot \nabla \mathbf{u}^n + \left(\frac{\mu \Delta \mathbf{u}^n}{\rho^{n+1}} \right)^{\pm} \right) \tag{B.5}$$

$$f_d = \nabla \cdot \left(\frac{\gamma \kappa \mathbf{n} + 2[\mu] \bar{\bar{\mathbf{D}}} \cdot \mathbf{n}}{\rho^{n+1}} \delta \right) + \left[\frac{\mu \Delta \mathbf{u}^n}{\rho^{n+1}} \right] \delta \tag{B.6}$$

where the following term has been decomposed in a continuous part and a discontinuous part

$$\nabla \cdot \left(\frac{\mu \Delta \mathbf{u}^n}{\rho^{n+1}} \right) = \nabla \cdot \left(\frac{\mu \Delta \mathbf{u}^n}{\rho^{n+1}} \right)^{\pm} + \left[\frac{\mu \Delta \mathbf{u}^n}{\rho^{n+1}} \right] \delta \tag{B.7}$$

Finally, following previous developments presented in [46], the equation (B.6) enables showing that equation (B.1) for the pressure field contains both a numerical approximation for imposing the correct jump condition on the pressure,

$$[p] = \gamma \kappa + 2[\mu] \mathbf{n} \cdot \bar{\bar{\mathbf{D}}} \cdot \mathbf{n} \tag{B.8}$$

and also for the jump condition on the pressure normal gradient such as

$$\left[\frac{\mathbf{n} \cdot \nabla p^{n+1}}{\rho^{n+1}} \right] = \left[\frac{\mu \mathbf{n} \cdot \Delta \mathbf{u}^n}{\rho^{n+1}} \right] \tag{B.9}$$

As a result of these developments, one can understand that secondary jump conditions due to the viscosity jump are involved by the resolution presented here. In particular, the jump condition on the normal derivative of the normal component velocity is imposed by keeping the viscosity inside the divergence operator in (B.1) as previously stated in [46]. As a consequence, this term has not to be further imposed in (43), since it is already contained in the divergence of $\mathbf{u}^*$. These elements are supported by theoretical and experimental benchmarks [66,46] for which space convergence toward the correct solution has been showed in several configurations involving a significant effect of the viscosity jump.

Appendix C. Capillary rising of a drop

The performed simulations are based on the theoretical study of [60] and the test-case in [57]. In the latter, we consider a drop of radius R in micro-gravity conditions ($\mathbf{g} = \mathbf{0}$) immersed in another fluid where a linear profile of surface tension γ is imposed along z direction:

$$\gamma(z) = \gamma_0 + \gamma' z \tag{C.1}$$

The constant surface tension gradient induces a movement of the drop at a steady-state velocity u_∞. Both fluids are supposed to be incompressible, and their thermo-physics properties are chosen such that Capillary, Marangoni and Reynolds numbers are small enough to consider a Stokes flow around the droplet which remains spherical. Thus, the flow remains axisymmetrical, and the conservation equation can be written in the moving referential as:

$$\mu \nabla^2 \mathbf{u} = \nabla p \,, \tag{C.2}$$

$$\nabla \cdot \mathbf{u} = \mathbf{0} \,, \tag{C.3}$$

with the asymptotic boundary conditions $\mathbf{u} \to u_\infty \mathbf{e_z}$ and $p \to 0$ as $r \to \infty$. The solution of those equations has been derived by Hadamard in [58], in the cylindrical coordinates:

$$u_r^- = \left(\frac{a^-}{\mu^-} \left(\frac{1}{r} - \frac{R^2}{r^3} \right) + u_\infty \left(1 - \frac{R^3}{r^3} \right) \right) \cos(\theta) \,, \tag{C.4}$$

$$u_\theta^- = -\left(\frac{a^-}{\mu^-} \left(\frac{1}{r} + \frac{R^2}{r^3} \right) + u_\infty \left(1 + \frac{1}{2} \frac{R^3}{r^3} \right) \right) \sin(\theta) \,, \tag{C.5}$$

$$p^- = \frac{a^-}{r^2} \cos(\theta) \,, \tag{C.6}$$

$$u_r^+ = -\left(\frac{a^+}{10\mu^+} \right) \left(r^2 - R^2 \right) \cos(\theta) \,, \tag{C.7}$$

$$u_\theta^+ = -\left(\frac{a^+}{10\mu^+} \right) \left(2r^2 - R^2 \right) \sin(\theta) \,, \tag{C.8}$$

$$p^+ = a^+ r \cos(\theta) + a_0^+ \,, \tag{C.9}$$

where u_r^- and u_r^+ are respectively the radial velocities outside and inside the droplet, u_θ^- and u_θ^+ the tangential velocities, p^- and p^+ the pressures, μ^- and μ^+ the dynamic viscosities, a^-, a^+ and a_0^+ three integration constants. These ones are calculated with the boundary conditions at the interface:

$$u_\theta^-(r = R, \theta) = u_\theta^+(r = R, \theta) \,, \tag{C.10}$$

$$\left[\mu\left(\frac{\partial u_\theta}{\partial r}-\frac{u_\theta}{r}\right)\right]_\Gamma=\frac{1}{R}\frac{\partial\gamma}{\partial\theta}\,, \tag{C.11}$$

$$\left[p-2\mu^-\frac{\partial u_r}{\partial r}\right]_\Gamma=\frac{2\gamma}{R}\,. \tag{C.12}$$

The no-slip condition gives the first equation. The second one takes into account the Marangoni effect through the discontinuity of the tangential viscous stress across the interface. The latter gives the pressure jump due to both viscosity jump and capillary pressure.

By replacing the theoretical expressions in eq. (C.12) and eq. (C.11), and combining the equations, we deduce the integration constants:

$$a^-=0\,, \tag{C.13}$$

$$a_0^+=\frac{2\gamma_0}{R}\,. \tag{C.14}$$

Then, eq. (C.10) gives:

$$a^+=\frac{15\mu' u_\infty}{R^2}\,. \tag{C.15}$$

The droplet velocity u_∞ is obtained by replacing the integration constants by their expression in eq. (C.11):

$$u_\infty=\frac{2}{3}\frac{\gamma' R}{(3\mu^+ + 2\mu^-)}\,. \tag{C.16}$$

This expression is actually the same as in [57] with $\gamma'=-\beta\gamma_0/l_z$. Finally, we obtain the following analytical expressions:

$$u_r^-=u_\infty(1-\frac{R^3}{r^3})\cos(\theta) \tag{C.17}$$

$$u_\theta^-=-u_\infty(1+\frac{1}{2}\frac{R^3}{r^3})\sin(\theta) \tag{C.18}$$

$$p^-=0 \tag{C.19}$$

$$u_r^+=\frac{3u_\infty}{2R^2}(r^2-R^2)\cos(\theta) \tag{C.20}$$

$$u_\theta^+=-\frac{3u_\infty}{2R^2}(2r^2-R^2)\sin(\theta) \tag{C.21}$$

$$p^+=\frac{15\mu^+ u_\infty}{R^2}r\cos(\theta)+\frac{2\gamma_0}{R} \tag{C.22}$$

References

[1] A. Dalmon, M. Lepilliez, S. Tanguy, A. Pedrono, B. Busset, H. Bavestrello, J. Mignot, Direct numerical simulation of a bubble motion in a spherical tank under external forces and microgravity conditions, J. Fluid Mech. 849 (2018) 467–497.
[2] M. Lepilliez, E.R. Popescu, F. Gibou, S. Tanguy, On two-phase flow solvers in irregular domains with contact line, J. Comput. Phys. 321 (2016) 1217–1251.
[3] A. Dalmon, M. Lepilliez, S. Tanguy, R. Alis, E. Popescu, R. Roumiguié, T. Miquel, B. Busset, H. Bavestrello, J. Mignot, Comparison between the fluidics experiment and direct numerical simulations of fluid sloshing in spherical tanks under microgravity conditions, Microgravity Sci. Technol. 31 (1) (2019) 123–128.
[4] J. Mignot, R. Pierre, M. Berhanu, B. Busset, R. Roumiguié, H. Bavestrello, S. Bonfanti, T. Miquel, L.O. Marot, A. Llodra-Perez, Fluid dynamic in space experiment, in: 68th International Astronautical Congress (IAC), Adelaide, Australia, 2017.
[5] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.
[6] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[7] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[8] R.J. Leveque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficient and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.
[9] R.J. Leveque, Z. Li, Immersed interface methods for stokes flow with elastic boundaries or surface tension, J. Sci. Comput. 18 (1997) 709–735.
[10] Z. Li, M.C. Lai, The immersed interface method for the Navier-Stokes equations with singular forces, J. Comput. Phys. 171 (2001) 822–842.
[11] P.G. Hu, L. Xue, S. Mao, R. Kamakoti, H. Zhao, N. Dittakavi, Z. Wang, Q. Li, K. Ni, Material point method applied to fluid-structure interaction (fsi)/aeroelasticity problems, in: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010, p. 1464.
[12] A.R. York II, D. Sulsky, H.L. Schreyer, Fluid-membrane interaction based on the material point method, Int. J. Numer. Methods Eng. 48 (2000) 901–924.
[13] G.H. Cottet, E. Maitre, A level set method for fluid-structure interaction with immersed surfaces, Math. Models Methods Appl. Sci. 16 (2006) 415–438.
[14] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[15] S. Ii, X. Gong, K. Sugiyama, J. Wu, H. Huang, S. Takagi, A full Eulerian fluid-membrane coupling method with a smoothed volume-of-fluid approach, Commun. Comput. Phys. 12 (2012) 544–576.
[16] K. Sugiyama, S. Ii, S. Takeuchi, S. Takagi, Y. Matsumoto, A full Eulerian finite difference approach for solving fluid-structure coupling problems, J. Comput. Phys. 230 (2011) 596–627.

[17] C.W. Hirt, B.D. Nichols, Volume of fluid (vof) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[18] D. Barthes-Biesel, J.M. Rallison, The time-dependent deformation of a capsule freely suspended in a linear shear flow, J. Fluid Mech. 113 (1981) 251–267.
[19] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[20] X.-D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domain, J. Comput. Phys. 160 (2000) 151–178.
[21] A.N. Gent, A.G. Thomas, Forms for the stored (strain) energy function for vulcanized rubber, J. Polym. Sci. 28 (1958) 625–628.
[22] M. Mooney, A theory of large elastic deformation, J. Appl. Phys. 11 (1940) 582–592.
[23] R.W. Ogden, Large deformation isotropic elasticity - on the correlation of theory and experiment for incompressible rubberlike solids, Proc. R. Soc. Lond. 326 (1972) 565–584.
[24] Yeoh, Characterization of elastic properties of carbon-black-filled rubber vulcanizates, Rubber Chem. Technol. 63 (1990) 792–805.
[25] L.R.G. Treloar, The elasticity of a network of long chain molecules, Trans. Faraday Soc. 39 (1943) 36–41.
[26] G. Huber, S. Tanguy, M. Sagan, C. Colin, Direct numerical simulation of nucleate pool boiling at large microscopic contact angle and moderate Jakob number, Int. J. Heat Mass Transf. (2017).
[27] L. Rueda Villegas, R. Alis, M. Lepilliez, S. Tanguy, A ghost fluid/level set method for boiling flows and liquid evaporation: application to the Leidenfrost effect, J. Comput. Phys. 316 (2016) 789–813.
[28] S. Tanguy, T. Menard, A. Berlemont, A level set method for vaporizing two-phase flows, J. Comput. Phys. 221 (2007) 837–853.
[29] S. Tanguy, M. Sagan, B. Lalanne, F. Couderc, C. Colin, Benchmarks and numerical methods for the simulation of boiling flows, J. Comput. Phys. 264 (2014) 1–22.
[30] A. Urbano, S. Tanguy, G. Huber, C. Colin, Direct numerical simulation of nucleate boiling in micro-layer regime, Int. J. Heat Mass Transf. 123 (2018).
[31] G. Huber, S. Tanguy, J. Bera, B. Gilles, A time splitting projection scheme for compressible two-phase flows. Application to the interaction of bubbles with ultrasound waves, J. Comput. Phys. 302 (2015) 439–468.
[32] Y. Ng, C. Min, F. Gibou, An efficient fluid-solid coupling algorithm for single-phase flows, J. Comput. Phys. 228 (2009) 8807–8829.
[33] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[34] F. Gibou, L.-T. Cheng, D. Nguyen, S. Banerjee, A level set based sharp interface method for the multiphase incompressible Navier-Stokes equations with phase change, J. Comput. Phys. 222 (2007) 536–555.
[35] F. Gibou, R. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (2002) 205–227.
[36] M. Kang, R. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.
[37] M. Sussman, K. Smith, M. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, J. Comput. Phys. 221 (2007) 469–505.
[38] M. Cisternino, L. Weynans, A parallel second order Cartesian method for elliptic interface problems, Commun. Comput. Phys. 12 (5) (2012) 1562–1587.
[39] A. Guittet, M. Lepilliez, S. Tanguy, F. Gibou, Solving elliptic problems with discontinuities on irregular domains - the Voronoi interface method, J. Comput. Phys. 298 (2015) 747–765.
[40] S. Hou, X.-D. Liu, A numerical method for solving variable coefficient elliptic equation with interfaces, J. Comput. Phys. 202 (2005) 411–445.
[41] R. Borges, M. Carmona, B. Costa, W. Don, An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws, J. Comput. Phys. 227 (2008) 3191–3211.
[42] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[43] A. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.
[44] J. Dendy, Black box multigrid, J. Comput. Phys. 48 (1982) 366–386.
[45] S. MacLachlan, J. Tang, C. Vuik, Fast and robust solvers for pressure-correction in bubbly flow problems, J. Comput. Phys. 227 (2008) 9742–9761.
[46] B. Lalanne, L.R. Villegas, S. Tanguy, F. Risso, On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method, J. Comput. Phys. 301 (2015) 289–307.
[47] Z. Tan, D.V. Le, K.M. Lim, B.C. Khoo, An immersed interface method for the incompressible Navier-Stokes equations with discontinuous viscosity across the interface, J. Sci. Comput. 31 (2009) 1798–1819.
[48] S. Xu, Z.J. Wang, Systematic derivation of jump conditions for the immersed interface method in three-dimensional flow simulation, J. Sci. Comput. 27 (2006) 1948–1980.
[49] Z. Tan, D.V. Le, Z. Li, K.M. Lim, B.C. Khoo, An immersed interface method for solving incompressible viscous flows with piecewise constant viscosity across a moving elastic membrane, J. Comput. Phys. 227 (2008) 9955–9983.
[50] C.C. de Langavant, A. Guittet, M. Theillard, F. Temprano-Coleto, F. Gibou, Level-set simulations of soluble surfactant driven flows, J. Comput. Phys. 348 (2017) 271–297.
[51] C. Min, On reinitializing level set functions, J. Comput. Phys. 229 (2010) 2764–2772.
[52] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive Cartesian grids, J. Comput. Phys. 225 (2007) 300–321.
[53] G. Russo, P. Smereka, A remark on computing distance functions, J. Comput. Phys. 163 (2000) 51–67.
[54] T. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (2004) 349–355.
[55] C. Tu, C.S. Peskin, Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods, SIAM J. Sci. Stat. Comput. 13 (1992) 1361–1376.
[56] C. Pozrikidis, Finite deformation of liquid capsules enclosed by elastic membranes in simple shear flow, J. Fluid Mech. 297 (1995) 123–152.
[57] M. Muradoglu, G. Tryggvason, A front-tracking method for computation of interfacial flows with soluble surfactants, J. Comput. Phys. 227 (2008).
[58] J. Hadamard, Motion of liquid drops (viscous), C. R. Acad. Sci. 152 (1911).
[59] W. Rybczynski, Uber die fortschreitende Bewegung einer flussigen Kugel in einem zahen Medium, Bull. Acad. Sci. Cracov. 40 (1911).
[60] N.O. Young, J.S. Goldstein, M.J. Block, The motion of bubbles in a vertical temperature gradient, J. Fluid Mech. 6 (1959) 330–356.
[61] L. Lee, R.J. Leveque, An immersed interface method for incompressible Navier-Stokes equations, J. Sci. Comput. 25 (2003) 832–856.
[62] C.D. Eggleton, A.S. Popel, Large deformation of red blood cell ghost in a simple shear flow, Phys. Fluids 10 (1998) 1834–1845.
[63] S. Ramanujan, C. Pozrikidis, Deformation of liquid capsule enclosed by elastic membranes in simple shear flow: large deformation and the effect of fluid viscosities, J. Fluid Mech. 361 (1998) 117–143.
[64] D. Barthes-Biesel, Capsule motion in flow: deformation and membrane buckling, C. R. Phys. 10 (2009) 764–774.
[65] L. Lac, D. Barthes-Biesel, N.A. Pelekasis, J. Tsamopoulos, Spherical capsules in three-dimensional unbounded stokes flows: effect of the membrane constitutive law and onset of buckling, J. Fluid Mech. 516 (2004) 303–334.
[66] B. Lalanne, N.A. Chebel, J. Vejrazka, S. Tanguy, O. Masbernat, F. Risso, Non-linear shape oscillations of rising drops and bubbles: experiments and simulations, Phys. Fluids 27 (2015).