An ALE-FE method for two-phase flows with dynamic boundaries

G.R. Anjos[1,*], N. Mangiavacchi[b], J.R. Thome[c]

[a] *COPPE/Department of Mechanical Engineering, Federal University of Rio de Janeiro, Brazil*
[b] *GESAR/Mechanical Engineering Department, State Univerisity of Rio de Janeiro, Graduate Program in Mechanical Engineering, Brazil*
[c] *Institute for Multiscale Thermofluids, School of Engineering, The University of Edinburgh, Scotland*

Abstract

The present work aims at developing a new flexible computational framework to simulate macro and microscale two-phase flows with dynamic boundaries. Such a technique is extremely useful for periodic and very large domains which requires exhaustive computational resources, consequently reducing the required numerical domain. In this article an interface tracking Finite Element (FE) method is used to solve the equations governing the motion of two immiscible incompressible fluids in the Arbitrary Lagrangian-Eulerian framework (ALE). The equations are written in axisymmetric coordinates, however the proposed moving boundary technique can be easily extended to 3-dimensional flows and other methods using the ALE framework such as the finite volume method. The two-phase interface separating the fluids is a subset of the domain mesh, therefore a layer of zero thickness is achieved assuring sharp transition of properties among phases. At the scale of interest, surface tension plays an important role and is thus considered in the flow equations. Several validations and results are presented for gravity dominated problem, including the sessile drop test and rising of spherical and Taylor bubbles, as well as the divergent

[*] Corresponding author
Email address: gustavo.rabello@mecanica.coppe.ufrj.br (G.R. Anjos)

Preprint submitted to Computer Methods in Applied Mechanics and EngineeringDecember 6, 2023

and sinusoidal channels, showing accuracy for modeling two-phase flows in large periodic domains.

Keywords: Arbitrary Lagrangian-Eulerian, Finite Element method,
Two-phase flows, Surface tension, Rising bubble, Corrugated channels

1. Introduction

Two-phase flow occurs frequently in nature and in industrial applications. This is for instance the case in the refrigeration industry, blood flow in biological systems or lately in the cooling of computer chips and power electronics. Two-phase cooling systems are usually superior to single-phase cooling systems, due to the fact that for most materials the latent heat of vaporization is one or two orders of magnitude higher than the thermal capacity of the liquid. The development of design tools for two-phase microchannels is becoming an important area of research. However, experiments are difficult to perform to investigate the dynamic nature of these flows because of the very small scales [1] and numerical simulations offer an alternative way to study such flows usually at a much lower cost. Therefore, the development of numerical methods able to simulate such flows in details is an important research tool.

The Finite Element Method (FEM) opens new ways in both interface tracking as well as interface capturing. Space time methods [2, 3] use finite elements to discretize both the spatial and time domain. They thereby allow for another kind of interface tracking where the deformation of the computational mesh is automatically included in the problem formulation. Meanwhile the eXtended Finite Element Method (XFEM) is ushering in a renaissance in Eulerian (fixed grid) two-phase flow. The XFEM is based on the idea of adding extra degrees of freedom to the fixed mesh problem, which are interpolated by discontinuous functions inside elements traversed by the interface. However, the XFEM approach suffers from ill conditioning of the associated discrete linear operator, which makes the solution of the linear system a more complicated task [4].

In two-phase flows, two main categories of modeling are available, namely ‘one-fluid’ and ‘two-fluids’ formulations. The former uses one set of equations to describe all the phases and it is assisted by a color function, which defines their regions in the domain. The latter describes each phase as a separate fluid with its own set of governing equations, thus requiring an additional mathematical formulation for their coupling. Within the ‘one-fluid’ formulation, the interface description may be divided in two sub-areas, namely Eulerian and Lagrangian descriptions. The basic difference between both methodologies is reflected on the modeling of the interface between the phases. For instance, the Volume of Fluid [5] and the Level-Set [6] methods are Eulerian methods. Improved coupling methods have also been used to simulate microscale two-phase flows [7]. On the other hand, the Lagrangian formulation describes the interface between fluids explicitly by computational elements. Such a description allows a sharp representation of the interface, but its drawback is the proper treatment of topological changes in the interface, since coalescence and break-ups are not inherent to its methodology.

Due to the shortcomings of purely Eulerian and purely Lagrangian formulations, the Arbitrary Lagrangian-Eulerian (ALE) description allows these two frameworks to be combined in one single formulation, thus the best benefits of each approach can be brought together, that is, the computational mesh nodes may move with the continuum in normal Lagrangian fashion, to be held fixed in Eulerian manner or somewhere in between the Eulerian and the Lagrangian description. The ALE description has shown to be suitable to describe fluid flow problems (see, for instance [8]) and this work extends its capability to two-phase gravity driven flows in axisymmetric coordinates with moving boundaries.

The interface tracking methods require constant remeshing to handle cases where the motion of the interface leads to strong mesh deformations. Remeshing comprises the handling of mesh connectivity and inserting/removing nodes in zones where the accuracy may change. Therefore, after the remeshing process, an interpolation is strictly required to project the solution onto the new available mesh. This interpolation leads to additional numerical errors, which are usually of diffusive nature. Some authors propose changing only the connectivity, e.g. by swapping edges or performing operations locally to avoid global remeshing [9]. However, these techniques lead to frequent insertion/deletion operations on vectors, which typically require doubly or single linked lists to perform better and may become less efficient than global remeshing. Moreover, [10] has continued the implementation of moving mesh technique to tackle bubble merging and breakup in multiphase flow simulations with an extra mesh adaptation criteria. Simulations of droplet pair collision is also shown. In [11] it is shown for some simple academic problems that remeshing at every time step can be more accurate than using an ALE scheme without remeshing. This is due to the fact that remeshing allows the mesh resolution to be increased by adding more points to the mesh. This can be used dynamically to refine the mesh in regions where refinement is needed and coarsen the mesh elsewhere. In [12], a high-order ALE schemes for incompressible flows are proposed for spatial discretization problems with moving boundaries. Several finite element were tested, showing accuracy for the proposed result of the sliding droplet. A vascular tumor growth has been modeled with a moving mesh approach in [13] using two different velocity-based methods applied to a two-phase flows, while in [14] and [15], the ALE method coupled to the finite element method has been applied to the study of insoluble surfactants, showing that in moving mesh context, complex physics can be successfully understood. Several moving mesh methods has been investigated in the extensive review article of [16] including nonlinear test cases, porous media and thin film equations and two-phase flow problems. In [17], five mesh-based tracking methods were compared for fluid flow applications, including free-surface flows.

In this work an accurate interface tracking method with moving boundaries is proposed in the context of the Arbitrary Lagrangian-Eulerian Finite Element method, which captures bubble/wall effects and handles cases with large ratios in physical parameters and interface deformations. By a sophisticated motion of the mesh points, this method reaches a detailed description of the interface at any time, while avoiding the strong mesh distortion of purely Lagrangian methods. Adaptive mesh refinement and remeshing are used to keep high quality mesh elements. An efficient way to compute the curvature and the surface tension term. This work is an extension of [18] where we propose a new moving mesh treatment to handle efficiently moving boundaries domains for two-phase flows in large and periodic domains. Numerical validations against exact solutions in axisymmetric geometries are presented as well as several results for rising of bubbles and drops with different fluid properties in challenging geometries.

This article is organized with an introduction of the moving boundary technique and the corresponding literature review, followed by the mathematical formulation using the equations in axisymmetric coordinates. Thus, the Finite Element Method is exposed as discretization technique to the mathematical equations followed by a section with all required mesh operations used in this work. Next, the result section shows several test cases to validate the proposed moving boundary methodology including rising in slowly divergent channel and a corrugated channel with experimental comparison. Finally, this text ends with conclusions.

2. Equations in Axisymmetric Formulation

The classical description of two-phase flow is based on the governing equations of continuum mechanics of flowing media or hydrodynamics. These are the incompressible Navier-Stokes equations including surface tension in axisymmetric formulation, which are presented below in non-dimensional form:

$$\rho(\mathbf{x})\frac{D\mathbf{v}}{Dt} = -\nabla p + \frac{1}{\mathrm{N}^{1/2}}\nabla\cdot\mu(\mathbf{x})\left[\nabla\mathbf{v}+\nabla\mathbf{v}^{T}\right] + \rho(\mathbf{x})\mathbf{g} + \frac{1}{\mathrm{Eo}}\mathbf{f_{st}} \qquad (1)$$

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

where $\rho(\mathbf{x})$ and $\mu(\mathbf{x})$ stand for the density and viscosity of the fluids in the numerical domain as a function of the position $\mathbf{x}$ occupied by the fluid, where $\mathbf{x}$ is the coordinate vector in the axisymmetric space defined by the axial x and the radial r coordinates. $\mathbf{v}$ is the velocity field with axial u and radial

v components, t represents time, p pressure, $\mathbf{g}$ is the gravity vector and $\mathbf{f_{st}}$ represents the surface tension force. The gradient ∇^T and the divergence $\nabla\cdot$ operators are defined in the axisymmetric formulation as:

$$\nabla^T = [\frac{\partial}{\partial r}; \frac{\partial}{\partial x}] \qquad \nabla\cdot = [\frac{1}{r}\frac{\partial}{\partial r}r; \frac{\partial}{\partial x}] \tag{3}$$

The non-dimensional numbers Archimedes (N) and Eötvös (Eo) are both defined as:

$$\mathrm{N} = \frac{\rho_0^2 g_0 D_0^3}{\mu_0^2} \qquad \mathrm{Eo} = \frac{\rho_0 g_0 D_0^2}{\sigma_0} \tag{4}$$

note that the above non-dimensional numbers are achieved by using the standard reference parameters when the velocity of the system is unknown, thus replacing the common non-dimensionalization $\mathbf{v}^* = \mathbf{v}/U_0$ by $\mathbf{v}^* = \mathbf{v}/\sqrt{g_0 D_0}$ and $\mathbf{x}^* = \mathbf{x}/L$ by $\mathbf{x}^* = \mathbf{x}/D_0$, where U_0, L, D_0 and g_0 are referential parameters of velocity, length, channel diameter and gravity ($g_0 = 9.81m/s^2$) respectively and $*$ stands for the non-dimensional quantities. The referential parameters are selected to be those for the liquid phase. Additionally, the non-dimensional Morton number (Mo) is defined due to its importance to characterize the shape of a bubble or (drop) and the relative effect of the viscous and surface tension forces, where σ_0 represents the referential surface tension:

$$\mathrm{Mo} = \frac{\mu_0^4 g_0}{\rho_0 \sigma_0^3} = \frac{\mathrm{Eo}^3}{\mathrm{N}^2} \tag{5}$$

The material derivative $D\mathbf{v}/Dt$ is represented in the Arbitrary Eulerian-Lagrangian framework as follows:

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}}{\partial t} + \left(\mathbf{v} - \hat{\mathbf{v}}\right) \cdot \nabla \mathbf{v} \tag{6}$$

where $\mathbf{v} - \hat{\mathbf{v}}$ is the difference between the fluid flow velocity $\mathbf{v}$ and the mesh velocity $\hat{\mathbf{v}}$. In the ALE context, the mesh velocity $\hat{\mathbf{v}}$ can be chosen arbitrarily and suitable to different flow situations. If mesh velocity $\hat{\mathbf{v}} = 0$, the *Eulerian* framework is achieved, while if $\hat{\mathbf{v}} = \mathbf{v}$, the *Lagrangian* framework is obtained.

An arbitrary framework is set whenever $\hat{\mathbf{v}} \neq \mathbf{v}$ or $\hat{\mathbf{v}} \neq 0$. Note that in the proposed formulation, the nodes at the interface between fluids are always moving with the fluid velocity in a *Lagrangian* way, since it is explicitly defined by a set of mesh nodes. Moreover, an Eulerian method for the advection term is required whenever the arbitrary framework is achieved. In this work, a Semi-Lagrangian method is therefore used to discretize such a term. In the following section the discretization of the interface between fluids is detailed as well as the modeling of the material derivative.

2.1. Surface Tension Force

According to [19], the non-dimensional surface tension force can be modeled as:

$$\mathbf{f_{st}} = \kappa \mathbf{n} \delta \tag{7}$$

where κ stands for curvature, $\mathbf{n}$ is the normal vector and δ the Dirac delta function which converts the surface tension force to a volume force. Note that in this present work we only consider constant surface tension, therefore the surface tension coefficient is embed at the Eötvös number Eo.

In 2-dimensional space, the mean curvature κ can be locally calculated from the variation of the normal or tangent vector along the curve that defines the interface. Such a definition is derived from the *Frenet*'s formulas ([20]) and it is written as:

$$\kappa_{2d}\mathbf{t} = -\frac{\partial \mathbf{n}}{\partial s} \qquad \text{or} \qquad \kappa_{2d}\mathbf{n} = \frac{\partial \mathbf{t}}{\partial s} \tag{8}$$

which represents the continuous description. In the axisymmetric case, a 3-dimensional curvature should be taken into account, which is the sum of the 2-dimensional curvature κ_{2d} represented by Eq. (8) in the $x - r$ plane and the curvature of the axisymmetric component κ_{axi} which is associated to the axis of revolution, and simply related to the angle ϕ between the interface normal vector and the distance to the symmetry axis R written as:

$$\kappa_{axi} = \frac{sin(\phi)}{R} \tag{9}$$

The total curvature of the axisymmetric form in the continuous description is then given as the sum of both curvatures:

$$\kappa = \kappa_{2d} + \kappa_{axi} = \kappa_{2d} + \frac{sin(\phi)}{R} \tag{10}$$

2.2. Boundary Conditions

Appropriated boundary conditions are required to find the particular solution of the partial differential equations presented above for the solution of the two-phase flow systems. Since geometry and the flow variables are assumed to be independent of the rotation angle θ, a symmetry boundary condition of

$$v = 0 \qquad \text{and} \qquad \frac{\partial u}{\partial r} = 0 \tag{11}$$

is required on the symmetry axis ($r = 0$). Moreover, the *no-slip* condition is used at the walls where u and v are set to zero velocity. In rising bubble test cases, the bubble is fixed in space, while the domain is moving with an Eulerian velocity, therefore the walls are also moving backward with the bubble's centroid velocity, thus a *slip* condition is required at non-symmetry walls. This is achieved by subtracting such a bubble's velocity to the walls. Eventually, pressure is explicitly prescribed by setting it to a referenced value $p = 0$ to facilitate the solution of the uncoupled system, which will be explained in the following sections. An additional boundary condition is required for the curvature calculation of the two nodes lying at the boundaries in the presented axisymmetric flow solver. In this work, we have computed curvature for those nodes considering the reflected neighbor nodes at the boundary, and thus computing curvature geometrically using the above method.

3. Finite Element Method

3.1. Weak Form of the Navier-Stokes Equations

Applying the Galerkin method to the Navier-Stokes equations in axisymmetric coordinates, one has to find velocity $\mathbf{v}$ with components $\mathbf{v} = (u, v)$ and pressure p both $\in \mathcal{U}_{u_D} \times \mathcal{V}_{v_D} \times \mathcal{Q}$ such that

$$m\left(\rho \frac{D\mathbf{v}}{Dt}, \mathbf{w}\right) + k\left(\frac{\mu}{\mathrm{N}^{1/2}}, \mathbf{v}, \mathbf{w}\right) + g\,(p, \mathbf{w}) = (\rho\,\mathbf{g}, \mathbf{w}) + (\frac{1}{\mathrm{Eo}}, \mathbf{f_{st}}, \mathbf{w}) \qquad (12)$$

$$d\,(q, \mathbf{v}) = 0 \qquad (13)$$

$$\text{with } \mathbf{v}(\mathbf{x}, t = t_0) = \mathbf{v}_0(\mathbf{x}) \qquad (14)$$

for all $(\mathbf{v}, q) \in \mathcal{U}_0 \times \mathcal{V}_0 \times \mathcal{Q}$ where $\mathbf{v}_0(\mathbf{x})$ is the initial condition. Additionally, m, k, g and d stand for the mass, viscous term, gradient and divergence matrices respectively. Now, consider the axisymmetric space $L^2(\Omega)$ for which

$$\int_\Omega w^2\, r\, d\mathbf{x} \;<\; \infty \qquad (15)$$

where $d\mathbf{x} = dx\, dr$. The function spaces $\mathcal{U}_{u_D}, \mathcal{V}_{v_D}$ and $\mathcal{Q}$ are defined as:

$$\mathcal{U}_{u_D} \;:=\; \{\, u \in H^1(\Omega) \mid u = u_D \quad \text{on} \quad \partial\Omega_D\} \qquad (16)$$

$$\mathcal{V}_{v_D} \;:=\; \{\, v \in H^1(\Omega) \mid v = v_D \quad \text{on} \quad \partial\Omega_D\} \qquad (17)$$

$$\mathcal{Q} \;:=\; \{\, q \in L^2(\Omega) \mid \int_\Omega q\, r\, d\mathbf{x} \;=\; 0\} \qquad (18)$$

The bilinear forms of the axisymmetric Navier-Stokes are then given by:

$$m\left(\rho\,\frac{D\mathbf{v}}{Dt},\mathbf{w}\right) = \int\limits_{\Omega} \rho\mathbf{w}\cdot\frac{D\mathbf{v}}{Dt}\; r\,d\mathbf{x} \tag{19}$$

$$k\left(\frac{\mu}{\mathrm{N}^{1/2}},\mathbf{v},\mathbf{w}\right) = 2\int\limits_{\Omega} \frac{\mu}{\mathrm{N}^{1/2}}\,\mathcal{D}(\mathbf{w}):\mathcal{D}(\mathbf{v})\,r\,d\mathbf{x} + 2\int\limits_{\Omega} \frac{\mu}{\mathrm{N}^{1/2}}\,\frac{w_2 v}{r}\,d\mathbf{x} \tag{20}$$

$$g\left(q,\mathbf{w}\right) = \int\limits_{\Omega} \nabla q\cdot\mathbf{w}\; r\,d\mathbf{x} \tag{21}$$

$$\left(\frac{1}{\mathrm{Eo}},\mathbf{f_{st}},\mathbf{w}\right) = \int\limits_{\Omega} \frac{1}{\mathrm{Eo}}\mathbf{f_{st}}\cdot\mathbf{w}\; r\,d\Omega \tag{22}$$

$$\left(\rho\mathbf{g},\mathbf{w}\right) = \int\limits_{\Omega} \rho\mathbf{g}\cdot\mathbf{w}\; r\,d\mathbf{x} \tag{23}$$

$$d\left(q,\mathbf{v}\right) = \int\limits_{\Omega} q\,\nabla\cdot\mathbf{v}\,r\,d\mathbf{x} + \int\limits_{\Omega} q\,v\,d\mathbf{x} \tag{24}$$

where $\mathcal{D}(\mathbf{v}) = \nabla\mathbf{v} + \nabla\mathbf{v}^T$. Integrating the viscous stress tensor $\overline{\overline{\tau}}_v = 2\,\mu\,\mathcal{D}(\mathbf{v})$ by parts allows for transfer of the derivatives to the test function $\mathbf{w}$. No boundary term appears because we consider only homogeneous Neumann boundary conditions for the viscous stress tensor. Therefore, the boundary $\partial\Omega$ of Ω is composed of two distinct regions $\partial\Omega_D$, $\partial\Omega_N$ on which Dirichlet and Neumann boundary conditions hold respectively:

$$\mathbf{v} = \mathbf{v}_D \quad \text{on} \quad \partial\Omega_D \qquad \text{and} \qquad \overline{\overline{\tau}}_v\mathbf{v}\cdot\mathbf{n} = 0 \quad \text{on} \quad \partial\Omega_N \tag{25}$$

Taking the pressure function space with zero average makes the solution unique. Otherwise the pressure p could not be determined uniquely, since only derivatives of the pressure appear in the equations and the addition of a constant to the pressure would still give a solution. What is usually done numerically instead of requiring the pressure to have zero average is to fix the pressure value at an arbitrary point of the boundary domain. Note that the fluid properties ρ and μ are kept constant element wise. Since interface between fluids is not crossing any finite element, the fluid properties are sharply defined at each phase, not requiring any additional function to smoothing out fluid properties near the interface. Once the discretization of the domain is accomplished, the system matrices are assembled in the classical finite element way and the solution of the time dependent 2-dimensional equations in axisymmetric coordinates is then found by successively solving the linear system in each time step for pressure and velocity. Due to the strong coupling between pressure and velocity, the numerical procedure implemented to solve the mentioned linear system uses the Projection Method based on the LU decomposition, which was first introduced by [21]. The aim of this method is to uncouple pressure and velocity and solve each quantity separately, thus reducing the large linear system size into smaller ones. In this work, both pressure and velocity linear systems are solved using a direct solver. The advantage on using direct solver is that the solution of the linear system is given in one iteration and solved to machine precision. The drawback is the need of extra computational resources. For 2-dimensional simulations our tests have shown excellent agreement between solution time and accuracy, and it represents approximately 43% of the overall simulation time for the test cases presented at the Results section.

3.2. Material derivative - advection term treatment

If one sets $\mathbf{v} = \hat{\mathbf{v}}$ to all mesh nodes, no additional method is required to solve the advection in the Navier-Stokes equation, since the material derivative is discretized in time and space by:

$$\frac{D\mathbf{v}}{Dt} = \frac{\mathbf{v}^{n+1} - \mathbf{v}^{n}}{\Delta t} \tag{26}$$

with the new position of mesh nodes given by:

$$\mathbf{x}^{n+1} = \mathbf{x}^{n} + \mathbf{v}^{n+1}\Delta t \tag{27}$$

On the other hand, if mesh velocity $\hat{\mathbf{v}}$ is set to an arbitrary value, an additional method is required to solve the remaining part of the advection term within the material derivative. In other words, the portion of velocity field that is not solved by Eq. (26,27), should be found by any other *Eulerian* method.

There are several ways to numerically tackle the discretization of the non-linear term $\mathbf{v} \cdot \nabla \mathbf{v}$, such as the Petrov-Galerkin method as described in [22] and [23], the characteristic Galerkin method [24], or even the pure Galerkin method if simulations are performed with low Reynolds numbers [25]. In this work, the proposed solution is achieved by discretization of the material derivative through the semi-Lagrangian method as implemented in [8, 26]. For the sake of completeness we briefly describe it here.

The semi-Lagrangian method has its own history dating from the end of the 1950's ([27]) and the beginning of the 1960's ([28] and [29]). However, the use of such a methodology for modeling fluid flow problems came later in the 1980's through the work of [30] and [31], in which predominately convective problems were investigated.

The semi-Lagrangian method uses the *Eulerian* framework, however the discrete equations are written in the *Lagrangian* framework using an integrating factor method based on the method of characteristics in which such a factor is an advection operator. This operator is shifted to a moving coordinate system from which the next time step quantity is calculated. Let ψ be a scalar function in its material derivative representation, given in axisymmetric form as:

$$\frac{D\psi}{Dt} = \frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} + v\frac{\partial \psi}{\partial r} \tag{28}$$

Note that the axisymmetric form is identical to the Cartesian form. Following this description, Eq. (28) may be discretized linearly in time at the point $\mathbf{x}_i$ by using an explicit first order scheme:

$$\frac{D\psi}{Dt} = \frac{\psi_i^{n+1} - \psi_d^n}{\Delta t} \tag{29}$$

where $\psi_d^n = \psi^n(\mathbf{x}_d, t^n)$ and $\mathbf{x}_d$ is the departure point. In the strong form, the substantial derivative is calculated along the characteristic trajectory, thus finding the point $\mathbf{x}_d$ by solving $D\psi/Dt = f$ backwards in time $t^{n+1} \geq t \geq t^n$

using the initial condition $\mathbf{x}(t^{n+1}) = \mathbf{x}_i$.

In the semi-Lagrangian context, a searching procedure is required to find the unknown departure points $\mathbf{x}_d$ in time t^n. This procedure may lead to excessive computational cost if it is not well designed, thus it should be treated with appropriate care. In this work, this searching procedure is implemented using neighbor-to-neighbor technique as found in [32] that maps each element node to the opposite element that shares the same triangle edge. Therefore, using area coordinates, it is possible to track the path from the current node's position $\mathbf{x}_i$ to its departure point $\mathbf{x}_d$ using less computational time. In this work, the semi-Lagrangian method is used whenever the mesh parameters presented in Sec. (4.2) are not null. In the same way described above, the semi-Lagrangian method is used to discretize the material derivative of the velocity field.

3.3. Discrete surface tension force

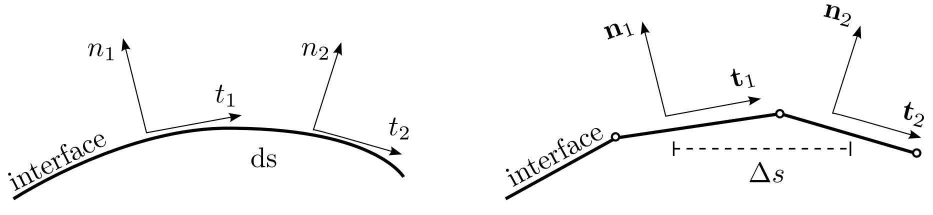
The discrete first order approximation of the non-dimensional surface tension force is given by:

$$\mathbf{f_{st}} = \kappa \mathbf{n} \nabla H \tag{30}$$

where κ stand for the curvature, $\mathbf{n}$ is the normal vector defined outward and ∇H is the gradient of the Heaviside function H, which is equal to one in the inner phase, zero in the outer phase and assumes an average value of 0.5 for nodes lying in the interface between fluids. To compute the curvature κ and the normal vectors $\mathbf{n}$, the interface nodes are used and a set of tangent vectors can be assembled such that:

$$\kappa \mathbf{n} = \frac{(\mathbf{t}_1 - \mathbf{t}_2)}{\Delta s} \tag{31}$$

where $\mathbf{t}_1$ and $\mathbf{t}_2$ are approximations of the unit tangent vectors in two consecutive interface edges and Δs is its approximate arc length. Figure (1a) shows the continuous representation of the mentioned formula and in Fig. (1b) the discrete representation of the 2-dimensional interface is presented. Two interface nodes are used to compute the tangent and normal vectors associated to an edge and a weighted sum is performed to set these vectors to the corresponding interface node. The approximate arc length is found by connecting the centroids of each segment.



(a)

(b)

Figure 1: The 2-dimensional *Frenet*'s formula for mean curvature ([20]). (a) The continuous description and (b) the discrete form used in the computational grid.

The axisymmetric component of the curvature is approximated by a signed distance radius as follows:

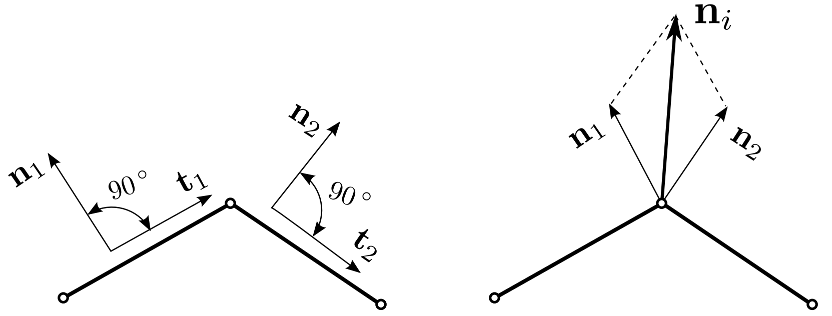
$$\kappa_{axi} = \frac{\sin(\phi)}{r} = \frac{1}{R} \tag{32}$$

The discrete form of the surface tension for the axisymmetric coordinate is then written as:

$$\mathbf{f}_{st} = \left[\frac{|\mathbf{t}_1 - \mathbf{t}_2|}{\Delta s} + \frac{1}{R} \right] \mathbf{n} \nabla H \tag{33}$$

In the above equation, the capillary force intensity $|\mathbf{t}_1 - \mathbf{t}_2|$ is divided by the approximate arc length Δs and added to the axisymmetric discrete component $1/R$ where R is the signed distance from the interface node to the node where the normal intersects the symmetry axis, with the sign being positive if the normal points away from the symmetry axis and negative if the normal points toward the symmetry axis. In this scheme, the vector $\mathbf{n}$ can be determined in several ways, for instance see [25] for a Level-Set approximation of the normal vector $\mathbf{n}$. If the interface representation is explicitly defined by nodes, as the case of the present work, a more accurate evaluation of the normal vector $\mathbf{n}$

can be achieved by orthogonalizing the same calculated unit tangent vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ as shown in Fig. (2a). The nodal normal vector $\mathbf{n}_i$ is found by summing the two normal vectors associated to the adjacent edges $\mathbf{n}_1$ and $\mathbf{n}_2$. The unit normal vector is therefore computed dividing $\mathbf{n}_i$ by its length. In 2-dimensional space, such a calculation is done relatively easy, since the interface mesh is always structured and thus the number of neighbors of each interface node is constant and equal to 2, except of two interface nodes lying at the symmetry axis. In such a case, the outward unit normal vector is set to be parallel to the $x-axis$. It is important to note that the normal vector is only used to evaluate the correct direction of application of the surface tension force at the proposed numerical scheme, thus an accurate computation of the normal vector is not strictly required. It is, however, a good choice to accurately compute those vectors since the mesh repair algorithm and the interface motion are carried out using the normal vector definition. An extensive study about different methods for accurately calculate normal vector can be found at [33].



(a) (b)

Figure 2: Normal vector evaluation in 2-dimensional spaces. (a) The normal vector of each edge may be found by rotating the previously calculated tangent vector by 90°. (b) The outward nodal normal vector $\mathbf{n}_i$ is found by summing the two normal vectors n_1 and n_2 and the respective unit vector is found by dividing by its norm.

In the axisymmetric formulation, extra care should be taken for the two interface nodes lying at the symmetry wall. The curvature κ_{2d} and normal vector $\mathbf{n}$ associated to these nodes are simply computed by reflecting the respective adjacent node in the symmetry axis, therefore creating a ghost node, and performing the normal computation as described above for the interface nodes where the axisymmetric component of the curvature κ_{axi} is zero.

These schemes are compatible with a linear continuous approximation of the pressure finite element space and, consequently, can be successfully applied to the non-dimensional Navier-Stokes equations (1).

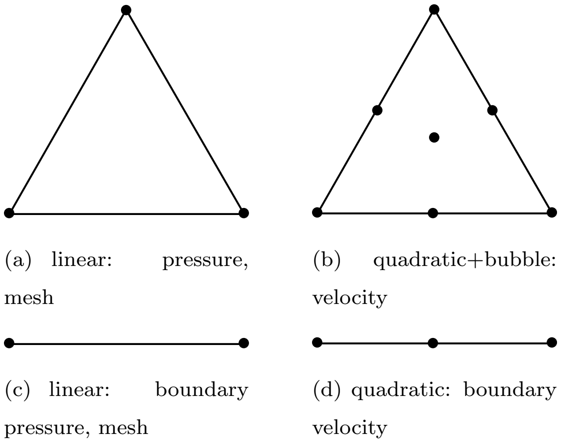
3.4. Triangle and boundary finite elements

When the fluid motion equations are solved using the FEM, the spatial domain Ω is subdivided into finite elements and the solution is sought in finite dimensional subspaces of $\mathcal{U}_{u_D}, \mathcal{V}_{v_D}$ and $\mathcal{Q}$. These discrete subspaces are denoted $\mathcal{U}^h_{u_D}, \mathcal{V}^h_{v_D}, \mathcal{Q}^h$ and they are composed of continuous functions which are piecewise-polynomials.

A combination of velocity/pressure shape functions is usually denoted as $PnPm$, where n stands for the degree of the velocity interpolation and m stands for the degree of the pressure interpolation. Particularly for the Navier-Stokes equations, not all velocity/pressure shape function pairs lead to a stable discretization method, thus they must fulfill the so called LBB conditions [34]. Several combinations of velocity/pressure shape functions have been demonstrated to fulfill the LBB condition.

Figure. 3 displays the finite element used in this work. The linear triangle element and the quadratic+bubble triangle element which is enriched by an extra node at the centroid were used to evaluate pressure and velocity respectively. The boundary linear and quadratic element were used for the boundary conditions of pressure and velocity respectively. Moreover, the linear triangle and linear boundary element were used to all mesh operations, therefore once the mesh is updated to a new mesh, the extra nodes are recreated to form the high-order elements. The combination of these two triangle elements generates the $P2^{bubble}P1$ element. The shape functions N_i, as function of the barycentric coordinates λ_i, are defined for both triangle elements, in which $N_i = \lambda_i$ for the linear elements and a combination of λ_i for the quadratic+bubble triangular element as follows:

$$\begin{aligned}
\text{corner node} : N_i &= \lambda_i(2\lambda_i - 1) + 3\lambda_1\lambda_2\lambda_3 \qquad i = 1, 2, 3 \\
\text{mid edge nodes} : N_4 &= 4\lambda_1\lambda_2 - 12\lambda_1\lambda_2\lambda_3 \\
N_5 &= 4\lambda_2\lambda_3 - 12\lambda_1\lambda_2\lambda_3 \\
N_6 &= 4\lambda_1\lambda_3 - 12\lambda_1\lambda_2\lambda_3 \\
\text{centroid} : N_7 &= 27\lambda_1\lambda_2\lambda_3
\end{aligned}$$



(a) linear: pressure, mesh

(b) quadratic+bubble: velocity

(c) linear: boundary pressure, mesh

(d) quadratic: boundary velocity

Figure 3: Interpolation nodes for triangular finite elements and boundary finite elements used in this work. The linear element was used to evaluate pressure at the corner nodes and mesh update, while the quadratic+bubble (enriched by the centroid) and the quadratic boundary element were used to evaluate velocity at all nodes. The combination of these two triangle elements generates the $P2^{bubble}P1$ element.

The shape functions N^b, as function of the barycentric coordinates λ^b, are defined for both boundary elements, in which $N^b = \lambda^b$ for the linear elements and a combination of λ^b for the quadratic boundary element as follows:

$$\begin{aligned}
\text{corner node} : N_i^b &= (2\lambda_i^b - 1)\lambda_i^b \qquad i = 1, 2 \\
\text{mid edge nodes} : N_3^b &= 4\lambda_1^b\lambda_2^b
\end{aligned} \tag{34}$$

The matrices of Eq. (24) are assembled with the $P2^{bubble}P1$ finite element and the integrals are evaluated numerically using 4 and 16 Gaussian points for the linear and quadratic+bubble element respectively.

3.5. Time step restriction

In the governing equations, the explicitly treated terms lead to a limitation of the chosen time step size. As mentioned in [31], the semi-Lagrangian method is unconditionally stable for the time steps. However, we restrict the corresponding time step to avoid excessive numerical diffusion, thus the adopted time step constraint for the semi-Lagrangian term is given by the expression below:

$$\Delta t_{sl} < \frac{h_{min}}{|\mathbf{v}_{max} - \hat{\mathbf{v}}_{max}|} \tag{35}$$

where h_{min} is the smallest triangle mesh edge length and $\mathbf{v}_{max} - \hat{\mathbf{v}}_{max}$ is the maximum value of the *Eulerian* velocity. On the other hand, the motion of the mesh nodes depends on the value of $\hat{\mathbf{v}}$ given by Eq. (41) and the time step constraint should prevent the nodes from moving more than one mesh cell, otherwise the mesh is corrupted due to element distortions and collision of nodes. The strategy of the moving mesh time constraint calculation is based on the velocity differences and the local mesh edge size, thus limiting one node to overlap the other or cross the edge of an element. Let us consider p_1 and p_2 as the vertices of a triangle edge h and $\Delta\hat{\mathbf{v}}_e = \hat{\mathbf{v}}_1 - \hat{\mathbf{v}}_2$ as the mesh velocity difference, thus the proposed Lagrangian time step should be bounded by:

$$\Delta t_l < \min\left(\frac{h}{|\Delta\hat{\mathbf{v}}_e|}\right) \tag{36}$$

Due to the explicit treatment, gravity and surface tension adds a constraint to the final time step which is related to the wave velocity propagating into the computational mesh. According to [19] and [35], such a criteria for both surface tension and gravity may be written as:

$$\Delta t_s < \left[\frac{\overline{\rho} h^3 \mathrm{Eo}}{2\pi}\right]^{1/2} \tag{37}$$

and

$$\Delta t_g < \left[\frac{1}{h_{min}}\right]^{1/2} \tag{38}$$

In the above equations, $\overline{\rho}$ is an average fluid density between the inner and outer fluids, h is the mesh characteristic length, h_{min} is the smallest mesh length and Eo is the *Eötvös* number as defined in Sec. (2). Thus, the final simulation time step constraint may be written as:

$$\Delta t \leq \min\left\{\Delta t_l, \Delta t_s, \Delta t_g, \Delta t_{sl}\right\} \tag{39}$$

4. Mesh Operations

In the present work an interface tracking method is used together with node insertion/deletion algorithms to maintain a suitable mesh quality. Such a mesh repair is performed in both the 2D domain mesh (triangles) and the boundary mesh (line segments) together with a mesh smoothing scheme. Different criteria are implemented to control the local mesh refinement. The mesh repairing is based on a target edge size which is a discrete scalar field (just like the flow quantities), i.e. defined for every mesh node. Such a target edge size is computed by solving a Helmholtz equation to ensures that the node distribution is smooth. Moreover, a mesh smoothing scheme is proposed to avoid excessive remeshing. Finally, a mass conservation algorithm is presented to keep two-phase flow mass conserved when strong mesh update occurs. The details of this proposed remeshing algorithm is described in the following sections.

4.1. Mesh nodes distribution

The triangle edge length determines whether insertion or deletion is required for a given predefined distance in a specific zone, therewith it is possible to avoid clustering and dispersion of computational nodes. In this work, the distribution edge lengths h are obtained by the solution of the following Helmholtz's equation:

$$\nabla^2 h = \frac{1}{k_h}(h - h_b) \tag{40}$$

where k_h is a diffusive parameter and h_b is the initial edge length distribution. Thus, the obtained solution h corresponds to a smooth distribution of nodes in the space. Note that for large values of k_h in the above equation, the right hand term tends to zero, thus resulting in Laplace's equation ($\nabla^2 h = 0$) in which the solution damps all the sudden changes in the distance between nodes. On the other hand, assuming a small value of k, the solution h approaches the initial node distribution h_b. The initial node distribution h_b may be set according to the flow requirements, where a particular zone may or may be not refined.

In this work, the boundary conditions to solve the Helmholtz equation are set to a chosen non-dimensional characteristic length and imposed as a Dirichlet type. Typical values of such a characteristic length is at order of 10^{-2}. The initial node distribution is defined according to a distance function inversely proportional to the interface between phases. Therefore, closer to the bubble's interface the clustering of nodes is higher. Another initial node distribution may be defined to set higher node density close to the domain boundaries, where velocity gradient is usually high. In this case, the boundary conditions is set to very small characteristic length (e.g. 10^{-4}) and bubble's interface to a higher value such as 10^{-2}, thus using the distance function, one can create an initial distribution h_b as mentioned before to be used in the Helmholtz equation, where solution delivers high density of nodes close to the walls.

The solution of the Helmholtz equation (40) has been shown to be extremely important in order to achieve a smooth distribution of nodes in the 2-dimensional domains. It is important to note that the solution of such equation has been extensively studied by many authors (see, for instance [36] and [37]), where it was shown that pollution errors may affect the solution when the wave number k_h increases, thus requiring extremely refined meshes. Due to the positive sign of the right hand side term h, the eigenvalues of the presented Helmholtz equations are complex and have only imaginary part, consequently avoiding pollution errors at the solution. Additionally, no refined meshes are required and its solution can be used to compute a smooth distribution of nodes.

The discretization of Eq. (40) follows the standard procedure found in finite element formulation for 2-dimensional spaces with the Galerkin method, thus, it does not need to be integrated in an axisymmetric fashion, however homogeneous Neumman boundary conditions should be used at the symmetry axis. Successive solution of the Helmholtz equation should be avoided if computer resources are limited.

4.2. Mesh smoothing

The mesh velocity $\hat{\mathbf{v}}$ determines the motion of the nodes of the finite element mesh. This velocity is obtained by a linear combination of the flow velocity itself and smoothing velocities, in which the latter is defined according to a criterion to redistribute the mesh nodes, thus minimizing the number of remeshing steps and avoiding heavy computation requirements. The transfinite mapping method (see [38]) and Laplacian smoothing are examples of mesh-update procedures ([39]). To achieve accurate results and avoid excessive remeshing, the mesh velocity should be set according specifically to the test case of interest, i.e. each two-phase flow must be treated specifically by adjusting the motion of the mesh nodes accordingly. The mesh velocity can be any geometrical scheme or even part of the fluid flow velocity $\mathbf{v}$. Indeed, it can also be a set of different schemes assembled in a manner aimed to redistribute the mesh nodes to avoid clustering and collapsing of finite elements, consequently interrupting the numerical simulation.

In this work, the 1D boundary mesh (interface and domain boundaries) and the 2D domain mesh are treated separately, thereby making the remeshing more flexible and adjustable to different flow situations. The mesh velocity $\hat{\mathbf{v}}$ is combined into a scheme that considers different mesh motions and is adjusted by parameters varying from 0 to 1 allowing flexibility and adaptability to different two-phase flows. These mesh motions covers the well known *Laplacian* smoothing operator that is applied to the triangular mesh, interface mesh and the axisymmetric boundary. Moreover, we have added a smoothing process based on the interface velocity that keeps interface node neighbors away enough from the interface nodes, thus avoiding element collapsing in demanding mesh deformation test cases. Additionally, to decrease the displacement of nodes in the tangential direction, one may remove partially, or even totally, its velocity from the total interface's velocity. We propose a set of mesh velocities that compose the final mesh velocity $\hat{\mathbf{v}}$ that can be included in Eq. (6). The 1D boundary and 2D domain mesh velocities are therefore treated as follows:

$$\hat{\mathbf{v}}(\mathbf{x}) = \begin{cases} \beta_1 \mathbf{v} + \beta_2 \mathbf{v}_v + \beta_3 \mathbf{v}_e & \text{if } \mathbf{x} \text{ does not belong to the interface} \\ \mathbf{v} - \gamma_1 (\mathbf{v} \cdot \mathbf{t})\mathbf{t} + \gamma_2 (\mathbf{v}_e \cdot \mathbf{t})\mathbf{t} & \text{if } \mathbf{x} \text{ belongs to the interface} \end{cases} \tag{41}$$

where $\mathbf{v}$ is the flow velocity, $\mathbf{v}_e$ is the Laplacian smooth velocity and $\mathbf{v}_v$ is an interface elastic mesh velocity. Such a velocity is the propagation of the interface mesh velocity to its neighbours, making interface neighbours to follow the same mesh motion of the interface, therefore avoiding squeezing of triangle elements around the interface between fluids. The strategy consists in finding all mesh nodes where the distance to the interface is bounded to a predefined distance parameter, including inner mesh nodes, and thus propagating the nearest interface node velocity to the mesh node itself. If such a described method is used at it is, all interface neighbours will move with same velocity of the closer interface node. However, one can use the same distance function to decrease linearly, quadratically or exponentially the amount of interface velocity associated to the neighbour mesh node, where closer mesh nodes will have more velocity and remote nodes less interface velocity. This scheme has been successfully applied to overcome the fast element distortion close to the interface where the velocity gradient may be high. Moreover, to decrease the displacement of nodes in the tangential direction at the interface, one may remove partially, or even totally, its tangential velocity from the total interface velocity. This can be achieved by removing the tangent component from the total interface mesh velocity $\mathbf{v} - (\mathbf{v} \cdot \mathbf{t})\mathbf{t}$. The parameter β controls the intensity of smoothing velocity in the inner and outer mesh and γ stands for the interface mesh regularization. With such a scheme, the parameters may be easily adjusted to fit the requirements of many specific simulation cases.

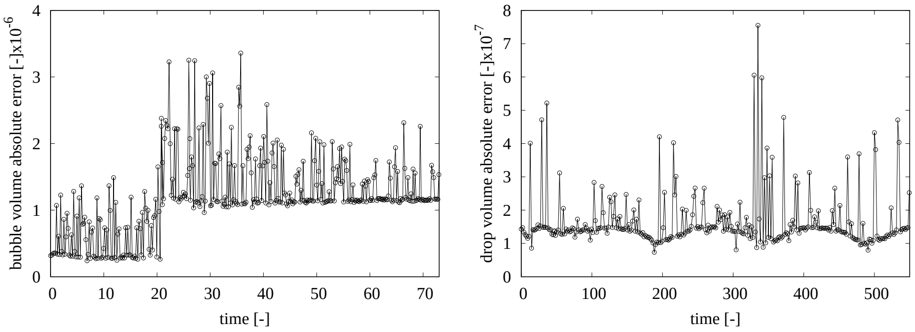
Large mesh deformation in two-phase flow simulations, specifically in gravity-driven simulations, suggests that the moving mesh technique requires arduous geometric maintenance, thereby demanding correspondingly high computational resources. The proper choice of the parameters in terms of mesh transport (β's and γ's) reduces considerably the frequency of the remeshing operations done in the numerical meshes.

4.3. Mesh repair

Unfortunately, mesh smoothing itself is not able to keep all the elements bounded to optimal shapes after numerous iterations. Furthermore, the moving front creates a poor distribution of interface nodes which can affect the accuracy of the computed curvature and, consequently, the final solution. Since the connectivity of the mesh is handled by the in-house code, a re-meshing technique is thus required to keep the elements aspect ratios in a satisfactory range as indicated by ([40], [41], [42], [43], [44]). The technique proposed here consists of changing the connectivity of the nodes and elements through "flipping" operations. Additionally, insertion and deletion of nodes is required when a coarse surface mesh is detected or when a dense cluster of surface nodes is not desired, respectively. A detailed description of the geometric operation is found in [18] and [45]. The criteria used when insertion and deletion of nodes is required is discussed as follows.

4.4. Mass conservation

Massive geometrical operations may lead to axisymmetric volume variation due to the insertion and deletion of nodes lying at the interface between fluids. To avoid accumulation of mass conservation errors a correction is done by moving the interface nodes in the direction of their associated normal vector, which are defined outward of the interface. Such a displacement is calculated based on the initial phase axisymmetric volume, which is compared to the current iteration, and thus a successive relaxation method is applied to find the final node positions. For a smooth interface geometry, the mass error after interface remeshing in one time step was calculated, corresponding to 0.03%, and 12 iterations were required to recover the initial bubble volume for a tolerance on the order of 10^{-8}. For severe interface remeshing, the mass error may increase and more steps are then necessary to recover the initial bubble's volume. Figure (4) shows the mass conservation error for the transient of two challenging problems with the proposed new technique of moving boundaries that will be presented at the results section. As can be seen at the figures, the bubble/drop volume absolute error is bounded to 10^{-6} and 10^{-8} for the rising bubble in slowly divergent channel and the motion of drop in periodically constricted capillary respectively. However it is possible to note that the error oscillates and grows when high interface deformation takes place due to the geometrical operations at the interface between fluids. It is also noted that more iterations at the relaxation method is required to control the bubble/drop's volume.



(a)

(b)

Figure 4: Mass conservation error evolution with time for two challenging problems with moving boundaries presented at the results section: (a) Rising bubble in slowly divergent channel and (b) motion of drop in periodically constricted capillary.

4.5. Moving boundaries

The ALE framework is commonly found in fluid-structure analysis due to its ability to displace mesh nodes according to the interactions between fluid and structure. Such a technique may also be applied to simulate two-phase flows with large domains with variable cross section. In this work we proposed a new method which combines the features of fluid-structure mesh dynamics for simulating two-phase flows where the cross section of the channel varies according to channel designs. Large periodic and non-periodic domains can be easily treated with such a technique, reducing dramatically the computational domain and consequently shorten the required simulation time. Since such a technique is not related to the discretization method, nor the space dimension, it can be also used in the finite volume method as long as the equations are written in the ALE framework in both 2-dimensional and 3-dimensional computations.

The proposed methodology consists in simulate the two-phase flow system using the moving referential frame, where bubble remains nearly fixed in space where flow and boundaries travels backwards. In a standard situation, a slip velocity condition is set to the walls according to the bubble's center of mass velocity, however this has been only applied to straight channels. In our work, the relative motion is also set to the wall boundaries where boundary nodes are displaced in a Lagrangian way for the radial direction relative to the bubble's referential point, keeping the axial direction with an Eulerian slip condition. The bubble's referential point can be any, for instance the bubble's center of mass or the bubble nose.

The proposed strategy is easily embed to the present code at the mesh smoothing method already presented. First, we find the new radial position of all wall boundary nodes relative to the bubble's referential point r^{n+1}_{wall}. Thus, we find a Lagrangian radial velocity simply dividing the difference between the initial wall boundary nodes position r^{n}_{wall} and the new position by the current time step as follows:

$$v = \frac{r_{wall}^{n+1} - r_{wall}^{n}}{\Delta t} \tag{42}$$

the radial velocity component v is used to move the points in a Lagrangian way, where the axial velocity u follows the standard Eulerian slip condition in the moving referential frame according to the bubble's referential point velocity. The new mesh update set of equations for the mesh motion is written as follows:

$$\hat{\mathbf{v}}(\mathbf{x}) = \begin{cases} \beta_1 \mathbf{v} + \beta_2 \mathbf{v}_v + \beta_3 \mathbf{v}_e & \text{if } \mathbf{x} \text{ does not belong to the interface} \\ \mathbf{v} - \gamma_1 (\mathbf{v} \cdot \mathbf{t})\mathbf{t} + \gamma_2 (\mathbf{v}_e \cdot \mathbf{t})\mathbf{t} & \text{if } \mathbf{x} \text{ belongs to the interface} \\ v & \text{if } \mathbf{x} \text{ belongs to the moving boundary} \end{cases} \tag{43}$$

here, the radial component of the velocity v is used to move nodes that belongs to the moving boundary. This scheme is successfully applied to the test cases delivering interesting results as can be seen in the results section.

The overall cost of the remeshing procedure (mesh smoothing, mesh repair operations, mesh nodes distribution and volume update) in the presented simulations represents 6% of the total simulation time. 2% is dedicated to the interpolation of the new nodes to the previous mesh. Such a small amount is due to the proposed moving frame/moving boundary technique where extensive remeshing is not required and the bubble does not move excessively causing node collapsing and squeezing of mesh elements.

5. Results

Several single- and two-phase flows benchmark tests were carried out to successfully validate the code, including the *Poiseuille* flow, *Drop under uniform translation* and two-phase flows in microchannel geometry as can be seen in [18]. In this section, numerical results obtained with the present Arbitrary Lagrangian-Eulerian Finite Element code in axisymmetric coordinate for incompressible two-phase flows is presented.

5.1. Sessile Drop

The next simulation was performed to validate the surface tension implementation and its coupling with pressure and gravity. A spherical drop with radius R = D/2, where D is the non-dimension diameter was initialized two diameters above the bottom of the domain and then released. Due to gravity, the drop, being heavier than the surrounding fluid, falls and hits the solid line. Before the contact between the solid line and the interface, the drop deforms to a quasi-steady state and approaches the wall with no significant shape changes. The drop's shape can be approximated by the classical Young-Laplace equation of capillarity which, in non-dimensional form, states that:

$$\sigma\left(\frac{1}{\mathrm{R}_1}+\frac{1}{\mathrm{R}_2}\right)=\Delta p_g=\Delta\rho g(x-x_0) \tag{44}$$

where R_1 and R_2 are the two principal radii at the apex of the drop, σ is the surface tension coefficient and Δp_g stands for the hydrostatic pressure difference across the interface, where $\Delta\rho = \rho_{in} - \rho_{out}$. Considering ϕ as the drop tangent angle with respect to the wall, for an axisymmetric drop:

$$\kappa=\left(\frac{1}{R_1}+\frac{1}{R_2}\right)=\frac{d\phi}{ds}+\frac{\sin(\phi)}{r} \tag{45}$$

where s is the coordinate along the interface and r is the radial coordinate. The equations for the equilibrium position of an axisymmetric liquid drop in non-dimensional form is written as follows:

$$\frac{d\phi}{ds}=\hat{\mathrm{Eo}}(p-x)-\frac{\sin(\phi)}{r} \tag{46}$$

$$\frac{dr}{ds}=\cos(\phi) \tag{47}$$

$$\frac{dx}{ds}=\sin(\phi) \tag{48}$$

Here, $\hat{p}$ stands for the dimensional reference pressure, $p=\hat{p}/\rho g\mathrm{L}$ as the non-dimensional pressure and $\hat{\mathrm{Eo}}=\mathrm{Eo}(\Delta\rho/\rho_{in})$ as the modified *Eötvös* number.

These equations are integrated numerically using the 4th. order Runge-Kutta method with appropriate initial conditions: $\phi = 0$, $p = 3.095$, $r = 0.205$ and $x = 0.037$ and are then integrated up to $\phi = 4.196$rad.

The simulation of the drop was performed assuming the following non-dimensional parameters: R = D/2, Eo = 2, N = 100, $\rho_{in} = 1.0$ and $\mu_{in} = 1.0$ for the drop and $\rho_{out} = 0.1$ and $\mu_{out} = 0.9$ for the external fluid. The non-dimensional domain limits were set to be 3.5D × 2.5D, where the first dimension stands for the direction of gravity, and discretized by approximately 2000 triangles and 1100 triangle vertices.

Lagrangian mesh motion is not desired due to extensive remeshing requirement, thus $\beta_1 = 0.0$. The drop is translating downward, thus the mesh nodes will be squeezed by the interface motion. Setting $\beta_2 = 1.0$ helps adjacent interface nodes travelling with partial interface velocity. $\beta_3 = 1.0$ is a good choice to keep mesh distribuition smooth using the Laplacian operator. For the interface mesh, $\gamma_1 = 1.0$ and $\gamma_2 = 0.1$ remove the tangent interface velocity and partially smooth the interface node distribution respectively, since interface remeshing is already delivering a quality mesh distribuion.

Figure (5) shows the evolution with time of the drop shape in five different time steps and the associated numerical mesh, including the explicit interface. From $t = 3.10$ and on, the drop's shape remains constant and the motion is purely translational, thus it can be compared with the numerical solution of Young-Laplace's equation.

The numerical solution of the drop's shape is compared to the solution of the Young-Laplace's equation (Eq. (46),(47) and (48)) and is shown in Fig. (6). The results show that the sessile drop is correctly predicted by the present implementation, due to the accurate balance of the gravity, pressure and surface tension force. The remeshing algorithm used 5% of the simulation time, while the solution of the linear systems used approximately 43%. Assembling of the finite element operators used 28%, while setting of boundary condition used 10% of the total simulation time for this test case.

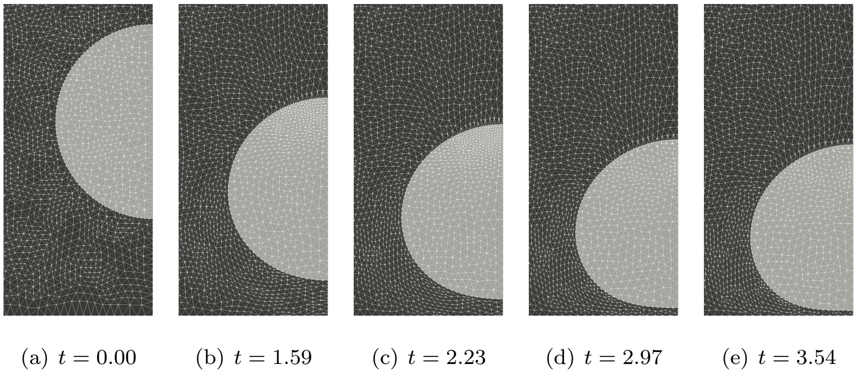


Figure 5: Bubble shape evolution with time for a drop in gravity field with dimensionless numbers N = 100, Eo = 2. (a) Initial bubble shape with $t = 0.00$. (b-d) Bubble shape change during transient solution. (e) Bubble shape of the long-term film drainage state at $t = 3.54$.

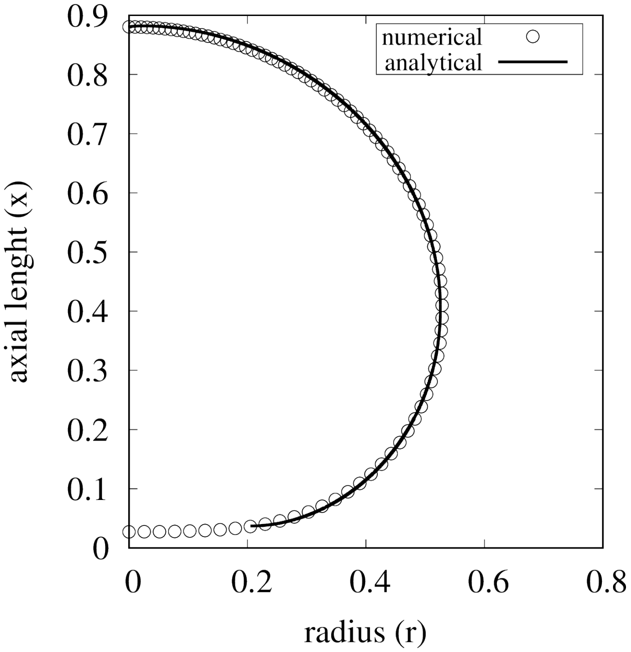


Figure 6: Comparison between the numerical solution of an axisymmetric sessile drop and the solution of its shape derived by the Young-Laplace equation of capillarity.

5.2. Rising of air bubble

The numerical results were compared to the widely cited experiments performed by [46]. Tests were carried out to predict the terminal velocity of a rising air bubble in aqueous sugar solutions for three different viscosities. According to the experiments, the volume of the generated bubbles was 9.3cm^3, thus the diameter is d = 2.61cm. The surface tension of 0.078N/m was that of the air-water-sugar interface, the bubble air's viscosity and density are $1.78{\times}10^{-5}$kg/ms and 1.225kg/m^3 respectively. We considered an average value for the aqueous solution density to be 1350kg/m^3, since the experiments presented measurement variations, and five different liquid viscosities $\{2.73, 1.28, 0.54\}$kg/ms, thus changing the final shape of the rising bubbles. The refinement levels used had approximately 2000 nodes, 3100 triangles, 350 interface nodes and 351 interface lines. The geometry of the domain consisted in a rectangle with dimensions of 12D × 5D with the higher dimension along the gravity direction was used and a bubble with radius R = 0.5 was placed at the top of the domain where the center of mass is located at r = 8D. A moving frame strategy was adopted here, where the bubble's front node at the axis of symmetry remains fixed in space and flow is moving backward. Such a strategy is interesting for simulating large domains, otherwise the bubble would touch the top lid of the numerical domain.

Figure (7) shows bubble shape transition and the center of mass velocity of the rising air bubble immersed in sugar water solution. The bottom edge of the bubble, where the curvature changes its sign varies accordingly to the viscosity. The mesh parameters used in these simulations is in accordance to the previous test case (Sessile Drop) were a fixed frame is used to simulate the translation of the bubble's centroid, therefore $\beta_1 = 0.0$, $\beta_2 = 0.8$, $\beta_3 = 1.0$ were set to the inner/outer mesh and $\gamma_1 = 1.0$ and $\gamma_2 = 0.1$ to the interface mesh. Note that we set $\beta_2 = 0.8$ differently of the previous test case, and we have not seen any quantitative difference compared to $\beta_2 = 1.0$.

The Eötvös number was constant for all simulations and were set to Eo = 116. For each case, the Archimedes number changes due to the viscosity of the liquid, as the ratio of density and viscosity as follows: (Case 1) N = 194.88,

$\rho_{in}/\rho_{out} = 9 \times 10^{-4}$, $\mu_{in}/\mu_{out} = 6.53 \times 10^{-6}$. (Case 2) N = 194.88, $\rho_{in}/\rho_{out} = 9 \times 10^{-4}$, $\mu_{in}/\mu_{out} = 1.39 \times 10^{-5}$. (Case 3) N = 1091.57, $\rho_{in}/\rho_{out} = 9 \times 10^{-4}$, $\mu_{in}/\mu_{out} = 3.29 \times 10^{-5}$.

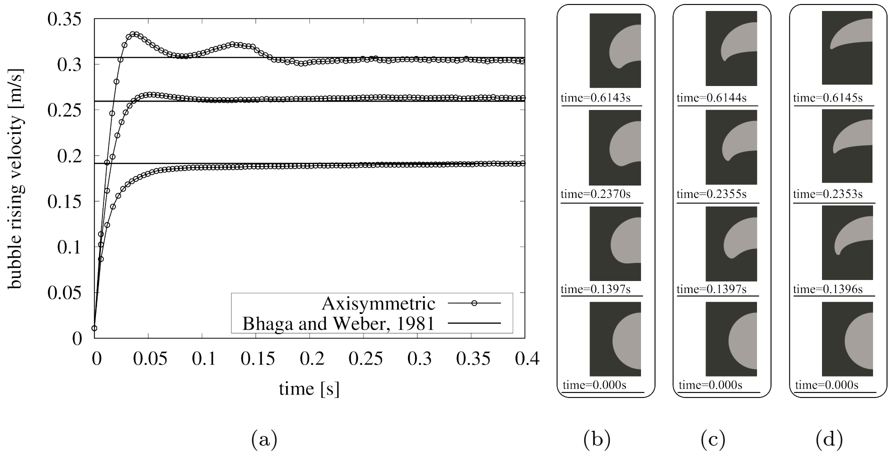


Figure 7: Time evolution of rising of an air bubble immersed in aqueous sugar solution. (a) Bubble rising velocity for three test cases and comparison with experiments performed by [46]. (b) $\mu = 2.73$kg/ms, (c) $\mu = 1.27$kg/ms and (d) $\mu = 0.539$kg/ms

In Fig. (8) the final bubble shape for the three presented test cases are shown with the associated finite element mesh, including the explicit representation of the interface with nodes and triangle edges. As can be seen in Figs. (8b,8c), strong mesh distortion is well modeled by a coarse mesh, delivering accurate results as compared to [46].

5.3. Rising of Taylor air bubble

To characterize the rising velocity of air bubbles in different solutions, we have identified 2 regions in the flow pattern map for the present numerical simulations. One solutions of sucrose and one of ethylene-glycol have been used. Table 1 summarizes the fluid properties used in the rising Taylor (elongated) bubble simulations.

The numerical solution for the rising bubble requires a long domain to be compatible to the experiments. According to [47], the development of the bub-

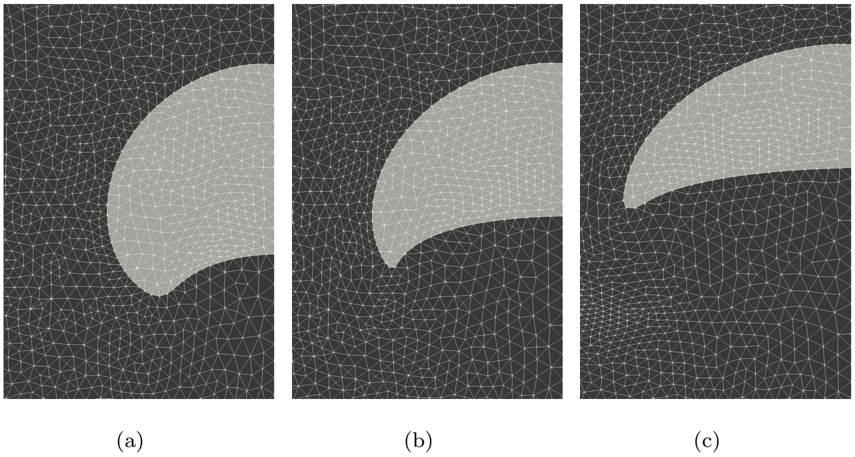


Figure 8: Final bubble shape and the associated finite element mesh for the rising of a single bubble in three different water-sugar solution where the dynamic viscosities are (a) $\mu = 2.73$kg/ms, (b) $\mu = 1.27$kg/ms and (c) $\mu = 0.539$kg/ms

ble's shape and, consequently, the terminal velocity requires that the numerical domain should be wide enough to avoid flow disturbances. Using the moving frame strategy, the bubble's front node remains fixed in space while the flow moves backward. The moving boundary technique described in this work is not used in the current test case since the radius of the channel remains constant during all the simulation. Since wall is affecting the hydrodynamic of the problem, the boundary condition should be updated according the bubble's referential point velocity, therefore the upper domain wall had slip velocity condition in x direction. The numerical domain used to simulate all the fluids given by Table (1) was set to D/2 × 10D, where D stands for the circular channel diameter. Due to the moving frame strategy, the initial Taylor bubble was fixing the bubble's front node at $x = 7.9$D to guarantee that the bubble's wake is well captured, thus allowing the bottom of the bubble to deform. The initial bubble's film thickness was fixed to $\delta = 0.1$ to all simulations. Care was taken to add an extra mesh node layer to solve the liquid film formed between the wall and a confined bubble.

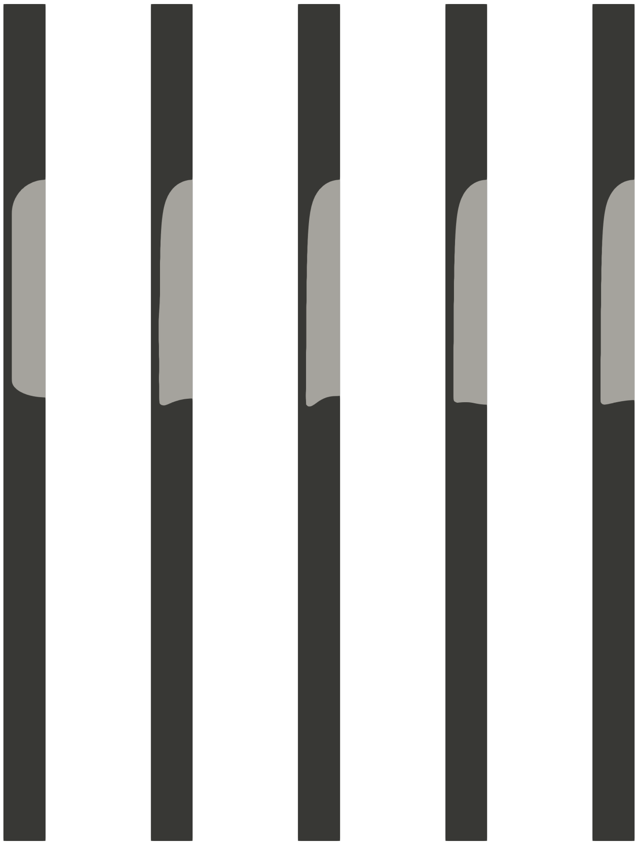
Table 1: Fluid properties for the rising of air Taylor bubble

fluid	properties		dimensionless number	
	ρ	μ	Eo	Mo
	[kg/m^3]	[μPa $\cdot$ s]	[-]	[-]
sucrose	1172	5.650	50	10^{-4}
ethylene glycol	1113	19.900	30	10^{-6}
air	1.789	1.225	—	—

Figure (9) shows time progression of a Taylor air bubble immersed in a sucrose solution. The same bubble shape and film thickness, as the previous cases, is used as an initial shape. In the transient evolution, the bubble's velocity reached its maximum velocity at time $t \approx 1$, and its terminal velocity at time $t \approx 3.7$ Also, it was shown that the bottom part of the bubble was pulled in and oscillated until convergence at $t \approx 7.4$. The mesh parameters used in this test case are similar to the previous test case and were set to $\beta_1 = 0.0$, $\beta_2 = 1.0$, $\beta_3 = 1.0$, $\gamma_1 = 1.0$ and $\gamma_2 = 0.1$. The dimensionless numbers were set to $\text{Mo} = 10^{-4}$, $\text{Eo} = 50$. Figure (10 presents the transient solution of the bubble's center of mass velocity. The initial number of mesh nodes (only vertices of the triangle) and mesh elements were 9611 and 18370 respectively, where the final were 9653 and 18445 for $t = 8.02$.

It was observed, an overshooting of the ascension velocity from time $t = 2$ to $t = 3$, due to the initial deformation of the bottom part of the bubble, and consequently acceleration of the center of mass. The result of the bubble's terminal velocity agreed to the prediction of the flow pattern map. The computed numerical error was found to be of 1.3% for $t > 6$.

Figure (11) show the evolution in time of the bubble's center of mass velocity, in which the computed terminal velocity approaches the value found in



(a) t = 0.00

(b) t = 1.52

(c) t = 3.07

(d) t = 4.14

(e) t = 8.02

Figure 9: Bubble shape evolution with time an air bubble in a sucrose solution with dimensionless numbers Mo $= 10^{-4}$, Eo $= 50$. (a) Initial bubble shape with $t = 0.0$. (b-d) Bubble shape change during transient solution. (e) Terminal bubble shape with $t = 8.02$.

the flow pattern map. The error was found to be 5.5%. The mesh parameters used in this simulation were $\beta_1 = 0.0$, $\beta_2 = 0.5$, $\gamma_1 = 0.5$ and $\gamma_2 = 0.1$ and the pair of dimensionless numbers was set to Mo $= 10^{-6}$,Eo $= 30$. Additionally, a numerical mesh detailed view is shown in Fig. (11b) where one can observe the tail of the air bubble. Note here that the interface is represented by nodes and

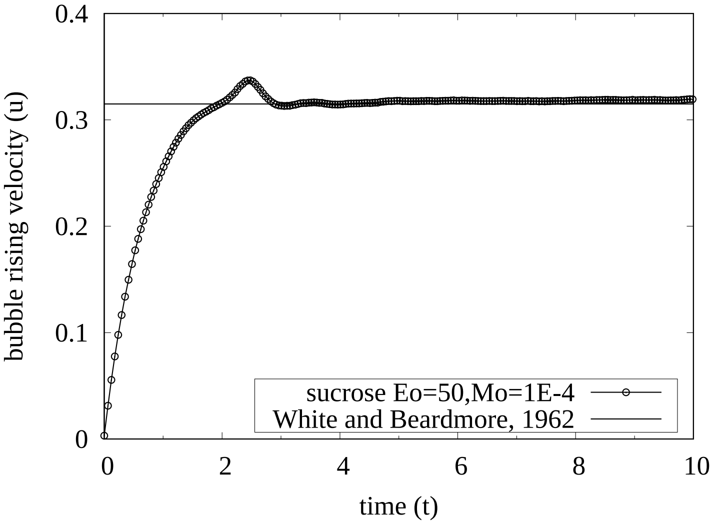


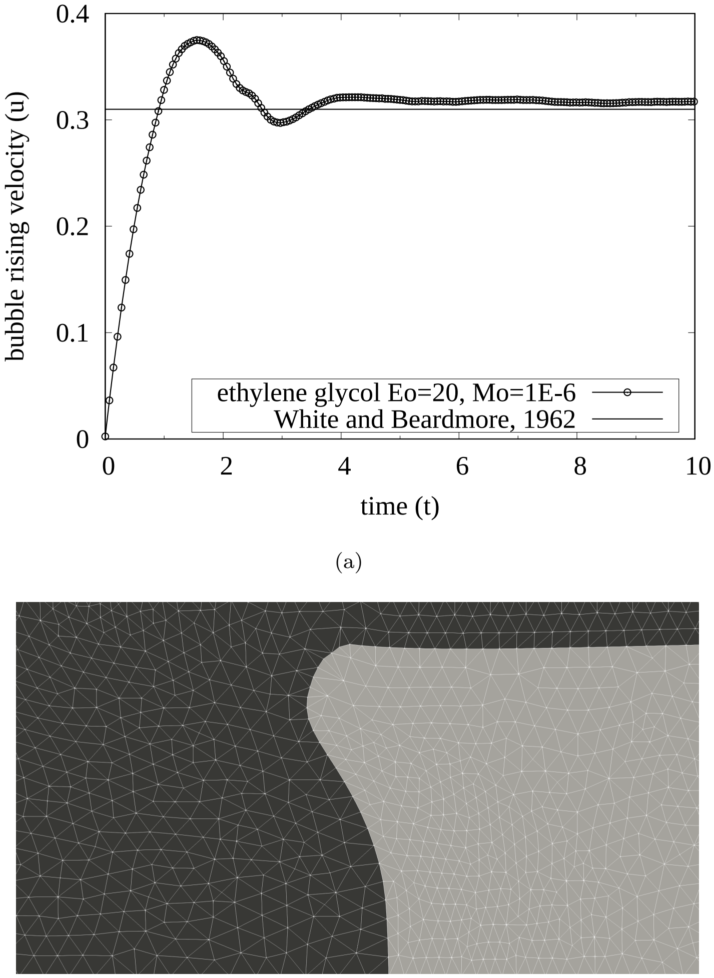
Figure 10: Rising of an air Taylor bubble immersed in a sucrose solution with dimensionless numbers set to Mo $= 10^{-5}$ and Eo $= 40$. The time evolution of the Bubble's center of mass velocity is compared to the terminal bubble's velocity found in [48]. Velocity and time are non-dimensional.

line segments both part of the numerical mesh. Moreover, it can be seen that the relative high interface deformation can be handled with the current methodology. The initial number mesh nodes (vertices of triangle) and mesh elements were 11495 and 21978 respectively. While at the end of the simulation these numbers were 11609 and 22196 for mesh nodes and mesh elements respectively for $t = 10.00$. The obtained result shows that the bubble shape and terminal velocity agrees well to experimental data.

It is important to note that the liquid film for the Taylor bubble simulations adds significantly the difficult of handling the mesh in the liquid film. At our simulations, the film thickness of the rising of the Taylor bubbles presented errors of $4 - 6\%$ compared to the correlation of [49].

5.4. Rising of air bubble in slowly divergent channel

The rising of an confined air bubble in a slowing divergent channel is numerically simulated in the current test case to demonstrate the moving boundary



(a)

(b)

Figure 11: Rising of an air Taylor bubble immersed in a ethylene-glycol solution with dimensionless pair numbers set to Mo $= 10^{-6}$,Eo $= 30$. (a) The time evolution of the Bubble's center of mass velocity is compared to the terminal bubble's velocity found in [48]. Velocity and time are non-dimensional. (b) Numerical mesh detailed for the tail of the air bubble at time $t = 2.48$.

technique proposed in this work. The divergent section is described by the following equation:

$$r_{wall}^{n+1} = \frac{\mathrm{D}}{2} + \frac{4.5}{100}\big(x + x_{ref}^{n+1} - \xi\big) \qquad \text{if} \quad 10 \leq x + x_{ref}^{n+1} \leq 30 \tag{49}$$

where D = 1 stands for the initial channel diameter, x is the axial coordinate, x_{ref} is the bubble referential point and ξ is the axial location of the initial divergent section. The new boundary node position r_{wall}^{n+1} is computed every time step as function of the bubble's referential point, thus the result is replaced in Eq. (42) to compute the radial velocity component v which will be used to move points using Eq. (43). Bubble's center of mass remains fixed in space and its referential point x_{ref} increments each time step n with axial bubble's center of mass velocity u_{ref} according to $x_{ref}^{n+1} = x_{ref}^{n} + u_{ref}dt$. The initial referential point is $x_{ref}^{0} = 0$. Flow and geometry move backward in the direction of gravity. In this test case, the divergent sections starts in $\xi = x = 10$ for a unit channel diameter D = 1 and ends at $x = 30$ when the channel diameters reaches D = 2.8, followed by a straight constant section with D = 2.8 until the end of the simulation with $x = 35$. The initial effective numerical domain length is L = 8D. Note that the first straight channel has length L = 10D and thus the initial effective numerical section has constant section between both ends. During all the simulation, the length L = 8D was kept constant, while the diameter of the channel changed according to Eq. (49).

Figure 12 shows a schematic representation of the domain geometry and initial bubble's location for this proposed test case. The simulation parameters were $\mu_{in} = 10^{-5}$Pa.s for dynamic viscosity of the gas phase, $\mu_{out} = 10^{-3}$Pa.s for dynamic viscosity of the liquid phase, while density of gas and liquid were $\rho_{in} = 1.225$kg/m^3 and $\rho_{out} = 1000$kg/m^3 respectively. The non-dimensional numbers were set to Mo = 10^{-5} and Eo = 40, resulting in N = 80000 as the Archimedes number. The Lagrangian velocity is set to zero to avoid backward mesh nodes displacement since the fluid flow is outflowing in the direction of gravity. Due to the moving frame, where the bubble's center of mass remains fixed in space, the parameter β_2 is set to zero since the two-phase interface does not push and pull neighbor nodes next to it. The Laplacian smooth parameter β_3 is kept to unit to maintain inner and outer meshes with smooth node distribution. The interface tangential velocity is not removed in the present test case (γ_1), allowing the free motion of the interface nodes along the interface path. The interface Laplacian smooth parameter γ_2 is set as the previous test cases. Therefore, the mesh parameter were set to $\beta_1 = 0.0$, $\beta_2 = 1.0$, $\beta_3 = 1.0$, $\gamma_2 = 0.0$, $\gamma_2 = 0.1$. The initial mesh node number was 7648 vertex nodes while the total initial number of triangle elements was 44505. Since the numerical domain width was increasing with time, the simulation ended up with 9739 nodes and 56957 triangle elements.

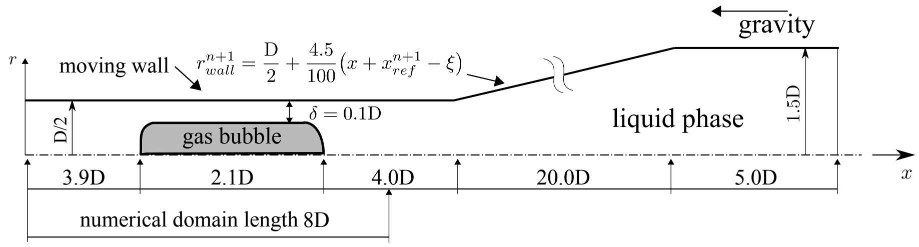

gravity
r
moving wall
$r^{n+1}_{wall} = \frac{D}{2} + \frac{4.5}{100}(x + x^{n+1}_{ref} - \xi)$
$\delta = 0.1$D
liquid phase
1.5D
D/2
gas bubble
x
3.9D
2.1D
4.0D
20.0D
5.0D
numerical domain length 8D

Figure 12: Schematic representation of the whole numerical domain geometry and initial bubble's location for the rising bubble in slowing divergent channel using the moving boundary technique for two-phase flows. The initial effective numerical domain was set to 0.5D × 8D wide.

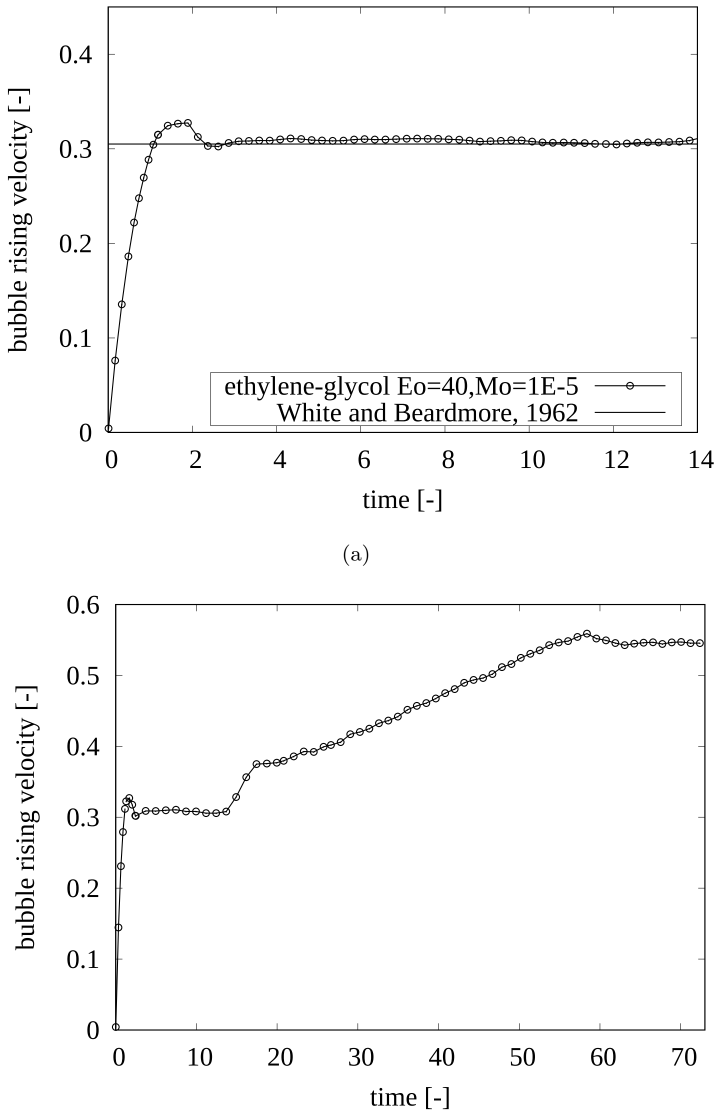
Figure (13) shows the bubble's center of mass nondimensional velocity at initial time steps $0 \leq t \leq 14$ at the initial straight section of the proposed simulation. Such an initial dynamics can be also compared to [48] as the previous section. For such, a numerical bubble's rising velocity of approximately 0.30 was found after $t = 4$ which is in accordance to the experiments of White and Beardmore. Moreover, the complete simulation of the rising of an air bubble can be seen from time $0 \leq t \leq 82.00$. As can be noted, its velocity increases when the divergent section begins and stabilizes again in a fixed value when reaches the second and last straight section. It is noted that from time $16 \leq t \leq 20$ the bubble's accelerates fast due to the initial divergent section where a stronger pressure gradient is identified. From time $t = 20$ to $t = 58$, the air bubble keeps accelerating due to the continuous expansion of the channel, however with a lower acceleration compared to the initial divergent section. From $t = 60$ and on, air bubble reached the second straight section and its non-dimensional axial velocity stabilizes at $u = 0.54$ for channel diameter D = 2.8.

The bubble's shape evolution with time for five different time steps are shown in Fig. (14). It can be confirmed that the inclined moving wall is reaching bubble's nose in $t \approx 14$. Note that in the moving boundary technique, the bubble's referential point is fixed in space, therefore while the air bubble travels, the channel walls move backward, consequently the diverging section approaches the bubble according to Eq. (49). In $t > 14$ the air bubble flows at the diverging section and it reaches the last straight channel when $t > 58$. It can also be noted that the proposed methodology solves with accuracy the transitory bubble shape, i.e. from Taylor bubble $t < 15$ to spherical cap $t > 15$. The mesh nodes (triangle vertices) and mesh elements at initial time step were 7648 and 14605 respectively, while at the end of simulation, 13600 mesh nodes and 26425 mesh elements were found for $t = 74.29$. Such an increase in number of nodes and elements can be automatically handled by the proposed method and it is due to the change of channel diameter.

The solution of the proposed numerical simulation using the moving boundary technique described in this work has shown to be extremely useful to simulate large domains with variable cross-sections at very short numerical domain. The complete simulation was carried out with a numerical domain of L = 8D during all simulation and a total simulation distance of $x = 32.66$.

5.5. Motion of drops in periodically constricted capillary

The following simulation is a challenging test case where the motion of drops in a periodic corrugated channel is investigated and compared to experiments found in [50]. Using the moving frame, where drop's remains nearly fixed in space, the moving boundary technique is used to reduce capillary length by moving the walls according to the drop virtual position as in relative motion. The channel wall is sinusoidal and its profile is described by the harmonic wave



(a)

(b)

Figure 13: Bubble's center of mass nondimensional velocity for the rising of an air bubble in a slowly divergent channel using the moving boundary technique. (a) initial time steps $0 \leq t \leq 14$ at the initial straight section where [48] can also be used as comparison to and (b) complete simulation $0 \leq t \leq 82.00$.

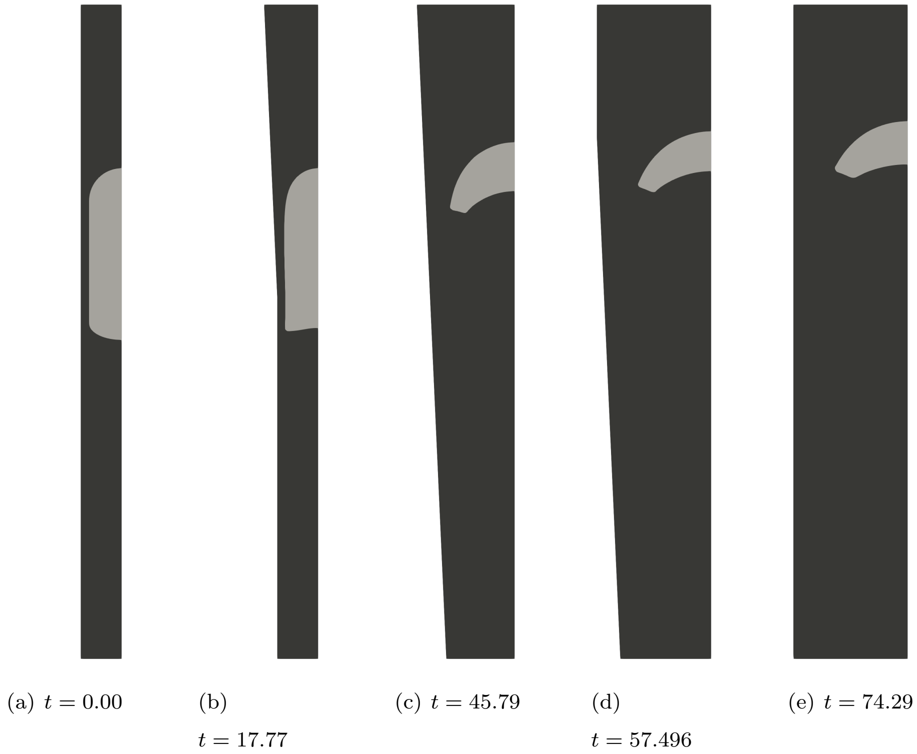

(a) $t = 0.00$ (b) $t = 17.77$ (c) $t = 45.79$ (d) $t = 57.496$ (e) $t = 74.29$

Figure 14: Bubble shape evolution with time an air bubble in a sucrose solution with dimensionless numbers Mo $= 10^{-5}$, Eo $= 40$. (a) Initial bubble shape with $t = 0.0$. (b-d) Bubble shape change during transient solution. (e) $t = 74.29$.

equation:

$$r_{wall}^{n+1} = \frac{\mathrm{D}}{2} + A\mathrm{sin}\left[\frac{2\pi}{\lambda}(x + x_{ref}^{n+1}) - \phi\right] \tag{50}$$

where D is the channel's diameter, A is the wave amplitude, λ is the wavelength and ϕ is its phase. The axial coordinate is x, while x_{ref} refers to the drop's referential point. Drop's nose remains fixed in space and its referential point x_{ref} increments each time step n with axial drop's nose velocity u_{ref} according to $x_{ref}^{n+1} = x_{ref}^{n} + u_{ref}dt$. The initial referential point is $x_{ref}^{0} = 0$. The new boundary node position r_{wall}^{n+1} is computed every time step as function of the drop's referential point, thus the result is replaced in Eq. (42) to compute the

Table 2: Fluid properties for the motion of drops in periodically constricted capillary

system	properties				
	ρ_{out}	ρ_{in}	μ_{out}	μ_{in}	σ
	$[\mathrm{kg/m^3}]$	$[\mathrm{kg/m^3}]$	$[\mu\mathrm{Pa}\cdot\mathrm{s}]$	$[\mu\mathrm{Pa}\cdot\mathrm{s}]$	$[\mathrm{mN/m}]$
GW3	1250	995	450	530	10.5
DEGG3	1110	995	28	530	3.2
DEGG12	1160	966	87	115	4.2

radial component velocity v which will be used to move points using Eq. (43). We have chosen to simulate three different systems namely DEGG3, DEGG12 and GW3 with geometric parameters set to A = 0.07, $\lambda = 4$ and $\phi = 0$. The fluids properties are found in Table (2) and the non-dimensional numbers are easily computed using Eqs. (4) and (5).

A schematic representation of the domain geometry and the simulation parameters is shown in Fig. (15) where the drop's position, the numerical domain length and the harmonic wave equation used to move the upper domain boundary are shown. As can be seen, the drop's center was placed 2.4D behind the front of the channel to avoid inflow disturbances. The numerical length was set to 8D to enable full drop's wake description. The drop's radius R_d was computed as a function of the κ_d parameter and the channel diameter as $\mathrm{R}_d = \kappa_d \mathrm{D}/2$. Several κ_d parameters have been used as the reader will find along this section.

In these corrugated tests cases, the mesh parameters β and γ were chosen according to the previous test case in Sec. (5.4). Since boundaries are in constant motion, the Laplacian smoothing parameter β_3 is an important setting to keep mesh with elastic capabilities. Therefore, the mesh parameter were set to $\beta_1 =$

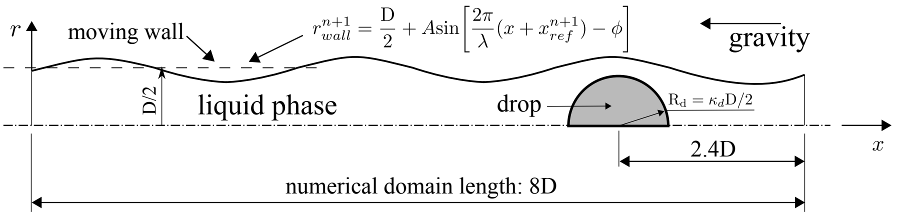

r
moving wall
$r_{wall}^{n+1} = \frac{D}{2} + A\sin\left[\frac{2\pi}{\lambda}(x + x_{ref}^{n+1}) - \phi\right]$
gravity
D/2
liquid phase
drop
$R_d = \kappa_d D/2$
2.4D
x
numerical domain length: 8D

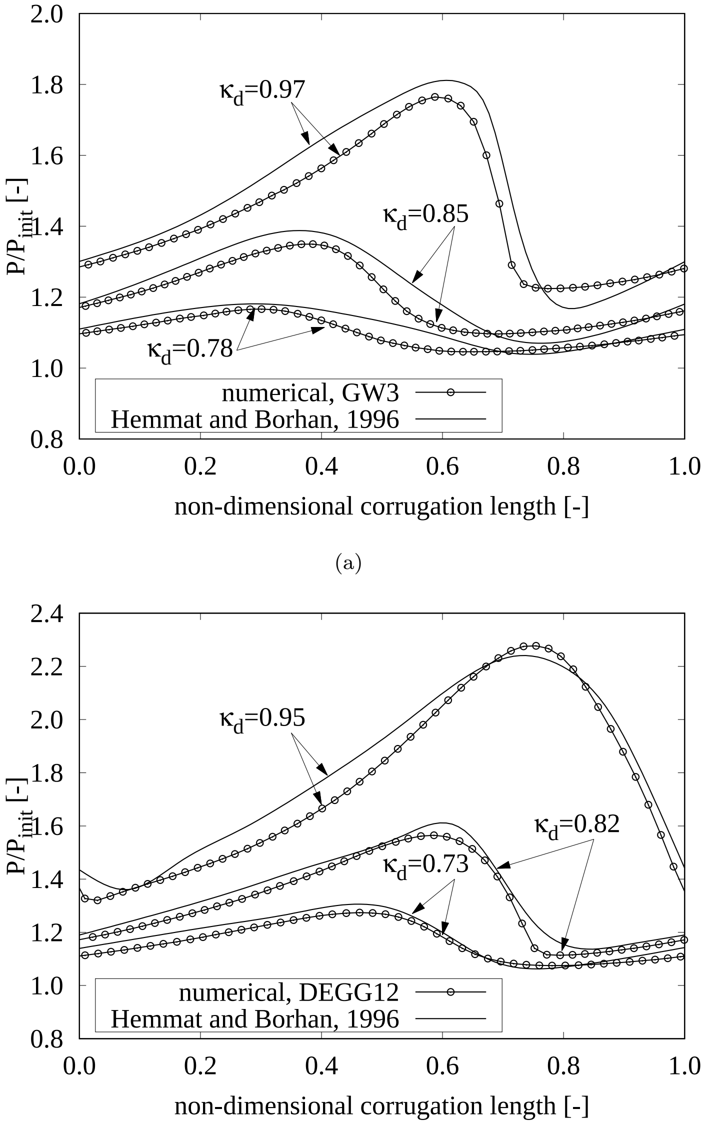
Figure 15: Schematic representation of the domain geometry and initial drop's location for the motion of drop in corrugated channel using the moving boundary technique for two-phase flows.

0.0, $\beta_2 = 1.0$, $\beta_3 = 1.0$, $\gamma_1 = 0.0$, $\gamma_2 = 0.1$.

Figure (16) shows the numerical results compared to the experimental data from [50] for two different fluids: DEGG12 and GW3 with properties found in Table (2). The x-axis is the non-dimensional corrugated length and the y-axis is the ratio of the computed instantaneous perimeter P and its initial perimeter P_{init} when simulation starts. The numerical results show good agreement to the experiments where the curves follow the same trend with experiments. It has been noted at our numerical simulations that the evolution of the ratio P/P_{init} from non-dimensional corrugated length 0.0 to approximately 0.6 of $\kappa_d = 97$ (GW3) and $\kappa_d = 0.95$ (DEGG12) does not obey a linear trend as found in the experiments results. This could be partly explained by the uncertainties in the experimental data.

In Fig. (17) we present numerical results for the motion of drop in DEGG3 system in corrugated channel for one corrugation length versus the ratio of axisymmetric drop perimeter for three different κ_d parameters: $\kappa_d = \{0.70, 0.80, 0.90\}$. These numerical simulations used the same geometry and mesh strategies adopted in the previous cases.

The drop's shape evolution with time for five different time steps for the GW3 system at one corrugated period are shown in Fig. (18). As can be seen, the sinusoidal moving walls are different from one to another figure while the drop's nose remains fixed in space. While the drop travels, the moving boundaries moves downward according to the harmonic wave equation Eq. (50). One of the



(a)

(b)

Figure 16: Comparison of the numerical results of the drop motion in corrugated channel with experiments found in [50] for one corrugated length for fluids: (a) GW3 and (b) DEGG12.

great advantages of the proposed moving boundary methodology in the context of the ALE-FE method is the reduced requirement for extensive remeshing.

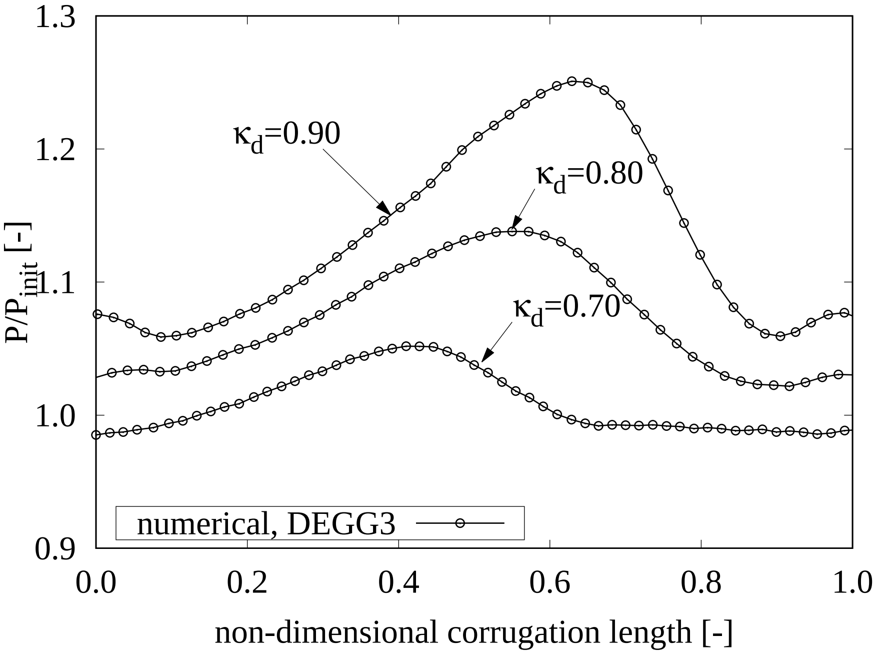
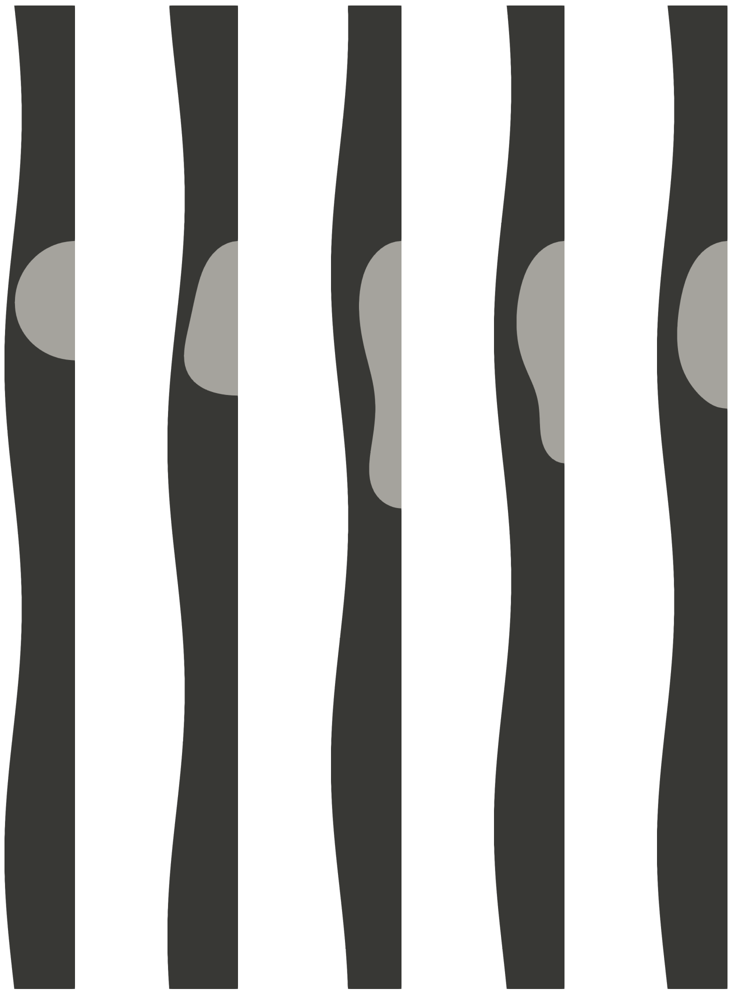


Figure 17: Numerical results for the motion of drop in corrugated channel for one corrugation length versus the ratio of axisymmetric drop perimeter for the system DEGG3.

For this simulation, the number of mesh nodes remained nearly constant. The initial number of mesh nodes and mesh elements were 2400 and 4220 respectively for this system, while at time $t = 1241.15$, where the drop has its maximum extension, the number of mesh nodes and mesh were 2628 and 4684 respectively.

6. Conclusions

In this article, an Arbitrary Lagrangian-Eulerian Finite Element method was presented for the simulation of axisymmetric two-phase flows with dynamic boundaries applied to several gravity-driven problems. This method discretizes the incompressible Navier-Stokes equations with the finite element method on unstructured triangular meshes in complex geometries. An adaptive mesh refinement method was used to accurately resolve the interface motion



(a) $t = 0.00$ (b) $t = 775.34$ (c) $t = 1241.15$ (d) $t = 1294.01$ (e) $t = 1315.14$

Figure 18: Time evolution of the two-phase flow system GW3 (glycerol-water, 96.2 wt%) as found in [50] showing the initial condition (a) $t = 0.00$ and one full corrugation length passage for non-dimensional time (b) $t = 775.34$, (c) $t = 1241.15$, (d) $t = 1294.01$ and (e) $t = 1315.14$.

and remeshing was performed to preserve the quality of the mesh elements. Using a discrete version of the Frenet-Serret formula for the curvature calculation was proved to be a good choice, as high accuracy could be achieved at very low computational cost. Validation of the method was presented for two-phase flow problems including the sessile test case and rising of single bubbles and drops where the numerical results were compared to exact solutions and experimental data of several authors. Moreover, highly demanding computational examples such as the flow in slowly divergent channel and in a periodically constricted capillary was proven to be easily addressed with the proposed moving boundary technique. Its extension to 3-dimensional flows is straightforward and follows the same procedure as detailed in this work. Additionally, any other method using the ALE equations such as the finite volume method can be rewritten to embed such a moving boundary capability since the proposed technique is based on the ALE formulation and it is not dependent on the discretization method. Based on the presented results, the current approach was shown to be an accurate simulation tool for capillary two-phase flows in periodic and large domains.

7. Acknowledgments

The authors thank FAPERJ (Research Support Foundation of the State of Rio de Janeiro) and CNPq (National Council for Scientific and Technological Development) its financial support. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 778104.

References

[1] G. Litak, G. Gorski, R. Mosdorf, A. Rysak, Study of dynamics of two-phase flow through a minichannel by means of recurrences, Mechanical Systems and Signal Processing 89 (2017) 48–57.

[2] A. Masud, T. Hughes, A space-time galerkin/least-squares finite element formulation of the navier-stokes equations for moving domain problems, Computer Methdos in Applied Mechancal Engineering 146 (1997) 91–126.

[3] T. Tezduyar, M. Behr, J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces - the deforming-spatial-domain/space-time procedure: I. the concept and the preliminary tests, Computer Methdos in Applied Mechancal Engineering 94 (3) (1992) 339–351.

[4] H. Sauerland, An xfem based sharp interface approach for two-phase and free-surface flows, Ph.D. thesis, RWTH Aachen (2013).

[5] C. Hirt, B. Nichols, Volume of fluid (vof) method for the dynamics of free boundaries, Journal of Computational Physics 39 (1981) 201–225.

[6] M. Sussman, P. Smereka, S. Osher, A level-set approach for computing solutions to incompressible two-phase flow, Journal of Computational Physics 114 (1994) 146–159.

[7] A. Ferrari, M. Magnini, J. R. Thome, A flexible coupled level set and volume of fluid (flexclv) method to simulate microscale two-phase flow in non-uniform and unstructured meshes, International Journal of Multiphase Flow 91 (2017) 276 – 295. `doi:https://doi.org/10.1016/j.ijmultiphaseflow.2017.01.017`. URL `http://www.sciencedirect.com/science/article/pii/S0301932216305389`

[8] G. Anjos, A 3d ale finite element method for two-phase flows with phase change, Ph.D. thesis, École Polytechnique Fédérale de Lausanne (July 2012).

[9] S. Quan, D. Schmidt, A moving mesh interface tracking method for 3d incompressible two-phase flows, Journal of Computational Physics 221 (2007) 761–780.

[10] S. Quan, J. Lou, D. Schmidt, Modeling merging and breakup in the moving mesh interface tracking method for multiphase flow simulations, Journal of Computational Physics 228 (2009) 2660–2675.

[11] C. Murea, Arbitrary lagrangian eulerian approximation with remeshing for navier-stokes equations, Int. J. Numer. Meth. Biomed. Engng. 26 (2010) 1435–1448.

[12] F. Montefuscolo, F. Souza, G. C. Buscaglia, High-order ale schemes for incompressible capillary flows, Journal of Computational Physics 278 (2014) 133–147.

[13] T. Lee, M. Baines, M. Langdon, S. Tindall, A moving mesh approach for modelling avascular tumour growth, Applied Numerical Mathematics 72 (1) (2013) 99–114.

[14] S. Ganesan, L. Tobiska, A coupled arbitrary lagrangian-eulerian and lagrangian method for computation of free surface flows with insoluble surfactants, Journal of Computational Physics 228 (2009) 2859–2873.

[15] S. Ganesan, L. Tobiska, Arbitrary lagrangian-eulerian finite element method for computation of two-phase flows with soluble surfactants, Journal of Computational Physics 231 (2012) 3685–3702.

[16] M. Baines, M. Hubbard, P. Jimack, Velocity-based moving mesh methods for nonlinear partial differential equations, Communications of Computational Physics 10 (2011) 509–576.

[17] S. Elgeti, H. Sauerland, Deforming fluid domains within the finite element method: Five mesh-based tracking methods in comparison, Archives of Computational Methods in Engineering 23 (2) (2016) 323–361. `doi:10.1007/s11831-015-9143-2`.
URL `https://doi.org/10.1007/s11831-015-9143-2`

[18] E. Gros, G. R. Anjos, J. R. Thome, Interface-fitted moving mesh method for axisymmetric two-phase flow in microchannels, International Journal for Numerical Methods in Fluids (2017) 201–217Fld.4413. `doi:10.1002/fld.4413`.
URL `http://dx.doi.org/10.1002/fld.4413`

[19] J. Brackbill, D. Kothe, A continuum method for modeling surface tension, Journal of Computational Physics 100 (1992) 335–354.

[20] E. Kreyszig, Differential Geometry, 1st Edition, Dover, New York, 1991.

[21] A. J. Chorin, Numerical solution of the navier-stokes equations, Mathematics of Computation 22 (1968) 745–762.

[22] A. Brooks, T. Hughes, Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations, Computer Methods in Applied Mechanics and Engineering 32 (1982) 199–259.

[23] Front Matter, John Wiley & Sons, Ltd, 2012, pp. i–xix. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118483565.fmatter, doi:10.1002/9781118483565.fmatter.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118483565.fmatter

[24] J. Lee, J. Peraire, O. Zienkiewicz, The characteristic-galerkin method for advection-dominated problems—an assessment, Computer Methods in Applied Mechanics and Engineering 61 (3) (1987) 359 – 369. doi:https://doi.org/10.1016/0045-7825(87)90100-9.
URL http://www.sciencedirect.com/science/article/pii/0045782587901009

[25] F. Souza, N. Mangiavacchi, A lagrangian level-set approach for the simulation of incompressible two-fluid flows, International Journal for Numerical Methods in Fluids 47 (2004) 1393–1401.

[26] G. Anjos, N. Mangiavacchi, N. Borhani, J. Thome, 3d ale finite-element method for two-phase flows with phase change, Heat Transfer Engineering 35 (5) (2013) 537–547. doi:10.1080/01457632.2013.833407.

[27] A. Wiin-Nielsen, On the application of trajectory methods in numerical forecasting, Tellus 11 (1959) 180–196.

[28] T. Krishnamurti, Numerical integration of primitive equations by a quasi-lagrangian advective scheme, Journal of applied Meteorology 1 (1962) 508–521.

[29] J. Sawyer, A semi-lagrangian method of solving the vorticity advection equation, Tellus 15 (1963) 336–342.

[30] A. Robert, A stable numerical integration scheme for the primitive meteorological equations, Atmosphere Oceans 19 (1981) 35–46.

[31] O. Pironneau, On the transport-diffusion algorithm and its applications to the navier-stokes equation, Numerische Mathematik 38 (1982) 309–332.

[32] R. Löhner, Applied Computational Fluid Dynamics Techniques: an Introduction Based on Finite Element Methods, 2nd Edition, John Wiley and Sons, 2008.

[33] R. Aubry, R. Löhner, On the 'most normal' normal, Communications in Numerical Methods in Engineering 24 (2008) 1641–1652.

[34] J. Donea, A taylor-galerkin method for convective transport problems, Int. J. Num. Methods Eng. 20 (1984) 101–19.

[35] A. Fortuna, Técnicas Computacionais para Dinâmica dos Fluidos, 1st Edition, edUSP, 2000.

[36] I. Harari, T. J. Hughes, Finite element methods for the helmholtz equation in an exterior domain: Model problems, Computer Methods in Applied Mechanics and Engineering 87 (1) (1991) 59 – 96. doi:https://doi.org/10.1016/0045-7825(91)90146-W. URL http://www.sciencedirect.com/science/article/pii/004578259190146W

[37] F. Ihlenburg, I. Babuška, Finite element solution of the helmholtz equation with high wave number part i: The h-version of the fem, Computers & Mathematics with Applications 30 (9) (1995) 9 – 37.

doi:https://doi.org/10.1016/0898-1221(95)00144-N.
URL http://www.sciencedirect.com/science/article/pii/089812219500144N

[38] W. Gordon, L. Thiel, Numerical grid generation, edited by J.F. Thompson (1982).

[39] M. Desbrun, M. Meyer, P. Schröder, A. Barr, Implicit fairing of irregular meshes using diffusion and curvature flow, in: Proceedings of Siggraph, 1999, pp. 317–324.

[40] J. Glimm, J. Grove, W. Lindquist, O. McBryan, G. Tryggvadson, The bifurcation of tracked scalar waves, SIAM Journal of Computations 9 (1) (1988) 61–79.

[41] S. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, Journal of Computational Physics 100 (1992) 25–37.

[42] J. Hua, J. Stene, P. Lin, Numerical simulation of 3d bubbles rising in viscous liquids using a front tracking method, Journal of Computational Physics 227 (6) (2008) 3358–3382.

[43] V. Jayaraman, H. Udaykumar, W. Shyy, Adaptive unstructured grid for three-dimensional interface representation, Numerical Heat Transfer, Part B: Fundamentals 32 (1997) 247–265.

[44] T. Tezduyar, Finite element methods for flow problems with moving boundaries and interfaces, Archives of Computational Methods in Engineering 8 (2) (2001) 83–130.

[45] E. Gros, Numerical modelling of two-phase flow with moving boundary fitted meshes, Ph.D. thesis, École Polytechnique Fédérale de Lausanne (June 2018).

[46] D. Bhaga, M. Weber, Bubbles in viscous liquids: Shapes, wakes and velocities, Journal of Fluid Mechanics 105 (1981) 61–85.

[47] J. Bugg, K. Mack, K. S. Rezkallah, A numerical model of taylor bubbles rising through stagnant liquids in vertical tubes, International Journal of Multiphase Flow 24 (1998) 271–281.

[48] E. White, R. Beardmore, The velocity of rise of single cylindrical air bubbles through liquids contained in vertical tubes, Chemical Engineering Science 17 (1962) 351–361.

[49] R. Brown, The mechanism of large bubbles intubes. i. bubbles velocities in stagnant liquids, Canadian Journal of Chemical Engineering 43 (1965) 217–223.

[50] M. Hemmat, A. Borhan, Buoyancy-driven motion of drops and bubbles in a periodically constricted capillary, Chemical Engineering Communications 148-150 (1) (1996) 363–384. `doi:10.1080/00986449608936525`.