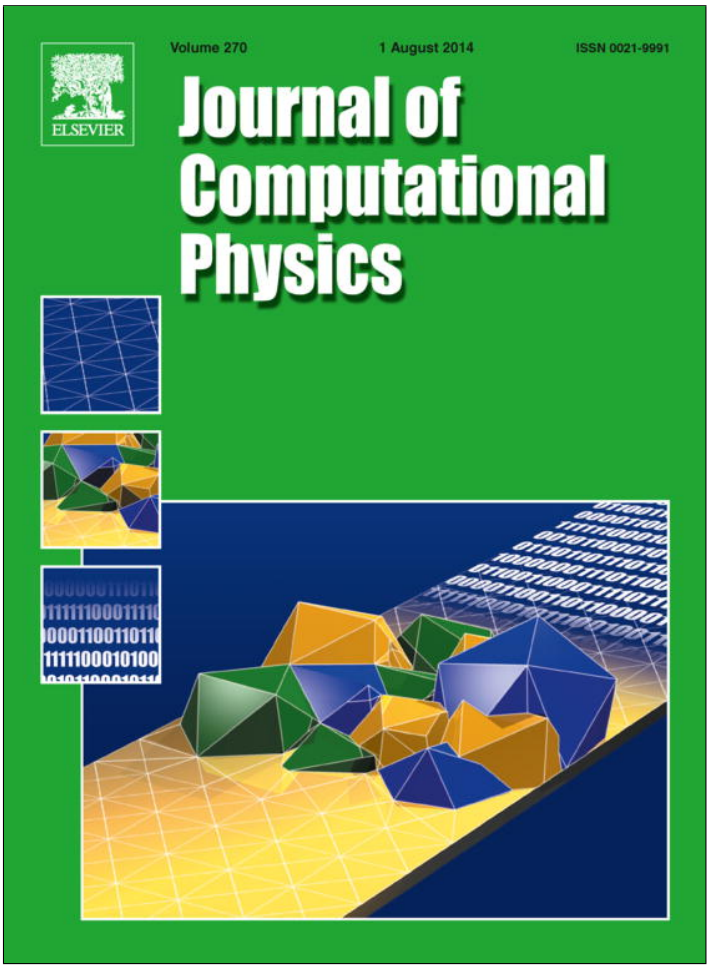
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.




This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/authorsrights

Author's personal copy

Journal of Computational Physics 270 (2014) 366–377

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

# A 3D moving mesh Finite Element Method for two-phase flows

G.R. Anjos [a,*], N. Borhani [a], N. Mangiavacchi [b], J.R. Thome [a]

[a] EPFL STI IGM LTCM, ME B1 345, Station 9, CH-1015 Lausanne, Switzerland
[b] Departamento de Engenharia Mecânica, Universidade do Estado do Rio de Janeiro, Rua São Francisco Xavier, 524, 20550-900, Rio de Janeiro, RJ, Brazil

ARTICLE INFO

*Article history:*
Received 8 August 2013
Received in revised form 20 February 2014
Accepted 31 March 2014
Available online 8 April 2014

*Keywords:*
Two-Phase flows
Surface tension
Curvature
Arbitrary Lagrangian–Eulerian
Finite Element Method
Adaptive mesh refinement

ABSTRACT

A 3D ALE Finite Element Method is developed to study two-phase flow phenomena using a new discretization method to compute the surface tension forces. The computational method is based on the Arbitrary Lagrangian–Eulerian formulation (ALE) and the Finite Element Method (FEM), creating a two-phase method with an improved model for the liquid–gas interface. An adaptive mesh update procedure is also proposed for effective management of the mesh to remove, add and repair elements, since the computational mesh nodes move according to the flow. The ALE description explicitly defines the two-phase interface position by a set of interconnected nodes which ensures a sharp representation of the boundary, including the role of the surface tension. The proposed methodology for computing the curvature leads to accurate results with moderate programming effort and computational cost. Static and dynamic tests have been carried out to validate the method and the results have compared well to analytical solutions and experimental results found in the literature, demonstrating that the new proposed methodology provides good accuracy to describe the interfacial forces and bubble dynamics. This paper focuses on the description of the proposed methodology, with particular emphasis on the discretization of the surface tension force, the new remeshing technique, and the validation results. Additionally, a microchannel simulation in complex geometry is presented for two elongated bubbles.

© 2014 Elsevier Inc. All rights reserved.

## 1. Introduction

Fluid flow can be expressed by two separate frameworks, namely *Lagrangian* and *Eulerian*. The former employs a referential frame moving with the flow and the latter is based on a fixed referential frame where the continuum medium moves with respect to the mesh nodes. Both frames are useful to describe fluid motion but, depending on the case, one may have more advantages than the other. The well-known *Eulerian* algorithms are widely used in fluid dynamics and can handle large deformations due to the fluid motion, but due to their fixed mesh description, the mesh resolution must be dense enough to capture all the desired flow scales. On the other hand, *Lagrangian* algorithms use the material frame to track the fluid motion. Each single nodal point of the frame moves with the velocity of the flow, thus avoiding the intrinsic numerical diffusion present in the discretization of the convective term. This approach's weakness lies in the fact that for large shearing deformations, a frequent remeshing is required, thus increasing dramatically the required computational resources. The

* Corresponding author.
*E-mail address:* gustavo.anjos@uerj.br (G.R. Anjos).

http://dx.doi.org/10.1016/j.jcp.2014.03.067
0021-9991/© 2014 Elsevier Inc. All rights reserved.

Author's personal copy



Arbitrary Lagrangian–Eulerian framework is a combined generalization of these two frameworks (see Howard et al. [1]). The conservation equations are re-written in a frame moving with an arbitrary velocity which may or may not coincide with the flow velocity. Such a combination benefits from the best aspects of each framework, thus avoiding the obstacles of a purely *Eulerian* or purely *Lagrangian* implementation.

In two-phase flow, these formulations have a strong influence on the modeling of the interface between the phases. In the *Eulerian* formulation, also called *interface capturing*, the interface is not explicitly described, but it is defined by special color functions which are advected by an additional transport equation. In the 1980s, Hirt and Nichols [2] proposed the *Volume of Fluid* method, or VOF, which consists on the description of the position of the interface by the volume ratio occupied by each phase in the computational cells. The Level-Set method in fluid mechanics, first presented by Sussman et al. [3], has become an important tool for two-phase flow modeling. The interface in this case is represented by the zero-level of the distance function that is advected by the flow field. The discretization of the Level-Set functions may lead to excessive numerical diffusion, thus requiring its frequent reinitialization and complex advection schemes to prevent such a problem. To overcome the drawbacks found in these methodologies, a hybrid method was proposed by Bourlioux [4] and Sussman and Puckett [5] where the curvature and the normal vectors are computed by the Level-Set functions while the interface is captured by the VOF function.

Unlike VOF and Level-Set methods, the *Lagrangian* description defines the interface between phases explicitly by computational elements called *interface tracking*. Among the Lagrangian methods, *volume-tracking* and *front-tracking* are the most widely used. The former uses marker particles for the reconstruction of the interface, combining precision and accuracy with a relatively low implementation investment [6,7]. The latter, first implemented by Glimm and co-authors [8], represent the interface with a set of interconnected nodes which move according to the fluid flow calculated in an *Eulerian* way. Such a description allows a sharp representation of the interface with high accuracy, but its drawback is the need of an explicit treatment of topological changes in the interface, such as for the case of coalescence and break-up of bubbles or drops. The group at the University of Massachusetts has suggested a complete moving mesh technique to simulate free-surface and two-phase flows [9,10] which differs from the fixed background mesh approach.

A front tracking technique using moving-least-squares to represent the interface between the fluids is proposed by Gois et al. [11]. The main difference suggested by the authors is the representation of the interface itself, for which no connectivity is required for the surface points, thus avoiding the complexity of the interface remeshing. In [12] is presented an Arbitrary Lagrangian–Eulerian formulation and the finite element method for 2-dimensional free surface flows. They discretized the numerical domain by employing the *MINI* triangular element. Furthermore, they implemented a robust node-repositioning algorithm to address the high distortion issues triggered by the moving mesh methodology. Recently, a numerical code has been developed to compute axisymmetric two-phase flows with a soluble surface in [13]. The conservation equations are discretized through the finite element method using an Arbitrary Lagrangian–Eulerian framework. A continuous piecewise quadratic polynomial enriched by a cubic bubble function is used to discretize velocity and a discontinuous piecewise linear expression is used to discretize pressure. Such an element combination enforces the stability criteria from the *Ladyshenskaya–Babuska–Brezzi* (LBB) condition [14,15].

Even though there are methods to handle 3D multiphase flows, this topic continues to be a challenging research area due to the difficulties in dealing with real applications. One such an application is two-phase flow in microchannels, where physical properties of the vapor and liquid phases are distinct by some order of magnitude, the vapor bubbles approach the channel wall and produce very thin films, surface tension effects play a major role and all these factors need to be accurately accounted for. In this work we propose a "one-fluid" moving mesh methodology, using the Arbitrary Lagrangian Eulerian framework and the Finite Element Method, to simulate 3-dimensional two-phase flows as an extension of the front-tracking codes cited above. An unstructured tetrahedron mesh is used to discretize the governing equations where the interface between the fluids is part of the volumetric mesh, represented by a set of nodes, edges and triangular faces advected by the fluid flow. A dynamic mesh control is performed to keep the finite elements bounded within an acceptable aspect ratio. This proposed new methodology is detailed in the following sections and then several test cases are used to evaluate its accuracy and then applied to a two-phase flow in microchannel.

## 2. Equations set

The Arbitrary Lagrangian Eulerian (ALE) vector form of the non-dimensional incompressible conservation equations is shown below, including the surface tension term $\mathbf{f}$ and the gravity term $\mathbf{g}$ which distinguish this formulation from the classical single phase form:

$$\rho(\phi)\left[\frac{\partial \mathbf{u}}{\partial t} + \mathbf{c}\cdot\nabla\mathbf{u}\right] = -\nabla p + \frac{1}{\mathrm{N}^{1/2}}\nabla\cdot\left[\mu(\phi)\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)\right] + \rho(\phi)\mathbf{g} + \frac{1}{\mathrm{Eo}}\mathbf{f} \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{2}$$

On the left hand side of Eq. (1), the convective velocity $\mathbf{c}$ represents the relative velocity between the flow field and the mesh, given by the following expression: $\mathbf{c} = \mathbf{u} - \hat{\mathbf{u}}$ where $\mathbf{u}$ stands for the computed flow field velocity and $\hat{\mathbf{u}}$ for the mesh

Author's personal copy



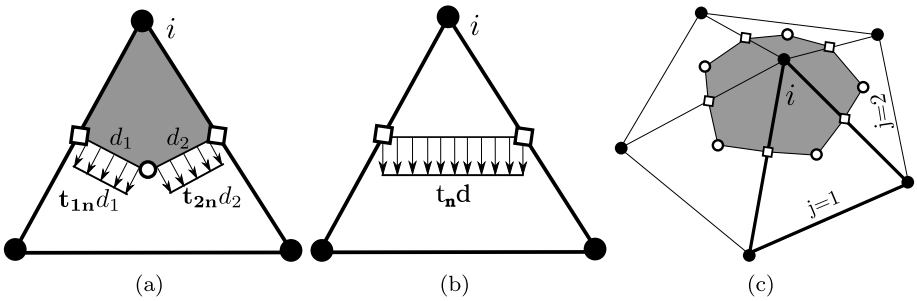



**Fig. 1.** (a) An elemental force estimation is done using the sum of the distributed forces $\mathbf{t_{1n}}d_1$ and $\mathbf{t_{2n}}d_2$. (b) An estimation of the node mean curvature is found by dividing the module of the sum of the calculated distributed forces ($|\mathbf{t_n}d|$) by the sum of the barycentric areas $A_j$. (c) Barycentric area of the 1-ring triangles connected to $i$.

velocity. Pressure is represented by $p$ and time by $t$. The fluid density $\rho(\phi)$ and viscosity $\mu(\phi)$ are defined as constant in each phase. The non-dimensional groups *Archimedes*, *Eötvös* and *Morton* are defined as:

$$\mathrm{N} = \frac{\rho_0^2 D^3 g}{\mu_0^2}, \qquad \mathrm{Eo} = \frac{\rho_0 g D^2}{\sigma_0}, \qquad \mathrm{Mo} = \frac{(\rho_l - \rho_g)\mu_0^4 g}{\rho_0^2 \sigma_0^3} \tag{3}$$

In the ALE formulation, the mesh velocity $\hat{\mathbf{u}}$ defines the new position of the mesh nodes in each iteration. A pure *Eulerian* description may be easily recovered when the mesh velocity $\hat{\mathbf{u}} = 0$. On the other hand, the fully *Lagrangian* description is reached when the mesh and the flow velocities coincide, mathematically meaning that $\hat{\mathbf{u}} = \mathbf{u}$. In the intermediary case, when $\hat{\mathbf{u}}$ assumes a non-zero arbitrary velocity, not equal to the flow velocity in general, the arbitrary framework (ALE) is obtained. Note that for two-phase flows it is convenient to enforce that the interface nodes laying between the phases should move according to the flow field velocity.

## 3. The discrete interface

The interface mesh implemented in this work is part of the tetrahedron domain mesh, where each interfacial triangle is a face of two adjacent tetrahedral elements. Combining ALE and FEM a sharp transition of properties is thus successfully achieved that does not require the use of any smoothing functions, consequently assuring accuracy on the balance of forces close to the interface. According to [16], the discrete surface tension force is given by $\mathbf{f} = \kappa \nabla H$, where the distributed interface force $\mathbf{f}$ is a volume force and its intensity $\kappa$ is calculated and applied on the direction of the gradient of a linear Heaviside function $\nabla H$ where $H$ is defined as 0 in one phase, 1 in the other phase and 0.5 to the vertices belonging to the interface. The mean curvature $\kappa$ is calculated at each interface node, where the nodal mean curvature $k_i$ is approximated by the elemental force integral over the 1-ring triangle neighbors divided by the corresponding barycentric area.

To calculate the nodal mean curvature $\kappa_i$, one locates the two unit normal vectors for each triangle connected to $i$, and then integrates them on the segments $d_1$ and $d_2$ as shown in Fig. 1(a). The segments $d_1$ and $d_2$ connect the triangle mid-edges to the triangle centroid. From the Stokes theorem, the distributed elemental force $\mathbf{t_n}d$ in Fig. 1(b) is equivalent to the integral of $\mathbf{t_n} \cdot \mathbf{n}$ over $d_1 + d_2$. Therefore, an estimation of nodal mean curvature $k_i$ is found by integrating the intensity of elemental force $\mathbf{t_n}d$ in the 1-ring triangle neighbors and weighting by the sum of the barycentric areas $A_j$ (Fig. 1(c)). Mathematically, $\kappa_i$ is calculated as follow:

$$\kappa_i = \frac{|\sum_{j=1}^{m} (\mathbf{t_n}d)_j|}{\sum_{j=1}^{m} A_j} \tag{4}$$

where $m$ is the number of element neighbors of $i$. The normal vector of the node $\mathbf{n}_i$ is approximated by the sum of the cross product of two the vectors $\mathbf{t_1}$ and $\mathbf{t_2}$, considering the 1-ring neighbor nodes consistently sorted. A similar approach of curvature estimation may be found in [17].

## 4. Dynamic mesh control

A proper choice of the mesh velocity and the meshing strategy is necessary to avoid the fast degradation of the computational elements and the onset of non-desirable numerical instabilities found in the pure *Lagrangian* framework. A local remeshing technique has been implemented, bounding the element aspect ratios within a satisfactory level with no significant time cost in two-phase flows, and is described below.

The equations are discretized over an unstructured non-regular Delaunay tetrahedral mesh. Two sets of data are stored during the simulation which are treated separately: the volumetric nodes and the surface mesh. The latter consists of two parts, the interface between the phases and the domain boundary, both created simultaneously by the software (GMesh, [18]). The code is then linked to a tetrahedral mesh generator (TETGEN, [19]) that uses the previous generated surface meshes and a volumetric set of nodes as input parameters. The initial volumetric node distribution may be set manually,

Author's personal copy



according to the requirements of the simulation, or by TETGEN, which may create a smooth distribution of nodes according to the edge lengths of the given surface meshes. Thus, the 3-dimensional connectivity array is exported to the code. Note that due to the separate treatment of the surface mesh and the volumetric nodes, each set of data must be handled on a different way. The volumetric connectivity array is created at every time step by TETGEN through the constrained Delaunay tetrahedralization, which respects the boundary limits of the 3-dimensional numerical domain $\Omega$. To achieve Delaunay properties along the entire mesh, flipping operations are performed, thus leading to better tetrahedron elements while the boundary and interface meshes remain unchanged. During the numerical simulation, the resulting velocity field may deteriorate the quality of the tetrahedrons, and therefore mesh smoothing, by an appropriate mesh velocity, is used to improve their shape and reduce the number of geometrical operations performed on the meshes.

### 4.1. Volumetric points

For each time step, the node distribution of the volumetric mesh is monitored and compared to the previous iteration. The tetrahedron edge length determines whether insertion or deletion is required for a given predefined distance in a specific zone. In this work, the distribution edge lengths $h$ are obtained by the solution of the following Helmholtz's equation: $\nabla^2 h = 1/k(h_b - h)$ where $k$ is a diffusive parameter and $h_b$ is the initial edge length distribution. Thus, the solution corresponds to a smooth distribution of nodes in space. For large values of $k$ in the above equation, the right hand term tends to zero, 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 point distribution $h_b$. Such a solution may require excessive computational resources and its computation is not recommended at every time step; its implementation is described below.

### 4.2. Mesh smoothing

The mesh velocity $\hat{\mathbf{u}}$ 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 a smoothing velocity, in which the latter is defined according to some criteria intended to redistribute the nodes, thus minimizing the number of remeshing steps and avoiding heavy computation requirements. The transfinite mapping method (see [20]) and Laplacian smoothing are examples of mesh-update procedures [17], the latter being chosen for this work.

Due to the separation of the domain and the surface mesh in the above procedure, the mesh distribution treatments may be combined into a scheme and adjusted by parameters varying from 0 to 1, which is a 3-dimensional generalization of the approach presented by Souza and Mangiavacchi [16] for 2-dimensional simulations. The domain and surface velocities are therefore treated as follows:

$$\hat{\mathbf{u}}(\mathbf{x}) = \begin{cases} \beta_1 \mathbf{u} + \beta_2 \mathbf{u}_v + \beta_3 \mathbf{u}_e & \text{if } \mathbf{x} \text{ does not belong to the interface} \\ \mathbf{u} - \gamma_1 (\mathbf{u} \cdot t)t + \gamma_2 (\mathbf{u}_e \cdot t)t & \text{if } \mathbf{x} \text{ belongs to the interface} \end{cases} \tag{5}$$

where $\mathbf{u}$ is the flow velocity, $\mathbf{u}_e$ is the Laplacian smooth velocity and $\mathbf{u}_v$ is the volumetric elastic mesh velocity, defined as a function of the interface velocity:

$$\mathbf{u}_{vi} = \frac{1}{n} \sum_{j \in N_1(j)} \mathbf{u}_j \tag{6}$$

where, $N_1$ is the set of the 1-ring neighbors to the $i$th node with $j$ neighbors. $v_j$ is the velocity associated to the $j$th node and $n$ is the number of mesh neighbors. This scheme has been successfully applied to overcome the fast element distortion close to the interface where the velocity gradient may be high. To decrease the displacement of nodes in the tangential direction, 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 surface mesh velocity $\mathbf{u} - (\mathbf{u} \cdot t)t$. Moreover, the parameter $\beta$ controls the intensity of smoothing velocity in the inner and outer volumetric mesh and $\gamma$ stands for the surface 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 suggest 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 (*beta*'s and *gamma*'s) reduces considerably the frequency of the remeshing operations done in the surface and volumetric meshes.

### 4.3. Surface remeshing

Mesh smoothing itself is not able to keep all elements well-shaped after numerous iterations. The moving front eventually creates a poor distribution of surface nodes which can affect the accuracy of the computed curvature and, consequently, the final solution. Since the connectivity of the surface mesh is handled by the code, a remeshing technique is thus required to keep the surface elements' aspect ratios in a satisfactory range. The technique proposed here consists of changing the

Author's personal copy



connectivity of surface nodes and elements through "flipping" operations. Additionally, insertion and deletion/collapse of nodes are required. Detailed information about these operations can be found in [21].

### 4.4. Volume correction

Excessive geometrical operations may lead to volume variation. A simple treatment has been implemented to compensate for the spurious mass variation, thus avoiding the accumulation of mass conservation errors. This correction is done by moving the surface vertices in the direction of their normal vector. Such a displacement is calculated based on the initial phase 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 surface, the mass error after surface remeshing in one time step was calculated, corresponding to 0.05%, and 21 iterations were required to recover the initial bubble volume for a tolerance on the order of $10^{-8}$. For severe surface remeshing, the mass error may increase and more steps are then necessary to recover the initial bubble's volume.

## 5. Discretization procedure

Consider the following spaces: $V = \{\mathbf{v} \in L^2(\Omega)\colon\ v_i \in H^1(\Omega), \forall i\}$ and $P = \{q \in L^2(\Omega)\colon\ \int_\Omega q\, d\Omega = 0\}$, where $H^1(\Omega)$ is a Sobolev space and the sub-spaces are $V_{u_r} = \{\mathbf{v} \in V\colon\ \mathbf{v}(x) = \mathbf{u}_\Gamma, x \in \Gamma_1\}$ and $V_0 = \{\mathbf{v} \in V\colon\ \mathbf{v}(x) = 0, x \in \Gamma_1\}$. The weak formulation of the problem can be written as: find $\mathbf{u}(x,t) \in V_{\mathbf{u}_\Gamma}$ and $p(x,t) \in P$ such that

$$m\left(\frac{D(\rho\mathbf{u})}{Dt}, \mathbf{w}\right) - G(p, \mathbf{w}) + k\left(\frac{\mu}{\mathrm{N}^{1/2}}, \mathbf{u}, \mathbf{w}\right) - m(\mathbf{g}, \mathbf{w}) - m\left(\frac{1}{\mathrm{Eo}}\mathbf{f}, \mathbf{w}\right) = 0 \tag{7}$$

$$D(q, u) = 0 \tag{8}$$

for all $\mathbf{w} \in V_0$, $q \in P$. The functionals in Eqs. (7) and (8) are the integration of the inertia ($m$), pressure gradient ($G$), velocity divergence ($D$) and viscous term ($k$), respectively. The gravity $\mathbf{g}$ and the surface tension force $\mathbf{f}$ are treated explicitly as source terms in the final set of equations.

The Galerkin method is applied to the variational form of the equations to discretize all terms except the non-linear convective term which is discretized by the semi-Lagrangian method. This method improves the overall stability and allows larger time steps to be used. In the semi-Lagrangian approach, each time step the nodes are moved with the flow velocity, similarly to the Lagrangian approach. Once the task is accomplished, the solution is interpolated onto the new ALE mesh.

The domain is discretized by an unstructured tetrahedral mesh which is initially Delaunay. The shape functions interpolating the discrete approximations are assumed to be linear for the pressure field and of 4th order for velocities, which is called a *mini* element, thus fulfilling the LBB condition.

The integration over the tetrahedral elements results in the following ODE system:

$$\mathbf{M}_\rho \frac{D\mathbf{u}}{Dt} + \frac{1}{\mathrm{N}^{1/2}}\mathbf{K}\mathbf{u} - \mathbf{G}p - \mathbf{M}_\rho\mathbf{g} - \frac{1}{\mathrm{Eo}}\mathbf{M}\mathbf{f} = \mathbf{bc}_1 \tag{9}$$

$$\mathbf{D}\mathbf{u} = \mathbf{bc}_2 \tag{10}$$

In the above equations, the material derivative is integrated by a backward Euler implicit scheme and $\mathbf{bc}_1$ and $\mathbf{bc}_2$ are the boundary conditions for velocity and pressure respectively. The quantities $\mathbf{u}$, $\mathbf{g}$ and $\mathbf{f}$ are vectors of dimension $3s_v$, where $s_v$ is the total number of free nodes in the discretized domain compatible to velocity, and $p$ is a vector of dimension $s_p$, where $s_p$ is the total number of free nodes in the discretized domain compatible to pressure. $\mathbf{M}$, $\mathbf{M}_\rho$, $\mathbf{K}$, $\mathbf{G}$ and $\mathbf{D}$ are the volume, mass, stiffness, gradient and divergence global matrices, respectively.

The ODEs above are solved using a discrete projection method based on the approximated block LU factorization which divides the original matrix into two different block triangular matrices, thus the coupled non-linear system is split into two smaller ones where the pressure and the velocity field are solved separately. This uncoupling method was first introduced by Chorin [22] and has been shown to be suitable for two-phase flow simulations. $\mathbf{B}$ and $\mathbf{r}^n$ are given by:

$$\mathbf{B} = \frac{\mathbf{M}_\rho}{\Delta t} + \frac{\mathbf{K}}{\mathrm{N}^{1/2}}\mathbf{r}^n = \frac{\mathbf{M}_\rho}{\Delta t}\mathbf{u}_d^n \tag{11}$$

where $t$ is time and $\mathbf{u}_d^n$ is the velocity calculated in the previous time step at the departure points. Thus, the complete solution procedure is as follows:

(a) compute the trial velocity $\tilde{\mathbf{u}}$ by solving:

$$\mathbf{B}\tilde{\mathbf{u}} = \mathbf{r}^n + \mathbf{bc_1} \tag{12}$$

(b) update $\tilde{\mathbf{u}}$ while considering the surface and gravity forces:

$$\tilde{\mathbf{u}}_{corr} = \tilde{\mathbf{u}} + \Delta t \mathbf{M}_{\rho L}^{-1}\left(\mathbf{M}_\rho\mathbf{g} + \frac{1}{\mathrm{Eo}}\mathbf{M}\mathbf{f}\right) \tag{13}$$

Author's personal copy



(c) compute $p^{n+1}$ from:

$$\mathbf{D}\mathbf{B}^{-1}\mathbf{G}p^{n+1} = -\mathbf{D}(\tilde{\mathbf{u}}_{corr}) + \mathbf{b}\mathbf{c}_2 \tag{14}$$

(d) find the solution velocity $\mathbf{u}^{n+1}$ from:

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}}_{corr} + \mathbf{B}_L^{-1}\mathbf{G}p^{n+1} \tag{15}$$

The subscript $L$ refers to the *Lumped* matrix, which consists in a diagonal approximation, thus avoiding the computation of the matrix inverse. Note that in Eq. (13) the source terms $\mathbf{f}$ and $\mathbf{g}$ are only included once the trial velocity $\tilde{\mathbf{u}}$ is computed, thus satisfying the balance with pressure. For the sake of consistency, the lumped approximation is also employed in the computation of $\mathbf{f}$, therefore $\mathbf{M}\mathbf{f} \approx \kappa \mathbf{G}H$. The two final linear systems (Eqs. (12) and (14)) are solved at each time step by the conjugate gradient method and the generalized minimum residual method, respectively. The Incomplete Cholesky and LU factorization are used as preconditioners to speed up the convergence of the final solution.

### 5.1. Time step restriction

In the ALE formulation, the convection term is driven by the difference between two velocities, namely $\mathbf{u}$ and $\hat{\mathbf{u}}$. The semi-Lagrangian time step constrain is given by the expression below:

$$\Delta t_{sl} < \min\left(\frac{h}{\mathbf{u} - \hat{\mathbf{u}}}\right) \tag{16}$$

where $h$ is the tetrahedral mesh edge length and $\mathbf{u} - \hat{\mathbf{u}}$ is the velocity used by the semi-Lagrangian method. The ALE time step restriction should be bounded by:

$$\Delta t_m < \min\left(\frac{h}{\hat{\mathbf{u}}}\right) \tag{17}$$

where, $\hat{\mathbf{u}}$ is the velocity of the moving mesh given by Eq. (5). Additionally, the explicit treatment of gravity and surface tension adds a constraint to the final time step. The criteria for both surface tension and gravity may be written as:

$$\Delta t_s < \left[\frac{\rho h^3 \mathrm{Eo}}{2\pi}\right]^{1/2} \quad \text{and} \quad \Delta t_g < \left[\frac{1}{h_{\min}}\right]^{1/2} \tag{18}$$

In the above equations, $\rho$ 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. Thus, the final simulation constraint may be written as:

$$\Delta t \leqslant \min\{\Delta t_m, \Delta t_s, \Delta t_g, \Delta t_{sl}\} \tag{19}$$

### 5.2. Boundary conditions

The domain length must be chosen by taking into consideration the velocity field as observed also by Tomiyama [23] and Hua et al. [24]. In the present simulations, the dimension of the domain was chosen to avoid undesirable wall effects that could potentially affect the final results. Thus, the no-slip condition was applied to all the solid walls by setting all the velocity components to zero. Additionally, inflow and outflow conditions were used for the shear-driven simulation (Section 6.5).

### 5.3. Mesh interpolation and time step iteration

The solution of the linear systems results in the computation of the velocity and pressure fields. Therefore, the remeshing operation for the volumetric and surface meshes is performed and the new connectivity array is imported to the numerical code. An interpolation is required to assign values to the fluid properties, velocity and pressure of these new included nodes. In the present methodology, a linear interpolation is carried out on the new nodes before the next iteration of the time marching algorithm. Results show acceptable accuracy during the transient solution for different test cases. It is well discussed in the literature that these costs may become extremely high, therefore revealing a major issue of the moving mesh method itself. It is indeed costly to operate in both volumetric and surface meshes simultaneously in addition to the required linear interpolation step; however, the code profiling has shown that the remeshing methodology applied to the test cases presented in the next section did not account for more than 30% of the total computational time at each time step. The complete time step procedure can be summarized in Algorithm 1. The relative time consumption for each item in Algorithm 1 in one loop is described as follows: (1) 1.25%, (2) 0.10%, (3) 0.07%, (4) 18.61%, (5) 20.91%, (6) 10.84%, (7) 17.92%, (8) 28.15%, (9) 2.15%.

Author's personal copy



**Algorithm 1** Time marching algorithm.

(1) compute ALE velocity $\hat{\mathbf{u}}(x)^n$ using $\mathbf{u}^n$;
(2) compute time step for $t^{n+1}$;
(3) move points according to $\hat{\mathbf{u}}(x)^n$;
(4) assemble finite element matrices;
(5) apply boundary conditions;
(6) build right hand side vectors (boundary conditions, $\mathbf{g}^n$ and $\mathbf{f}^n$);
(7) solve linear systems for pressure $p$ and velocity $\mathbf{u}$;
(8) volumetric/surface remeshing;
(9) linear interpolation.

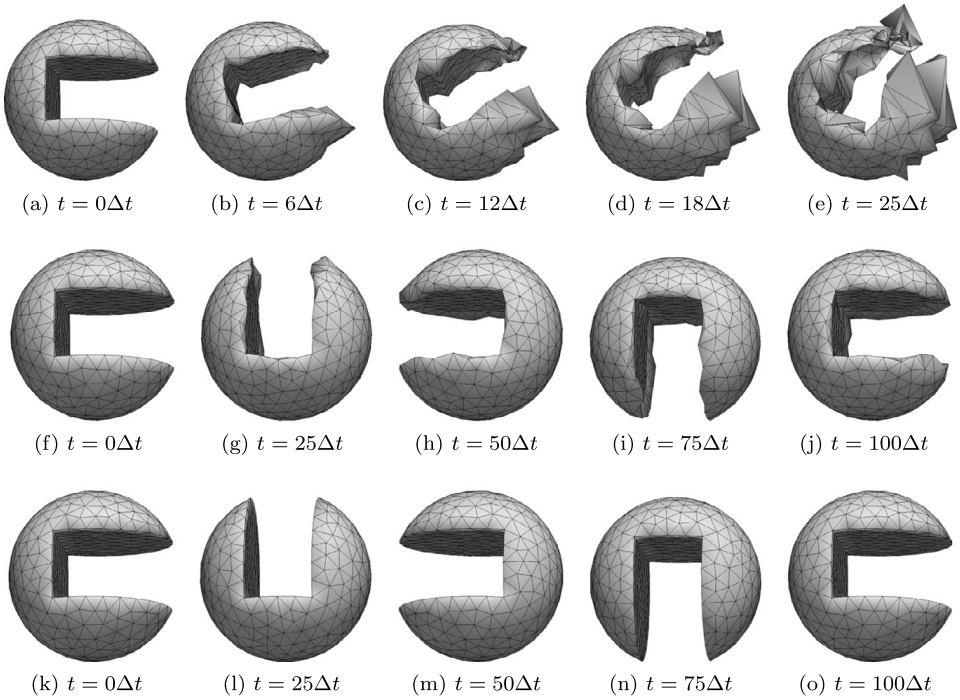



**Fig. 2.** Zalesak slotted sphere test case for mesh length size $h = 0.1$ and different mesh parameters. (a)–(e) $\gamma_1 = 1$, $\gamma_2 = 0$, (f)–(j) $h = 0.1$, $\gamma_1 = 0$, $\gamma_2 = 1$ and (k)–(o) $\gamma_1 = 0$, $\gamma_2 = 0$.

## 6. Validations and results

This section describes numerical results for incompressible two-phase flows obtained with the present 3D Arbitrary Lagrangian–Eulerian Finite Element code. Additionally, single and two-phase flow benchmark tests were carried out to successfully validate the code versus a *Poiseuille* flow, a backward facing step, a lid-driven cavity and a rotating disk flow, a two-phase static droplet and a sessile drop, showing low parasitic currents and accurate pressure differences between the phases [21].

### 6.1. Zalesak sphere

The rigid body rotation of a Zalesak sphere is presented in this section for different surface mesh parameters ($\gamma$'s) and mesh refinement levels ($h$'s). A sphere of radius 0.5 is centered at $(0.0, 0.0, 0.0)$ with a slot of width 0.4 and length 0.82, and rotates according to the velocity field given by $\mathbf{u}(x, y, z) = (-y\ x\ 0)$. By choosing $T = 2 * \pi$ and time step size $dt = T/100$, a full rotation of the rigid body is achieved after 100 iterations. We therefore performed a critical evaluation of the surface mesh parameters $\gamma_1$ and $\gamma_2$ by keeping the same edge length size $h$ as shown in Fig. 2. In Figs. 2(a)–2(e) with $\gamma_1 = 1.0$ and $\gamma_2 = 0.0$, the tangent component of the surface velocity is totally removed, the error in the transport of the points increases and the simulation is ceased after $t = 25\Delta t$. In Figs. 2(f)–2(j) with $\gamma_1 = 0.0$ and $\gamma_2 = 1.0$, the interface is transported in a Lagrangian way and a vertex smoothing operation is performed to rearrange the node distribution. The edges of the slotted sphere are slightly diffused at $t = 25\Delta t$; however the error of the computed area and volume is kept within 0.5% at $t = 100\Delta t$. An accurate rotation of the slotted sphere is presented in Fig.s 2(k)–2(o) for $\gamma_1 = 0.0$ and $\gamma_2 = 0.0$. By choosing these values, the interface connectivity is maintained and the nodes are transported with no significant loss of mass.

Two meshes with higher refinement levels were also tested. The rigid body is fully rotated for both examples with no noticeable numerical diffusion. A summary of the mesh length sizes, the test conditions, the computed area and volume after 1 full rotation is presented in Table 1 for mesh parameters $\gamma_1 = 1$ and $\gamma_2 = 0$. The analytical surface area is $A_A = 3.7120$ and

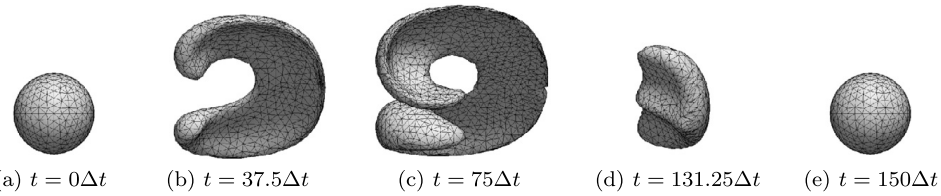
Author's personal copy



**Table 1**
3-dimensional Zalesak's sphere mesh parameters and errors for area and volume.

| Length [$h$] | Surf mesh | | Area [−] | $L_1$ norm[a] [%] | Vol [−] | $L_1$ norm[b] [%] |
|---|---|---|---|---|---|---|
| | Nodes | Tris | | | | |
| 0.2000 | 162 | 316 | 3.5934 | 3.1950 | 0.27581 | 7.8298 |
| 0.1000 | 696 | 1384 | 3.6770 | 0.9428 | 0.29239 | 2.2891 |
| 0.0500 | 2115 | 4222 | 3.7038 | 0.0220 | 0.29766 | 0.5280 |
| 0.0250 | 7376 | 14744 | 3.7096 | 0.0646 | 0.29881 | 0.1436 |
| 0.0125 | 29210 | 58412 | 3.7112 | 0.0021 | 0.29912 | 0.0410 |

[a] $L_1 = 100(|Aa - A|)/A_a$.
[b] $L_1 = 100(|V_A - V|)/V_A$.



(a) $t = 0\Delta t$ (b) $t = 37.5\Delta t$ (c) $t = 75\Delta t$ (d) $t = 131.25\Delta t$ (e) $t = 150\Delta t$

**Fig. 3.** Triangulated surface mesh of the 3-dimensional vortex field test case with mesh parameters $h = 0.03$, $\gamma_1 = 0$, $\gamma_2 = 0$.

the analytical volume is $V_A = 0.29924$. As expected, the $L_1$ norm of the error for the computed area and volume decreases when the mesh is refined.

### 6.2. 3-dimensional vortex field

In this mesh test case we consider the deformation of an initially spherical bubble under the influence of a 3-dimensional vortex field. An initially placed sphere of radius 0.15 centered at $(0.35, 0.35, 0.35)$ in a cubic computational domain with dimension length $L = 1$. The same test case is also reported by [11] and [25] for Front-Tracking algorithms. The imposition of the 3-dimensional deformation field is given by the following vectorial equation:

$$\mathbf{u}(x, y, z, t) = \begin{pmatrix} 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z) \\ -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z) \\ -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z) \end{pmatrix} \cos\left(\frac{\pi t}{T}\right) \tag{20}$$

where $t$ is the time step size and $T$ is the total simulation time. Note that the velocity field reverses in $t = T/2$ and, therefore the sphere shape recovers its initial shape when $t = T$. In our simulations $T = 3$ and $t = 0.2$, thus requiring a total of 150 time steps. Three different mesh refinement levels were tested: $h = 0.030, 0.015, 0.007$. In Fig. 3 the coarse mesh is shown for several simulation times. At time $75\Delta t$ the remeshing does not handle accurately the zones with high curvatures, but nevertheless the error is limited to the characteristic length size of the mesh, and therefore it is reasonable to expect improvement for refined meshes. In fact, accurate results for the two refined meshes were noted for $t = 75\Delta t$ with surface mesh parameters set to $\gamma_1 = 0$ and $\gamma_2 = 0$.

Four mesh properties were evaluated during the simulation for three levels of mesh refinement, namely surface area, bubble's volume, surface curvature and number of surface nodes. The surface area does not show significant disparity for the three meshes, however the change in volume is more pronounced for the coarsest mesh $h = 0.030$, where we found 3% variation during simulation. For $h = 0.015$ and $h = 0.007$ the volume variation were 1% and 0.2% respectively. The mean curvature for the three meshes presented the same behavior during the simulations. An increase of 400% and 200% of total nodes for $h = 0.007$ and $h = 0.015$, respectively, was noted in $t = 75\Delta t$. For the coarsest mesh no significant variation was computed.

### 6.3. Oscillating drop

The simulation consists of an initially 3D axisymmetric ellipsoidal drop immersed in another fluid in the absence of gravity. The perturbed drop oscillates with the frequency $\omega$ given by:

$$\omega^2 = \frac{24\sigma}{(3\rho_{in} + 2\rho_{out})R^3} \tag{21}$$

and an amplitude decay $a(t) = a_0 e^{-t/\tau}$, where $\rho_{in}$ stands for the drop density, $\rho_{out}$ represents the density of the surrounding fluid and $R$ is the unperturbed drop radius. The time is given by $t$ and $\tau = R/5\nu$ where $R$ is the unperturbed drop radius and $\nu$ is the kinematic viscosity. For this simulation $a_0$ was set to $0.1R$.

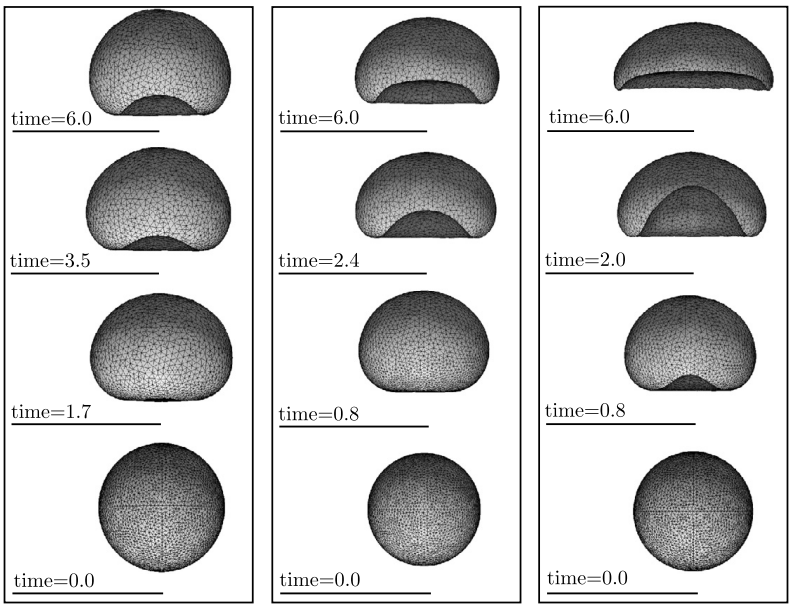
Author's personal copy



**Table 2**
Mesh parameters and error for the convergence analysis of the oscillating drop test case.

| Length [$h$] | Vol mesh | | Surf mesh | | Freq. [–] | $L_1$ norm[a] [%] |
|---|---|---|---|---|---|---|
| | Nodes | Tets | Nodes | Tris | | |
| 0.10 | 1097 | 5338 | 724 | 1440 | 7.2973 | 8.7538 |
| 0.08 | 1491 | 7726 | 924 | 1840 | 7.4673 | 6.6120 |
| 0.04 | 3472 | 19 673 | 1876 | 3744 | 7.8073 | 2.3599 |
| 0.02 | 5914 | 34 698 | 2883 | 5758 | 7.9773 | 0.2339 |

[a] $L_1 = 100(|\omega_A - \omega|)/\omega_A$.




(a) $Mo = 848$ (b) $Mo = 41.1$ (c) $Mo = 1.31$

**Fig. 4.** Bubble's shape evolution with time for the rising of an air bubble immersed in three aqueous sugar solutions.

The non-dimensional parameters used for this simulation were $\sigma = 1$ and $R = 0.5$ as the drop's radius, $\rho_{in} = 1.0$, $\mu_{in} = 1.0$ for the drop and $\rho_{out} = 0.001$ and $\mu_{out} = 0.001$ for the external fluid. The domain limits were set to be $8D \times 8D \times 8D$. A summary of the simulation conditions for different mesh refinement levels is presented in Table 2. Additionally the oscillating frequency $\omega$ and the $L_1$ norm of the frequency error compared to the analytical value of $\omega_A = 7.99603$ are shown. The comparison shows acceptable result for $h = 0.04$ and good result for $h = 0.02$.

### 6.4. Rising bubble

The resilience of the remeshing procedure implemented in this work and the difference of fluid properties (viscosity and density) was tested to validate our proposed methodology. The numerical results were compared to the widely cited experiments performed by Bhaga and Weber [26] and a numerical database generated by an axisymmetric front-tracking code developed in [27] using a fixed background Cartesian grid and a moving interface. Tests were carried out to predict the shape and terminal velocity of a rising air bubble in aqueous sugar solutions for three different liquid viscosities $\{2.73, 1.28, 0.54\}$ kg/ms. According to the experiments, the volume of the generated bubbles was 9.3 cm$^3$, therefore the diameter is $d = 2.61$ cm. The surface tension of 0.078 N/m was that of the air–water–sugar interface, the viscosity and density of the air are 0.0000178 kg/ms and 1.225 kg/m$^3$. We considered an average density value for the aqueous solution to be 1350 kg/m$^3$, since the experiments presented measurement variations. The Morton number Mo for the three cases presented here from the higher to lower liquid viscosity are $\mathrm{Mo} = \{848, 41.1, 1.31\}$. Two different mesh refinement levels were used, namely coarse and refined. The former had approximately 5000 volumetric nodes, 33 000 tetrahedrons, 1400 surface nodes and 2800 surface triangles, and the later had approximately 18 000 volumetric nodes, 111 000 tetrahedrons, 4000 surface nodes and 8000 surface triangles. The geometry of the domain consisted of a cuboid with dimensions of $8D \times 8D \times 15D$ with the higher dimension along the gravity direction and a bubble with diameter $D = 1.0$ placed at the bottom of the domain with its center of mass located at $z = 3$.

As can be seen in Fig. 4, the lower is the viscosity, the stronger is the bubble's shape deformation, requiring substantial remeshing due to the sharp edges at the rim of the bubble. In Fig. 5, the terminal velocity solution for the coarse and refined mesh was compared to the experimental results of Bhaga and Weber for the three test cases presented here, the errors were within 1% while the numerical results of Hua and Lou were instead within an error of 5% for the higher

Author's personal copy



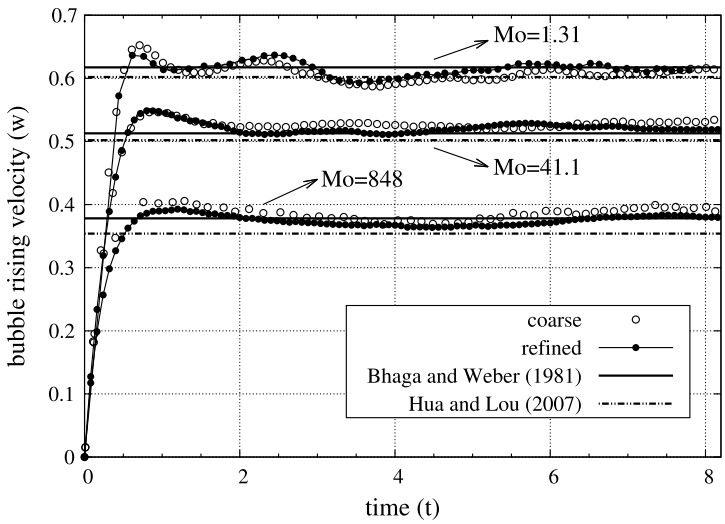



**Fig. 5.** Bubble's center of mass velocities for an air bubbles immersed in three aqueous sugar solution. The straight solid line represents the terminal velocity found by Bhaga and Weber, [26] and the double points are data extracted at [27] for the axisymmetric front-tracking method. Velocity and time are non-dimensional.

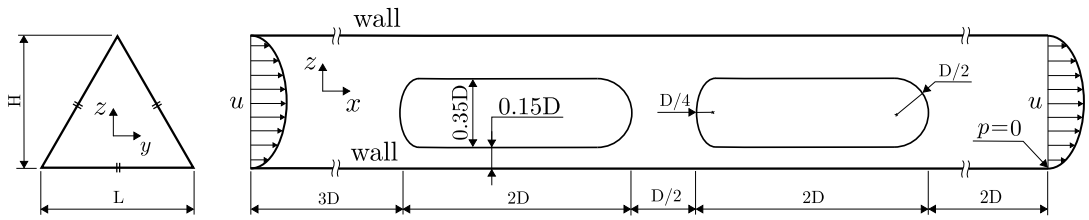



**Fig. 6.** Schematic representation of the geometry, boundary and initial condition and for the simulation of two bubbles in a triangular cross-section microchannel.

viscosity case and 2% for all the others. The mesh parameters used in these simulations were $\beta_1 = \{0.0, 0.0, 0.0\}$, $\beta_2 = \{0.8, 1.0, 1.0\}$, $\beta_3 = \{0.1, 0.3, 1.0\}$ for the volumetric mesh and $\gamma_1 = \{1.0, 1.0, 1.0\}$ and $\gamma_2 = \{0.1, 0.2, 0.2\}$ for the surface mesh. The non-dimensional parameters were set to $\mathrm{Eo} = \{116, 116, 116\}$, $\mathrm{N} = \{42.89, 194.88, 1091.57\}$.

### 6.5. Shear-driven flow of two bubbles in triangular microchannel

We now turn to simulating shear-driven flow in complex geometry for two initially elongated bubbles. The fluid R1234ze is to be used in the scope of the microchannel interlayer cooling of 3D stacked computer chips (Nano-Tera/CMOSAIC, project sponsoring the present work), and properties were taken at temperature 25 °C. In order to simulate such a condition, the moving referential frame is used to shorten the numerical domain, so that the bubbles remain nearly steady in the $x$-direction while the surrounding fluid and the boundaries move in the opposite direction at an average of the bubbles' centroid velocities. A fully developed velocity profile was set on the left and right walls by previously simulating the same condition in a single-phase flow and, therefore, interpolating the solution into the two-phase flow boundaries. On the remaining walls, the no slip condition was used, and pressure was imposed in a single node to provide uniqueness of solution. The initial velocity condition was set in the entire domain using the fully developed velocity profile.

Fig. 6 shows the geometry, initial and boundary conditions for the simulation of the two bubbles in a cross-sectional triangular microchannel. The equivalent diameter $D_{eq}$ of the triangular channel was set to 100 μm and calculated through the formula $D_{eq} = 4 * area/perimeter$, while the channel's mean velocity was set to $u_{mean} = 0.611$ m/s, Due to the prescription of flow velocity, Eq. (1) was slightly modified since the inflow velocity $u_{mean}$ and the microchannel diameter $D_{eq}$ were the relevant parameters, therefore the simulation Reynolds, Froud, Weber and Morton numbers were, $Re = \rho_l u_{mean} D_{eq}/\mu = 370.8$, $Fr = u_{mean}/\sqrt{gD_{eq}} = 19.5$, $We = \rho_l u_{mean}^2 D_{eq}/\sigma = 5.5$ and $\mathrm{Mo} = 2.265 \times 10^{-11}$. We emphasize that the Morton number in this simulation is small (same order air–water systems), showing the applicability of the proposed methodology to flows appearing in real applications. Note that the volumetric mesh is more refined next to the bubble's interface to capture high velocity gradients and the thin liquid film between the interface and wall. Earth gravity condition was set in the $z$-direction.

The time evolution of the two bubbles' shapes is found in Fig. 7. It was observed that the nose of the right bubble is slightly narrower if compared to the left one. For simulation times $t > 1.10$, it was observed that the bubbles' shapes were affected by the microchannel's triangular cross-sectional geometry. Additionally, it was verified that gravity did not play an important role in the numerical solution.

Fig. 8(a) shows the transient prediction of the centroid's velocity for the two bubbles. Similar evolution of the film thickness for the two bubbles, as can be seen in Fig. 8(b). The final liquid film thicknesses were found to be $\delta = 0.074$

Author's personal copy



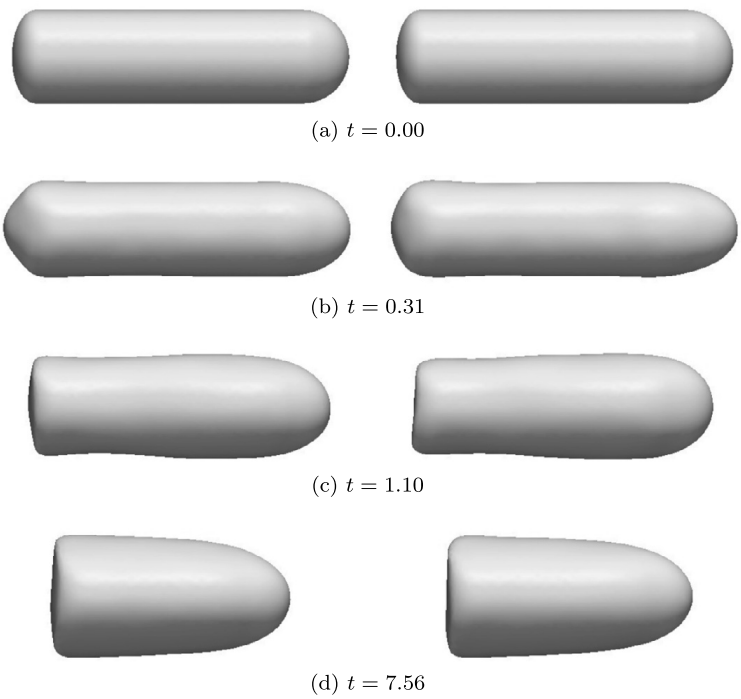



**Fig. 7.** Time evolution of bubbles shapes for refrigerant R1234ze in a triangular microchannel. Flow direction from left to right. Time is non-dimensionalized.

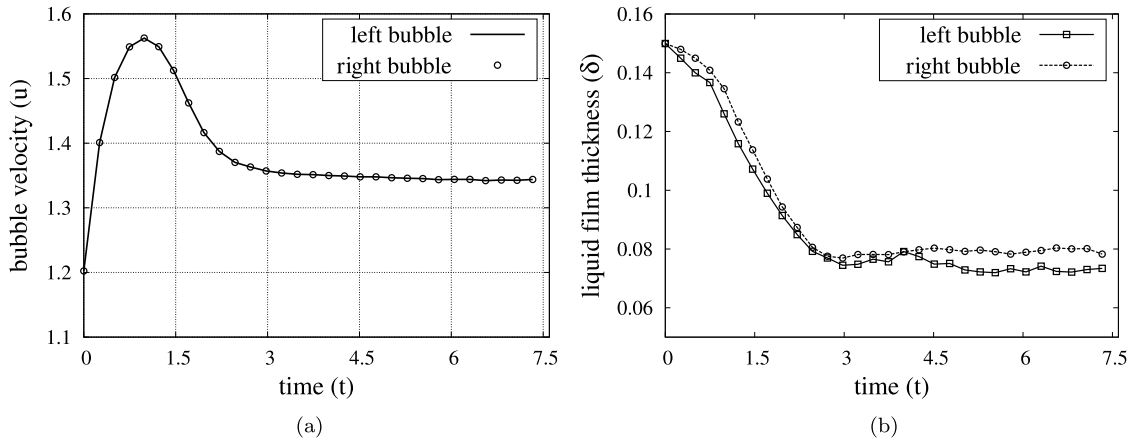



**Fig. 8.** Transient solution of the (a) centroid's velocities and (b) liquid film thickness computed for two bubbles of refrigerant R1234ze.

and $\delta = 0.080$ for the left and right bubble respectively. From time $t \approx 3$ until the end of the simulation, the two bubbles remained equally distanced.

For this simulation, the volumetric mesh parameters $\beta_1 = 0.0$ and $\beta_2 = 0.0$ were set to zero, while $\beta_3 = 1.0$ to guarantee the volumetric nodes distant to the interface. The surface mesh parameters were set to $\gamma_1 = 1.0$ and $\gamma_2 = 1.0$. The domain's discretization used approximately 69 000 tetrahedrons and 8200 nodes. The surface mesh (boundary and bubbles) had approximately 16 000 triangles in which 9000 were part of the interface meshes.

## 7. Conclusions

This paper presents a methodology for simulating incompressible two-phase flows within the Finite Element Method context in which the mesh moves in an Arbitrary Lagrangian–Eulerian fashion. The combined ALE-FEM methodology provides a sharp representation of the interface between the phases, not only for the geometrical representation itself but also for the definition of the phase properties, resulting in a model which accurately describes the actual physical conditions.

The proposed treatment of the computational mesh, together with the splitting of the surface meshes and the volumetric points, has shown to be an excellent choice, thus avoiding the obstacles of handling the remeshing process over the tetrahedron mesh, and thus allowing the utilization of a standard Delaunay tetrahedralization library. Moreover, the new adaptive meshing strategy achieves good control of the mesh quality during the simulations by keeping the volumetric and surface elements bounded to a satisfactory shape, thus preserving the accuracy of the calculation. However, repeated linear

Author's personal copy



interpolations for the velocity and pressure fields on the volumetric and surface mesh may decrease the accuracy in certain domain regions.

The surface remeshing algorithm underwent two important interface transport test cases which are well documented in the literature, namely Zalesak's sphere and 3-dimensional vortex field. Results have shown that the proposed technique can describe strong interface deformations with a small number of surface nodes with reasonable accuracy. The oscillating drop benchmark test has also shown good agreement with its respective analytical solutions for the oscillating frequency. The rising bubble tests were successfully compared to a numerical and experimental database found in the literature for different mesh refinement levels. Moreover, a simulation in complex geometry was carried out to predict bubble shape, centroid velocities and film thicknesses of two bubbles in a triangular microchannel. Therefore, these results highlight the proposed ALE-FEM scheme's suitability to use as an accurate and adaptable two-phase flow simulation strategy for single and isolated multiple bubbles.

## Acknowledgements

This research was funded by the Swiss National Science Foundation (SNF) in Nano-Tera RTD project CMOSAIC (ref. 123618) financed by the Swiss Confederation and scientifically evaluated.

## References

[1] H. Hu, N. Patankar, M. Zhu, Direct numerical simulations of fluid–solid systems using the arbitrary Lagrangian–Eulerian technique, J. Comput. Phys. 169 (2001) 427–462.
[2] C. Hirt, B. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[3] M. Sussman, P. Smereka, S. Osher, A level-set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[4] A. Bourlioux, A coupled level set volume-of-fluid algorithm for tracking material interfaces, in: Proceedings of the 6th International Symposium on Computational Physics, Lake Tahoe CA, 1995.
[5] M. Sussman, E. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2000) 301–337.
[6] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.
[7] A. Amsden, F.H. Harlow, The SMAC method: a numerical technique for calculating incompressible fluid flows, Tech. Rep., Los Alamos, 1970.
[8] J. Glimm, J. Grove, W. Lindquist, O. McBryan, G. Tryggvadson, The bifurcation of tracked scalar waves, SIAM J. Comput. 9 (1) (1988) 61–79.
[9] B. Perot, R. Nallapati, A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows, J. Comput. Phys. 184 (1) (2002) 192–214.
[10] S. Quan, D. Schmidt, A moving mesh interface tracking method for 3D incompressible two-phase flows, J. Comput. Phys. 221 (2006) 761–780.
[11] J. Gois, A. Nakano, L. Nonato, G. Buscaglia, Front tracking with moving-least-squares surface, J. Comput. Phys. 227 (2008) 9643–9669.
[12] H. Braess, P. Wriggers, Arbitrary Lagrangian–Eulerian finite element analysis of free surface flow, Comput. Methods Appl. Mech. Eng. 190 (2000) 95–109.
[13] S. Ganesan, L. Tobiska, Arbitrary Lagrangian–Eulerian finite element method for computation of two-phase flows with soluble surfactants, J. Comput. Phys. 231 (2012) 3685–3702.
[14] I. Babuska, Error-bounds for finite element method, Numer. Math. 16 (1971) 322–333.
[15] F. Brezzi, On the existence, uniqueness and approximation of saddle-point problems arising from Lagrange multipliers, RAIRO, Anal. Numér. R2 (1974) 129–151.
[16] F. Souza, N. Mangiavacchi, A Lagrangian level-set approach for the simulation of incompressible two-fluid flows, Int. J. Numer. Methods Fluids 47 (2004) 1393–1401.
[17] M. Desbrum, M. Meyer, P. Schrder, A. Barr, Implicit fairing of arbitrary meshes using diffusion and curvature flow, in: Proceedings of Siggraph, 1999, pp. 317–324.
[18] C. Geuzaine, J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, Int. J. Numer. Methods Eng. 79 (11) (2009) 1309–1331.
[19] H. Si, Adaptive tetrahedral mesh generation by constrained Delaunay refinement, Int. J. Numer. Methods Eng. 46 (7) (2008) 856–880.
[20] W. Gordon, L. Thiel, Transfinite mappings and their application to grid generation, Appl. Math. Comput. 10–11 (1982) 171–233, http://dx.doi.org/10.1016/0096-3003(82)90191-6.
[21] G. Anjos, A 3D ALE Finite Element Method for Two-Phase Flows with Phase Change, Ph.D. thesis, École Polytechnique Fédérale de Lausanne, 2012.
[22] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. (ISSN 1088-6842) 22 (1968) 745–762.
[23] A. Tomiyama, Struggle with computational bubble dynamics, in: Proceedings of Third Conference on Multiphase Flow – ICMF, 1998.
[24] J. Hua, J. Stene, P. Lin, Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method, J. Comput. Phys. 227 (6) (2008) 3358–3382.
[25] J. Du, B. Fix, J. Glimm, X. Jia, X. Li, Y. Li, L. Wu, A simple package for front tracking, J. Comput. Phys. 213 (2006) 613–628.
[26] D. Bhaga, M. Weber, Bubbles in viscous liquids: shapes, wakes and velocities, J. Fluid Mech. 105 (1981) 61–85.
[27] J. Hua, J. Lou, Numerical simulation of bubble rising in viscous liquid, J. Comput. Phys. 222 (2007) 769–795.