HAL
open science

A higher-order finite element method with unstructured anisotropic mesh adaption for two phase flows with surface tension

Modesar Shakoor, Chung Hae Park

To cite this version:

Modesar Shakoor, Chung Hae Park. A higher-order finite element method with unstructured anisotropic mesh adaption for two phase flows with surface tension. Computers and Fluids, 2021, 230, pp.105154. 10.1016/j.compfluid.2021.105154 . hal-03478730

HAL Id: hal-03478730

https://hal.science/hal-03478730v1

Submitted on 14 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A higher-order finite element method with unstructured anisotropic mesh adaption for two phase flows with surface tension

Modesar Shakoor* and Chung Hae Park

IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Materials and Processes, F-59000 Lille, France

December 14, 2021

Abstract

A novel finite element framework is proposed for the numerical simulation of two phase flows with surface tension. The Level-Set (LS) method with piece-wise quadratic (P2) interpolation for the liquid-gas interface is used in order to reach higher-order convergence rates in regions with smooth interface. A balanced-force implementation of the continuum surface force model is used to take into account the surface tension and to solve static problems as accurately as possible. Given that this requires a balance between the discretization used for the LS function, and that used for the pressure field, an equal-order P2/P2/P2 scheme is proposed for the Navier-Stokes and LS advection equations, which are strongly coupled with each other. This fully implicit formulation is stabilized using the residual-based variational multiscale framework. In order to improve the accuracy and obtain optimal convergence rates with a minimum number of elements, an anisotropic mesh adaption method is proposed where the unstructured mesh is kept as fine as possible close to the zero iso-value of the P2 LS function. Elements are automatically stretched in regions with flat interface in order to keep the complexity fixed during the simulation. The accuracy and efficiency of this approach are demonstrated for two and three dimensional simulations of a rising bubble.

Keywords: multiphase flow, level-set, anisotropic mesh, variational multiscale, higher-order, balanced-force

1 Introduction

Trending industrial applications such as advanced manufacturing (*e.g.*, liquid composite moulding, additive manufacturing) feature complex multiphase flows where capillary effects have a significant influence at different scales [1, 2, 3]. An increasing research effort has been made in the last decade towards the development of robust and reliable Finite Element (FE) simulation tools for the modeling of multiphase flows with surface tension.

First, stabilized FE methods are necessary in order to solve Navier-Stokes equations for a wide range of flow regimes, including turbulent [4]. These methods can also be used to satisfy the inf-sup condition when using incompatible velocity-pressure discretizations [5]. Recent developments based on the Variational MultiScale (VMS) framework are particularly relevant in that regard [6]. For instance, the Residual-Based VMS (RB-VMS) framework has been proven to be applicable both to reaction-diffusion-advection equations and to the incompressible Navier-Stokes equations in convection-dominated regimes for various choices of discretization [7, 8].

*Corresponding author

Second, as shown in Ref. [9], advanced mesh adaption and motion techniques are required to maintain a conformal FE mesh at interfaces between the different phases of the flow during the simulation. Instead, a front-tracking method is generally employed in order to represent and track these moving and deforming interfaces. Most surface tension models additionally introduce a mean curvature dependent surface force term at liquid-gas interfaces [10]. The volume of fluid method [11], which tracks the local volume fraction of each phase during the simulation, is hence not sufficient, because it does not capture the morphology of the interface. The interface morphology can be captured by using the Level-Set (LS) method [12], which approximates the boundary of each phase through its signed distance function, a.k.a. LS function. Indeed, the normal vector and the mean curvature of an interface are related to the first and second derivatives of its distance function [13]. The properties of volume conservation in the volume of fluid method are however lost when switching to the LS method, which has led researchers to consider combinations of those two methods [14]. Alternatively, higher-order discretizations using discontinuous Galerkin methods [15] or isogeometric analysis [8], anisotropic mesh adaption [16, 1] and semi-Lagrangian particles [17] have been considered to deal with the same issue, as well as to improve the order of accuracy at the interface.

Because the liquid-gas interface is usually not discretized with a conformal FE mesh, the implementation of the surface force term is not straightforward. Both the discontinuity in material properties (*i.e.*, density and viscosity) and the surface tension term can be integrated accurately by reconstructing the separate liquid and gas domains and the interface at elements intersected by the interface using the extended FE method [18, 19]. This interface reconstruction and integration step can be avoided using the Continuum Surface Force (CSF) model, where the surface force is transformed into a volume force distributed in a small layer around the interface through the Dirac delta function [10]. Similarly, the discontinuity in material properties can be distributed in this small layer through a regularized Heaviside function [13]. It is important to point out that the regularized Heaviside function depends on the LS function, and that the delta function depends on the first derivative of the regularized Heaviside function, which can be computed analytically [10, 13].

The regularized transition of the material properties relies on the distance property of LS functions [13]. This distance property is usually lost during interface motion and deformation. Thus, it should be restored through an operation called LS reinitialization. This can be achieved directly through geometric distance computation [20] or indirectly by solving an Eikonal equation using fast-marching methods [21]. Alternatively, indirect LS reinitialization methods based on the solution of nonlinear hyperbolic, parabolic or elliptic equations have been proposed in the literature [22]. The choice of LS reinitialization method is not obvious and there is no method that has been widely accepted in the literature. Geometric reinitialization is theoretically more robust, because the exact distance to the interface can be computed [20]. However, it is computationally expensive and its generalization to different FE types and orders is not straightforward [23]. Similarly, the generalization of fast-marching methods to unstructured FE meshes requires generalized upwinding criteria [21]. Indirect methods based on a nonlinear hyperbolic equation rely on an iterative process which has been shown to be unstable in the presence of interface singularities [20]. Recently, LS reinitialization methods based on nonlinear parabolic or elliptic equations have been developed. Even if their robustness has not been demonstrated in the frame of multiphase flow simulations yet [22], these indirect methods are promising as they can easily be implemented for any FE type and order.

Spurious non-physical currents near the interface have been reported in the literature, even for static cases [24, 23, 25, 26]. As analyzed in details in Refs. [25, 27], these currents originate from a mismatch between the pressure discretization and the surface tension model, and numerical errors in the mean curvature calculation. To address the first issue, a balanced-force algorithm can be employed. In this algorithm, the delta function and the normal vector in the CSF model are replaced by the numerical gradient of the regularized Heaviside function, and this gradient as well as the pressure gradient are projected to the same discretization space [24]. To avoid those projections, the same discretization can be used for the LS function and the pressure. Since the pressure discretization space cannot be larger than that of the velocity, the balanced-force implementation of higher-order LS methods is more difficult.

Last, interface motion in the LS method is modeled through advection equations for LS functions. The advection velocity is no other than the flow velocity computed by the flow solver. The flow model includes the CSF model, which depends on the second derivatives of the LS function and, in a balanced-force im-

plementation, on the first derivatives of the LS-dependent regularized Heaviside function. Thus, a choice of time stepping scheme and coupling should be made. To avoid severe restrictions on the time step [13], various implicit schemes have been considered in the literature. It has been shown that both a semi-implicit and a fully implicit implementation of the CSF model can be achieved without strong coupling of the flow and LS advection solvers [28]. To eliminate all restrictions on the time step, fully implicit implementations with strong coupling of the two solvers have been proposed [29, 8].

In this work, we propose an adaptive fully implicit RBVMS method for two phase flow with surface tension relying on a piece-wise quadratic (P2) discretization of the velocity, the pressure and the LS function as well, in order to reduce the computational cost while maintaining a good accuracy of the numerical solutions. All variables, *i.e.*, the velocity, the pressure and the LS function are solved simultaneously because Navier-Stokes equations with the CSF model are strongly coupled to the LS advection equation, as opposed to most conventional approaches. The use of the same discretization space for all variables is only possible because of RBVMS stabilization, and enables us to obtain a balanced-force implementation in a straightforward manner without the procedure of pressure gradient projection. Moreover, a large time step can be adopted through the fully implicit RBVMS method while avoiding divergence of the solution. In order to improve the accuracy near the interface, the unstructured mesh is dynamically adapted during the simulation based on an anisotropic P2 error estimator.
The governing equations for incompressible two phase flow with surface tension are summarized in Sec. 2, and the proposed numerical method is presented in Sec. 3. In Sec. 4, the performance of the proposed method in terms of accuracy and efficiency is demonstrated for two and three dimensional simulations of a rising bubble. It is shown that our fully implicit implementation leads to a reduction of the number of time steps compared to other methods proposed in the literature, while anisotropic unstructured mesh adaption allows us to achieve a more optimal FE discretization with a higher-order convergence rate.

2 Governing equations

2.1 Navier-Stokes equations

The computation domain is denoted as $\Omega \subset \mathbb{R}^d, d = 2, 3$. The Navier-Stokes momentum and mass conservation equations for isothermal, Newtonian, incompressible flow in Ω are

$$\left\{ \begin{array}{rcl} \rho \left(\dfrac{\partial \mathbf{v}}{\partial t} + \mathbf{v}.\nabla \mathbf{v} \right) - \nabla.\sigma(\mathbf{v}, p) - \mathbf{f}_g + \mathbf{f}_s & = & \mathbf{0}, \\ \nabla.\mathbf{v} & = & 0. \end{array} \right. \tag{1}$$

In these equations $\mathbf{v}$ is the velocity vector, p the pressure, t the time, ρ the mass density, and μ the viscosity. The Cauchy stress tensor σ is defined as

$$\sigma(\mathbf{v}, p) = \mu(\nabla \mathbf{v} + \nabla^T \mathbf{v}) - p\mathbf{I}, \tag{2}$$

where $\mathbf{I}$ is the identity matrix. The force $\mathbf{f}_g = -\rho g \mathbf{e}_y$ is the gravity force, with g the gravity acceleration. The force $\mathbf{f}_s$ is the surface tension force and is described in Subsec. 2.3.
Those equations will be accompanied by either Dirichlet or homogeneous Neumann boundary conditions in the paper.

2.2 LS method

The computation domain is split into two parts Ω_1 and Ω_2, each occupied by a different fluid, with

$$\Omega_1 \cup \Omega_2 = \Omega, \text{ and } \Omega_1 \cap \Omega_2 = \Gamma_{1,2}.$$

Both domains as well as their interface evolve in time according to the motion of the two fluids:

$$\Omega_1 = \Omega_1(t), \Omega_2 = \Omega_2(t), \text{ and } \Gamma_{1,2} = \Gamma_{1,2}(t).$$

Instead of using a conformal FE mesh of $\Gamma_{1,2}$, we introduce the LS function ϕ [12] which is the signed distance function to $\Gamma_{1,2}$:

$$\phi(\mathbf{x},t) = \begin{cases} +dist(\mathbf{x},\Gamma_{1,2}(t)), & \mathbf{x} \in \Omega_1(t), \\ -dist(\mathbf{x},\Gamma_{1,2}(t)), & \mathbf{x} \in \Omega_2(t), \\ 0, & \mathbf{x} \in \Gamma_{1,2}(t). \end{cases} \tag{3}$$

The first (resp. second) fluid hence occupies the part of the domain with positive (resp. negative) ϕ, while the interface is defined as the zero isolevel of ϕ. The evolution of the interface can then be modeled by advecting ϕ with the flow velocity:

$$\frac{\partial \phi}{\partial t} + \mathbf{v}.\nabla\phi = 0. \tag{4}$$

Fluid properties in Eq. (1) can now be defined in terms of the LS function [30]:

$$\begin{aligned} \rho(\mathbf{x},t) = \rho(\phi(\mathbf{x},t)) &= H(\phi(\mathbf{x},t))\rho_1 + (1 - H(\phi(\mathbf{x},t)))\rho_2 \\ \mu(\mathbf{x},t) = \mu(\phi(\mathbf{x},t)) &= H(\phi(\mathbf{x},t))\mu_1 + (1 - H(\phi(\mathbf{x},t)))\mu_2 \end{aligned} \tag{5}$$

where ρ_1, ρ_2 and μ_1, μ_2 are the mass densities and viscosities of each fluid, respectively, and H is the Heaviside function:

$$H(\phi) = \begin{cases} 1, & \phi > 0, \\ \frac{1}{2}, & \phi = 0, \\ 0, & \phi < 0. \end{cases} \tag{6}$$

2.3 CSF model

The surface tension force $\mathbf{f}_s$ can be expressed through the CSF model [10] as

$$\mathbf{f}_s = \sigma_s \kappa_s \mathbf{n}_s \delta \tag{7}$$

where σ_s is the surface tension coefficient, κ_s is the mean curvature of the interface, $\mathbf{n}_s$ its normal vector, and $\delta = \frac{\mathrm{d}H}{\mathrm{d}\phi}$ is the so-called interface density. The computation of this derivative requires a regularization of the Heaviside function, as described in Subsec. 3.1.
The normal vector and the mean curvature can be directly computed from LS function ϕ using

$$\mathbf{n}_s = \mathbf{n}_s(\phi) = \frac{\nabla\phi}{|\nabla\phi|}, \kappa_s = \kappa_s(\phi) = -\nabla.\mathbf{n}_s(\phi) = \frac{\nabla\phi.\nabla\nabla\phi.\nabla\phi - |\nabla\phi|^2 \mathrm{tr}(\nabla\nabla\phi)}{|\nabla\phi|^3}, \tag{8}$$

where $\nabla\nabla\phi$ is the Hessian matrix of ϕ and $|.|$ the Euclidean norm. Due to the sign of the LS function in Eq. (3), the normal vector will be pointing towards phase 1, which means that the liquid phase is phase 1 while the gas phase is phase 2.

The numerical formulation described in the following section solves Eqs. (1) and (4) to approximate $\{\mathbf{v}, p, \phi\}$. It can be observed that the LS advection equation depends linearly on the flow velocity, while the momentum equation involves a nonlinear dependence of the mass density and the viscosity on the LS function. This is added to the nonlinearity of the auto-advection term and of the surface tension force.

3 Numerical methods

In this section, we present the different numerical methods to solve the equations described in Sec. 2. Those equations are solved using the RBVMS framework where we choose equal-order FE discretizations $\{\mathbf{v}_h, p_h, \phi_h\}$ for the velocity, pressure and LS solutions. This approximation space is denoted by $\mathcal{V}_h$. The parameter h is proportional to the length of the longest edge of the FE mesh.

3.1 Balanced-force implementation

As mentioned in the introduction, to avoid dealing explicitly with the discontinuity in Eqs. (6) and (7), a regularized Heaviside function H_ϵ will be used in practice instead of H in Eq. (5):

$$H_\epsilon(\phi) = \begin{cases} 1, & \phi > \epsilon, \\ \frac{1}{2}\left(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\phi}{\epsilon}\right)\right), & |\phi| \leq \epsilon, \\ 0, & \phi < -\epsilon. \end{cases} \tag{9}$$

As a result, the transition of fluid properties within a small layer of thickness 2ϵ around the interface can become smooth. The parameter $\epsilon \sim \mathcal{O}(h)$ should be as small as possible to have an accurate transition of material properties while it cannot be smaller than $\frac{h}{2}$ (as compared to h for P1 FEs [29]).
In general [13, 30, 8], the δ function in Eq. (7) is also replaced by a regularized function δ_ϵ which is defined by the analytical derivative of H_ϵ, namely $\delta_\epsilon = \frac{\mathrm{d}H_\epsilon}{\mathrm{d}\phi}$. This function may be weighed to deal with large ratios of mass density [10, 23].
However, such formulation leads to oscillations in the case of a static bubble or droplet submitted to no gravity [24]. In this case Eq. (4) has no effect and Eq. (1) reduces to $\nabla p = \sigma_s \kappa_s \mathbf{n}_s \delta$. Regardless of the numerical method used to estimate the mean curvature, if the discretization space of ∇p_h does not match that of $\mathbf{n}_s(\phi_h)\delta(\phi_h)$, numerical errors will induce spurious non-physical currents [24].
To avoid these oscillations, we replace $\mathbf{n}_s(\phi_h)\delta(\phi_h)$ by the numerical gradient of the Heaviside function $\nabla H_\epsilon(\phi_h)$, which leads to the following balanced-force implementation of the CSF model:

$$\mathbf{f}_s(\phi_h) = \sigma_s \kappa_s(\phi_h) \nabla H_\epsilon(\phi_h). \tag{10}$$

where $H_\epsilon(\phi_h)$ is defined in Eq. (9) and $\kappa_s(\phi_h)$ is computed using Eq. (8). This is a balanced-force implementation because we are using equal-order discretizations for the pressure, the LS function and the regularized Heaviside function.
In order to compute $\kappa_s(\phi_h)$ directly through Eq. (8), ϕ_h needs to be at least piece-wise quadratic. Alternatively, we could use integration by part to eliminate the computation of second derivatives of the LS function [18, 19]. Nevertheless, the choice of a piece-wise quadratic LS function is adopted in this paper to improve mass and energy conservation [29]. As a consequence, p_h also needs to be piece-wise quadratic and, even though RBVMS stabilization is used, $\mathbf{v}_h$ needs to be at least piece-wise quadratic because of the inf-sup condition. If we wanted to use a higher-order discretization for ϕ_h and obtain an improved mass conservation [15], we could avoid those restrictions by projecting $\nabla H_\epsilon(\phi_h)$ down to the pressure gradient discretization space. However, this would deteriorate the accuracy of the discrete CSF model. Consequently, we adopt a P2/P2/P2 discretization.

3.2 Fully implicit RBVMS framework

The stabilization of advection-dominated equations, in particular Navier-Stokes equations in the convection-dominated regime, has been a very active research topic for the last decades [4, 6, 5]. The VMS framework proposed in 1998 has given interesting insights on how this stabilization could be achieved by approximating those equations simultaneously at two scales (*i.e.*, coarse and fine) and projecting the fine scale onto the coarse scale [6]. The coarse one is simply the one discretized by the mesh, while the subgrid scale should be modeled separately. The difficulty resides in how the fine scale is modeled and how it is projected to stabilize the coarse scale. This issue can be addressed through the RBVMS framework which has been proposed for incompressible Navier-Stokes equations [7] and later on extended to the simulation of multiphase flow with surface tension [18, 8].
In the RBVMS framework for multiphase flow [3], the velocity, pressure and LS approximations are written

as

$$
\begin{aligned}
\mathbf{v} &\approx \mathbf{v}_h + \mathbf{v}' \\
p &\approx p_h + p' \\
\phi &\approx \phi_h + \phi'
\end{aligned}
\tag{11}
$$

where $\{\mathbf{v}_h, p_h, \phi_h\}$ are the coarse scale solutions and $\{\mathbf{v}', p', \phi'\}$ are the fine scale solutions. The discrete space of velocity, pressure and LS test functions $\{\mathbf{w}_h, q_h, \psi_h\}$ is denoted as $\mathcal{W}_h$. The semi-discrete weak form of Eqs. (1) and (4) follows [8]:
Find $\{\mathbf{v}_h, p_h, \phi_h\} \in \mathcal{V}_h^{d+2}$, such that,

$$
\left\{
\begin{aligned}
&\int_{\Omega_h} \mathbf{w}_h.\rho(\phi_h)\left(\frac{\partial \mathbf{v}_h}{\partial t} + \mathbf{v}_h.\nabla\mathbf{v}_h\right)\mathrm{d}\Omega \\
&+\int_{\Omega_h} \nabla\mathbf{w}_h : \mu(\phi_h)\left(\nabla\mathbf{v}_h + \nabla^T\mathbf{v}_h\right)\mathrm{d}\Omega - \int_{\Omega_h} p_h\nabla.\mathbf{w}_h\mathrm{d}\Omega \\
&-\int_{\Omega_h} \mathbf{w}_h.\mathbf{f}_g(\phi_h)\mathrm{d}\Omega + \int_{\Omega_h} \mathbf{w}_h.\mathbf{f}_s(\phi_h)\mathrm{d}\Omega \\
&-\sum_{K\in\mathcal{T}_h}\int_K \left(\rho(\phi_h)\mathbf{v}_h.\nabla\mathbf{w}_h\right).\mathbf{v}'(\mathbf{v}_h, p_h, \phi_h)\mathrm{d}\Omega \\
&+\sum_{K\in\mathcal{T}_h}\int_K \rho(\phi_h)\mathbf{w}_h.\left(\nabla\mathbf{v}_h.\mathbf{v}'(\mathbf{v}_h, p_h, \phi_h)\right)\mathrm{d}\Omega \\
&-\sum_{K\in\mathcal{T}_h}\int_K \rho(\phi_h)\nabla\mathbf{w}_h : \left(\mathbf{v}'(\mathbf{v}_h, p_h, \phi_h)\otimes\mathbf{v}'(\mathbf{v}_h, p_h, \phi_h)\right)\mathrm{d}\Omega \\
&-\sum_{K\in\mathcal{T}_h}\int_K \nabla.\mathbf{w}_h p'(\mathbf{v}_h, p_h, \phi_h)\mathrm{d}\Omega && = 0, \forall\mathbf{w}_h \in \mathcal{W}_h^d, \\
&-\int_{\Omega_h} q_h\nabla.\mathbf{v}_h\mathrm{d}\Omega \\
&+\sum_{K\in\mathcal{T}_h}\int_K \nabla q_h.\mathbf{v}'(\mathbf{v}_h, p_h, \phi_h)\mathrm{d}\Omega && = 0, \forall q_h \in \mathcal{W}_h, \\
&+\int_{\Omega_h} \psi_h\left(\frac{\partial\phi_h}{\partial t} + \mathbf{v}_h.\nabla\phi_h\right)\mathrm{d}\Omega \\
&-\sum_{K\in\mathcal{T}_h}\int_K \nabla\psi_h.\mathbf{v}_h\phi'(\mathbf{v}_h, \phi_h)\mathrm{d}\Omega && = 0, \forall\psi_h \in \mathcal{W}_h,
\end{aligned}
\right.
\tag{12}
$$

where we have indicated explicitly the nonlinear dependence of some terms on the solution. The boundary integral term is omitted due to homogeneous Neumann boundary conditions. The fine scale stabilization terms are integrated element-wise, where $\mathcal{T}_h$ is the FE mesh which is a set of elements spanning the discrete domain $\Omega_h \approx \Omega$. The fine scale stabilization term for the conservation equation makes this formulation inf-sup stable and enables us to use equal-order FEs for velocity and pressure. Readers are referred to Ref. [7] for the details regarding the remaining fine scale terms. In the RBVMS framework, the fine scale solutions are simply modeled as proportional to the residuals of the strong form coarse scale equations:

$$
\begin{aligned}
\mathbf{v}'(\mathbf{v}_h, p_h, \phi_h) &= -\frac{\tau_{\mathbf{v}}(\mathbf{v}_h, \phi_h)}{\rho(\phi_h)}\left(\rho(\phi_h)\left(\frac{\partial\mathbf{v}_h}{\partial t} + \mathbf{v}_h.\nabla\mathbf{v}_h\right) - \nabla.\sigma(\mathbf{v}_h, p_h, \phi_h) - \mathbf{f}_g(\phi_h) + \mathbf{f}_s(\phi_h)\right), \\
p'(\mathbf{v}_h, p_h, \phi_h) &= -\tau_p(\mathbf{v}_h, \phi_h)\rho(\phi_h)\nabla.\mathbf{v}_h, \\
\phi'(\mathbf{v}_h, \phi_h) &= -\tau_\phi(\mathbf{v}_h)\left(\frac{\partial\phi_h}{\partial t} + \mathbf{v}_h.\nabla\phi_h\right),
\end{aligned}
\tag{13}
$$

where σ is the Cauchy stress tensor defined in Eq. (2). Here we have again indicated explicitly the dependence of the fine scale solutions on the coarse scale solutions as well as that of the following stabilization parameters:

$$\begin{aligned}
\tau_{\mathbf{v}}(\mathbf{v}_h, \phi_h) &= \left(\left(\frac{2}{\Delta t} \right)^2 + \mathbf{v}_h . \mathbf{G} \mathbf{v}_h + C_I \left(\frac{\mu(\phi_h)}{\rho(\phi_h)} \right)^2 \mathbf{G} : \mathbf{G} \right)^{-\frac{1}{2}}, \\
\tau_p(\mathbf{v}_h, \phi_h) &= \left(\mathrm{tr}(\mathbf{G}) \tau_{\mathbf{v}}(\mathbf{v}_h, \phi_h) \right)^{-1}, \\
\tau_\phi(\mathbf{v}_h) &= \left(\left(\frac{2}{\Delta t} \right)^2 + \mathbf{v}_h . \mathbf{G} \mathbf{v}_h \right)^{-\frac{1}{2}}.
\end{aligned} \tag{14}$$

The constant C_I depends on the type and order of the FE, Δt is the time step, and $\mathbf{G} = \mathbf{B}^T \mathbf{B}$ is the element metric tensor, with $\mathbf{B}$ the transposed inverse of the gradient of the mapping from reference element to mesh element [7]. Note that those stabilization terms are defined and computed at quadrature points within each element, contrary to other stabilization methods where they may be defined as constant element-wise [4, 5]. For the time discretization, we use the backward Euler scheme which is unconditionally stable but requires a Newton-Raphson scheme to deal with the nonlinear terms in Eq. (12), the fine scale solutions in Eq. (13) and the stabilization parameters in Eq. (14). In order to avoid the difficulty of computing all analytical derivatives, we adopt numerical differentiation. This procedure is conducted locally inside each element by iteratively adding and then removing a small perturbation δ_{ND} to the node-wise values of each of the solutions $\{\mathbf{v}_h, p_h, \phi_h\}$. The drawback of this approach is that the Newton-Raphson scheme will not converge up to a very low tolerance depending on the value of δ_{ND}.

If the prescribed time step Δt_{max} is too big or the solution is irregular, the Newton-Raphson scheme might not converge in a prescribed maximum number of iterations or even diverge and obtain solutions which lead to a larger residual from one nonlinear iteration to another. If such situation arises, we restart the Newton-Raphson scheme with half the time step and repeat this procedure until we find a time step small enough for the Newton-Raphson scheme to obtain a converged solution in less than 10 iterations. Inversely, when a time increment can be solved at once with no restart, we multiply the time step by 1.5, unless it leads to a larger value than Δt_{max}. In all simulations presented in this paper, this automatic time step control algorithm was always successful in making the Newton-Raphson scheme converge in less than 10 iterations. As observed in Sec. 4, this automatic variation of the time step did not lead to an increase of the number of time increments higher than a factor of two.

3.3 LS reinitialization

The use of the regularized Heaviside function as defined in Eq. (9) requires ϕ to remain a signed distance function throughout the simulation. As mentioned in the introduction, the distance property is lost during interface motion and deformation, and should be restored using LS reinitialization. Because the implicit scheme enables the use of a large prescribed time step Δt_{max}, we reinitialize ϕ_h at every time step.
Recently developed indirect methods based on a nonlinear elliptic equation are promising because, in theory, this nonlinear elliptic equation can be easily solved using an FE method with any FE type or order [22]. Additionally, it should be theoretically possible to integrate this approach directly into the fully implicit RBVMS framework as an additional regularity constraint [31] and even to add a mass conservation constraint to reduce mass loss during advection and reinitialization [31].
In the present work, we adopt a more straightforward approach as we opt for direct geometric reinitialization [20]. The implementation of a P2 LS function is technically challenging because the intersection of each element of the mesh with the zero isolevel of ϕ_h is not necessarily planar. To avoid this problem, we hierarchically subdivide each P2 FE twice and reconstruct a fine surface mesh of the zero isolevel of ϕ_h as a P1 mesh of straight line segments (2D) or flat triangles (3D). More details on this method can be found in Ref. [23].
The distance of any point to the interface $\Gamma_{1,2}$ is approximated by looking for the closest projection among all elements of the surface mesh [32]. This search can be conducted efficiently using a space partitioning

technique [33, 20]. In a recent work [34], it has been shown that Graphics Processing Units (GPUs) can be much more efficient for this kind of task than conventional CPU implementations. Therefore, we opt for a brute force search where we project all mesh nodes on all elements of the surface mesh, with no space partitioning at all. This search is conducted independently for each mesh node on a separate GPU thread. Additionally, we let the LS reinitialization procedure automatically truncate the LS function around the interface so that its values be restricted to $[-2\epsilon, 2\epsilon]$.

3.4 Anisotropic P2 mesh adaption

Thanks to the higher-order discretization of the solution $\{\mathbf{v}_h, p_h, \phi_h\}$, it is expected to improve the accuracy and the conservation of the LS function at each time increment compared to P1 methods. The accuracy should be controlled near to the interface, in the region of thickness 2ϵ where we operate the fluid properties transition defined in Eqs. (9) and (5) and integrate the surface tension force defined in Eq. (10).
The challenge is that the interface is evolving during the simulation. We hence propose to dynamically adapt and refine the unstructured mesh during the simulation to track the interface and keep elements refined in the transition region. To reach this objective, we define a sensor variable $s_h = (H_\epsilon(\phi_h^n), H_\epsilon(2\phi_h^n - \phi_h^{n-1}))$, where ϕ_h^n is the discrete LS function at the start of a time increment (before Newton-Raphson solution) and $2\phi_h^n - \phi_h^{n-1}$ is an extrapolation of ϕ_h^{n+1}. We do not include the velocity and pressure fields in the sensor variable in order to avoid difficulties in taking into account errors of different natures. This will be investigated in a future work as it could be relevant for multiphysics problems. It can be noted that, because of the use of an equal-order P2/P2/P2 FE method, the convergence rate for the velocity and pressure fields should be improved in the transition region.

Isotropic mesh adaption can be based on any scalar *a priori* or *a posteriori* estimation of the approximation error at integration points or mesh nodes. For instance, the well-known Zienkiewicz-Zhu error estimator [35] considers the point-wise difference between the numerical solution and the improved approximation recovered using an operation called Superconvergent Patch Recovery (SPR). This SPR operation which is used subsequently in this work, consists in recovering a higher-order and higher-regularity approximation of the solution around each mesh node. For a current P1 approximation at a given node, the recovered approximation should be of order 2 [36], namely it should embed a recovered node-wise continuous gradient and Hessian matrix. This recovered approximation is fitted in a least-squares sense to the values of the numerical solution at neighboring mesh nodes which is called the patch.
Alternatively, the fine scale solutions of the VMS framework described in Subsec. 3.2 can be used as estimates of the error in the coarse scale approximations [6, 7].
The latter interpolation error can be used as a target for error estimation because the approximation error is bounded by it. In particular, the interpolation error for a P1 solution can be estimated from the Hessian matrix of this solution and that for a P2 solution from its third derivatives [37] which can be recovered node-wise using a SPR one order higher than that of the derivatives to be recovered [38], namely a SPR of order 3 for second derivatives and order 4 for third derivatives.
Elements can be refined and coarsened based on the point-wise estimation of the approximation or interpolation error.

For anisotropic mesh adaption, additional directional information on the error is required not only to refine and coarsen elements, but also to stretch them differently along each direction. Improved convergence rates are obtained with unstructured anisotropic mesh adaption, because the mesh can be distorted and oriented locally depending on the estimated error [38, 39, 37].
In the P1 case, the interpolation error estimator reduces to a Hessian-based error estimator [40, 41, 16]. A particular feature of this estimator is that the Hessian matrix defines locally a scalar product and a distance measure which are distorted compared to the Euclidean ones [42]. This distortion or stretching depends on the eigenvalues and eigenvectors of the Hessian matrix. The eigenvectors can also be regarded as the axis of an ellipsoid and the square root of the inverse of the eigenvalues as the radii of this ellipsoid [43]. An edge-based version of the Hessian-based error estimator has been proposed recently and applied to multiphase

flow simulations [39, 1].
To the best of the authors' knowledge, there has been no anisotropic error estimator based on the fine scale solutions of the VMS framework in the literature except the anisotropic versions of the Zienkiewicz-Zhu estimator [44, 45]. Anisotropic *a priori* error estimators for multiphase flow simulations directly targeting the error in the geometric representation of the interface through the LS function have been proposed in Refs. [46, 47]. In the following, we opt for an error estimator targeting the interpolation error of the sensor variable. This will allow us not only to adapt the mesh during the simulation but also pre-adapt the mesh before solving any equation so as to ensure that the interpolation of the initial LS function is accurate.

To prepare an unstructured mesh for the initial time increment or any current time increment, it is required to estimate the interpolation error of s_h, and then adapt the mesh to distribute this error uniformly on the domain, for a given complexity. Indeed, we will not try to prescribe an error tolerance, which may blow up the computational cost if the interface area increases during the simulation. Instead, we keep the complexity (the number of P1 nodes of the mesh) constant during the simulation and expect the error estimator to capture only a certain level of detail which can be afforded with the prescribed complexity. Such approaches have been proposed for instance in Refs. [40, 39] for anisotropic P1 interpolation.
Approaches to dynamically adapt an unstructured mesh to control the interpolation error for P1 FEs have been widely discussed in the literature whereas there has been few works on higher-order discretizations, with the exception of Refs. [37, 48].

Recently, the continuous mesh framework for metric-based mesh adaption [49, 50] has been extended to the higher-order case [37, 48]. In the following, we briefly describe this framework and our proposed implementation. We denote by s_i the component $i = 1 \dots \dim(s)$ of the continuous sensor variable ($\dim(s) = 2$ in this work) and by Πs_i its discrete P2 interpolation. The approximation error $||s_i - s_{h,i}||_{L^2(K)}$ is bounded within an element $K \in \mathcal{T}_h$ by the interpolation error $||s_i - \Pi s_i||_{L^2(K)}$ which is itself bounded by the L^2 norm of the third derivatives $||\nabla\nabla\nabla s_i||_{L^2(K)}$:

$$||s_i - \Pi s_i||_{L^2(K)} \leq \mathcal{C}(K)||\nabla\nabla\nabla s_i||_{L^2(K)}, \forall K \in \mathcal{T}_h, \forall i = 1 \dots \dim(s).$$

An approximation of the third derivatives of s_i is recovered node-wise using SPR which is naturally extended for the P2 sensor variable by recovering the continuous first, second, third and fourth derivatives of s_i around each node [38]. Hence, 15 unknowns are solved at each node in 2D and 35 in 3D. To ensure that this problem is well-defined, we use patches with at least twice as many nodes as unknowns for the least-squares fit. The node-wise continuous third derivatives of the two-dimensional sensor variable are extracted from this SPR operation.
The main challenge with respect to the P1 case is that $\nabla\nabla\nabla s_i$ is not a symmetric positive definite second-order tensor and thus does not define a quadratic form, a scalar product and a distance measure. The idea proposed in Ref. [37] is to find a suitable symmetric positive definite matrix $\mathbf{Q}(\mathbf{x})$ at each point $\mathbf{x} \in \Omega$ such that

$$|\nabla\nabla\nabla s_i(\mathbf{x}).\mathbf{y}.\mathbf{y}.\mathbf{y}| \leq |\mathbf{y}^T\mathbf{Q}(\mathbf{x})\mathbf{y}|^{\frac{3}{2}}, \forall \mathbf{y} \in \mathbb{R}^d, \forall i = 1 \dots \dim(s). \tag{15}$$

In order to have a close bound, the determinant of $\mathbf{Q}(\mathbf{x})$ should be as small as possible. This constrained minimization problem hence consists in finding a symmetric positive definite matrix $\mathbf{Q}(\mathbf{x})$ such that $\det \mathbf{Q}(\mathbf{x})$ is minimal and the bound in Eq. (15) is satisfied. It was solved in Ref. [37] using a so-called logarithm transformation and a simplex minimization algorithm. Alternatively, a simpler approach consisting in directly finding a quadratic form approximating $\nabla\nabla\nabla s(\mathbf{x})$ through a least-squares fitting was proposed in Ref. [48]. In this work, we develop a similar approach which consists in directly approximating $\nabla\nabla\nabla s(\mathbf{x})$ as a quadratic form through the following geometric averaging operation:

$$\mathbf{Q}(\mathbf{x}) = \left(\exp\left(\frac{1}{\dim(s)d}\sum_{i=1}^{\dim(s)}\sum_{j=1}^{d}\log\left(\left(\nabla\nabla\frac{\partial s_i}{\partial x_j}(\mathbf{x})\right)^{-\frac{1}{2}}\right)\right)\right)^{-2}. \tag{16}$$

This geometric averaging operation is widely used to average or interpolate metric fields in various applications [42, 43]. It is called geometric because $\nabla\nabla\frac{\partial s_i}{\partial x_j}(\mathbf{x})$ is a symmetric positive definite matrix and can be associated to an ellipsoid whose radii are given by the eigenvalues of $(\nabla\nabla\frac{\partial s_i}{\partial x_j}(\mathbf{x}))^{-\frac{1}{2}}$. It is shown in Sec. 4 that this simple averaging operation preserves the directions embedded in $\nabla\nabla\nabla s(\mathbf{x})$ and, in particular, predicts correctly the decrease of the interpolation error in regions with flat interface.
The conversion of the directional errors carried by the metric field $\mathbf{Q}$ into a metric field $\mathbf{M}$ of directional mesh sizes requires the solution of a constrained minimization problem consisting in minimizing the total error while uniformly distributing local errors and controlling the complexity. The solution of this constrained minimization problem in the higher-order case can be expressed as [37]

$$\mathbf{M}(\mathbf{x}) = N^{\frac{2}{d}} \left(\int_\Omega (\det(\mathbf{Q}(\mathbf{x}))^{\frac{3}{6+d}} \mathrm{d}\mathbf{x} \right)^{-\frac{2}{d}} (\det(\mathbf{Q}(\mathbf{x})))^{-\frac{1}{6+d}} \mathbf{Q}(\mathbf{x}), \tag{17}$$

where N is the prescribed number of P1 nodes. It is reminded that the asymptotic convergence rate in L^2 norm with respect to the mesh size for a uniform isotropic P2 mesh is 3, which means a rate of $\frac{3}{d}$ with respect to N.

The metric field defined in Eq. (17) can be used to compute the distorted lengths of each edge of the current FE mesh and determine whether they are too long (distorted length larger than one) or too short (distorted length smaller than one). The distorted volumes of each element of the current mesh can also be computed. An optimal mesh for a metric field $\mathbf{M}$ has a minimum number of elements whose distorted volumes are as uniform as possible and edges whose distorted lengths are as close as possible to one.
To solve this multi-objective optimization problem, we use a mesh adaption algorithm developed in previous works [51, 9]. Note that we keep the P2 nodes at edge centers in this optimization and the unstructured mesh remains linear (only the interpolation is quadratic). For a given element, the quality criterion which is used in the algorithm considers the element's distorted volume and the average of the distorted lengths of the element's edges. All conformal topological modifications including node re-positioning and edge splitting are tested in small patches in the neighborhood of each node and edge of the mesh, and the modification which maximizes the quality criterion is accepted. This iterative algorithm with local modifications ends when no more local modification can be found to improve the quality of the mesh. The details on the mathematical background of this algorithm can be found in Ref. [51], while the implementation details can be found in Ref. [9].
Once the mesh adaption algorithm terminates, the velocity field and the LS function should be transferred from the old mesh to the new (adapted) one. First, a space partitioning technique is used to locate efficiently the element of the old mesh containing each node of the new mesh. The values of the velocity and the LS function are then computed at each node of the new mesh using P2 interpolation from the containing element of the old mesh. At this step, the P2 discretization of both the velocity field and the LS function can reduce any errors and diffusion induced by mesh adaption and fields transfer. Finally, the new mesh and the transferred velocity field and LS function can be used for the current time increment.

4 Results and discussion

The proposed method has been implemented in a *C* code (as of Ref. [52]), whereas the LS reinitialization procedure mentioned in Subsec. 3.3 has been implemented in *NVIDIA's CUDA* programming language for GPU computing and the mesh adaption algorithm presented in Subsec. 3.4 has been implemented in *C++* (as of Ref. [53]). For computationally demanding operations such as numerical differentiation and integration of the discrete RBVMS weak form presented in Subsec. 3.2, implementations rely on shared-memory multi-threaded parallelism using *OpenMP* [54]. The *lis* library to solve the linear system at each iteration of the Newton-Raphson scheme also relies on *OpenMP* [55]. In particular, we employ the generalized minimum

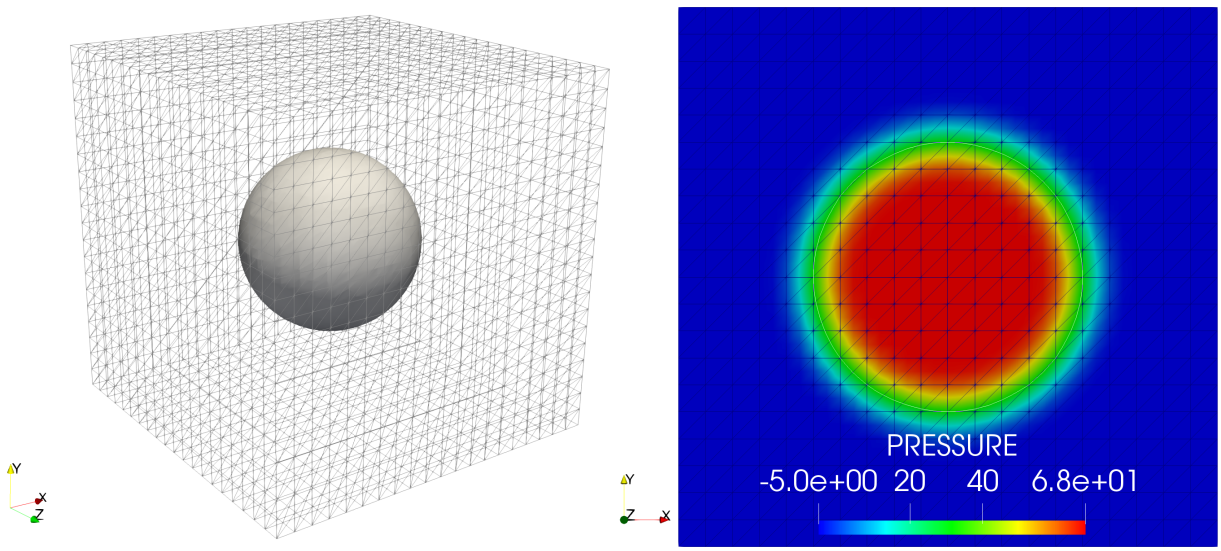
residual method preconditioned using an incomplete LU factorization with threshold. The tolerance for this linear solver is set to 10^{-10}, the perturbation for numerical differentiation δ_{ND} is set to 10^{-8}, and the tolerance for the Newton-Raphson solver is set to 10^{-6}.
Following the notations in Ref. [56], we note NDOF the number of degrees of freedom for a given simulation. It is reminded that depending on the boundary conditions, there are at most $d+2$ DOF per node of the FE mesh. The number of time increments for a simulation is denoted as NTS, and the number of elements of the mesh as NEL.

4.1 Single static inviscid droplet (3D)

This first set of simulations is conducted to assess the efficiency of the balanced-force implementation of the CSF model as well as the accuracy of curvature computation using P2 FEs in comparison with other works [57, 24, 8, 26]. Following these works, we define the computation domain as a cube of dimensions $8 \times 8 \times 8$, discretized with meshes of $40 \times 40 \times 40$ and $20 \times 20 \times 20$ elements. A spherical droplet of radius $R = 2$ is placed at the center of the computation domain as shown in Fig. 1(a) and no-slip conditions are applied at all the computation domain boundaries. Non-dimensionalized fluid and flow properties $\rho_1 = 1$, $\rho_2 = 10^{-1}$, $\mu_1 = \mu_2 = 0$, $g = 0$, $\sigma_s = 73$ are used. The simulation is conducted for 50 time increments using a time step $\Delta t = 0.001$. Under these conditions, the inviscid droplet should remain static and the pressure difference Δp between outside and inside the droplet should converge to that given by the Young-Laplace equation:

$$\Delta p = \sigma_s \kappa_s = 73, \text{ with } \kappa_s = \frac{2}{R} = 1.$$

As an error measure, we use the maximum Euclidean norm of the velocity vector in the domain $\max_{\mathbf{x}\in\Omega}(||v(\mathbf{x},t)||_2)$, after one time increment and after 50. At the same instants, we also report the error of the difference between the highest and the lowest pressures $\text{Error}(\Delta p_{max}(t))$.



(a) (b)

Figure 1: 3D single static inviscid droplet: (a) uniform 40^3 mesh with droplet placed at its center, (b) pressure field computed at $t = 0.050$ using the 20^3 uniform P2 mesh and exact curvature.

First, using the exact curvature in the numerical scheme, both errors should be below the tolerance used for the Newton-Raphson scheme, owing to the balanced-force implementation of the CSF model. In this

case, we can also conduct the simulation using equal-order P1 FEs, because we do not need to compute the curvature. After 50 time increments, errors might increase depending on the accuracy of the numerical method, in particular the LS reinitialization method.
Results are reported in the first part of Tab. 1. At equal NDOF, both errors of pressure and of velocity are below the tolerance used for the Newton-Raphson scheme, even after 50 time increments, regardless of the order of the FE method. This result is in agreement with the theory and validates our balanced-force implementation. Contrary to other works [26], we cannot reduce this error down to machine precision, which is mainly due to the numerical differentiation that uses $\delta_{ND} = 10^{-8}$.
Note that in Tab. 1 we have also reported the regularization thickness ϵ. There can be significant oscillations of the pressure field near the interface if ϵ is much smaller than the mesh size. For instance, the pressure field obtained using exact curvature and a value of ϵ corresponding to a regularization over four elements is shown in Fig. 1(b).

Mesh	40^3	20^3	40^3	20^3
ϵ	0.4	0.8	0.4	0.8
Curvature	Exact	Exact	Numerical	Numerical
FE	P1	P2	P2	P2
NDOF	315,799	315,799	2,541,999	315,799
Error($\Delta p_{max}(t = 0.001)$)	$< 10^{-8}$	$< 10^{-8}$	5.27×10^{-1}	1.44×10^{0}
Error($\Delta p_{max}(t = 0.050)$)	1.64×10^{-7}	3.60×10^{-8}	1.31×10^{1}	2.72×10^{0}
$\max_{\mathbf{x}\in\Omega}(\|\|v(\mathbf{x}, t = 0.001)\|\|_2)$	$< 10^{-8}$	$< 10^{-8}$	3.33×10^{-2}	2.32×10^{-2}
$\max_{\mathbf{x}\in\Omega}(\|\|v(\mathbf{x}, t = 0.050)\|\|_2)$	8.61×10^{-8}	4.43×10^{-8}	1.07×10^{1}	1.75×10^{0}

Table 1: 3D single static inviscid droplet simulation results for different FE meshes and orders, using either exact or numerical curvature. Errors below $\delta_{ND} = 10^{-8}$ are not reported.

Second, using numerically computed curvature, errors will depend on the interpolation error of the LS function and its second derivatives. In the literature, most authors rely on derivatives recovery techniques which can smooth the mean curvature [57, 24, 25, 26, 3]. A higher-order and higher-regularity isogeometric analysis method has also been proposed [8]. Here, we use a P2 discretization of the LS function without derivatives recovery nor smoothing technique.
Results are reported in the second part of Tab. 1. There are clearly numerical errors in the computation of the mean curvature, which may significantly affect simulation results depending on the discretization at the vicinity of the interface. These errors accumulate with time steps and are hence higher after 50 steps. This error accumulation effect seems to be alleviated with a second order time stepping scheme [8] which could be worth considering.
In the literature [57], various derivatives recovery and smoothing techniques using the finite difference method have been reported to lead to errors of the velocity after one time increment between 8.55×10^{-2} and 3.49×10^{-1} for the 40^3 mesh. After 50 time increments, those errors have been reported to vary between 3.86×10^{-1} and 2.55×10^{0} depending on the smoothing technique. These works did not employ a balanced-force algorithm. For example, the velocity errors between 4.02×10^{-3} and 4.87×10^{-3} after one time increment, and between 4.02×10^{-2} and 1.63×10^{-1} after 50 time increments have been reported in Ref. [24] using the finite difference method with different smoothing techniques for the 40^3 mesh. Overall, our results after one time increment are better than those obtained with non balanced-force implementations, whereas they are not as accurate as those reported in Ref. [24]. It seems that the balanced-force implementation is necessary even though it is not sufficient to eliminate spurious currents. Thus, the balanced-force implementation should be combined with curvature smoothing techniques.
This speculation is confirmed by the fact that our errors are at least two orders superior to those reported in Ref. [8] using equal-order P1 FEs and a projection technique to recover the mean curvature. Although such technique would make the numerical differentiation in our fully implicit scheme non-local and hence

more difficult to implement, it may be worth considering in the future.
Ref. [8] reported that isogeometric analysis with its higher-order basis functions led to a more accurate mean curvature computation but did not lead to a reduction of spurious currents, compared to a P1 discretization. This is also confirmed by the results using numerical curvature in Tab. 1.
However, higher-order methods are still relevant to improve the accuracy on the pressure field. Errors after 50 time increments of 1.39×10^0 for the 40^3 mesh and 3.93×10^0 for the 20^3 mesh have been reported in Ref. [8] using the equal-order P1 FE method with projection, and 7.90×10^{-1} and 1.37×10^0 using isogeometric analysis. Our results are of the same order but we observe a drastic increase of the error after 50 time increments, which emphasizes again the need to improve the time stepping scheme in future developments.

As a conclusion, this first set of simulations validates our balanced-force implementation of the CSF model, because all computations using exact mean curvature lead to errors both in velocity and pressure which are below the tolerance used for the Newton-Raphson solver. Regarding computations with numerically computed mean curvature, spurious non-physical currents are not completely eliminated. However, the proposed equal-order P2 framework enables a good compromise between the accuracy of the pressure field and the simplicity of the scheme because we do not use any projection or curvature smoothing technique.

4.2 Single bubble rise (2D)

This second set of simulations considers the dynamics of bubble rise in a 2D domain. The simulation setup shown in Fig. 2 has been proposed in a benchmark for multiphase flow simulation methods [56]. The computation domain is a rectangle of dimensions 1×2, discretized with meshes of 40×80, 80×160 and 160×320. A circular bubble of radius $R = 0.25$ is initially placed at $(0.5, 0.5)$. Free-slip boundary conditions are applied at lateral boundaries, while no-slip conditions are applied at top and bottom boundaries. Two cases corresponding to two sets of non-dimensionalized fluid and flow properties [56] are used:

- for case 1, properties are $\rho_1 = 1000$, $\rho_2 = 100$, $\mu_1 = 10$, $\mu_2 = 1$, $g = 0.98$, $\sigma_s = 24.5$,
- and for case 2, $\rho_1 = 1000$, $\rho_2 = 1$, $\mu_1 = 10$, $\mu_2 = 0.1$, $g = 0.98$, $\sigma_s = 1.96$.

Case 1 corresponds to the Reynolds number of Re=35 and the Eötvös number of Eo=10, while case 2 corresponds to Re=35 and Eo=125. For the definitions of these dimensionless numbers in this particular setup, the reader is referred to Ref. [56]. For each case, we measure and report the relative L^2 errors of the bubble area V_b, the bubble center of mass y coordinate y_b and the bubble rise velocity v_b integrated over time. Those parameters are respectively given by

$$V_b(t) = \int_\Omega H(-\phi(\mathbf{x}, t))\mathrm{d}\Omega,$$

$$y_b(t) = \frac{\int_\Omega H(-\phi(\mathbf{x}, t))\mathbf{x}\mathrm{d}\Omega}{V_b(t)}.\mathbf{e}_y,$$

$$v_b(t) = \frac{\int_\Omega H(-\phi(\mathbf{x}, t))\mathbf{v}(t)\mathrm{d}\Omega}{V_b(t)}.\mathbf{e}_y.$$

The relative L^2 error integrated over time is

$$\mathrm{Error}(q) = \sqrt{\frac{\int_0^T (q(t) - q_{ref}(t))^2 \mathrm{d}t}{\int_0^T (q_{ref}(t))^2 \mathrm{d}t}},$$

where $T = 3$ and $q = V_b, y_b,$ or v_b. Simulations with different space (NDOF) and time (NTS) discretizations are conducted with and without anisotropic mesh adaption. The reference solution $V_{b,ref}$ is πR^2, as the flow is incompressible. Since we do not have analytical solutions for $y_{b,ref}$ and $v_{b,ref}$, we use the numerical

solutions corresponding to the simulation with best V_b accuracy for each case.
The asymptotic convergence rate with respect to the square root of NDOF is 3, as explained in Subsec. 3.4, while the asymptotic convergence rate with respect to NTS is 1. Even though the numerical scheme is unconditionally stable, these optimal convergence rates will not be reached if the mesh size is too small compared to the time step. In Ref. [56], convergence rates corresponded to a simultaneous reduction of the mesh size and the time step. Furthermore, for a given NDOF, increasing NTS may not necessarily increase the accuracy as more time increments mean more LS reinitializations. Indeed, the LS reinitialization method (Subsec. 3.3) slightly shifts the interface at each reinitialization.

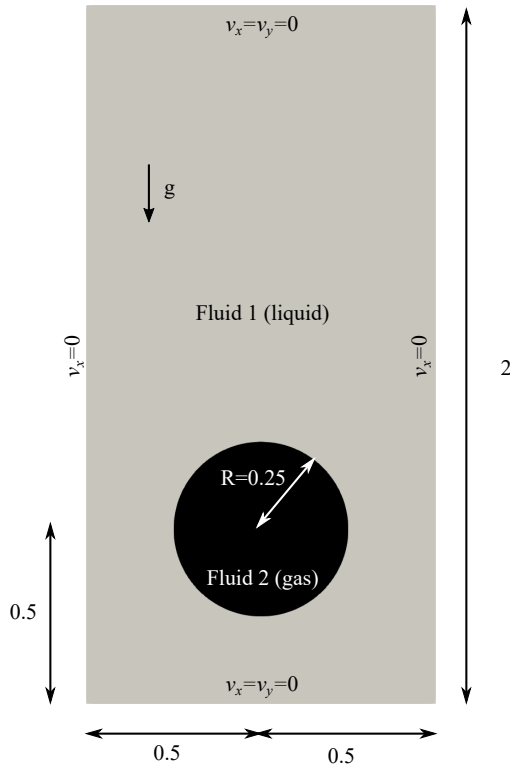
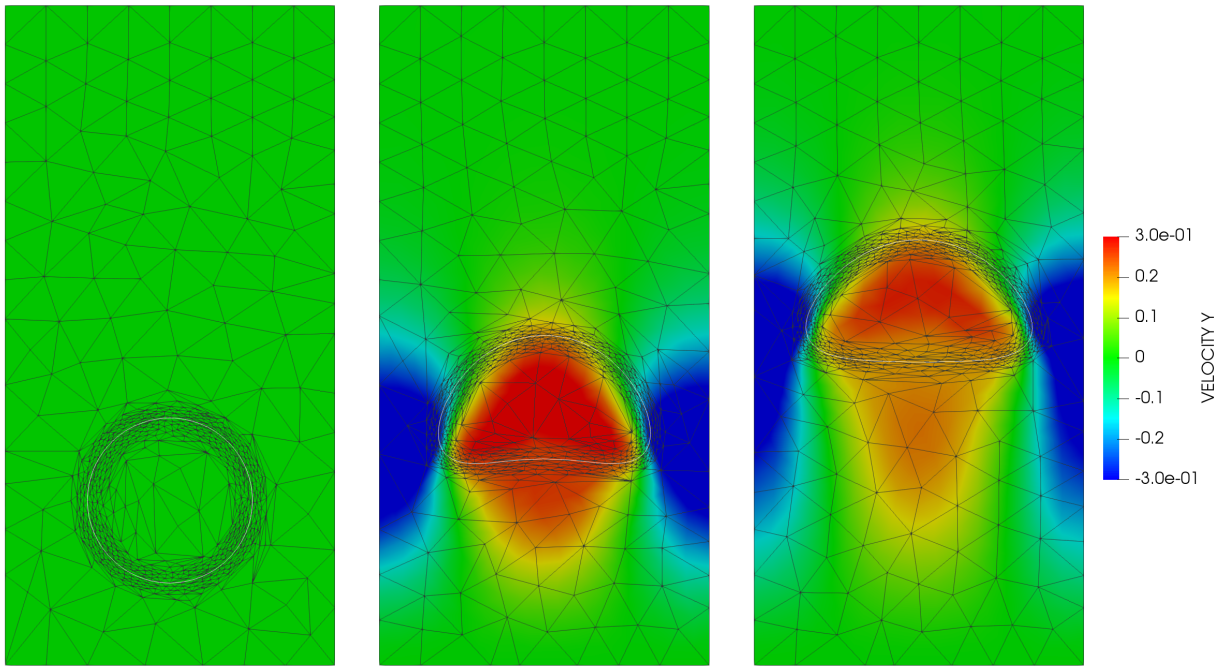


Figure 2: Setup for the 2D single bubble rise simulations.

4.2.1 Case 1

In case 1, the bubble which was initially circular becomes nearly elliptical with a flat bottom surface. One of our solutions computed using an adaptive mesh is shown in Fig. 3. Anisotropic mesh adaption is particularly relevant for this kind of interface morphology because elements can be stretched along directions where the third derivatives of the LS function are close to zero, to reduce the computational cost. This can be seen in Fig. 3(c) in the lower part of the bubble. This anisotropic stretching of elements is also relevant to control the computational cost, because there is a longer perimeter to cover in Fig. 3(c) than in Fig. 3(a), while the number of elements is almost the same, as explained in Subsec. 3.4.
A closer look at the final shapes of the bubble for some of the simulations with adaptive mesh is presented in Fig. 4. A good convergence is obtained by dint of the fine mesh which is maintained near the interface, using anisotropic mesh adaption.

Errors in the computation of the parameters of interest are reported in Tab. 2. Although NTS was prescribed to be equal to 100, 200, 400 and 800, the Newton-Raphson problem was too stiff for the simulations



(a) (b) (c)

Figure 3: Velocity field in rise direction $\mathbf{v}.\mathbf{e}_y$ for the 2D single bubble rise case 1 simulation using an adaptive mesh with NDOF $\approx$ 11500 and NTS = 407 at: (a) $t = 0$, (b) $t = 1.5$, (c) $t = 3.0$.

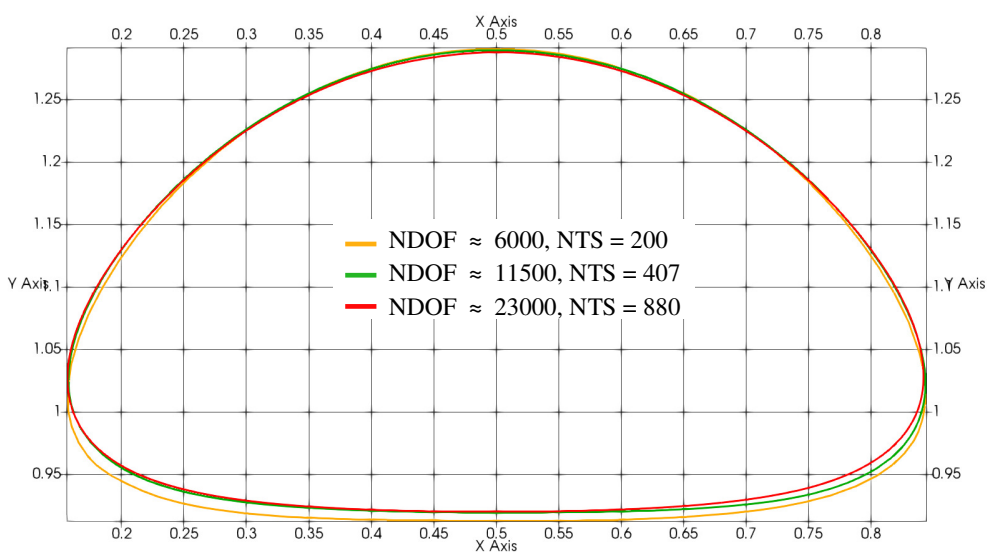


Figure 4: Liquid-gas interface for some of the 2D single bubble rise case 1 simulations with adaptive mesh at $t = 3$.

with very fine meshes for the given time step. Thus, the time step was automatically decreased to obtain a solution as explained in Subsec. 3.2. The most extreme case was the simulation with uniform 80×160 mesh and a prescribed NTS of 100 which was automatically increased to nearly 200. Results from this simulation were not reported in Tab. 2 to avoid any confusion.

Mesh	20×40	40×80	80×160	Adaptive	Adaptive	Adaptive
ϵ	0.1	0.05	0.025	0.05	0.025	0.0125
NEL	1,600	6,400	25,600	$\approx$ 740	$\approx$ 1,400	$\approx$ 2,800
NDOF	12,962	51,522	205,442	$\approx$ 6,000	$\approx$ 11,500	$\approx$ 23,000
NTS	102	100		100	104	165
Error(V_b)	5.45×10^{-2}	4.79×10^{-2}		4.89×10^{-2}	4.68×10^{-2}	4.32×10^{-2}
Error(y_b)	4.45×10^{-3}	2.46×10^{-3}		2.13×10^{-3}	1.81×10^{-3}	3.41×10^{-3}
Error(v_b)	8.05×10^{-2}	3.01×10^{-2}		2.74×10^{-2}	1.74×10^{-2}	1.62×10^{-2}
NTS	207	200	200	200	204	261
Error(V_b)	3.30×10^{-2}	2.40×10^{-2}	2.33×10^{-2}	2.05×10^{-2}	2.13×10^{-2}	2.14×10^{-2}
Error(y_b)	4.89×10^{-3}	2.51×10^{-3}	1.95×10^{-3}	2.99×10^{-3}	$\mathbf{7.48 \times 10^{-4}}$	1.32×10^{-3}
Error(v_b)	7.64×10^{-2}	2.44×10^{-2}	1.37×10^{-2}	2.38×10^{-2}	$\mathbf{9.25 \times 10^{-3}}$	9.59×10^{-3}
NTS	406	400	400	400	407	463
Error(V_b)	2.71×10^{-2}	1.06×10^{-2}	1.10×10^{-2}	2.25×10^{-3}	6.18×10^{-3}	8.22×10^{-3}
Error(y_b)	7.45×10^{-3}	2.24×10^{-3}	1.85×10^{-3}	1.52×10^{-3}	$\mathbf{4.51 \times 10^{-4}}$	1.86×10^{-3}
Error(v_b)	8.39×10^{-2}	2.12×10^{-2}	1.05×10^{-2}	1.60×10^{-2}	$\mathbf{5.07 \times 10^{-3}}$	8.04×10^{-3}
NTS	802	800	800	800	804	880
Error(V_b)	3.10×10^{-2}	$\mathbf{1.99 \times 10^{-3}}$	3.92×10^{-3}	2.11×10^{-2}	$\mathbf{7.17 \times 10^{-3}}$	$\mathbf{1.67 \times 10^{-3}}$
Error(y_b)	1.03×10^{-2}	$\mathbf{9.72 \times 10^{-4}}$	2.05×10^{-3}	1.49×10^{-3}	9.10×10^{-4}	
Error(v_b)	7.80×10^{-2}	1.71×10^{-2}	9.20×10^{-3}	1.48×10^{-2}	7.09×10^{-3}	

Table 2: Relative L^2 errors for 2D single bubble rise case 1 simulations using different space (NDOF) and time (NTS) discretizations, with and without anisotropic mesh adaption. All results with convergence rate with respect to the square root of NDOF higher than 2.5 are highlighted in bold.

As shown in the left part of Tab. 2 where no mesh adaption was used, there is a compromise to find between the time discretization and the space discretization. Optimal convergence rates in V_b and y_b are reached for instance for the intermediate 40×80 mesh using the finest time discretization (highlighted in bold in Tab. 2). This confirms that if the mesh is not fine enough, interface shifts due to LS reinitialization become the dominant source of error and lead to an increased error with smaller time steps.

Overall good mass conservation is obtained because of the use of a P2 LS function. This is drastically improved using an adaptive mesh. For instance, using an adaptive mesh with NDOF $\approx 6,000$ (right part of Tab. 2), the error of V_b for NTS = 400 is lower than the errors obtained using a uniform mesh with same NTS, even using the finest uniform mesh. A factor of reduction in terms of NDOF close to 35 is hence achieved.

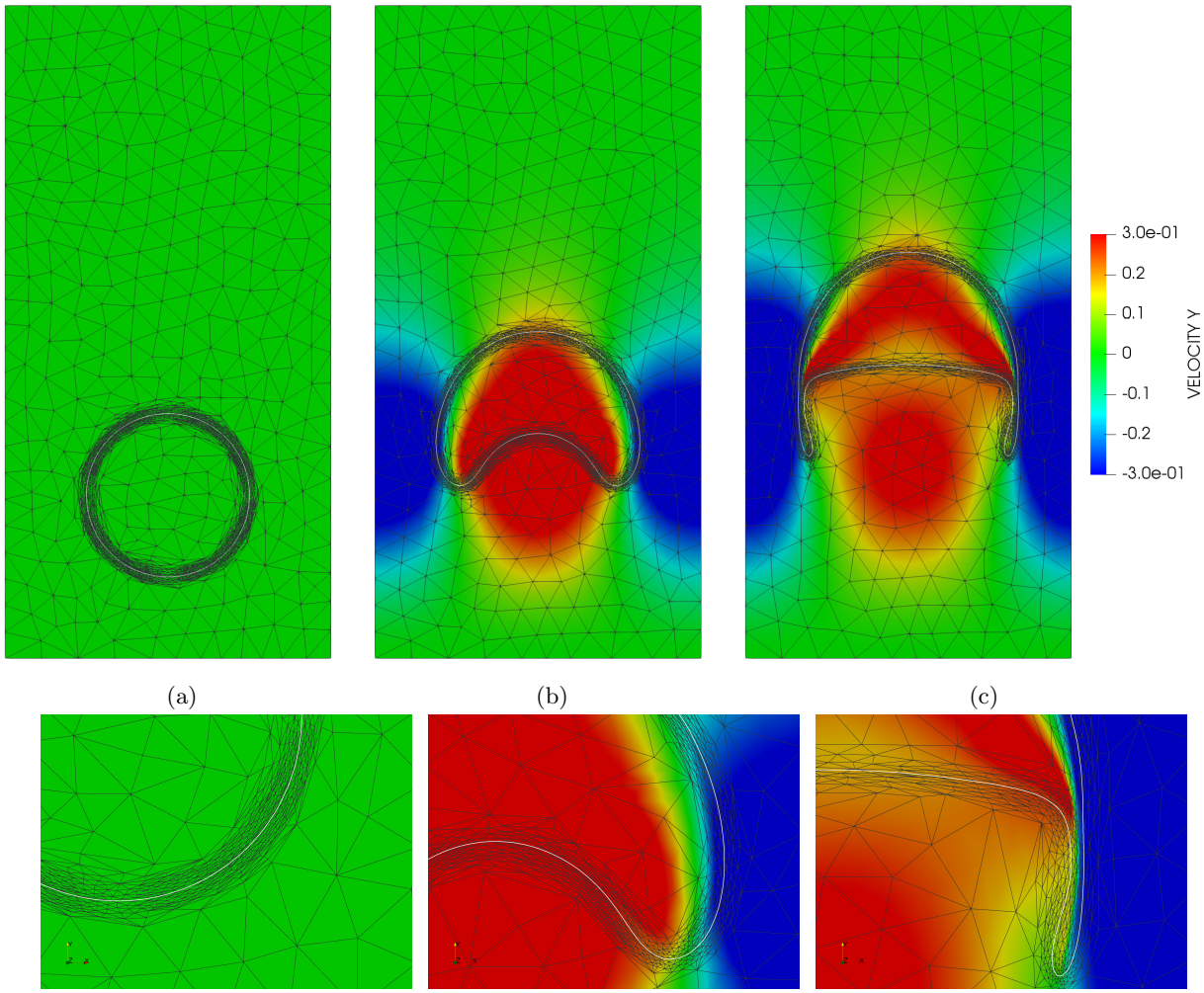
Anisotropic mesh adaption is also relevant to reach optimal convergence rates for a lower NDOF than uniform meshes. This can be seen in the last line (NTS $\approx$ 800) of Tab. 2 where an optimal convergence rate on V_b is obtained using an adaptive mesh with NDOF $\approx 11,500$. Using an adaptive mesh with NDOF $\approx 11,500$ and NTS = 204, all errors are lower than the errors obtained using a uniform mesh with same NTS, which results in a factor of reduction in terms of NDOF close to 15.

For case 1, there was very good agreement between the simulation results obtained with the different computational methods assessed in the benchmark [56]. This is also verified for our results. A more thorough comparison is proposed in the following for case 2 where significantly different results were obtained in the benchmark depending on the computational method.

4.2.2 Case 2

In case 2, thin bubble filaments develop at lateral extremities of the bubble and distort significantly, while the main part of the bubble remains nearly elliptical with a flat bottom surface. One of the solutions obtained using an adaptive mesh is shown in Fig. 5. The great advantage of anisotropic mesh adaption is demonstrated once again, because both thin filaments are captured and tracked by the adaptive mesh, as shown in Fig. 5(f).

A closer look at the final shapes of the bubble in some simulations is presented in Fig. 6. Convergence is more difficult for this case as the filaments become thinner and thinner with a smaller mesh size and a smaller ϵ. This change of thickness of the bubble filaments with finer meshes affects slightly the whole shape of the bubble. For the simulation with the highest NDOF and NTS, the lateral extremities of the bubble seem to transform into new bubbles which could potentially emerge and breakup.



(a) (b) (c)

(d) (e) (f)

Figure 5: Velocity field in rise direction $\mathbf{v}.\mathbf{e}_y$ for the 2D single bubble rise case 2 simulation using an adaptive mesh with NDOF $\approx$ 23000 and NTS = 407 at: (a) $t = 0$, (b) $t = 1.5$, (c) $t = 3.0$. Zoomed images on the adapted mesh at: (d) $t = 0$, (e) $t = 1.5$, (f) $t = 3.0$.

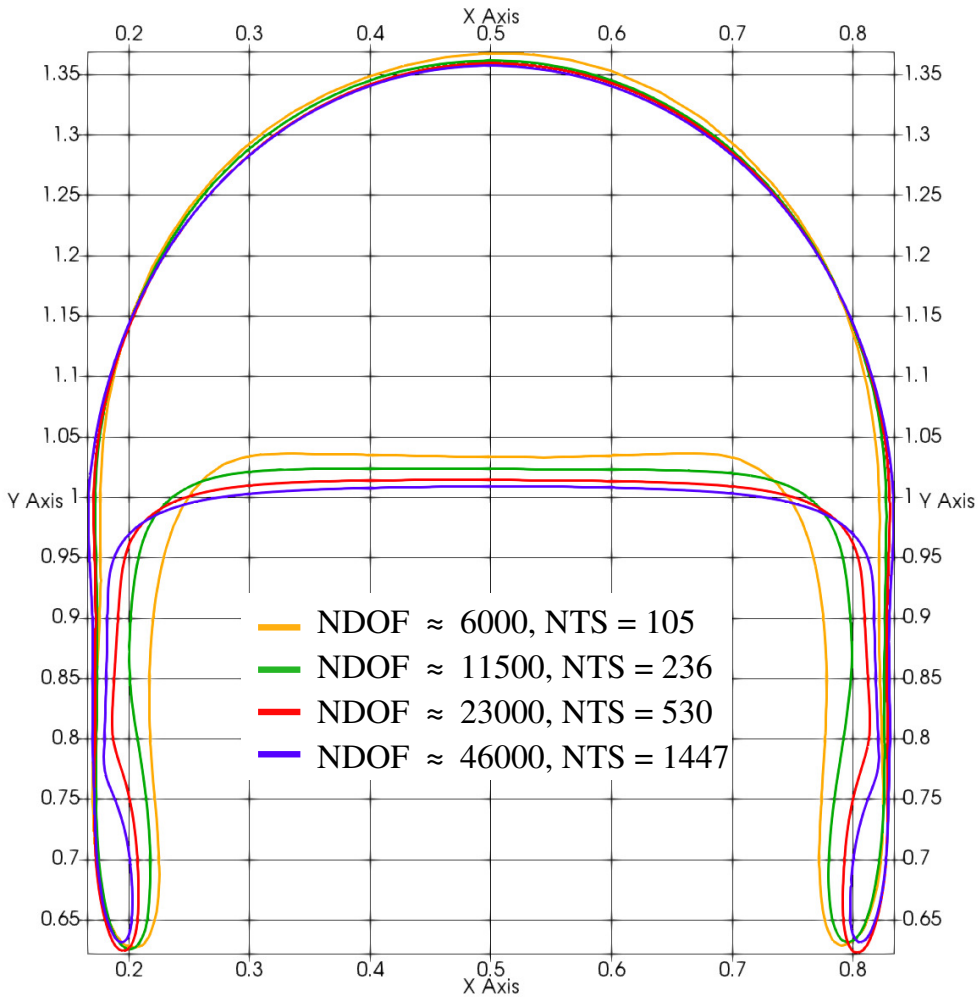


Figure 6: Liquid-gas interface for some of the 2D single bubble rise case 2 simulations with adaptive mesh at $t = 3$.

Some codes assessed in the benchmark predicted a breakup of the thin filaments [56]. This could be clearly observed by looking at an additional parameter of interest, *viz.* the bubble circularity S_b which is defined as

$$S_b(t) = \frac{2\pi}{V_b(t)} \sqrt{\frac{\int_\Omega \delta(-\phi(\mathbf{x}, t))\mathrm{d}\Omega}{\pi}}.$$

After the point of breakup, some codes did not show clear convergence on circularity curves and significant differences were found when comparing curves computed using different codes. We provide an explanation of those differences in the following.

The TP2D code used the LS method with bilinear FEs for the LS function and a second order time stepping scheme with a weak coupling between Navier-Stokes equations and the LS advection equation. The FreeLIFE code also used the LS method with a second order time stepping scheme and a weak coupling, but with linear FEs for the LS function. An additional algorithm was implemented to ensure global mass conservation. The MooNMD code used linear FEs with conformal meshing of the interface and a second order time stepping scheme with a weak coupling. An Arbitrary Lagrangian-Eulerian (ALE) method was implemented to achieve local mass conservation better. See Ref. [56] for more references on each code and how they were used in the benchmark.

Based on these facts, the code that could theoretically achieve the best performance regarding local mass conservation was the MooNMD code owing to its ALE method. It is also the only code that could reach a clear convergence on circularity curves for the numerical parameters used in the benchmark. Our method can achieve a similar performance by virtue of the use of a P2 LS function and anisotropic mesh adaption. This is demonstrated by looking at the errors of V_b in Tab. 3 and for case 1 in Tab. 2.

Neither the MooNMD code nor our method predict breakup of the filaments. Moreover, as shown in Fig. 7(a), the circularity evolution predicted by our method converges at $t = 3$ to a value a little greater than 0.5 which is similar to that predicted by the MooNMD code [56]. Our results are also in agreement with other studies where no breakup of the filaments was observed for this benchmark case [58, 59, 60].

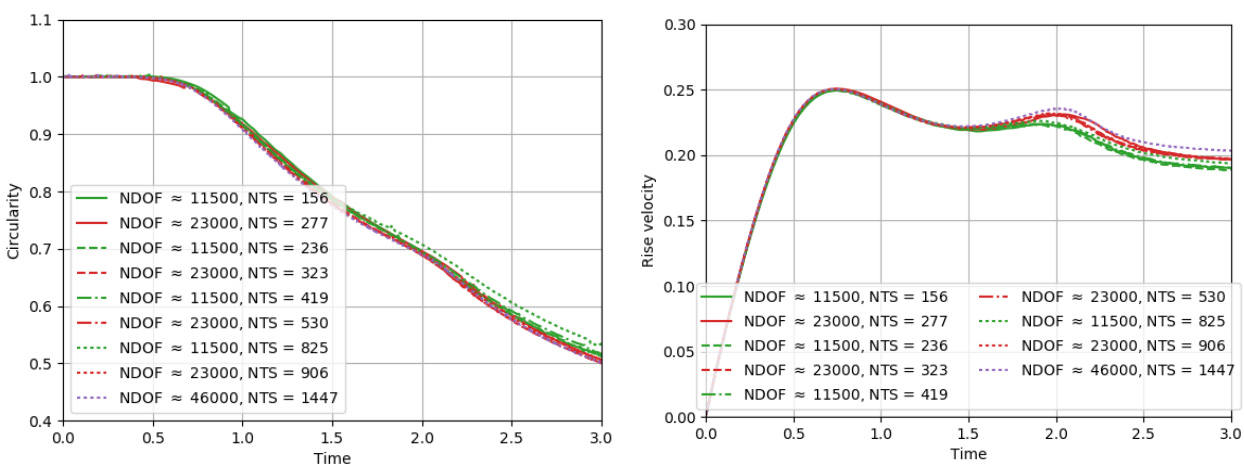
The accuracy of the velocity field predicted by our method could still be improved, as shown in Fig. 7(b). The rise velocity converges to a final value above 0.2, which matches with the results found in the benchmark for all codes. The convergence is nevertheless not as fast as for the circularity, as shown in Tab. 3 where the results with optimal convergence rate are highlighted in bold. We do not obtain any convergence rate on the square root of NDOF which is close to three for the rise velocity. A multi-criterion error estimator for anisotropic P2 mesh adaption which would take into account the velocity field should hence be considered in future developments.

We note however that we used NTSs which were lower than that used by the MooNMD code (between 3,000 and 6,000). This is a limitation of the ALE approach, which is avoided with the Eulerian method used herein with anisotropic mesh adaption. Owing to the P2/P2/P2 scheme, we can reach a high NDOF with a relatively low NEL compared to that used by all codes assessed in the benchmark. This is of particular interest for our approach regarding mesh adaption, because the computational cost of mesh adaption is related to NEL (Subsec. 3.4).

Following the diagonal in Tab. 3, the total computation times on a machine with a 20-core *Intel Xeon E5-2630 v4 @ 2.20GHz* CPU and a 3072-core *NVIDIA GTX TITAN X* GPU were about 10min, 28min, 1h28min and 4h32min, respectively. For all case 1 and case 2 simulations, the computation time was mainly spent on the Newton-Raphson solution of the RBVMS formulation whereas the computational cost of LS reinitialization was negligible (smaller than 1%). For the simulations with an adaptive mesh, the computation time of error estimation, SPR, metric computation, mesh adaption and fields transfer represented between 1 and 15% of the total computation time. In addition, we did not observe any significant mass loss due to P2 fields transfer after mesh adaption. Instead, we observed an improved mass conservation at equal NDOF with P2 mesh adaption. Therefore, we can conclude that there is a great merit in using an adaptive P2 mesh.

ϵ	0.05	0.025	0.0125	0.00625
NEL	$\approx$ 740	$\approx$ 1,400	$\approx$ 2,800	$\approx$ 5,600
NDOF	$\approx$ 6,000	$\approx$ 11,500	$\approx$ 23,000	$\approx$ 46,000
NTS	105	156		
Error(V_b)	4.38×10^{-2}	4.31×10^{-2}		
Error(S_b)	2.38×10^{-2}	1.39×10^{-2}		
Error(y_b)	1.02×10^{-2}	$\mathbf{3.61 \times 10^{-3}}$		
Error(v_b)	8.70×10^{-2}	$\mathbf{3.84 \times 10^{-2}}$		
NTS	203	236	323	
Error(V_b)	1.54×10^{-2}	1.83×10^{-2}	2.10×10^{-2}	
Error(S_b)	2.27×10^{-2}	$\mathbf{9.59 \times 10^{-3}}$	4.73×10^{-3}	
Error(y_b)	1.13×10^{-2}	5.14×10^{-3}	$\mathbf{1.52 \times 10^{-3}}$	
Error(v_b)	8.05×10^{-2}	4.12×10^{-2}	$\mathbf{1.65 \times 10^{-2}}$	
NTS	403	419	530	
Error(V_b)	3.13×10^{-3}	2.70×10^{-3}	6.79×10^{-3}	
Error(S_b)	2.00×10^{-2}	1.21×10^{-2}	$\mathbf{2.77 \times 10^{-3}}$	
Error(y_b)	1.46×10^{-2}	$\mathbf{5.45 \times 10^{-3}}$	2.37×10^{-3}	
Error(v_b)	8.34×10^{-2}	$\mathbf{3.64 \times 10^{-2}}$	1.76×10^{-2}	
NTS	804	825	906	1447
Error(V_b)	1.86×10^{-2}	8.34×10^{-3}	$\mathbf{1.64 \times 10^{-3}}$	1.19×10^{-3}
Error(S_b)	2.31×10^{-2}	2.25×10^{-2}	$\mathbf{3.39 \times 10^{-3}}$	
Error(y_b)	1.39×10^{-2}	$\mathbf{4.39 \times 10^{-3}}$	2.67×10^{-3}	
Error(v_b)	7.21×10^{-2}	$\mathbf{2.75 \times 10^{-2}}$	1.56×10^{-2}	

Table 3: Relative L^2 errors for 2D single bubble rise case 2 simulations using different space (NDOF) and time (NTS) discretizations, with anisotropic mesh adaption. All results with convergence rate with respect to the square root of NDOF higher than 2.5 are highlighted in bold.



(a) (b)

Figure 7: Circularity (a) and rise velocity (b) evolution for the 2D single bubble rise case 2 simulations using an adaptive mesh with various NDOF and NTS.

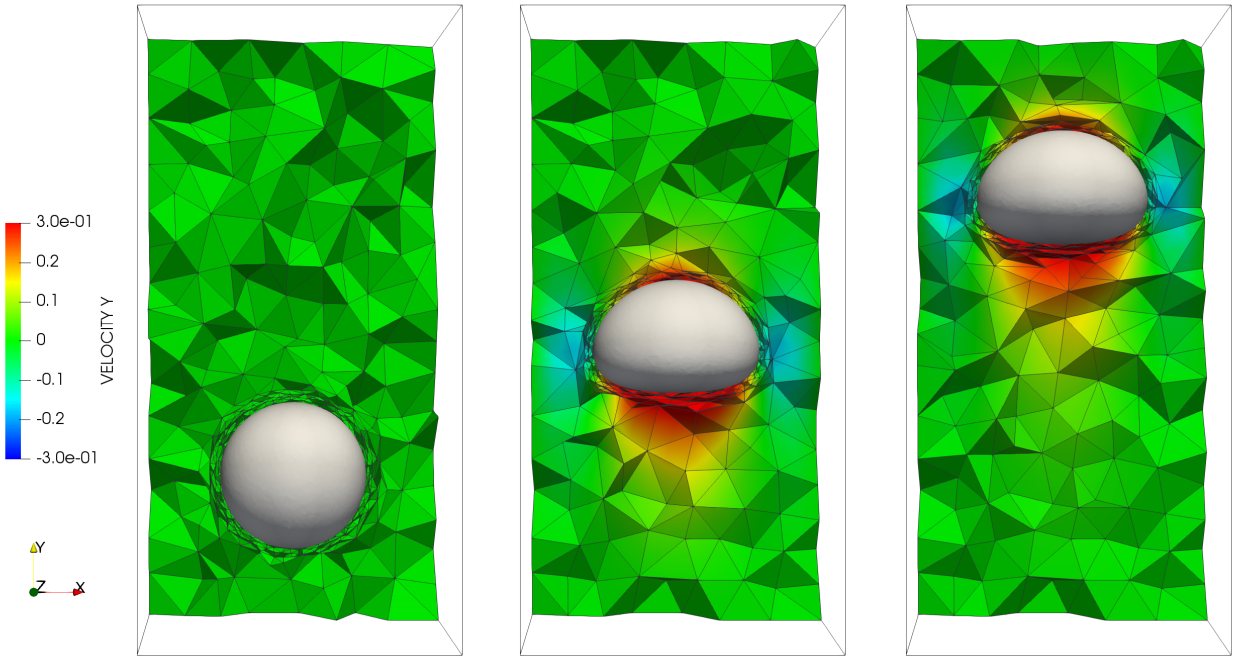
4.3 Single bubble rise (3D)

Different 3D versions of the single bubble rise benchmark have been proposed in the literature [61, 62, 63, 26]. Here we opt for a simple 3D extension where the computation domain is a box of dimensions $1 \times 2 \times 1$. A spherical bubble of radius $R = 0.25$ is initially placed at $(0.5, 0.5, 0.5)$. No-slip boundary conditions are applied at all boundaries. Fluid and flow properties with two cases are identical to those of the 2D version. This corresponds to the 3D setup used in Refs. [61, 26]. Note that 3D results are not expected to be comparable to 2D results due to the change of boundary conditions and the extra dimension.

Two discretizations are used for each case. The coarse discretization has an adaptive mesh of NEL $\approx 25,000$ and NDOF $\approx 175,000$ for a NTS ≈ 100, while the fine discretization has respectively NEL $\approx 56,000$ and NDOF $\approx 375,000$ for a NTS ≈ 200. The value of ϵ is 0.05 for the coarse discretization and 0.025 for the fine one.

The results with the fine discretization are shown for case 1 in Fig. 8 and for case 2 in Fig. 9. The final shape of the bubble for each case is similar to that obtained by other authors in the literature [61, 26]. Contrary to the 2D version, there is no excessive stretching of the bubbles into filaments that might breakup for case 2, which is also in agreement with the literature [61, 26].

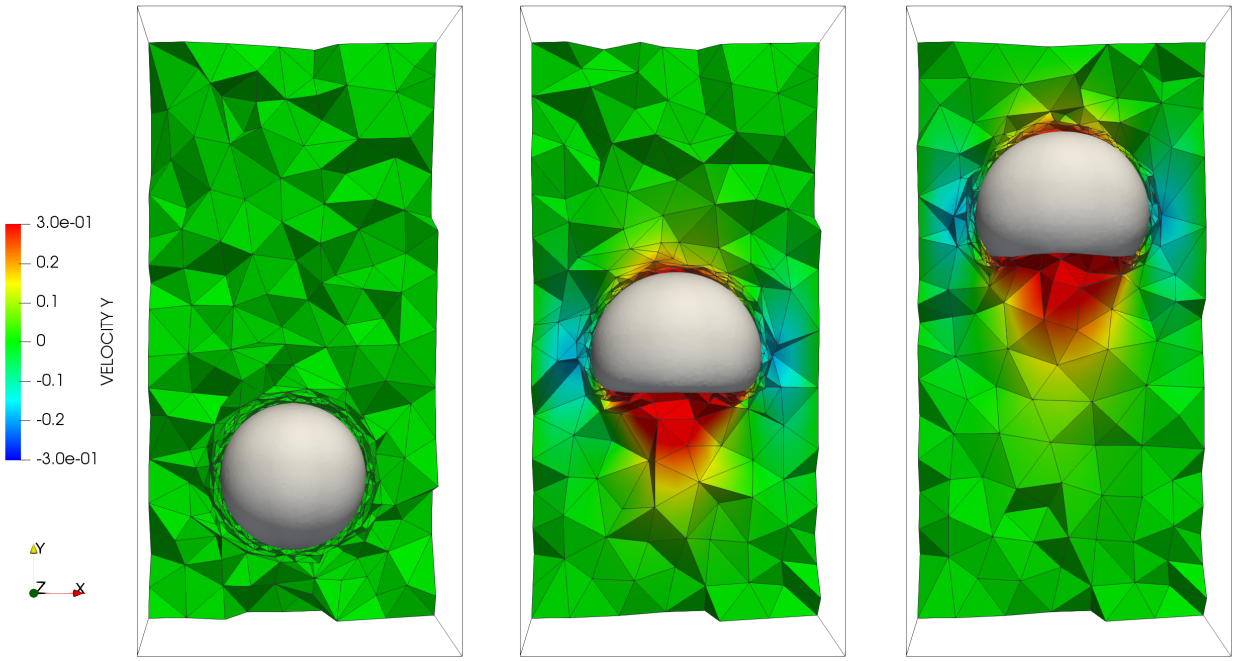
A closer look at the final shapes is shown in Fig. 10. There is clearly an overestimation of the bubble volume in both cases for the coarse discretization, which can be attributed to the low order of the time stepping scheme based on the 2D simulations. The final shapes are nevertheless smooth, even using the coarse discretization. In comparison with other results for a similar number of elements found in the literature [61, 26, 8], this result shows the great advantage of using an anisotropic adaptive P2 interpolation for the LS function.



(a) (b) (c)

Figure 8: Velocity field in rise direction $\mathbf{v}.\mathbf{e}_y$ for the 3D single bubble rise case 1 simulations using an adaptive mesh with NDOF ≈ 375000 and NTS ≈ 200 at: (a) $t = 0$, (b) $t = 1.5$, (c) $t = 3.0$.

The computed parameters of interest are the same as for the 2D case, except that $V_b(t)$ now represents



(a) (b) (c)

Figure 9: Velocity field in rise direction for the 3D single bubble rise case 2 simulations using an adaptive mesh with NDOF $\approx$ 375000 and NTS $\approx$ 200 at: (a) $t = 0$, (b) $t = 1.5$, (c) $t = 3.0$.

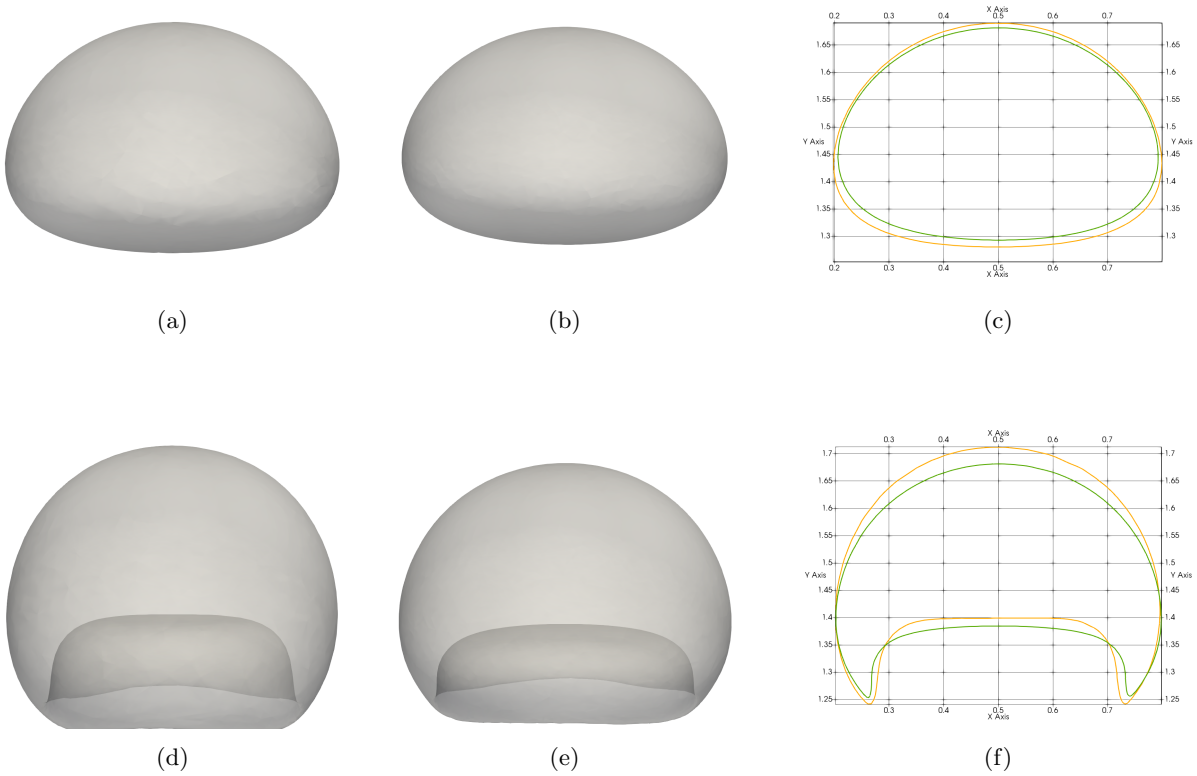
the bubble volume and the bubble circularity S_b is replaced by the bubble sphericity given by:

$$S_b(t) = \frac{4\pi}{V_b(t)} \left(\sqrt[3]{\frac{3}{4\pi} \int_\Omega \delta(-\phi(\mathbf{x}, t)) \mathrm{d}\Omega} \right)^2 .$$

Time evolution of each parameter of interest is reported for both cases in Fig. 11. For case 1, the errors of V_b are 1.32×10^{-1} for the coarse discretization and 5.43×10^{-2} for the fine discretization, while for case 2 they are respectively 1.38×10^{-1} and 5.65×10^{-2}. As expected, the errors are slightly higher for case 2, while this difference is quite low compared to the differences between the two simulation results shown in Fig. 10. This proves that our numerical method is robust and can achieve a similar accuracy for very different bubble deformations.

The error of y_b is very low for both cases, as shown in Fig. 11(c,d) where the overall position of the bubble is well captured. In order to describe the bubble's shape, a finer discretization is required, as shown by the sphericity curves in Fig. 11(a,b). Small oscillations can be seen for case 1 in Fig. 11(a), which are due to local mass loss during fields transfer after mesh adaption. Similar oscillations can be seen for case 2 in Fig. 11(b). Their amplitude is clearly low, however compared to the overall evolution of the sphericity. The rise velocity is underestimated with the coarse discretization, as shown in Fig. 11(e,f). This was also observed in Ref. [26]. The differences between the two discretizations are however quite low. Both the instant at which the maximum rise velocity is reached and the value of the maximum rise velocity are well predicted with the coarse discretization.

All 3D single bubble rise simulations were run on a machine with a 20-core *Intel Xeon E5-2630 v4 @ 2.20GHz* CPU, and a 3072-core *NVIDIA GTX TITAN X* GPU. The total computation time was about 9h for coarse discretization simulations and 34h for fine discretization simulations. About 65% of this computation



(a) (b) (c)

(d) (e) (f)

Figure 10: Shapes with transparent interface at $t = 3$ for the 3D single bubble rise simulations: (a) case 1 using the coarse discretization, (b) case 1 using the fine discretization, (d) case 2 using the coarse discretization, (e) case 2 using the fine discretization. Midsection of the interface using the coarse (orange) and fine (green) discretizations: (c) superposition of (a) and (b) for case 1, (f) superposition of (d) and (e) for case 2.

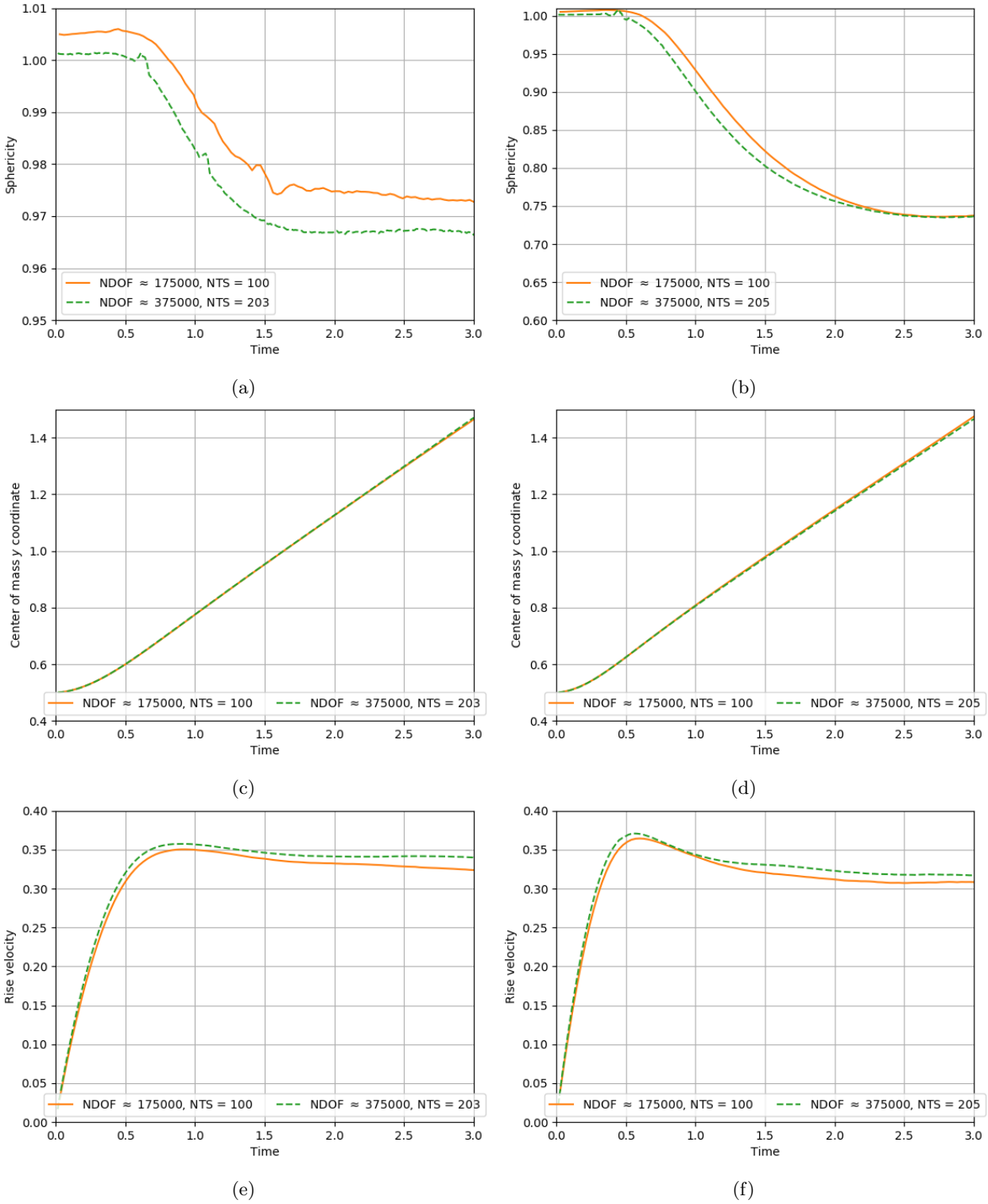


Figure 11: 3D single bubble rise simulation results using an adaptive mesh with various NDOF and NTS. Case 1: (a) sphericity, (c) center of mass y coordinate, (e) rise velocity. Case 2: (b) sphericity, (d) center of mass y coordinate, (f) rise velocity. Scales may differ from left to right.

time was spent on Newton-Raphson solution, 7% on the mesh adaption algorithm, 23% on fields transfer after mesh adaption and 5% on LS reinitialization. The computation time of the other operations such as error estimation and the SPR operation was negligible.
The computational cost caused by the modification of the mesh at each time increment was clearly more significant for these 3D simulations. However, this was mainly due to the fields transfer operation and in particular to the search of the containing element of the old mesh for each node of the new mesh. This could be significantly reduced by the implementation of a parallel version of this search on the GPU, as we did for LS reinitialization.
The increase in the computational cost by anisotropic P2 mesh adaption is marginal and acceptable given the drastic improvements of accuracy in the 2D simulations.

5 Conclusions

Incompressible two phase flows with surface tension were modeled through the Navier Stokes equations, the Level-Set (LS) method and the Continuum Surface Force (CSF) model. The equal-order piece-wise quadratic (P2) finite element discretization was used for the simultaneous solution of the velocity, the pressure and the LS function. The Navier-Stokes equations with the CSF model and the LS advection equation were strongly coupled in a residual-based variational multiscale formulation and solved using a Newton-Raphson scheme with numerical differentiation. The fully implicit formulation was shown to be robust and stable even for large time steps.
A balanced-force implementation was used for the CSF model. This implementation could be achieved in a straightforward manner with no need to rely on the numerically complex projections of pressure gradient, because the same discretization was used for the pressure field and the regularized Heaviside function. This implementation was shown to reduce errors of both the velocity and pressure fields below the tolerance used for the Newton-Raphson solver on a static problem using exact curvature.
The original approach based on anisotropic P2 mesh adaption was proposed to automatically refine, stretch and orient elements along the interface. As the interface was represented by the zero isolevel of an LS function, the new error estimator targeting the P2 interpolation error of the regularized Heaviside function was developed to define a directional error metric field. The continuous mesh framework was used to convert local directional errors into local directional mesh sizes with a control of the global complexity. This strategy was shown to be more accurate than a uniform fixed mesh for a same number of degrees of freedom, and to enable reductions of the number of degrees of freedom with a factor up to 15 for a same accuracy. In addition, the convergence rates and the mass conservation properties could be improved by virtue of P2 interpolation.

Acknowledgments

The authors would like to acknowledge the financial support of the French Agence Nationale de la Recherche to COMP3DRE project (grant number: Projet-ANR-16-CE08-0042).

References

[1] R. Nakhoul, P. Laure, L. Silva, and M. Vincent, "A multiphase Eulerian approach for modelling the polymer injection into a textured mould," International Journal of Material Forming, vol. 11, pp. 53–66, jan 2018.

[2] M. Mehdikhani, L. Gorbatikh, I. Verpoest, and S. V. Lomov, "Voids in fiber-reinforced polymer composites: A review on their formation, characteristics, and effects on mechanical performance," Journal of Composite Materials, p. 002199831877215, nov 2018.

[3] J. Yan, W. Yan, S. Lin, and G. Wagner, "A fully coupled finite element formulation for liquid–solid–gas thermo-fluid flow with melting and solidification," Computer Methods in Applied Mechanics and Engineering, vol. 336, pp. 444–470, jul 2018.

[4] A. N. Brooks and T. J. 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, vol. 32, pp. 199–259, sep 1982.

[5] R. Codina, "Stabilized finite element approximation of transient incompressible flows using orthogonal subscales," Computer Methods in Applied Mechanics and Engineering, vol. 191, pp. 4295–4321, aug 2002.

[6] T. J. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy, "The variational multiscale method—a paradigm for computational mechanics," Computer Methods in Applied Mechanics and Engineering, vol. 166, pp. 3–24, nov 1998.

[7] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, and G. Scovazzi, "Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows," Computer Methods in Applied Mechanics and Engineering, vol. 197, pp. 173–201, dec 2007.

[8] J. Yan, S. Lin, Y. Bazilevs, and G. Wagner, "Isogeometric analysis of multi-phase flows with surface tension and with application to dynamics of rising bubbles," Computers & Fluids, vol. 0, pp. 1–13, apr 2018.

[9] M. Shakoor, P.-O. Bouchard, and M. Bernacki, "An adaptive level-set method with enhanced volume conservation for simulations in multiphase domains," International Journal for Numerical Methods in Engineering, vol. 109, pp. 555–576, jan 2017.

[10] J. Brackbill, D. Kothe, and C. Zemach, "A continuum method for modeling surface tension," Journal of Computational Physics, vol. 100, pp. 335–354, jun 1992.

[11] C. Hirt and B. Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," Journal of Computational Physics, vol. 39, pp. 201–225, jan 1981.

[12] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," Journal of Computational Physics, vol. 79, pp. 12–49, nov 1988.

[13] M. Sussman, P. Smereka, and S. Osher, "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow," Journal of Computational Physics, vol. 114, pp. 146–159, sep 1994.

[14] M. Sussman, "A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles," Journal of Computational Physics, vol. 187, pp. 110–136, may 2003.

[15] D. A. Di Pietro, S. Lo Forte, and N. Parolini, "Mass preserving finite element implementations of the level set method," Applied Numerical Mathematics, vol. 56, pp. 1179–1195, sep 2006.

[16] C. Bui, C. Dapogny, and P. Frey, "An accurate anisotropic adaptation method for solving the level set advection equation," International Journal for Numerical Methods in Fluids, vol. 70, pp. 899–922, nov 2012.

[17] D. Enright, F. Losasso, and R. Fedkiw, "A fast and accurate semi-Lagrangian particle level set method," Computers & Structures, vol. 83, pp. 479–490, feb 2005.

[18] U. Rasthofer, F. Henke, W. Wall, and V. Gravemeier, "An extended residual-based variational multiscale method for two-phase flow including surface tension," Computer Methods in Applied Mechanics and Engineering, vol. 200, pp. 1866–1876, may 2011.

[19] H. Sauerland and T.-P. Fries, "The extended finite element method for two-phase and free-surface flows: A systematic study," Journal of Computational Physics, vol. 230, pp. 3369–3390, may 2011.

[20] M. Shakoor, B. Scholtes, P.-O. Bouchard, and M. Bernacki, "An efficient and parallel level set reinitialization method – Application to micromechanics and microstructural evolutions," Applied Mathematical Modelling, vol. 39, pp. 7291–7302, dec 2015.

[21] J. A. Sethian and A. Vladimirsky, "Fast methods for the Eikonal and related Hamilton- Jacobi equations on unstructured meshes.," Proceedings of the National Academy of Sciences of the United States of America, vol. 97, pp. 5699–703, may 2000.

[22] C. Basting and D. Kuzmin, "A minimization-based finite element formulation for interface-preserving level set reinitialization," Computing, vol. 95, pp. 13–25, may 2013.

[23] E. Marchandise, P. Geuzaine, N. Chevaugeon, and J.-F. Remacle, "A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics," Journal of Computational Physics, vol. 225, pp. 949–974, jul 2007.

[24] M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, and M. W. Williams, "A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework," Journal of Computational Physics, vol. 213, no. 1, pp. 141–173, 2006.

[25] S. Zahedi, M. Kronbichler, and G. Kreiss, "Spurious currents in finite element based level set methods for two-phase flow," International Journal for Numerical Methods in Fluids, vol. 69, pp. 1433–1456, jul 2012.

[26] S. Lin, J. Yan, D. Kats, and G. J. Wagner, "A volume-conserving balanced-force level set method on unstructured meshes using a control volume finite element formulation," Journal of Computational Physics, vol. in press, nov 2018.

[27] F. Denner, F. Evrard, R. Serfaty, and B. G. van Wachem, "Artificial viscosity model to mitigate numerical artefacts at fluid interfaces with surface tension," Computers & Fluids, vol. 143, pp. 59–72, jan 2017.

[28] S. Hysing, "A new implicit surface tension implementation for interfacial flows," International Journal for Numerical Methods in Fluids, vol. 51, no. 6, pp. 659–672, 2006.

[29] F. Pochet, K. Hillewaert, P. Geuzaine, J.-F. Remacle, and É. Marchandise, "A 3D strongly coupled implicit discontinuous Galerkin level set-based method for modeling two-phase flows," Computers & Fluids, vol. 87, pp. 144–155, oct 2013.

[30] M. Sussman, E. Fatemi, P. Smereka, and S. Osher, "An improved level set method for incompressible two-phase flows," Computers & Fluids, vol. 27, pp. 663–680, jun 1998.

[31] C. Basting, D. Kuzmin, and J. N. Shadid, "Optimal control for reinitialization in finite element level set methods," International Journal for Numerical Methods in Fluids, vol. 84, pp. 292–305, jun 2017.

[32] D. H. Eberly and P. J. Schneider, Geometric Tools for Computer Graphics. Kaufmann, Morgan, 2003.

[33] J. L. Bentley, "Multidimensional binary search trees used for associative searching," Communications of the ACM, vol. 18, pp. 509–517, sep 1975.

[34] T. Park, S.-H. Lee, J.-H. Kim, and C.-H. Kim, "CUDA-based Signed Distance Field Calculation for Adaptive Grids," in 2010 10th IEEE International Conference on Computer and Information Technology, pp. 1202–1206, IEEE, jun 2010.

[35] O. C. Zienkiewicz and J. Z. Zhu, "A simple error estimator and adaptive procedure for practical engineerng analysis," International Journal for Numerical Methods in Engineering, vol. 24, pp. 337–357, feb 1987.

[36] Z. Zhang and A. Naga, "A New Finite Element Gradient Recovery Method: Superconvergence Property," SIAM Journal on Scientific Computing, vol. 26, pp. 1192–1213, jan 2005.

[37] O. Coulaud and A. Loseille, "Very High Order Anisotropic Metric-Based Mesh Adaptation in 3D," Procedia Engineering, vol. 163, pp. 353–365, 2016.

[38] Y. Belhamadia, A. Fortin, and E. Chamberland, "Anisotropic mesh adaptation for the solution of the Stefan problem," Journal of Computational Physics, vol. 194, pp. 233–255, feb 2004.

[39] T. Coupez, "Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing," Journal of Computational Physics, vol. 230, pp. 2391–2405, apr 2011.

[40] F. Alauzet, A. Loseille, A. Dervieux, and P. Frey, "Multi-Dimensional Continuous Metric for Mesh Adaptation," in Proceedings of the 15th International Meshing Roundtable (P. P. Pébay, ed.), pp. 191–214, Springer Berlin Heidelberg, 2006.

[41] C. Dobrzynski and P. Frey, "Anisotropic Delaunay Mesh Adaptation for Unsteady Simulations," in Proceedings of the 17th International Meshing Roundtable (R. V. Garimella, ed.), pp. 177–194, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[42] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-Euclidean metrics for fast and simple calculus on diffusion tensors," Magnetic Resonance in Medicine, vol. 56, pp. 411–421, aug 2006.

[43] P. Laug and H. Borouchaki, "Construction d'un champ continu de métriques," Comptes Rendus Mathematique, vol. 351, pp. 639–644, aug 2013.

[44] P. E. Farrell, S. Micheletti, and S. Perotto, "An anisotropic Zienkiewicz-Zhu-type error estimator for 3D applications," International Journal for Numerical Methods in Engineering, vol. 85, pp. 671–692, feb 2011.

[45] G. M. Porta, S. Perotto, and F. Ballio, "Anisotropic mesh adaptation driven by a recovery-based error estimator for shallow water flow modeling," International Journal for Numerical Methods in Fluids, vol. 70, pp. 269–299, sep 2012.

[46] G. Compère, E. Marchandise, and J.-F. Remacle, "Transient adaptivity applied to two-phase incompressible flows," Journal of Computational Physics, vol. 227, pp. 1923–1942, jan 2008.

[47] Z. Xie, D. Pavlidis, P. Salinas, J. R. Percival, C. C. Pain, and O. K. Matar, "A balanced-force control volume finite element method for interfacial flows with surface tension using adaptive anisotropic unstructured meshes," Computers & Fluids, vol. 138, pp. 38–50, oct 2016.

[48] A. Carabias, A. Belme, A. Loseille, and A. Dervieux, "Anisotropic goal-oriented error analysis for a third-order accurate CENO Euler discretization," International Journal for Numerical Methods in Fluids, vol. 86, pp. 392–413, feb 2018.

[49] A. Loseille and F. Alauzet, "Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error," SIAM J. Numer. Anal., vol. 49, no. 1, pp. 38–60, 2011.

[50] A. Loseille and F. Alauzet, "Continuous Mesh Framework Part II: Validations and Applications," SIAM Journal on Numerical Analysis, vol. 49, pp. 61–86, jan 2011.

[51] C. Gruau and T. Coupez, "3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric," Computer Methods in Applied Mechanics and Engineering, vol. 194, pp. 4951–4976, nov 2005.

[52] "ISO/IEC 9899:1999," tech. rep., International Organization for Standardization, 1999.

[53] "ISO/IEC 14882:2011," tech. rep., International Organization for Standardization, 2011.

[54] "OpenMP Application Program Interface Version 4.5," tech. rep., OpenMP Architecture Review Board, 2015.

[55] A. Nishida, "Experience in developing an open source scalable software infrastructure in japan," in Computational Science and Its Applications – ICCSA 2010 (D. Taniar, O. Gervasi, B. Murgante, E. Pardede, and B. O. Apduhan, eds.), pp. 448–462, Springer Berlin Heidelberg, 2010.

[56] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska, "Quantitative benchmark computations of two-dimensional bubble dynamics," International Journal for Numerical Methods in Fluids, vol. 60, pp. 1259–1288, aug 2009.

[57] N. W. Williams, D. B. Kothe, and E. G. Puckett, "Accuracy and Convergence of Continuum Surface Tension Models," in Fluid Dynamics of Interfaces, pp. 294–305, Cambridge University Press, 1998.

[58] S. Aland and A. Voigt, "Benchmark computations of diffuse interface models for two-dimensional bubble dynamics," International Journal for Numerical Methods in Fluids, vol. 69, pp. 747–761, may 2012.

[59] V. Doyeux, Y. Guyot, V. Chabannes, C. Prud'homme, and M. Ismail, "Simulation of two-fluid flows using a finite element/level set method. Application to bubbles and vesicle dynamics," Journal of Computational and Applied Mathematics, vol. 246, pp. 251–259, jul 2013.

[60] Y. Wang, C. Shu, J. Shao, J. Wu, and X. Niu, "A mass-conserved diffuse interface method and its application for incompressible multiphase flows with large density ratio," Journal of Computational Physics, vol. 290, pp. 336–351, jun 2015.

[61] M. A. Safi, N. Prasianakis, and S. Turek, "Benchmark computations for 3D two-phase flows: A coupled lattice Boltzmann-level set study," Computers and Mathematics with Applications, vol. 73, no. 3, pp. 520–536, 2017.

[62] V. Karyofylli, M. Frings, S. Elgeti, and M. Behr, "Simplex space-time meshes in two-phase flow simulations," International Journal for Numerical Methods in Fluids, vol. 86, pp. 218–230, jan 2018.

[63] T. Metivet, V. Chabannes, M. Ismail, and C. Prud'homme, "High-Order Finite-Element Framework for the Efficient Simulation of Multifluid Flows," Mathematics, vol. 6, p. 203, oct 2018.