XFEM with Hanging Nodes for Two-phase Incompressible Flow

K.-W. Cheng[1,2,*]

Aachen Institute for Advanced Study in Computational Engineering Science, RWTH Aachen University, Schinkelstr. 2, 52058 Aachen, Germany

T.-P. Fries[3]

Institute for Computational Analysis of Technical Systems, RWTH Aachen University, Schinkelstr. 2, 52058 Aachen, Germany

Abstract

This study develops the h-version of the Extended Finite Element Method (XFEM) applied to the simulation of two-fluid incompressible flow in two and three dimensions. A multilevel adaptive mesh refinement realized via hanging nodes on 1-irregular meshes is employed in the vicinity of the two-fluid interface. The sign-enrichment is used for the XFEM approximation which accurately accounts for the jump in the pressure field. The level-set method is used for the implicit representation of the interface. The Laplace-Beltrami technique is employed for the modelling of the surface tension, which avoids the explicit computation of the curvature. An emphasis of this work is on how the interplay between the interface movement (in terms of a time-dependent level-set function), the adaptive refinement and the enriched XFEM approximations, is realized. This study also demonstrates that the approximation of the normal vector to the interface, required for the computation of the surface tension, can have a significant impact on the accuracy of the solver. Several two- and three-dimensional test cases are investigated.

Keywords: XFEM, hanging nodes, two-phase flow, level-set, surface tension

1. INTRODUCTION

Fluid flows with moving interfaces encompass a myriad of physical phenomena and industrial processes such as ocean waves and liquid films in coating and drying processes.

*Corresponding author
Email address: kwok-wah-cheng@uiowa.edu (K.-W. Cheng)
[1]Present address: IIHR-Hydroscience and Engineering, University of Iowa, Iowa City, IA 52242, USA
[2]Research supported by German Research Foundation; contract/grant number: GSC 111
[3]Research supported by German Research Foundation; Emmy-Noether Programme

These phenomena are collectively known as two-fluid flow. In modelling two-fluid flow, several challenges are encountered. First, jumps in the fluid density and viscosity across the interface need to be accounted for in the numerical scheme. Second, as surface tension effects play an integral part in two-fluid flow phenomena, this necessitates an accurate modelling and discretization of the surface tension. Third, as topological changes of the interface can occur as it evolves with time, the numerical method employed must have the capability of capturing such changes as the breaking apart or merging of bubbles. Finally, a well-known problem in two-fluid flow simulations is the occurrence of unphysical spurious velocities in the vicinity of the interface. Various sources have been cited as being responsible for the existence of such parasitic currents (see e.g. [1, 2]). For example, the approximation of the jump in the pressure field, the approximation of the curvature, the approximation of the normal vectors to the interface required for the computation of the surface tension and the approximation of the geometry of the interface. It is conceivable that unphysical movements of the interface can be generated by such spurious velocities. Therefore, the development of a numerical scheme which suppresses (or at least minimizes) such parasitic currents is highly desirable.

Among the different numerical methods available for two-fluid flow simulation, the finite element (FE) method based on the weak formulation offers several intrinsic advantages over the traditional finite difference (FD) approach based on the strong formulation. First, the singular surface tension term is naturally included as a boundary condition in the weak form via a well-defined boundary integral. This is in contrast to the FD approach where the surface tension term is introduced into the right-hand side of the momentum equation as an interface-concentrated source term via a Dirac delta function. As such, numerical smoothing of the delta function is needed and this inevitably results in interface smearing and loss of accuracy [3]. Second, the strong formulation requires the computation of the derivative of the viscosity which is discontinuous across the interface. This again leads to the necessity of evaluating some kind of regularized delta function. In the finite element context, this derivative is shifted to the test function, thus precluding the need to evaluate a singular viscosity term. Other advantages of the FEM are (i) minimal regularity requirements on the dependent variables to ensure the existence of the unique solution (spatial derivatives of the dependent variables can often be shifted to the test function via integration by parts) and (ii) local mesh adaptivity which allows one to use a refined mesh only in regions where it is needed (i.e. in the vicinity of the moving interface), thus achieving computational efficiency. Studies where the FEM has been applied with success to simulate two-phase incompressible flow include Smolianski [3], Tornberg [4, 5], Hysing [6] and Ganesan [1].

In incompressible two-fluid flow, the velocity and pressure fields and/or their gradients are discontinuous across the interface between the fluids. As the interface evolves with time and cuts through the elements, the standard finite element approximation performs poorly unless constant remeshing is carried out such that the element edges are aligned with the moving interface throughout the simulation. The Extended Finite Element Method (XFEM) [7, 8], on the other hand, ensures high accuracy even if the discontinuities lie within the elements, due to the enriched approximation space which is able to reproduce the discontinuous solution properties within the elements. This allows for the convenient use of a fixed mesh throughout the simulation. Studies where the XFEM has been employed in two-fluid flow simulations include Chessa and Belytschko [9, 10, 11], Groß and Reusken [12], Fries [13] and Zlotnik and Díez [14]. In this study, in addition

to the enrichment of the pressure space using the *sign-enrichment*, a multi-level adaptive mesh refinement using a hierarchy of hexahedral elements with hanging nodes on 1-irregular meshes, is realized in the vicinity of the moving interface. The use of XFEM with hanging nodes has previously been applied to 2D problems in solid and fluid mechanics by the authors in [15]. There, the focus is on the treatment of hanging nodes in the context of the XFEM. The fluid simulations have the character of a feasibility study without going into detail on the governing equations and their discretization. In this study, however, the method is explained in detail and the focus is on the interplay of the refinement algorithm with the interface movement that affects the level-set function and the enriched XFEM approximations. Furthermore, the h-XFEM is extended to three-dimensional two-fluid simulations in this work.

An important feature of two-fluid flow simulation is the description of the moving interface. The level-set (LS) method [16] is employed in this study, which defines the interface as the zero-level of a signed-distance function. The function is then advected with the velocity of the fluid via a pure advection equation. It is remarkable to point out that the LS method has been the method of choice for describing moving interfaces when used in conjunction with the XFEM (see e.g. Chessa and Belytschko [9, 17] and Fries [13]). This is because both methods require only a fixed mesh to be used throughout the simulation and the LS function can also be used to construct enrichment functions within the XFEM framework.

This study employs the Laplace-Beltrami technique [18, 19] for the modelling of the surface tension term so as to avoid the explicit computation of the curvature. An important finding of this study is that the computation of the normal vector to the interface has a significant bearing on the accuracy of the solver. This is verified by tracking the mass conservation of the two immiscible fluids as the simulation evolves. It is discovered that normal vectors computed using the level-set function yield better mass conservation properties compared to those computed based on the discretized interface.

Summarizing, the following numerical strategies are employed: (i) the *Eulerian* formulation for the kinematical description of the flow field, (ii) the primitive variable formulation which seeks the velocity and pressure unknowns simultaneously, (iii) the Streamline Upwind Petrov-Galerkin (SUPG) stabilization [20] to suppress any oscillations due to the existence of the nonlinear convective term, (iv) the Pressure-Stabilizing Petrov-Galerkin (PSPG) stabilization [21, 22] to circumvent the LBB condition, thus allowing for equal-order interpolations (bilinear in 2D and trilinear in 3D) for both pressure and velocity, (v) the Crank-Nicolson method for time-stepping, (vi) the discretization of time *before* space due to the time-dependence of the enrichment function in the XFEM [23], (vii) the level-set (LS) method for the description of the moving two-fluid interface, (viii) a strong coupling between the Navier-Stokes and level-set equations akin to the recursive *predictor-corrector* method, (ix) the method of solving a partial differential equation to steady state to regain the signed-distance property of the level-set function (Sussman *et al.* [24]), (x) a multi-level adaptive mesh refinement (realized via hanging nodes on 1-irregular meshes) applied in the vicinity of the moving interface to improve the resolution of the interface and also to capture any high gradients in the solution, (xi) the enrichment of the pressure space using the *sign-enrichment* to capture the discontinuous jump in the pressure field across the interface due to the existence of surface tension and (xii) the Laplace-Beltrami technique [18, 19] for the modelling of the surface tension term so as to avoid the explicit computation of the curvature.

The paper is organized as follows. Section 2 presents the governing Navier-Stokes and level-set transport and reinitialization equations used in this study with accompanying boundary/interfacial conditions. Section 3 considers the topic of spatial discretization where the XFEM is employed for the enrichment of the pressure space. The topic of subcell quadrature is then considered in detail for both 2D and 3D. Section 4 is devoted to a discussion of the multilevel mesh refinement procedure used in this study. Section 5 describes the flow solver which includes the temporal discretization, the remeshing procedure and the strong coupling between the level-set and Navier-Stokes equations. For clarity, flowcharts are provided to illustrate the procedures. The modelling of the surface tension using the Laplace-Beltrami technique is discussed in Section 6. Finally, the numerical results for several 2D and 3D test cases are presented in Section 7. This paper ends with a summary and conclusions in Section 8.

2. GOVERNING EQUATIONS

The governing incompressible Navier-Stokes and the level-set transport equations are presented in this section. The equations are accompanied by appropriate boundary conditions including the interfacial conditions for the two-fluid interface.

2.1. The incompressible Navier-Stokes equations

We consider a n-dimensional domain $\Omega \subset \mathbb{R}^n$ with the boundary $\Gamma = \partial\Omega$. The geometrical situation is depicted in Figure 1. The boundary Γ is decomposed into the Dirichlet and Neumann boundary, $\Gamma_{\boldsymbol{u}}$ and $\Gamma_{\boldsymbol{h}}$, respectively, such that $\Gamma_{\boldsymbol{u}} \cup \Gamma_{\boldsymbol{h}} = \Gamma$ and $\Gamma_{\boldsymbol{u}} \cap \Gamma_{\boldsymbol{h}} = \emptyset$. The normal vector on Γ is denoted by $\boldsymbol{n}$. The domain Ω contains two different, immiscible and incompressible Newtonian fluids in Ω_1 and Ω_2, respectively, so that $\Omega = \Omega_1 \cup \Omega_2$. We consider Ω to be a time-independent, closed container whereas $\Omega_1(t)$ and $\Omega_2(t)$ change in time. The (moving) interface between the two fluids is denoted by Γ_d. $\hat{\boldsymbol{n}}$ is the normal vector on Γ_d and points from Ω_1 to Ω_2.

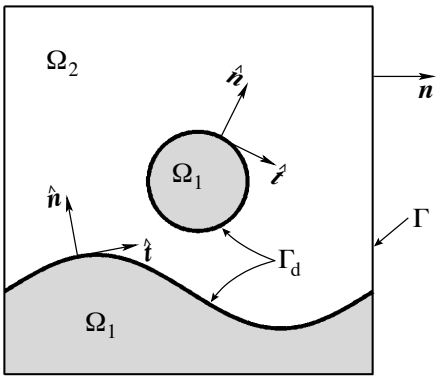


Figure 1: The two fluids in Ω_1 and Ω_2, separated by the interface Γ_d.

The governing equations are now given in strong form. Let $u_i(\boldsymbol{x},t)$ be the velocities and $p(\boldsymbol{x},t)$ the pressure; ϱ_k and μ_k with $k=(1,2)$ are the density and dynamic viscosity of the two fluids, respectively. The fluids inside $\Omega_k \times (0,t_{\text{end}})$, $k=(1,2)$, are modeled by the instationary, incompressible Navier-Stokes equations in velocity-pressure formulation

$$\varrho_k\Big(\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j}\Big) - \frac{\partial \sigma_{ij}}{\partial x_j} = f_{k,i}, \tag{1}$$

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{2}$$

where (1) denotes the momentum equations and (2) denotes the continuity equation (incompressibility constraint). The stress tensor σ_{ij} of the Newtonian fluids is given as

$$\sigma_{ij}(u_i,p) = -p\delta_{ij} + 2\mu_k\varepsilon_{ij}, \text{ with } \varepsilon_{ij} = \frac{1}{2}\big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\big), \tag{3}$$

where δ_{ij} is the identity tensor and ε_{ij} is the strain rate tensor (symmetric part of the velocity gradient tensor). The term $f_{k,i} = \varrho_k a_i$ where a_i is the acceleration, represents the externally applied force per unit volume. Dirichlet and Neumann boundary conditions on the outer boundary of Ω are

$$u_i = \hat{u_i} \quad \text{on } \Gamma_{\boldsymbol{u}} \times (0,t_{\text{end}}), \tag{4}$$

$$\sigma_{ij}n_j = \hat{h_i} \quad \text{on } \Gamma_{\boldsymbol{h}} \times (0,t_{\text{end}}), \tag{5}$$

where $\hat{u_i}$ and $\hat{h_i}$ are prescribed velocities and stresses. As an initial condition, a divergence-free velocity field $\hat{u}_{0_i}$ is specified over Ω,

$$u_i(\boldsymbol{x},0) = \hat{u}_{0_i}(\boldsymbol{x}) \quad \text{in } \Omega \text{ at } t=0. \tag{6}$$

Furthermore, if only Dirichlet boundary conditions are prescribed (i.e. $\Gamma_h = \emptyset$), the pressure p is present only as a gradient in the Navier Stokes momentum equations (1) and is therefore determined up to only a constant. In such cases, there is a need to impose the value of p at an arbitrary point in the computational domain to uniquely determine the pressure field [25].

The situation at the interface Γ_d is now considered in more detail. The density and viscosity fields

$$\varrho(x_i,t) = \begin{cases} \varrho_1 & \forall x_i \in \Omega_1(t), \\ \varrho_2 & \forall x_i \in \Omega_2(t). \end{cases} \qquad \mu(x_i,t) = \begin{cases} \mu_1 & \forall x_i \in \Omega_1(t), \\ \mu_2 & \forall x_i \in \Omega_2(t). \end{cases} \tag{7}$$

change discontinuously at Γ_d. As a consequence, also the state variables such as the velocities and pressure fields involve discontinuities at the interface. Discontinuities may be classified into strong or weak. In the case of strong discontinuities, a jump and a change in the gradient is present in the field. For weak discontinuities there is only a kink in the field, i.e. the field is continuous with a discontinuous gradient.

The interfacial conditions express the continuity of mass and balance of momentum across the interface. From momentum balance, the stress boundary condition at the interface between the two fluids in Ω_1 and Ω_2 can be expressed as [26]

$$\big(\sigma_{ij}^1 - \sigma_{ij}^2\big)\hat{n}_j = \gamma\kappa\hat{n}_i + \frac{\partial\gamma}{\partial x_i}, \tag{8}$$

where γ is the surface tension coefficient (material parameter) and κ is the curvature of Γ_d. If we do not consider the variation of the surface tension coefficient γ, we obtain

$$\left(p^1 - p^2\right)\hat{n}_i + \left(\bar{\sigma}^1_{ij} - \bar{\sigma}^2_{ij}\right)\hat{n}_j = \gamma\kappa\hat{n}_i, \tag{9}$$

where $\bar{\sigma}_{ij} = 2\mu_k\varepsilon_{ij}$ is the *viscous* stress tensor. On the other hand, application of mass conservation leads to the following interfacial condition for the velocity [3]

$$(u^1_i - u^2_i)\hat{n}_i = 0 \tag{10}$$

which depicts the continuity of the *normal* velocity across the interface. Further, from physical considerations alone, owing to the effects of viscosity, continuity of the *tangential* velocity is also assumed [3]. Summarizing, the following conditions apply at the interface

$$[u_i]_{\Gamma_d} = 0 \quad \text{on } \Gamma_d \times (0, t_{\text{end}}), \tag{11}$$

$$[\sigma_{ij}]_{\Gamma_d}\hat{n_j} = \gamma\kappa\hat{n}_i \quad \text{on } \Gamma_d \times (0, t_{\text{end}}), \tag{12}$$

where the jump notation $[\cdot]_{\Gamma_d}$ for a quantity $f(x_i)$ is defined as

$$[f(x_i)]_{\Gamma_d} = \lim_{x_i \in \Omega_1 \to x_i \in \Gamma_d} f(x_i) - \lim_{x_i \in \Omega_2 \to x_i \in \Gamma_d} f(x_i). \tag{13}$$

The interfacial condition (11) states that the velocities are continuous across Γ_d, or, in other words, that the jump in the velocity field is zero. The second interface condition (12) states that the surface tension balances the jump of the normal stress at the interface. As a consequence of (1)-(2) and (11)-(12), the velocity fields $u_i(\boldsymbol{x}, t)$ are weakly discontinuous across Γ_d, whereas the pressure field $p(\boldsymbol{x}, t)$ has a strong discontinuity at the interface. In the case where no surface tension is considered, $\gamma = 0$ and the jump in the pressure field vanishes which implies that $p(\boldsymbol{x}, t)$ has a kink across Γ_d.

2.2. The level-set transport equation

In this study, we employ the level-set method (LSM) for the implicit representation of the two-fluid interface [27]. An implicit interface representation defines the interface as the isocontour of some implicit function (called the level-set function). An important advantage of the LSM is that it can readily handle dynamic interfaces which exhibit topological changes (pinching apart of the interface or merging of several pieces of the interface) in a straightforward manner without the need for any special *ad-hoc* procedure (as is often employed in an explicit interface-tracking approach).

The level-set function used in this study is the *signed-distance* function defined as

$$\phi_{\text{sd}}(\boldsymbol{x}, t) = \begin{cases} -\phi_{\text{d}} & \forall \boldsymbol{x} \in \Omega_1(t) \\ 0 & \forall \boldsymbol{x} \in \Gamma_{\text{d}}(t) \\ +\phi_{\text{d}} & \forall \boldsymbol{x} \in \Omega_2(t) \end{cases} \tag{14}$$

where

$$\phi_{\text{d}}(\boldsymbol{x}, t) = \min_{\boldsymbol{x}^* \in \Gamma_{\text{d}}} \|\boldsymbol{x} - \boldsymbol{x}^*\|, \quad \boldsymbol{x} \in \Omega \tag{15}$$

is, for every point $\boldsymbol{x}$, the shortest distance to the interface. For ease of notation, we will refer to $\phi_{\text{sd}}(\boldsymbol{x}, t)$ simply as $\phi(\boldsymbol{x}, t)$ from here onward. Suppose that the interface, $\Gamma_d(t)$ is moving with velocity $\boldsymbol{u}(\boldsymbol{x}, t)$. Since the interface is defined by

$$\phi(\boldsymbol{x}(t), t) = 0, \tag{16}$$

we can take the time derivative and obtain

$$\frac{\partial \phi}{\partial t} + u_j \frac{\partial \phi}{\partial x_j} = 0. \tag{17}$$

The last relation (17), which represents an advection equation for the moving interface, is also known as the *level-set transport equation*. Since (17) is a pure advection (hyperbolic) equation, if inflow/outflow boundaries exist in the computational domain, only Dirichlet boundary conditions may be specified at the inflow boundary [28]. However, in all test cases considered in this work, no inflow/outflow boundaries exist and hence, no boundary conditions for (17) are prescribed.

An important property of the interface is the mean curvature κ of the interface, defined as the divergence of the normal $\hat{\boldsymbol{n}}$ to the interface as

$$\kappa\left(\boldsymbol{x}(t), t\right) = \nabla \cdot \hat{\boldsymbol{n}} \ , \ \ \boldsymbol{x} \in \Gamma_{\mathrm{d}}. \tag{18}$$

We can also express κ in terms of the level-set function ϕ as

$$\kappa\left(\boldsymbol{x}(t), t\right) = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right), \tag{19}$$

which, due to the property $|\nabla \phi| = 1$ possessed by the signed-distance function, can be further simplified as

$$\kappa\left(\boldsymbol{x}(t), t\right) = \Delta \phi, \tag{20}$$

where $\Delta = \nabla \cdot \nabla$ is the *Laplacian* operator.

2.3. The level-set reinitialization equation

At the beginning of the simulation, the level-set function ϕ is initialized to be the signed-distance function ϕ_{sd} to the interface Γ_d. However, as the interface evolves with time, the level-set function will inevitably get distorted due to the advection, especially in the vicinity of the interface, thus losing the desirable property $|\nabla \phi| = 1.0$. This makes the determination of the interface position via interpolation less accurate (see e.g. [3, 29]). We note that the closer the level-set function is to the signed-distance function, the more accurate is the approximation. This provides the motivation for reinitializing the level-set function periodically, in addition to obtaining simpler expressions for the mean curvature (20). It is to be noted that if the level-set function gets too steep in the vicinity of the interface, the interface propagation problem amounts to the transport of a nearly discontinuous function, which requires a very fine mesh to suppress numerical oscillations near the interface [30]. Such oscillations affect not only the interface shape but also the quality of the computed normal and curvature.

Reinitialization of the level-set function has been an active area of research in the level-set method, see e.g. [31, 32, 33, 34]. Reinitialization is simply the process of replacing ϕ by the signed-distance ϕ_{sd} which has the same zero contour as ϕ but is well-behaved (i.e. possessing the desirable property $|\nabla \phi| = 1.0$), and then using this new function for the advection until the next reinitialization. A straightforward way to reinitialize the level-set function is to locate the position of the interface by some interpolation technique and then construct the signed-distance function to this interpolated front, see e.g. [3, 13]. The accuracy of the approximated interface location depends on the accuracy of the

interpolation method used. If a piecewise linear (or piecewise planar in 3D) interface representation is used, kinks will exist in the reinitialized level-set function. Therefore, this procedure has the disadvantage that it may introduce unwanted oscillations in the curvature (20) which is represented as the Laplacian of the level-set function [13]. This is highly undesirable in situations where such geometric quantities play a crucial role (as for two-fluid flow where the curvature is needed to compute the surface tension).

In Sussman *et al.* [31], the reinitialization is formulated as solving a partial differential equation together with initial condition given by

$$\frac{\partial \phi}{\partial \tau} + \text{sign}(\phi)(|\nabla \phi| - 1) = 0 \ , \ \ \forall (\boldsymbol{x}, \tau) \in \Omega \times (0, T_{steady}) \ , \tag{21}$$

$$\phi(\boldsymbol{x}, 0) = \phi^0 \ , \tag{22}$$

where

$$\text{sign}(\phi) = \begin{cases} -1 & \text{if } \phi < 0 \\ 0 & \text{if } \phi = 0 \\ 1 & \text{if } \phi > 0, \end{cases} \tag{23}$$

ϕ^0 is the initial level-set function at the beginning of the reinitialization and τ is the pseudo time. The boundary conditions are the same as for the level-set transport equation. The above equation is to be solved to steady state when $|\nabla \phi| = 1.0$ with T_{steady} being the time taken to reach steady state. In principle, ϕ remains unchanged at the interface (where $\text{sign}(\phi) = 0$), therefore, the zero isocontour of ϕ and ϕ^0 are the same. Away from the interface, $|\nabla \phi|$ will converge to 1.0. This algorithm avoids explicitly locating the position of the interface by interpolation.

The reinitialization equation (21) can be recast into a more illuminating form [35]

$$\frac{\partial \phi}{\partial \tau} + \boldsymbol{w} \cdot \nabla \phi \quad = \text{sign}(\phi), \tag{24}$$

where

$$\boldsymbol{w} = \text{sign}(\phi) \frac{\nabla \phi}{|\nabla \phi|} \tag{25}$$

is the characteristic velocity of the hyperbolic equation (24). We observe that $\boldsymbol{w}$ is pointing *outward* from the interface in the direction of the normal. This implies that $\phi(\boldsymbol{x}, \tau)$ will be reinitialized to $|\nabla \phi| = 1$ near the interface first. The implication is that we typically need much fewer pseudo time-steps in order that the property $|\nabla \phi| = 1$ is satisfied for grid points close to the interface which is what is needed in practice. That is, grid points far away from the interface are inconsequential for the description of the interface. In this study, both the interpolation-reinitialization and PDE-reinitialization (24) schemes will be investigated and compared.

3. SPATIAL DISCRETIZATION AND QUADRATURE

This section presents the FEM/XFEM approximations for the dependent variables (i.e. velocity, pressure and level-set function) in the governing equations. This is then followed by a description of the quadrature procedure used in this study.

3.1. Spatial discretization

As mentioned in the Section 2.1, the velocity field and pressure field possess a kink and a jump across the moving interface, respectively when surface tension is considered. The classical FEM relies on the approximation properties of mapped polynomials. Optimal accuracy is achieved for *smooth* solutions and inner-element jumps and kinks lead to a drastic decrease in accuracy. It is therefore crucial in the classical FEM to align the element edges of the mesh with the interfaces where discontinuities exist. Furthermore, for strong discontinuities, a complete decoupling of the elements adjacent to the interface is important. In applications where the interfaces are moving, the mesh has to be updated so that the elements always align with the moving interface (i.e. interface tracking). The Extended Finite Element Method (XFEM) [8] is a numerical method for modelling arbitrary discontinuities in finite elements by extending the piecewise polynomial approximation space of the standard FEM to include discontinuous function spaces in local regions of the computational domain which exhibit discontinuities. This is achieved via a partition-of-unity concept [36].

In this study, the pressure space is enriched with the (shifted) sign-enrichment leading to the following approximation for pressure

$$p^h(\boldsymbol{x},t) = \sum_{i\in I} N_i(\boldsymbol{x})\, p_i + \sum_{i\in I^*} N_i^*(\boldsymbol{x})\,[\psi_{\text{sign}}(\boldsymbol{x},t) - \psi_{\text{sign}}(\boldsymbol{x}_i,t)]a_i, \tag{26}$$

where $\psi_{\text{sign}}(\boldsymbol{x},t)$ is defined as

$$\psi_{\text{sign}}(\boldsymbol{x},t) = \text{sign}(\phi(\boldsymbol{x},t)) = \begin{cases} -1 & \text{if } \phi(\boldsymbol{x},t) < 0 \\ 0 & \text{if } \phi(\boldsymbol{x},t) = 0 \\ 1 & \text{if } \phi(\boldsymbol{x},t) > 0. \end{cases} \tag{27}$$

The approximation for the velocity is given as

$$\boldsymbol{u}^h(\boldsymbol{x},t) = \sum_{i\in I} N_i(\boldsymbol{x})\,\boldsymbol{u}_i, \tag{28}$$

and the approximation for the level-set function is given as

$$\phi^h(\boldsymbol{x},t) = \sum_{i\in I} N_i(\boldsymbol{x})\,\phi_i. \tag{29}$$

The notations used in the above equations follow standard convention, see Figure 2. We employ standard bilinear shape functions (2D) and trilinear shape functions (3D) for both $N_i(\boldsymbol{x})$ and $N_i^*(\boldsymbol{x})$ in the pressure approximation (26) and also for $N_i(\boldsymbol{x})$ in the velocity approximation (28) and level-set approximation (29). The use of equal-order interpolations for both pressure and velocity is possible due to the adopted PSPG stabilization which circumvents the LBB condition.

3.2. Quadrature

An important consequence of extending the pressure FE approximation by the enrichment function $\psi_{\text{sign}}(\boldsymbol{x},t)$ is that the jump in the enrichment function is now inherited by the resulting shape functions of the enrichment part $N_i^*(\boldsymbol{x}) \cdot \psi_{\text{sign}}(\boldsymbol{x},t)$. This has important implications in the quadrature of the weak form. Standard Gaussian quadrature

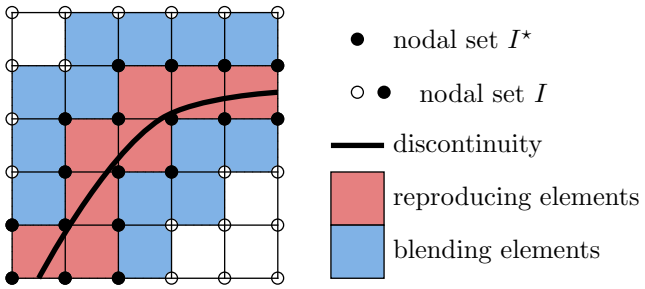


Figure 2: Illustration of reproducing elements, blending elements and nodal set $I^\star$ for a computational domain crossed by a discontinuity.

for the weak form requires smoothness of the integrands. In the presence of jumps or kinks, the accuracy of the Gaussian quadrature is drastically reduced. Therefore, for discontinuous enrichments, a *subcell* integration is typically employed for the quadrature of the weak form (see e.g. [8, 7, 37, 38]).

In this study, the interface is described implicitly by the discretized level-set function which is interpolated by standard Lagrange FE shape functions so that the zero-level is given by

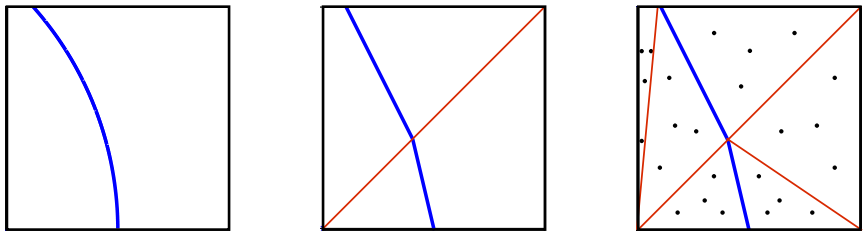
$$\phi^h(\boldsymbol{x}, t) = \sum_{i \in I} N_i(\boldsymbol{x})\, \phi_i(t) = 0. \tag{30}$$

A linear interpolation is employed (i.e. shape functions $N_i(\boldsymbol{x})$ are associated with 3-node triangular or 4-node tetrahedral elements) which results in piecewise linear and planar interfaces in 2D and 3D, respectively. Such piecewise linear/planar interfaces facilitate the subcell integration process by enabling a decomposition of the cut elements into polygons in 2D and polyhedra in 3D. It is to be noted, however, that such piecewise linear/planar interfaces are in general curved when projected to the real domain. The subcell integration procedure in 2D is illustrated in Figure 3. The first figure on the left (a) shows the original interface crossing the element. This cut element is then subdivided into two triangular subcells in (b) so that linear interpolation can be employed in each, resulting in a piecewise linear representation of the interface (shown by the blue segments). Finally, the quadrilateral subcells are further subdivided so that only triangular subcells result for the integration.

For 3D, in order to have only piecewise planar interfaces, the hexahedral element is decomposed into six tetrahedral subcells (Figure 4) so that the interpolated interface is planar within each subcell where linear interpolation is employed (Figure 5).

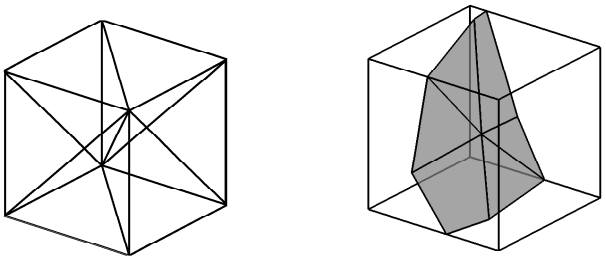
4. MULTI-LEVEL MESH REFINEMENT WITH HANGING NODES

In two-fluid flow simulations, the most important phenomena occur at the interface (e.g. surface tension, topological changes, jump in the pressure field and steep gradients in the velocity field). It is thus important that such phenomena get resolved adequately



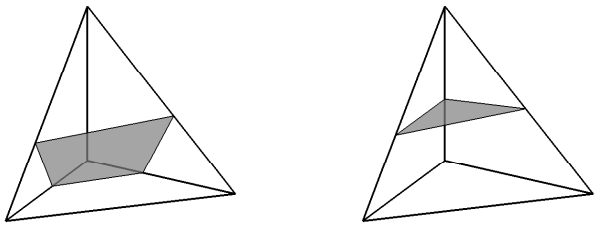
(a) Original interface (b) Piecewise linear interface (c) Gauss points

Figure 3: (a) shows the original interface. In order to consider only polygons to facilitate integration, the bilinear element is subdivided into two triangles where linear interpolation is applied in each, resulting in a piecewise linear representation of the interface in (b). For the quadrature, further subdivisions are made so that only triangular subcells result.



(a) (b)

Figure 4: (a) Decomposition of a hexahedral element into 6 tetrahedra, (b) piecewise planar embedded interface.



(a) (b)

Figure 5: An embedded *planar* interface in a tetrahedral subcell divides the subcell into (a) two pentahedra and (b) a tetrahedron and a pentahedron.

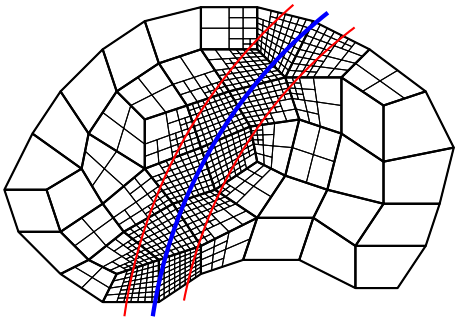


Figure 6: Mesh refinement with hanging nodes in the vicinity of the interface (shown as a thick blue line). The finest resolution is realized in a band around the interface (bounded by the two red lines) Note that elements with hanging nodes are not cut (and therefore not enriched).

by the underlying mesh. This provides the motivation for employing an adaptive mesh refinement in the vicinity of the moving interface. This section is an extension of a recent 2D study by the same authors [15] to cover the 3D situation applied to two-fluid flow simulations.

4.1. Definition of 1-irregular mesh

Local mesh refinements with hanging nodes are particularly easy to realize compared to conforming refinements. Hanging nodes are those which exist not only in element corners but also on element edges/faces of their neighboring elements. In this study, we accept hanging nodes only in the center of element edges (in 2D) and center of element faces and edges (in 3D) (2-to-1 property, 1-irregular mesh [39]). The refinement criteria employed herein are based on physical considerations: we refine *a priori* near interfaces where steep gradients are expected or where the resolution is to be increased. In particular, we only discuss the situation where the enrichment is realized in standard finite elements without hanging nodes, which is in contrast to Fries *et al.* [15] where hanging nodes may also be enriched due to the consideration of specific applications (e.g. crack problems where the crack path can cross elements with hanging nodes). This is justified since for two-phase flow simulations, elements with hanging nodes are never enriched. The only enriched elements are those cut by the moving interface where the underlying mesh has the highest resolution (and therefore no hanging nodes), see Figure 6.

In this study, we consider the case where DOFs are present at hanging nodes and valid shape functions for all regular and hanging nodes have to be found so that (i) conformity of the shape functions and (ii) the partition of unity property are achieved. We employ the shape functions of Gupta [40] in 2D and Morton *et al.* [41] in 3D. It is then found that the XFEM may be applied in a straightforward manner: the same enrichment functions and the same set of enriched nodes are used as in the standard case of the XFEM on conforming meshes. Special care is needed for the quadrature in this approach as—already without the enrichment—these special element shape functions have kinks in the reference element.

We now describe the refinement procedure in three dimensions (refer to [15] for the 2D situation). Let Ω be a bounded volume of $\mathbb{R}^3$. We start with a conforming (coarse) mesh that discretizes the domain by n_{el}^0 shape regular hexahedral elements. The collection of these elements is called $\mathcal{P}_0 = \{1, ..., n_{\text{el}}^0\}$. The element domain (volume) of element $k \in \mathcal{P}_0$ is denoted by $w_k^0 \subseteq \Omega$. Assume that based on this mesh, n_{ref} refinement steps are desired. The result of each refinement step is the collection of shape regular elements $\mathcal{P}_l, 1 \leq l \leq n_{\text{ref}}$, with related element domains $w_k^l \subseteq \Omega, k \in \mathcal{P}_l$.

Algorithmically, each collection $\mathcal{P}_l$ is obtained by applying the following procedure:

1. a set of elements $\mathcal{W}_{\text{ref}}^{l-1} \subseteq \mathcal{P}_{l-1}$ is constructed which marks elements for (another) refinement.
2. each (parent) element in $\mathcal{W}_{\text{ref}}^{l-1}$ is refined by sub-dividing it into eight (children) elements with nodes at the centroid, at the centers of the four faces and at the centers of the eight edges.
3. further elements may have to be refined until the 2-to-1 property is fulfilled, i.e. all element vertices are shared by other element vertices or are in the center of element edges or element faces. It is noted that combinations of these may also exist.

We note that the eight sub-elements resulting from an element refinement replace their parent element. One may associate the term "refinement level" with each element indicating the number of refinements that have been realized. Clearly, different refinement levels are present in each collection of elements $\mathcal{P}_l$. It is a direct consequence of (3) that all neighboring elements of each element have a maximum difference in the refinement level of 1.

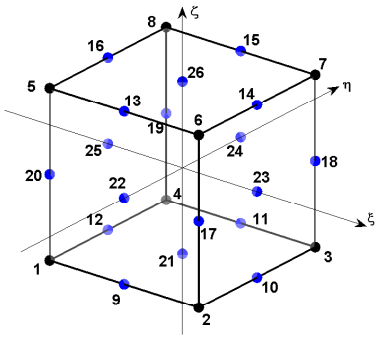
For the final collection of elements, $\mathcal{P}_{n_{\text{ref}}}$, let us now define nodes which will be associated with shape functions later on. For trilinear elements as considered here, nodes i with coordinates $\boldsymbol{x}_i$ are placed right at the vertices. The nodal set I involves all the nodes in the domain. Furthermore, the hanging nodes are given in the subset $I_h \equiv \left(I_h^{\text{face}} \bigcup I_h^{\text{edge}}\right) \subset I$. All other nodes are called regular nodes and are in the set $I_\Gamma = I \setminus I_h$. Hanging nodes in the subset $I_h^{\text{face}} \subset I_h$ reside in the center of element faces and are each associated with a quadruple (4−tuple) which defines the other four regular nodes that share the face with the hanging node.

$$\mathcal{Q}_k^{\text{face}} = \{(i,j,m,n) : \text{nodes } i,j,m,n \in I_\Gamma \text{ which share the face with hanging node } k\}, k \in I_h^{\text{face}}. \tag{31}$$

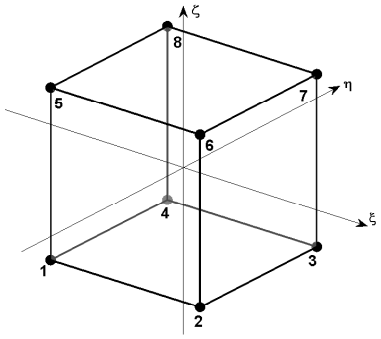
Hanging nodes in the subset $I_h^{\text{edge}} \subset I_h$ reside in the center of element edges and are each associated with a pair (2−tuple) which defines the other two regular nodes that share the edge with the hanging node.

$$\mathcal{Q}_k^{\text{edge}} = \{(i,j) : \text{nodes } i,j \in I_\Gamma \text{ which share the edge with hanging node } k\}, k \in I_h^{\text{edge}}. \tag{32}$$

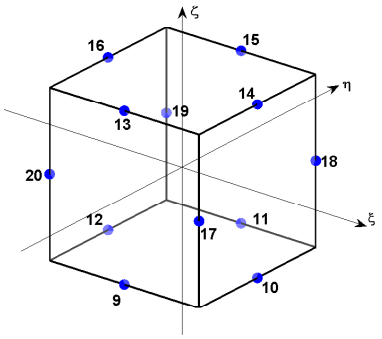
Altogether, an element can have a maximum of 18 hanging nodes of which 12 reside in the centers of the edges (Figure 7(c)) and 6 reside in the centers of the faces (Figure 7(d)). The remaining 8 nodes are the regular nodes (Figure 7(b)), making the total number of possible nodes to be 26 (Figure 7(a)).



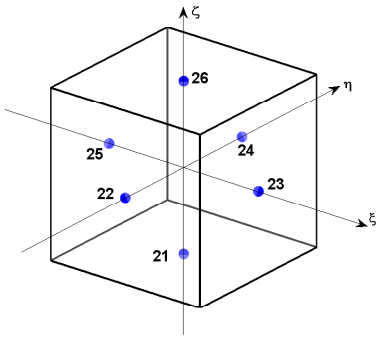
(a) Hanging nodes (blue) with regular nodes (black).



(b) Regular nodes only.

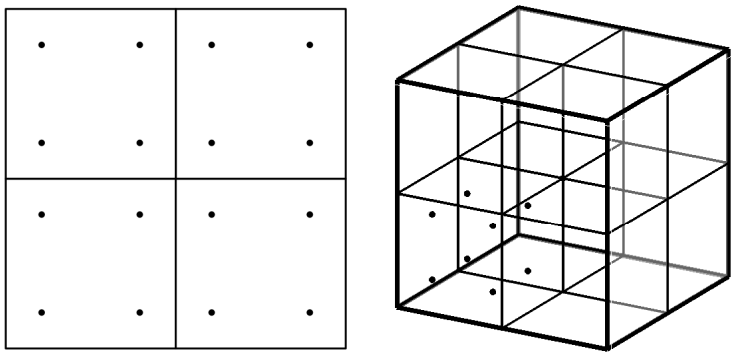


(c) Hanging nodes $k \in I_h^{\text{edge}}$



(d) Hanging nodes $k \in I_h^{\text{face}}$

Figure 7: Regular and hanging nodes for an element.



(a) 2D (b) 3D

Figure 8: Integration points in a reference element with (at least one) hanging node. (a) shows the 2D situation with 4 subcells and (b) shows the 3D situation with 8 subcells where, for clarity, integration points are shown in only one subcell.

4.2. Refinement criterion

As noted above, each refinement step $l+1$ starts with the marking of elements for (further) refinement, i.e. the construction of an element set $\mathcal{W}_{\text{ref}}^l, 0 \leq l \leq n_{\text{ref}}-1$. For two-fluid flow where steep gradients are expected near an interface or where the resolution of the interface is crucial for the overall solution, the mesh refinement is realized in the vicinity of the interface. In this study, the signed-distance function $\phi(\boldsymbol{x}, t)$ (14) is employed to define the refinement criterion. Assume that a refinement of all elements which lie *completely* within the distance $d \in \mathbb{R}^+$ from the interface is desired, then

$$\mathcal{W}_{\text{ref}}^l = \left\{ k \in \mathcal{P}_l : \max_{\boldsymbol{x} \in w_k^l} |\phi(\boldsymbol{x}, t)| < d \right\}. \tag{33}$$

4.3. Conforming shape functions

Shape functions for three-dimensional 1-irregular meshes are proposed by Morton *et al.* [41]. Shape functions on two-dimensional 1-irregular meshes as proposed by Gupta [40] are described in detail in [15]. The definition is based on the reference element $\Omega^* = (-1,1)\times(-1,1)\times(-1,1)$ with (i) 8 corner nodes, (ii) 12 (potential) hanging nodes in the centers of element edges and (iii) 6 (potential) hanging nodes in the centers of element faces, see Figure 7. The corresponding element shape functions in the physical domain Ω are obtained by a bilinear mapping (2D) or trilinear mapping (3D) from the reference domain. The shape functions $N_i(\boldsymbol{x})$ for each node at $\boldsymbol{x}_i$ result from the usual assembly of the element shape functions. It is trivial to show that these shape functions build a partition-of-unity over an element. Since certain shape functions possess kinks across the planes spanned by the ξ, η, ζ-axes of the coordinate system, a special quadrature is required in elements with hanging nodes: Gauss points are placed in each of the 8 subcells of the reference element. Figure 8 shows the integration points for both the 2D and 3D situations.

5. THE FLOW SOLVER

The temporal discretizations for the Navier-Stokes, level-set and reinitialization equations are first discussed in Section 5.1. The appropriate function spaces for the dependent variables together with the weak formulations are also presented. Section 5.2 discusses the remeshing strategy and Section 5.3 describes the strong coupling procedure between the Navier-Stokes and the level-set transport equations.

5.1. Temporal discretization

In general, two choices are available when considering time-stepping in the context of the FEM according to whether time or space is discretized first. When time-dependent problems (involving first-order derivatives in time) are first discretized with respect to space, this leads to a system of coupled first order ODEs in time. Also called the *semi-discrete* method, this approach requires that the shape functions do not depend on time, i.e. the approximation is expressed as

$$u^h(\boldsymbol{x}, t) = \sum_i N_i(\boldsymbol{x}) u_i(t), \tag{34}$$

where it can be seen that time-dependence is accounted for by the nodal values of the dependent variable. However, such a formulation is not possible in the context of the XFEM with moving discontinuities due to the inherent time-dependence of the enrichment functions [23]. Therefore, in this study, time is discretized before space.

In this work, the trapezoidal rule will be employed for time-stepping. The trapezoidal rule is one of the θ-family of one-step time-stepping methods with θ set to $1/2$. It is a second-order accurate semi-implicit scheme and is unconditionally stable. The trapezoidal rule (sometimes also referred to as the Crank-Nicolson method) is expressed as

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \theta G_i^{n+1} + (1-\theta) G_i^n, \quad \theta = \frac{1}{2}, \tag{35}$$

Applying the trapezoidal rule to (1), it is immediately clear that

$$G_i = -\frac{1}{\varrho_k}\left[\varrho_k u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} - f_{k,i}\right], \tag{36}$$

from which the time-discretized momentum equations can then be expressed as

$$\varrho_k u_i^{n+1} + \theta \Delta t \left[\varrho_k u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} - f_{k,i}\right]^{n+1} = \varrho_k u_i^n + (1-\theta)\Delta t \varrho_k G_i^n. \tag{37}$$

To ease notation, we denote the RHS by h_i^n which leads to the following expression for the time-discretized momentum equations

$$\frac{1}{\theta \Delta t}\varrho_k u_i^{n+1} + \left[\varrho_k u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} - f_{k,i}\right]^{n+1} = \frac{1}{\theta \Delta t} h_i^n, \tag{38}$$

together with the continuity equation

$$\frac{\partial u_j^{n+1}}{\partial x_j} = 0, \tag{39}$$

which implies that the incompressibility constraint is enforced in the current time-step.

We define the trial and test spaces, $\mathcal{S}^h$ and $\mathcal{V}^h$ for the velocities u_i and pressure p as

$$\mathcal{S}^h_{u_i} = \left\{ u^h_i \middle| \, u^h_i \in \left(\mathcal{H}^{1h}\right)^d, \; u^h_i = \hat{u}^h_i \text{ on } \Gamma_{\boldsymbol{u}} \right\}, \tag{40}$$

$$\mathcal{V}^h_{u_i} = \left\{ w^h_i \middle| \, w^h_i \in \left(\mathcal{H}^{1h}\right)^d, \; w^h_i = 0 \text{ on } \Gamma_{\boldsymbol{u}} \right\}, \tag{41}$$

$$\mathcal{S}^h_p = \mathcal{V}^h_p = \left\{ q^h \middle| \, q^h \in \mathcal{L}^{2h} \right\}, \tag{42}$$

where $\mathcal{H}^{1h} \subseteq \mathcal{H}^1$ is the finite dimensional *Sobolev* space with the space $\mathcal{H}^1$ being the set of functions which are, together with their first derivatives, square-integrable in Ω. Further, $\mathcal{L}^{2h} \subseteq \mathcal{L}^2$ is the finite dimensional *Hilbert* space where the space $\mathcal{L}^2$ is the set of functions which are square-integrable in Ω. Since there are no explicit Dirichlet boundary conditions on the pressure, the trial and test function spaces for pressure (42) are identical. The test functions w^h_i and q^h for the velocity and pressure can be expressed as

$$w^h_i \in \mathcal{V}^h_{u_i} := \text{span}_{j \in I}\{N_j(\boldsymbol{x})\}, \tag{43}$$

$$q^h \in \mathcal{V}^h_p := \text{span}_{i \in I}\{N_i(\boldsymbol{x})\} \cup \text{span}_{i \in I^*}\{M_i(\boldsymbol{x}, t)\}, \tag{44}$$

respectively, where $M_i(\boldsymbol{x}, t) = N^*_i(\boldsymbol{x}) \cdot \psi_{\text{sign}}(\boldsymbol{x}, t)$ are the local enrichment functions defined on the enriched nodal set I^*.

To further ease notation, the superscript $n+1$ is dropped from this point onward. Unless otherwise stated, all field variables are assumed to be associated with time-level $n+1$. Combining (38) and (39), the weak formulation taking into account the SUPG/PSPG stabilizations is expressed as: Find $u^h_i \in \mathcal{S}^h_{u_i}$ and $p^h \in \mathcal{S}^h_p$ such that $\forall w^h_i \in \mathcal{V}^h_{u_i}$ and $\forall q^h \in \mathcal{V}^h_p$,

$$\begin{aligned} &\frac{1}{\theta \Delta t} \int_\Omega w^h_i \varrho_k u^h_i \, \mathrm{d}\Omega + \int_\Omega w^h_i \varrho_k u^h_j \frac{\partial u^h_i}{\partial x_j} \, \mathrm{d}\Omega - \int_{\Gamma_h} w^h_i h_i \, \mathrm{d}\Gamma + \int_\Omega q^h \frac{\partial u^h_j}{\partial x_j} \, \mathrm{d}\Omega \\ &+ \sum_{e=1}^{n_{el}} \int_{\Omega^{el}_e} \tau_e \left(u^h_j \frac{\partial w^h_i}{\partial x_j} + \frac{1}{\varrho_k} \frac{\partial q^h}{\partial x_i} \right) \cdot \left[\frac{1}{\theta \Delta t} \varrho_k u^h_i + \varrho_k u^h_j \frac{\partial u^h_i}{\partial x_j} - \frac{\partial \sigma^h_{ij}}{\partial x_j} - f^h_{k,i} - \frac{1}{\theta \Delta t} h^n_i \right] \mathrm{d}\Omega \\ &+ \int_\Omega \sigma^h_{ij} \frac{\partial w^h_i}{\partial x_j} \, \mathrm{d}\Omega - \int_\Omega w^h_i f^h_{k,i} \, \mathrm{d}\Omega = \frac{1}{\theta \Delta t} \int_\Omega w^h_i h^n_i \, \mathrm{d}\Omega, \end{aligned} \tag{45}$$

with $u^h_i(\boldsymbol{x}, 0) = \mathbf{0}$ as the initial condition. We note that the additional stabilization terms are found within the summation $\sum_{e=1}^{n_{el}}$ with the expression within the square brackets denoting the residual of the time-discretized equation (38). These terms stabilize oscillations in advection dominated regions and enable equal-order interpolations of the velocity and pressure by circumventing the LBB condition [42, 43]. The stabilization parameter τ_e is element-specific and is defined as [44]

$$\tau_e = \left[\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2 |\boldsymbol{u}^h_{avg}|_2}{h_e} \right)^2 + \left(\frac{4\nu}{h^2_e} \right)^2 \right]^{-1/2}, \tag{46}$$

where $|\boldsymbol{u}^h_{avg}|_2$ is the L_2-norm of the average velocity computed element-wise, ν is the kinematic viscosity which takes on the weighted average of the two fluids' viscosities when the element is cut by the interface and h_e is computed element-wise as

$$h_e = \sqrt{2} \cdot A_e^{\text{el}} / h_{\text{diag}}, \tag{47}$$

with A_e^{el} being the element area and h_{diag} being the largest diagonal distance between the nodes of the element [45].

Two important remarks are in place at this point. First, we note that the surface tension term, which is embedded in the term h_i^n, is evaluated as a boundary integral at the previous time-level n as

$$\int_{\Gamma_d^n} \gamma \kappa w_i \hat{n}_i \, \mathrm{d}\Gamma. \tag{48}$$

That is, a fully explicit treatment of the surface tension is employed in this study ($\theta = 0$ for the surface tension term). This is because of the partitioned treatment of the interface position (level-set function) and the flow variables (i.e. velocity and pressure) which implies that only the interface position at the *previous* time-level is available when solving the Navier-Stokes equations for the flow variables at the current time-level. This inevitably introduces a *capillary* time-step size limit on Δt defined as [46]

$$\Delta t_{\text{num}}^{(\text{ca})} = \sqrt{\frac{\overline{\varrho} h^3}{\gamma}}, \tag{49}$$

where $\overline{\varrho}$ represents the average fluid density at the interface and h represents the mesh size. As can be seen, this condition is rather restrictive for a small mesh size and/or a large surface tension coefficient. Efforts to lift such a restriction have been realized, for example, in the work of Hysing [6] which proposes a semi-implicit treatment of the surface tension. However, we adopt the simpler explicit treatment of surface tension while observing the numerical capillary time-step size limit (49).

The second remark is that an examination of (45) reveals that terms which include test and trial functions from both time-levels may coexist within the same integral. This implies that one may need to consider the situation of having two interfaces crossing a single element in the subcell quadrature. However, in this study, we avoid this delicate situation by adopting a fully implicit treatment of the pressure (i.e. $\theta = 1$ for the pressure term). This strategy, coupled with the fact that the velocity space is not enriched, leads to the much simpler situation of having to deal with only one interface in the subcell quadrature.

The level-set transport equation (17) and the reinitialization equation (24) are treated in a similar vein. That is, the trapezoidal rule is used for time-stepping together with SUPG stabilization. The stabilization parameter τ_e is still defined as in (46) but with $\nu = 0$. The trial $\mathcal{S}_\phi^h$ and test $\mathcal{V}_\phi^h$ function spaces are defined as:

$$\mathcal{S}_\phi^h \quad = \quad \mathcal{V}_\phi^h = \left\{ \psi^h \middle| \, \psi^h \in \mathcal{H}^{1h} \right\}. \tag{50}$$

Note that the trial and test function spaces coincide since no Dirichlet boundary conditions are prescribed in all the test cases considered in this study. The test function ψ^h

can be expressed as

$$\psi^h \in \mathcal{V}_\phi^h \quad := \quad \text{span}_{i \in I}\{N_i(\boldsymbol{x})\}, \tag{51}$$

where standard bilinear shape functions (2D) and trilinear shape functions (3D) are employed for $N_i(\boldsymbol{x})$.

5.2. Remeshing for moving interfaces

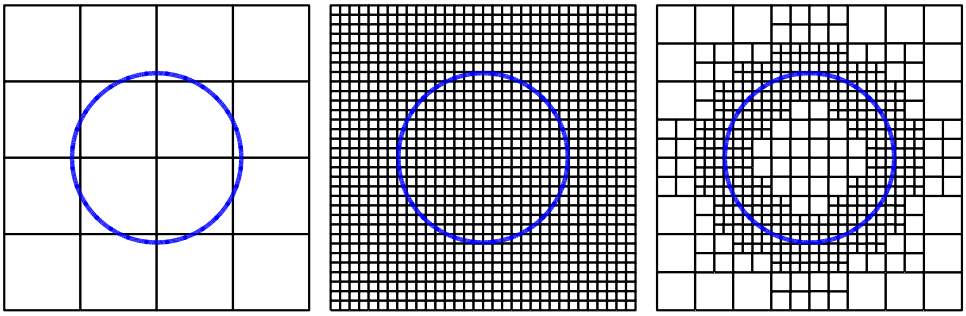
This section discusses the remeshing procedure in the context of moving interfaces adopted in this study. Recall from (33) that the mesh refinement is realized in a finite band around the interface, see Figure 6. This is necessary since it is possible that, within a single time-step, the new interface position (for the current time-step $n+1$) may venture outside the region of the finest mesh on the refined mesh, thus leading to a loss in accuracy. For optimal accuracy, the interface position should always lie within the band where the highest refinement level is realized. The width of the band to be used depends on factors such as the velocity of propagation of the interface and the time-step size.

The remeshing procedure is now described. We first define the three kinds of meshes used in this study: a coarse mesh Π^{c}, a fine mesh Π^{f} and a refined mesh Π^{r}, see Figure 9. A coarse mesh is the original uniform mesh on which mesh refinement is to be applied. A refined mesh is an adaptive mesh where mesh refinement is realized in the vicinity of the interface. A fine mesh is a uniform mesh whose resolution is equivalent to the finest resolution on the refined mesh. Further, we denote the level-set function defined on a coarse mesh as ϕ^{c} and those defined on a fine mesh and refined mesh as ϕ^{f} and ϕ^{r}, respectively.

The remeshing procedure is described in Algorithm 1 and also shown as a flowchart in Figure 10. It is to be emphasized that the level-set defined on a fine mesh ϕ^{f} is employed to realize the mesh refinement. In other words, the criterion (33) should be applied on ϕ^{f}. This is to ensure that the reconstructed interface based on the refined mesh Π^{r} will be just as accurate as the one reconstructed from a uniform fine mesh Π^{f}. However, both the Navier-Stokes and level-set transport equations are never solved on this fine mesh Π^{f}. It is merely used for the purpose of constructing the refined mesh Π^{r} on which both equations are solved. For moving interfaces, projection of the nodal field values (i.e. velocity, pressure and level-set) from one refined mesh to another will inevitably introduce projection errors. However, such errors can be avoided in the vicinity of the interface by refining in a *finite band* around the interface so that projection of these field values is not required for nodes close to the interface.

5.3. Strong coupling

A segregated (partitioned) approach is adopted in this study for the flow-interface coupling. This means that the flow is first computed with a fixed interface and then the computed velocity field is used to advect the interface to a new position. The procedure is repeated until a certain convergence in the interface position is achieved before moving on to the next time step. Two different iteration loops are employed: an outer loop for the strong coupling and an inner Picard iteration for the nonlinear Navier-stokes solver. The outer loop is iterated until the level-set position converges as indicated by the relative L_2 norm of the difference between successive positions falling below a certain



(a) Coarse mesh Π^{c} (b) Fine mesh Π^{f} (c) Refined mesh Π^{r}

Figure 9: The coarse mesh (a) and the fine mesh (b) are used to construct the refined mesh (c). The blue line depicts the actual interface.

Algorithm 1 Procedure for remeshing.

1. If $n = 0$, construct $\phi^{\mathrm{r,n}}$ on $\Pi^{\mathrm{r,n}}$ based on $\phi^{\mathrm{f},0}$ on Π^{f}, Otherwise, $\phi^{\mathrm{r,n}}$ on $\Pi^{\mathrm{r,n}}$ for $n > 0$ available from previous time-level n (Step 7).
2. Strong coupling procedure (refer to Section 5.3) yields deformed level-set $\phi_{\mathrm{def}}^{\mathrm{r,n+1}}$ on $\Pi^{\mathrm{r,n}}$.
3. Project $\phi_{\mathrm{def}}^{\mathrm{r,n+1}}$ from $\Pi^{\mathrm{r,n}}$ to Π^{f} to obtain $\phi_{\mathrm{def}}^{\mathrm{f,n+1}}$.
4. Reinitialize $\phi_{\mathrm{def}}^{\mathrm{f,n+1}}$ on Π^{f} to obtain $\phi^{\mathrm{f,n+1}}$.
5. Construct $\Pi^{r,n+1}$ based on $\phi^{\mathrm{f,n+1}}$ on Π^{f}.
6. Project $\phi^{\mathrm{f,n+1}}$ from Π^{f} to $\Pi^{\mathrm{r,n+1}}$ to obtain $\phi^{\mathrm{r,n+1}}$.
7. Set $\phi^{\mathrm{r,n}} = \phi^{\mathrm{r,n+1}}$ and $\Pi^{\mathrm{r,n}} = \Pi^{\mathrm{r,n+1}}$ and return to Step 1.

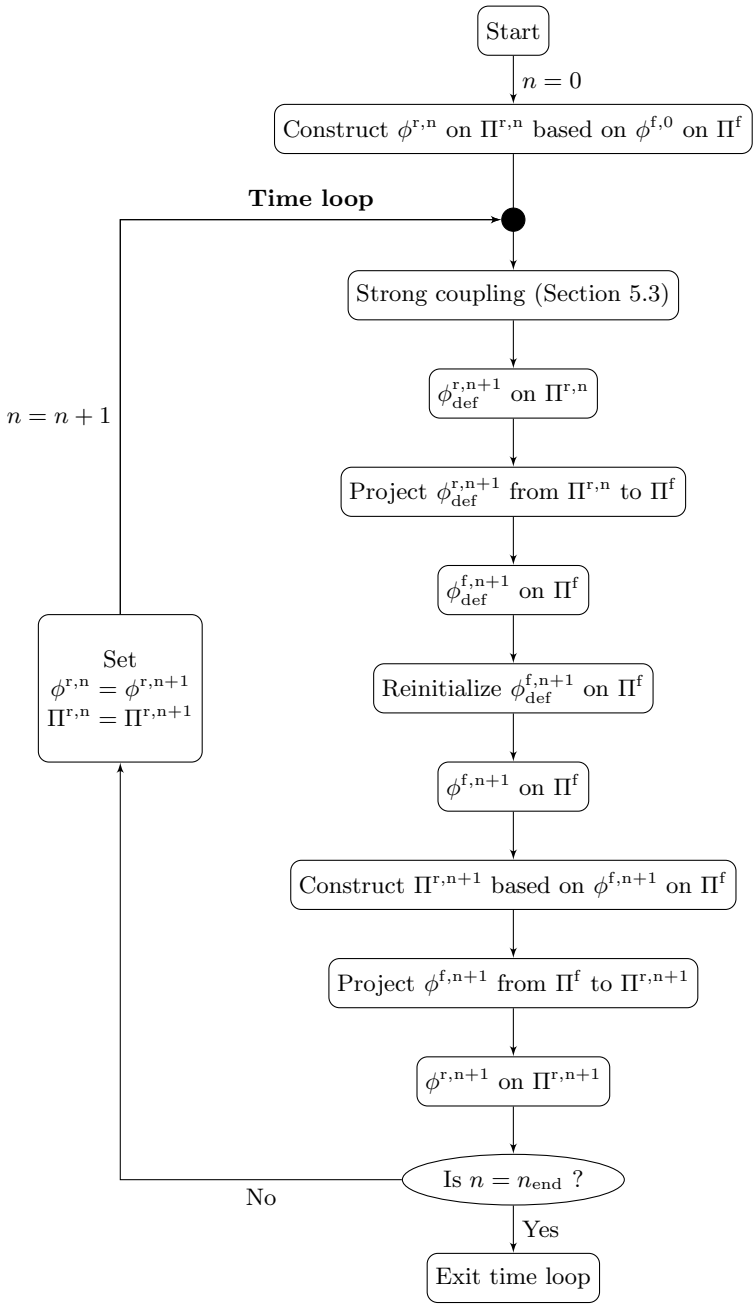


Figure 10: Flowchart for the remeshing procedure.

user-defined tolerance. This can also be seen as an iterative *predictor-corrector* method. The inner loop for the N-S solver is iterated until the L_2 norm of the difference between successive velocity fields falls below a certain user-defined tolerance.

6. MODELLING OF SURFACE TENSION

This section gives a description of the modelling of the surface tension term. The Laplace-Beltrami technique for reformulating the surface tension term is first described. This is then followed by a description of the computation of the normal vectors to the discretized interface Γ_d^h.

6.1. Laplace-Beltrami discretization

We recall that the surface tension term in the weak form of the Navier-Stokes equations is expressed in terms of a boundary integral given in (48). It is to be noted that $\hat{n}_i$ is the unit normal vector to the interface Γ_d between the two fluids in Ω_1 and Ω_2 (pointing from Ω_1 into Ω_2) and that $\boldsymbol{f}_{\text{st}} = \gamma\kappa\hat{n}_i$ is the surface tension force (per unit area in 3D and per unit length in 2D) always pointing toward the center of curvature of the interface [3]. The surface tension term expressed in (48) demands a reasonably smooth level-set function $\phi(\boldsymbol{x})$ since it requires the evaluation of the curvature κ which is given as the Laplacian of the level-set function (20). Therefore, in this study, the *Laplace-Beltrami* technique [18, 19] will be employed to reformulate the surface tension term in order to avoid the need to compute the curvature κ. However, we note that the computation of the normal vector $\hat{\boldsymbol{n}}$ to the interface still cannot be avoided.

A brief description of the Laplace-Beltrami technique is now given. To express the surface tension term (48) in terms of the Laplace-Beltrami operator, we first define the tangential gradient of a vector function $\boldsymbol{f}(\boldsymbol{x})$, which is differentiable in an open neighbourhood of the interface Γ_d, as

$$\underline{\nabla}\boldsymbol{f} = \frac{\delta f_i}{\delta x_j} = \frac{\partial f_i}{\partial x_j} - \left(\hat{n}_k \frac{\partial f_i}{\partial x_k}\right)\hat{n}_j, \quad x_i \in \Gamma_d. \tag{52}$$

The Laplace-Beltrami operator on $\boldsymbol{f}(\boldsymbol{x})$ is then given as

$$\underline{\Delta}\boldsymbol{f} = \underline{\nabla}\cdot(\underline{\nabla}\boldsymbol{f}) = \frac{\delta}{\delta x_k}\left(\frac{\delta f_i}{\delta x_k}\right), \quad x_i \in \Gamma_d. \tag{53}$$

We now invoke a theorem of differential geometry [47] which states

$$\underline{\Delta}\boldsymbol{id} = \frac{\delta}{\delta x_k}\left(\frac{\delta id_i}{\delta x_k}\right) = \kappa\hat{n}_i, \quad x_i \in \Gamma_d, \tag{54}$$

where $\boldsymbol{id}$ is the identity mapping on Γ_d. Substituting (54) into (48) and after some manipulation, the surface tension term can be expressed as

$$\int_{\Gamma_d} \gamma\kappa w_i^h \hat{n}_i \,\mathrm{d}\Gamma = \gamma\int_{\Gamma_s}\left(w_i^h\frac{\delta id_i}{\delta x_k}\right)\hat{n}_k' \,\mathrm{d}\Gamma - \gamma\int_{\Gamma_d}\left(\frac{\delta w_i^h}{\delta x_k}\frac{\delta id_i}{\delta x_k}\right)\mathrm{d}\Gamma, \tag{55}$$

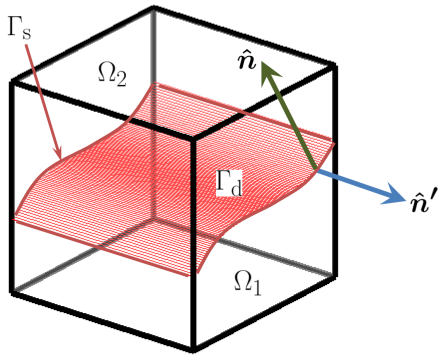


Figure 11: 3D representation of the computational domain. Γ is the boundary of the computation domain and Γ_d represents the two-fluid interface separating the domains Ω_1 and Ω_2. $\Gamma_s = \Gamma_d \cap \Gamma$ is the boundary of the interface Γ_d. $\hat{\boldsymbol{n}}'$ is the unit normal vector to Γ_s and $\hat{\boldsymbol{n}}$ is the unit normal vector to Γ_d.

where $\Gamma_s = \Gamma_d \cap \Gamma$ is the boundary given by the intersection of the two-fluid interface Γ_d with the boundary of the computational domain Γ and $\hat{n}'_k$ is the unit normal vector to Γ_s, see Figure 11.

A few remarks are in place at this stage. First, for closed interfaces where $\Gamma_d \cap \Gamma = \emptyset$, the first term on the RHS of (55) vanishes. Second, although the application of the Laplace-Beltrami technique has rid the original surface tension expression of the curvature κ, there is still the need to compute the normal vector $\hat{\boldsymbol{n}}$ to the two-fluid interface Γ_d as is apparent from (55). Finally, the normal vector $\hat{\boldsymbol{n}}'$ to the boundary Γ_s can also be interpreted as the *tangential* vector to the two-fluid interface Γ_d along the boundary Γ_s, see Figure 11.

6.2. Construction of normal vectors to Γ_d^h

This study considers two different methods of computing the normal vectors to the discretized interface Γ_d^h. Recall that the discretized interface (in the reference domain) is either piecewise linear in 2D or piecewise planar in 3D. In the first method, the tangential vectors to the piecewise linear/planar interface in the reference domain are first constructed and then mapped to the (in general) piecewise curved interface in the *real* domain. The normal vectors in the real domain are then computed by exploiting their orthogonality to the mapped tangential vectors. The normal vectors computed in this manner will always be orthogonal to the interface Γ_d^h. A detailed description can be found in [29].

In the second method, the normal vector is computed using its definition as the gradient of the level-set function. We recall from Section 2.2 that the normal vector to the interface can also be computed as

$$\hat{\boldsymbol{n}}^{\text{real}} = \frac{\nabla \phi}{|\nabla \phi|}. \tag{56}$$

Since the discrete level-set ϕ^h function values are only available at the nodes, some form of interpolation will be required to determine the value of $\hat{\boldsymbol{n}}^{\text{real}}$ along the interface. If

a Cartesian mesh is used, the finite central difference can be employed to determine the first spatial derivatives of the discrete level-set function ϕ^h at the nodes. Using standard FD notation, the components of $\nabla\phi^h$ at the nodes can be computed as

$$(\phi_x)_{i,j,k} = \left(\frac{\partial\phi}{\partial x}\right)_{i,j,k} = \frac{\phi_{i+1,j,k} - \phi_{i-1,j,k}}{2\Delta x}, \tag{57}$$

$$(\phi_y)_{i,j,k} = \left(\frac{\partial\phi}{\partial y}\right)_{i,j,k} = \frac{\phi_{i,j+1,k} - \phi_{i,j-1,k}}{2\Delta y}, \tag{58}$$

$$(\phi_z)_{i,j,k} = \left(\frac{\partial\phi}{\partial z}\right)_{i,j,k} = \frac{\phi_{i,j,k+1} - \phi_{i,j,k-1}}{2\Delta z}. \tag{59}$$

The magnitude of the gradient at each node can then be computed as

$$|\nabla\phi|_{i,j,k} = \sqrt{(\phi_x)^2_{i,j,k} + (\phi_y)^2_{i,j,k} + (\phi_z)^2_{i,j,k}}\ . \tag{60}$$

Thereafter, the i-component of the unit normal vector $\hat{n}_i^{\text{real}}(\boldsymbol{x})$ to the interface at a point $\boldsymbol{x}$ can be computed using a trilinear interpolation

$$\hat{n}_i^{\text{real}}(\boldsymbol{x}) = \sum_{j=1}^{8} N_j(\boldsymbol{x}) \left(\frac{1}{|\nabla\phi|}\frac{\partial\phi}{\partial x_i}\right)_j. \tag{61}$$

It is to be noted that the above computation is performed only for elements which are cut by the interface. For general unstructured meshes, the normal vector to the interface can be computed using

$$n_i^{\text{real}}(\boldsymbol{x}) = \sum_{j=1}^{8} \frac{\partial N_j(\boldsymbol{x})}{\partial x_i}\phi_j. \tag{62}$$

Thereafter, the *unit* normal vector can be computed as

$$\hat{n}_i^{\text{real}}(\boldsymbol{x}) = \frac{n_i^{\text{real}}}{|\boldsymbol{n}^{\text{real}}|}. \tag{63}$$

The latter method of computing the normal vectors for general unstructured meshes can also be applied for Cartesian meshes. However, for Cartesian meshes, the use of the FD method to first compute the spatial derivatives at the nodes eliminates the need to differentiate the trilinear shape functions. This reduces the differentiability requirement on the shape functions employed for the interpolation. To improve accuracy, one can also resort to higher-order shape functions. This has been realized, for example, in Groß and Reusken [12] where quadratic shape functions are employed in (62) for tetrahedral elements. A final remark is that the normal vectors computed using the level-set function as discussed in this section are not necessarily perpendicular to the piecewise linear/planar interface. However, this inconsistency does not necessarily imply a lower accuracy, as will be demonstrated in the numerical results.

7. NUMERICAL RESULTS

This section presents the numerical results. Both 2D and 3D test cases are considered. The 2D test cases include: (i) a static bubble without externally applied forces where convergence studies are performed, (ii) a rising bubble with topological changes and (iii) a rising bubble which maintains its shape due to a high surface tension. The 3D test cases include: (iv) the 3D version of the static bubble test case and (v) a rising n-butanol drop where terminal (constant) velocities are extracted and compared with theoretical predictions. The test cases (ii) and (iii) have previously been studied by Fries [13] who employed the intrinsic XFEM [48]. The test case (i) has been studied by Smolianski [3] and Hysing [6], both of whom employed the standard finite element method (FEM). Its 3D version (iv) has been considered by Groß and Reusken [12] using the XFEM. Finally, test case (v) has been considered by Groß [49]. As far as possible, we select test cases where theoretical predictions are available so that useful comparisons with the numerical results can be made. This is, for example, the case for (i), (iv) and (v).

7.1. 2D Static bubble

A stationary circular bubble at equilibrium is considered. The setup of the test case follows the description in [3, 6]. The radius of the bubble is $r = 0.25$ m and is positioned at the center of the computational domain Ω which is a square container with side $4r$ as shown in Figure 12. The densities and viscosities of the two fluids in Ω_1 and Ω_2 are identical with $\varrho_1 = \varrho_2 = 1$ kg/m^3 and $\mu_1 = \mu_2 = 1$ kg/s/m. No externally applied forces are considered and no-slip conditions are assumed along the walls of the container. The pressure $p = 0$ N/m^2 is fixed at the top left corner of the domain. The surface tension coefficient is set as $\gamma = 0.01$ kg/s^2.

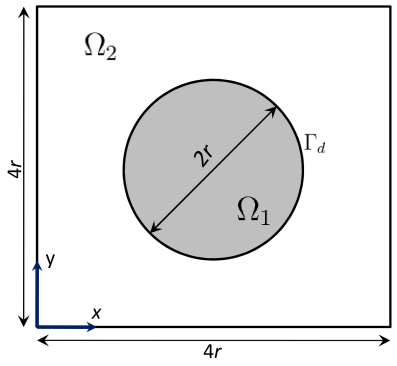


Figure 12: Problem statement for the static bubble test case.

This test case is characterized by 3 non-dimensional numbers [3]: the density ratio ϱ_2/ϱ_1, the viscosity ratio μ_2/μ_1 and the Laplace number defined as

$$\mathrm{La} = \frac{\varrho_2 \gamma d}{\mu_2^2}, \tag{64}$$

where d is the diameter of the bubble. For this test case, $\varrho_2/\varrho_1 = 1$, $\mu_2/\mu_1 = 1$ and $\mathrm{La} = 0.005$. Due to surface tension effects, the numerical capillarity time-step size limit (49) is applicable, where $\Delta t^{(\mathrm{ca})}_{\mathrm{num}} = 0.005$ s. The time step in the Crank-Nicholson scheme is then chosen as $\Delta t = 0.002$ s and the simulation is performed until $t = 4$ s with 2000 time steps.

Since the bubble is stationary, the velocity is zero and the interfacial condition (9) reduces to

$$p_1 - p_2 = \frac{\gamma}{r}, \tag{65}$$

which is known as the Laplace-Young law. This implies that a pressure jump of $p_1 - p_2 = 0.04\ \mathrm{N/m}^2$ across the two-fluid interface is to be expected. The zero velocity field and the theoretical pressure jump computed above will be used as the reference solutions upon which the errors are computed.

For the spatial discretization, mesh sizes of $5 \times 5, 10 \times 10, 20 \times 20$ and 40×40 with a constant refinement level $n_{\mathrm{ref}} = 2$ are employed. This translates to resolutions of $\frac{1}{20}, \frac{1}{40}, \frac{1}{80}, \frac{1}{160}$ in the vicinity of the interface, see Figure 13. Three different methods of computing the normal vector $\hat{\boldsymbol{n}}$ in the surface tension term are considered:

Method A

The normal vector $\hat{\boldsymbol{n}}$ is computed according to procedure described in Section 6.2. The normal vector results from a systematic mapping of the tangential vectors from the reference to the real domains. As such, $\hat{\boldsymbol{n}}$ is always perpendicular to the piecewise linear interface.

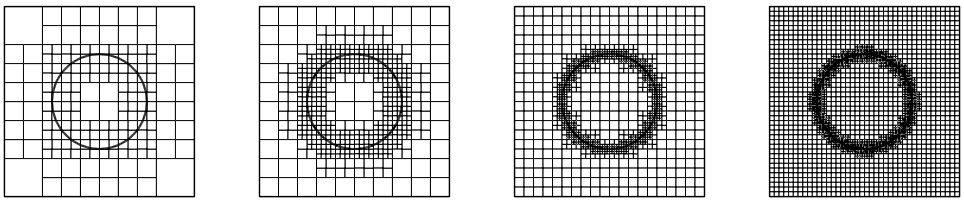
Method B

The normal vector $\hat{\boldsymbol{n}}$ is computed according to the procedure described in equations (62) − (63). The normal vector on the interface is interpolated from the nodal level-set values using the first derivatives of the 2D bilinear shape functions.

Method C

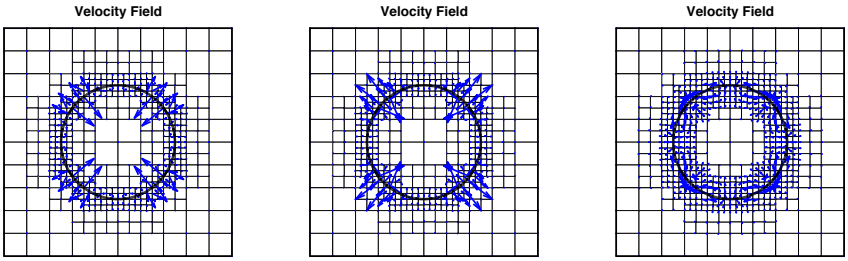
The normal vector $\hat{\boldsymbol{n}}$ is computed according to the procedure described in equations (57) − (61). The level-set function is first differentiated using a standard FD central differencing. Thereafter, the normal vector on the interface is interpolated from the nodal values of the FD-differentiated level-set function using the 2D bilinear shape functions. This method is only applicable for Cartesian meshes.

The pressure approximation is sign-enriched according to (26). The criterion (33) is employed where all elements within a certain distance $d = 0.016$ m from the interface are refined. The level-set is kept fixed throughout the simulation and no reinitialization is required. The errors in the pressure and velocity fields for this test case come from three possible sources: (i) approximation error for the discontinuous pressure, (ii) geometrical error of the interpolated interface and (iii) discretization error of the surface tension force. Due to these error sources, the computed velocity field is not exactly zero. Figure 14 displays the (non-zero) velocity field for a 10×10 mesh computed using the three methods at the final time $t = 4$ s. Visual inspection reveals that the spurious velocities are much smaller in magnitude for method C. This observation will be further augmented by the computed convergence rates in the sequel. The pressure field computed using Method



(a) 5×5 (b) 10×10 (c) 20×20 (d) 40×40

Figure 13: Meshes used for the static bubble test case. A constant refinement level $n_{\text{ref}} = 2$ is maintained.



(a) Method A (b) Method B (c) Method C

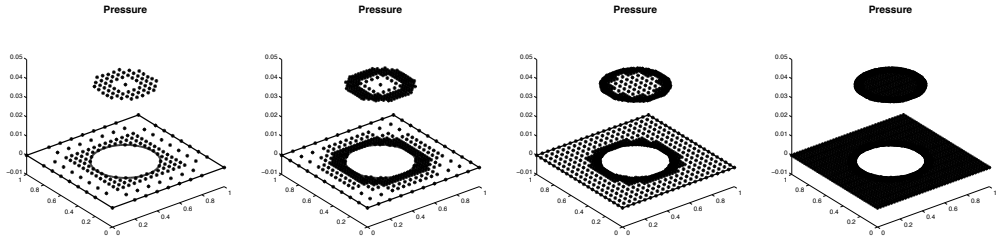
Figure 14: Velocity field for the static bubble test case at $t = 4$ s computed using three different methods of computing the normal vectors.

C is shown in Figure 15 for all four mesh sizes. The jump in the pressure field across the interface is apparent. Figure 16 shows the variation of the pressure field along the cross-section $y = 0.5$ m.

We first compute the L^∞-norm of the non-dimensional velocity error defined as

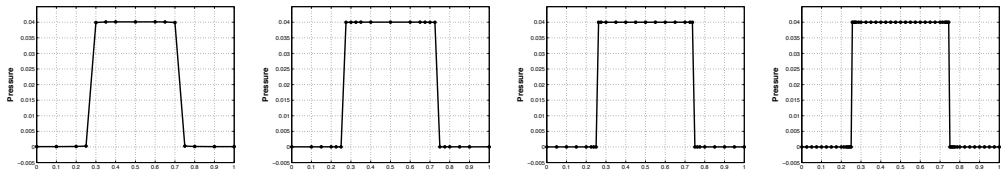
$$||\boldsymbol{u}^h \frac{\mu}{\gamma}||_{L^\infty} = \max \left| \boldsymbol{u}^h \frac{\mu}{\gamma} \right|. \tag{66}$$

The results from the three methods A, B and C described above are compared with the results of Smolianski [3] (abbreviated SMO) and Hysing [6] (abbreviated HS), see Figure 17. The study 'HS' employs a bilinear approximation for the velocity and a constant approximation for the pressure. The study 'SMO' employs equal-order linear approximations for both velocity and pressure. Both studies 'SMO' and 'HS' employ a standard FEM without any enrichment of the approximation. As can be observed from the figure, all methods yield convergence rates close to 1.0 (only the study 'SMO' yields a suboptimal rate of 0.7). The enrichment of the pressure space used in this study does not lead to an improvement in the convergence rate of the velocity error, though accuracy is drastically improved.



(a) 5×5 (b) 10×10 (c) 20×20 (d) 40×40

Figure 15: Pressure field for the static bubble test case at $t = 4$ s computed using Method C for different mesh sizes.



(a) 5×5 (b) 10×10 (c) 20×20 (d) 40×40

Figure 16: Pressure field for the static bubble test case at $t = 4$ s along the cross-section $y = 0.5$m computed using Method C for different mesh sizes.

Among the three methods A, B and C, method C yields the best accuracy overall and method A the least accurate. Recall that the normal vectors computed using method A are consistently perpendicular to the piecewise linear interface. On the other hand, the normal vectors computed using methods B and C are not necessarily perpendicular to the piecewise linear interface. However, the results above show that methods B and C nevertheless lead to a higher accuracy. We surmise that this is because the piecewise linear interface is only a crude approximation to the real interface position and the normal vectors computed using method A are not necessarily a more accurate representation of the true normal vectors (even though they are consistently perpendicular to the interpolated interface). Finally, we observe that method C yields a higher accuracy than method B. Recall that method B relies on the interpolation properties of the *differentiated* bilinear shape functions (62) which are necessarily less accurate than the original bilinear shape functions used in method C. Therefore, using a FD central differencing to first compute the derivative of the level-set function shifts the burden of differentiation from the bilinear shape functions in method C. However, it is to be noted that such a strategy (i.e. finite difference of the level-set) is possible only for a Cartesian mesh considered in this test case. This simple test case demonstrates the importance of an appropriate discretization of the surface tension term.

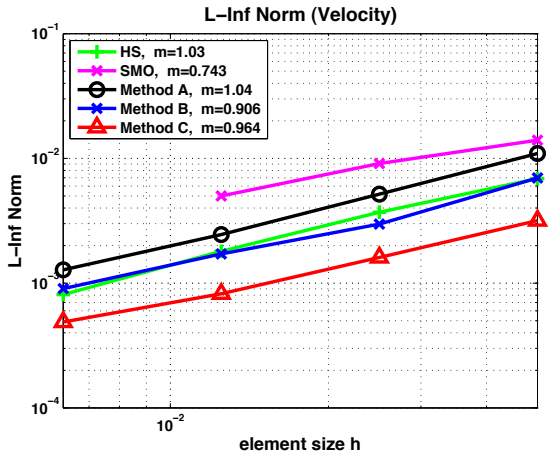
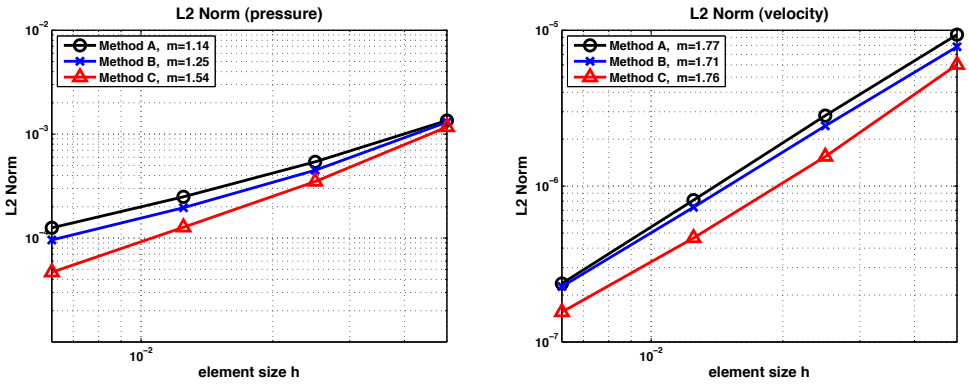


Figure 17: Comparison of L^∞-norm for the non-dimensional velocity for the static bubble test case.

We now compute the L_2-norm of the velocity error defined as

$$\| \boldsymbol{u}^{ex} - \boldsymbol{u}^h \|_{L_2} = \sqrt{\sum_{\text{all } \Omega^e} \int_{\Omega^e} \sum_{i=1}^{n^{\text{sd}}} \left(u_i^{ex} - u_i^h\right)^2 \, \mathrm{d}\Omega^e}, \tag{67}$$

where $u_i^{ex} = 0$ and u_i^h are the exact and approximated velocity in the x_i-direction, respectively. Ω^e refers to the element domain and n^{sd} is the number of spatial dimensions.



(a) Convergence of pressure.

(b) Convergence of velocity.

Figure 18: Convergence plots for L_2-norm for (a) pressure and (b) velocity.

The L_2-norm of the pressure error is defined as

$$\| p^{ex} - p^h \|_{L_2} = \sqrt{\sum_{\text{all } \Omega^e} \int_{\Omega^e} \left(p^{ex} - p^h\right)^2 \, \mathrm{d}\Omega^e}, \tag{68}$$

where p^{ex} and p^h are the exact and approximated pressure fields, respectively. Figure 18(a) and (b) display the convergence plots in the L_2-norm for the pressure and velocity errors, respectively. Similar to the convergence of the L^∞-norm for the velocity error, the convergence plots for the L_2-norm exhibit the same trends, i.e. method C yields the best accuracy, followed by method B and method A. Once again, for the convergence plot of the L_2-norm of the velocity error, the convergence rates of all three methods do not differ significantly, though accuracy is very much improved for method C. However, for the L_2-norm of the pressure error, method C not only yields a higher accuracy but also a higher convergence rate compared to the other two methods. The best convergence rate achieved is around 1.5 for the pressure error and around 1.75 for the velocity error. However, both are suboptimal compared to the supposedly optimal rate of 2.0 using a bilinear approximation used in this study.

7.2. 2D Rising bubble with large Eötvös number

A rising bubble with a large Eötvös number is considered next. The setup of the test case follows the description in [13] and is shown in Figure 19. The diameter of the bubble is $d = 10^{-0.5}$ m. The densities of the two fluids in Ω_1 and Ω_2 are $\varrho_1 = 1$ kg/m^3 and $\varrho_2 = 1000$ kg/m^3 and the viscosities are $\mu_1 = 10^{-3.5}$ kg/s/m and $\mu_2 = 10^{-1.5}$ kg/s/m. A gravitational acceleration of $g_y = -0.01$ m/s^2 is considered. Slip-conditions are assumed along the walls of the tank, and $p = 0$ N/m^2 is fixed at the top left corner of the domain. The surface tension coefficient is set as $\gamma = 0.001$ kg/s^2 and the situation is observed for $t = (0\text{s}, 25\text{s})$.

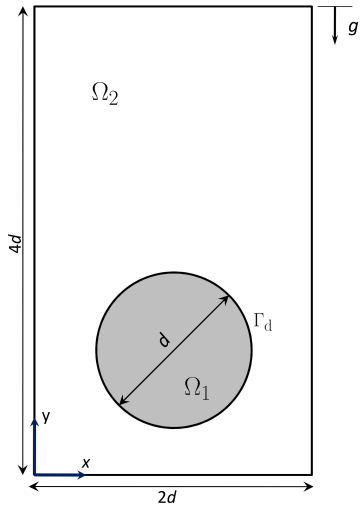


Figure 19: Problem statement for the rising bubble test case.

This test case is characterized by 4 non-dimensional numbers [4, 3]: the density ratio ϱ_2/ϱ_1, the viscosity ratio μ_2/μ_1, the Morton number defined as

$$M = \frac{g\mu_2^4}{\varrho_2\gamma^3}, \tag{69}$$

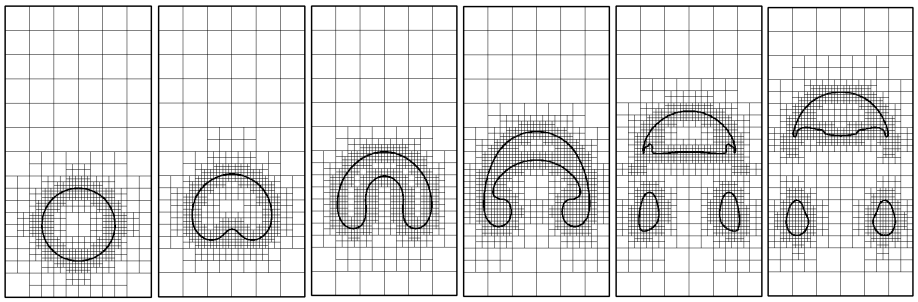
and the Eötvös number

$$Eo = \frac{\varrho_2 g d^2}{\gamma}, \tag{70}$$

where d is the diameter of the bubble and g is the magnitude of the gravitational acceleration. The Morton number is used to characterize the shape of the evolving bubble and the Eötvös number (or Bond number) is the ratio of body forces to surface tension forces. For this test case, $\varrho_2/\varrho_1 = 1000$, $\mu_2/\mu_1 = 100$, $M = 0.01$ and $Eo = 1000$. The relatively large Eo number implies that body forces dominate the capillary forces.

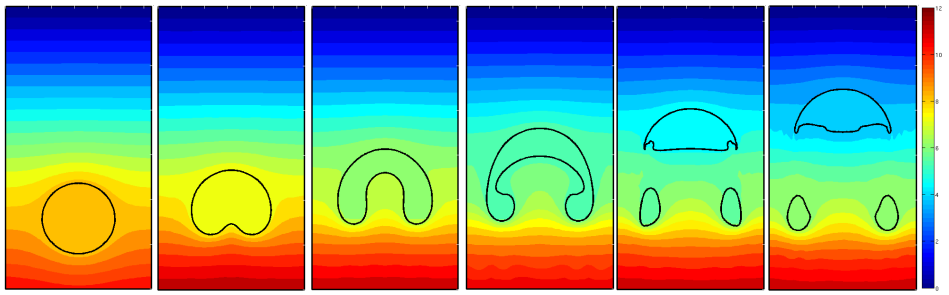
In addition to comparing the three methods of computing the normal vectors described in the previous test case (Section 7.1), two methods of reinitialization are also considered: (i) the method as described in Section 2.3 where the PDE (24) is solved to steady state and (ii) the standard reinitialization where a new signed-distance function is constructed based on the interpolated front. In the sequel, method (i) is abbreviated "PDE-reinitialized" and method (ii) is abbreviated "Interpolation-reinitialized".

Due to surface tension effects, the numerical capillarity time-step size limit (49) is applicable, where $\Delta t_{\text{num}}^{(\text{ca})} = 1$ s. This large capillarity time-step limit is due to the small surface tension coefficient and the relatively large diameter of the bubble. Nevertheless, the time step in the Crank-Nicholson scheme is chosen as $\Delta t = 0.025$ s in order to capture the evolution of the interface. For the spatial discretization, a coarse mesh size of 6×12



(a) $t = 0$ s (b) $t = 5$ s (c) $t = 10$ s (d) $t = 15$ s (e) $t = 20$ s (f) $t = 25$ s

Figure 20: Evolution of the mesh for the rising bubble test case. The two-fluid interface is shown as a black solid line.



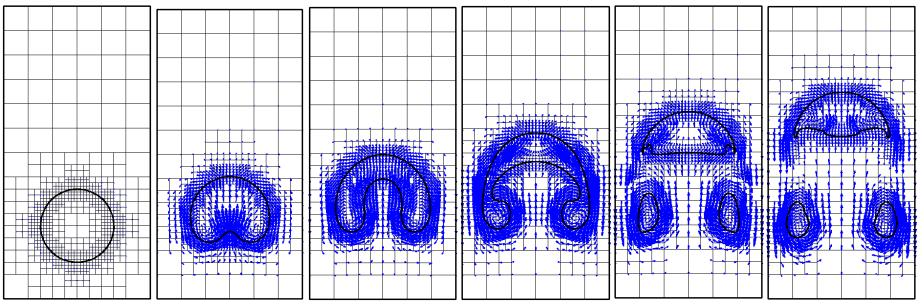
(a) $t = 0$ s (b) $t = 5$ s (c) $t = 10$ s (d) $t = 15$ s (e) $t = 20$ s (f) $t = 25$ s

Figure 21: Evolution of the pressure field for the rising bubble test case. The two-fluid interface is shown as a black solid line.

elements is chosen. Refinement levels $n_{\text{ref}} = 2, 3$ and 4 are considered. This translates to resolutions of $2d/24, 2d/48, 2d/64$ in the vicinity of the interface.

The pressure approximation is sign-enriched according to (26). The criterion (33) is employed where all elements within a certain distance $d = 0.015$ m from the interface are refined. The evolutions of the mesh, pressure and velocity fields for $n_{\text{ref}} = 3$ using Method C for computing the normal vectors with a PDE-reinitialized methodology are shown in Figure 20, Figure 21 and Figure 22, respectively.

Since both fluids are immiscible and incompressible, the masses (or areas) of the two fluids should be conserved over time. Figure 23(a) compares the mass conservation of the bubble in Ω_1 over time for the three methods of computing the normal vector using an interpolation-reinitialized level-set and a refinement level $n_{\text{ref}} = 3$. Figure 23(b) makes the same comparison but using a PDE-reinitialized level-set. Two important observations can be made here. First, a PDE-reinitialized level-set leads to a much better mass conservation compared to an interpolation-reinitialized level-set. Second, all three methods of computing the normal vector lead to only small differences in the mass conservation. It seems that the different ways of computing the normal vector do



(a) $t = 0$ s (b) $t = 5$ s (c) $t = 10$ s (d) $t = 15$ s (e) $t = 20$ s (f) $t = 25$ s

Figure 22: Evolution of the velocity field for the rising bubble test case. The two-fluid interface is shown as a black solid line.

not have a significant bearing when surface tension effects are relatively unimportant compared to the body forces (as manifested by a large Eötvös number).

Finally, Figure 24(a) compares the mass conservation for three different levels of refinement $n_{\text{ref}} = 1, 2, 3$ with corresponding DOFs shown in Figure 24(b). A PDE-reinitialized level-set together with method C for computing the normal vector are employed. As can be seen, the finer the mesh resolution, the better is the mass conservation, as expected.

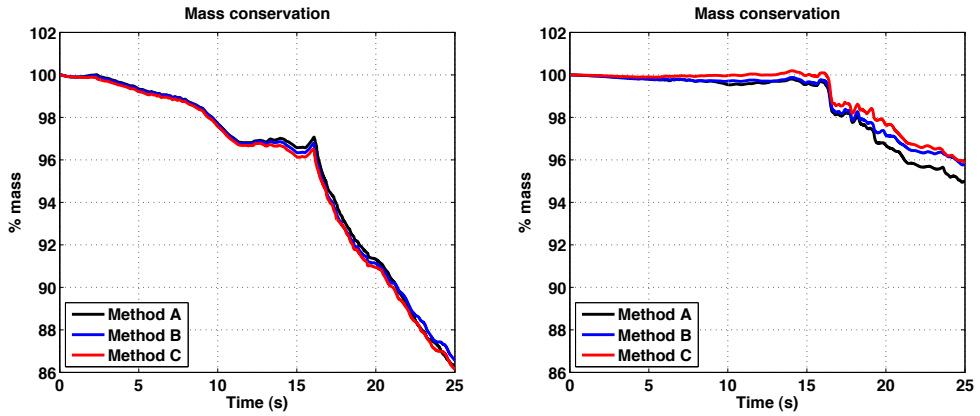
7.3. Rising bubble with small Eötvös number

A rising bubble with a small Eötvös number is considered next. The setup of the test case follows the description in [13] and is shown in Figure 19. The diameter of the bubble is now $d = 0.01$ m. The fluid properties are identical to those of the previous test case and will therefore not be repeated. Slip-conditions are assumed along the walls of the tank, and $p = 0$ N/m^2 is fixed at the top left corner of the domain. The situation is observed for $t = (0\text{ s}, 15\text{ s})$.

This test case is also characterized by the same 4 non-dimensional numbers as described in the previous test case. Since only the size of the bubble changes, the dimensionless numbers $\varrho_2/\varrho_1 = 1000$, $\mu_2/\mu_1 = 100$ and $M = 0.01$ are identical to those for the previous test case. Only the Eötvös number $Eo = 1$ is reduced. The relatively small Eo number implies that the capillary forces now dominate the body forces.

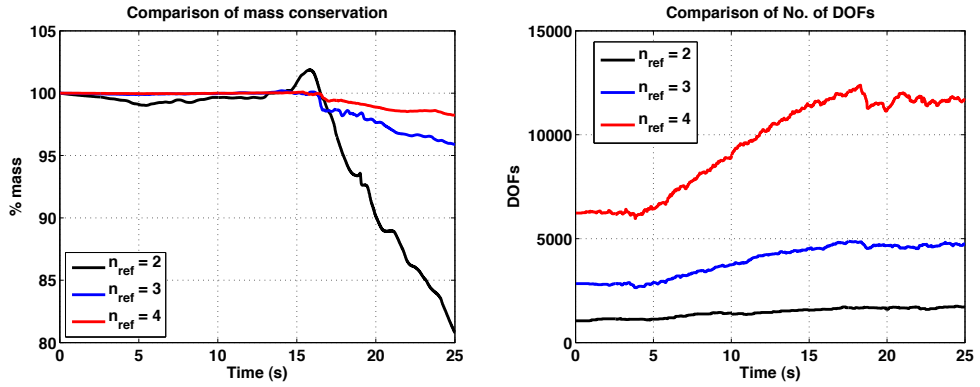
Similar to the previous test case, we compare the three different methods of computing the normal vectors and also the two different reinitialization strategies: "PDE-reinitialized" and "Interpolation-reinitialized". Due to surface tension effects, the numerical capillarity time-step size limit (49) is applicable, where $\Delta t^{(\text{ca})}_{\text{num}} = 0.006$ s. This rather restrictive capillarity time-step limit results from the small diameter of the bubble and is typical for cases where capillary forces dominate. The time step in the Crank-Nicholson scheme is therefore chosen as $\Delta t = 0.006$ s. For the spatial discretization, a coarse mesh size of 6×12 elements is chosen. A refinement level of $n_{\text{ref}} = 3$ is considered. This translates to a resolution of $2d/48$ in the vicinity of the interface.

The pressure approximation is sign-enriched according to (26). The criterion (33) is employed where all elements within the distance $d = 0.001$ m from the interface are



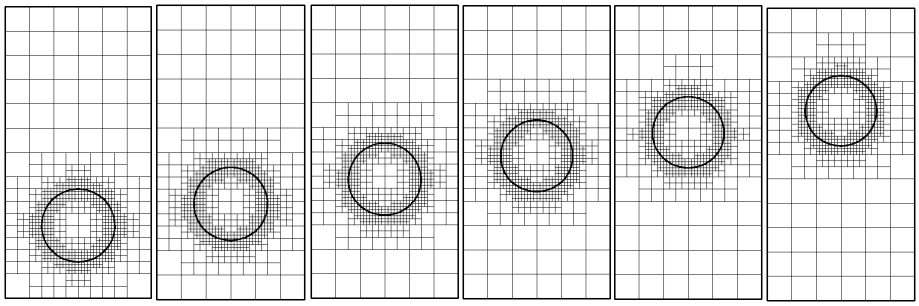
(a) Interpolation-reinitialized (b) PDE-reinitialized

Figure 23: Comparison of mass conservation for the rising bubble test case using (a) an interpolation-reinitialized level-set and (b) a PDE-reinitialized level-set.



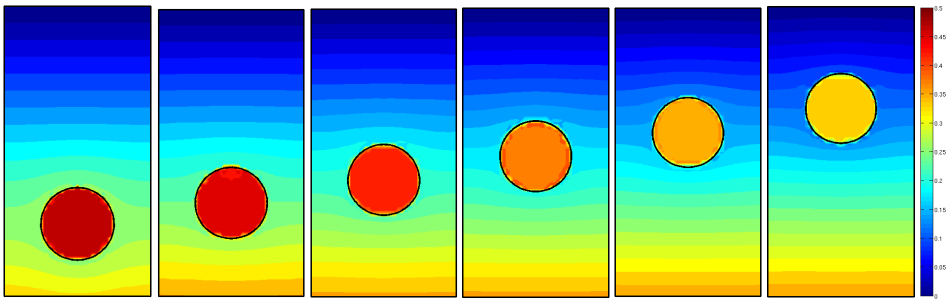
(a) Comparison of mass conservation. (b) Comparison of DOFs.

Figure 24: Comparison of (a) mass conservation and (b) number of DOFs for the rising bubble test case between three different levels of refinement $n_{\text{ref}} = 2, 3, 4$. A PDE-reinitialized level-set together with method C for computing the normal vector are employed.



(a) $t = 0$ s (b) $t = 5$ s (c) $t = 10$ s (d) $t = 15$ s (e) $t = 20$ s (f) $t = 25$ s

Figure 25: Evolution of the mesh for the rising bubble test case. The two-fluid interface is shown as a black solid line.

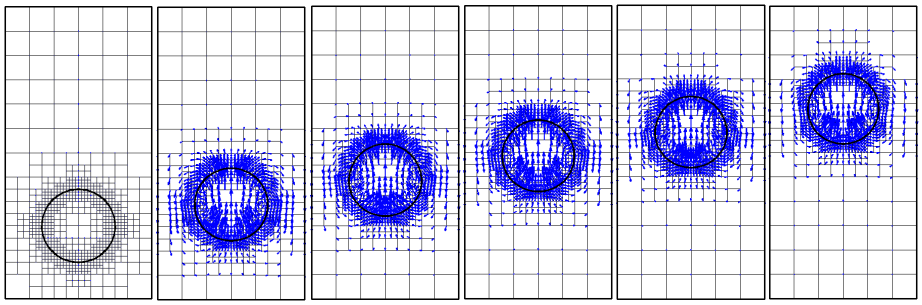


(a) $t = 0$ s (b) $t = 5$ s (c) $t = 10$ s (d) $t = 15$ s (e) $t = 20$ s (f) $t = 25$ s

Figure 26: Evolution of the pressure field for the rising bubble test case. The two-fluid interface is shown as a black solid line.

refined. The evolutions of the mesh, pressure and velocity fields for $n_{\text{ref}} = 3$ using Method C for computing the normal vectors with a PDE-reinitialized methodology are shown in Figure 25, Figure 26 and Figure 27, respectively. As can be observed from Figure 26, the pressure within the rising bubble is much higher than that of the surrounding fluid. Further, the circular shape of the bubble seems to be maintained as it rises up the container. Both phenomena can be explained by the large surface tension forces for this test case.

Figure 28(a) compares the mass conservation of the bubble in Ω_1 over time for the three methods of computing the normal vector using an interpolation-reinitialized level-set and a refinement level $n_{\text{ref}} = 3$. Figure 28(b) makes the same comparison but using a PDE-reinitialized level-set. Two important observations can be made here. First, a PDE-reinitialized level-set leads to a much better mass conservation compared to an interpolation-reinitialized level-set (except for the case where normal vectors are computed by method A). Second, contrary to the previous test case, the three different methods of computing the normal vector lead to significant differences in the mass con-



(a) $t = 0$ s (b) $t = 5$ s (c) $t = 10$ s (d) $t = 15$ s (e) $t = 20$ s (f) $t = 25$ s

Figure 27: Evolution of the velocity field for the rising bubble test case. The two-fluid interface is shown as a black solid line.

servation, with method C yielding the best results. We deduce that the way in which the normal vector is computed is important when surface tension effects dominate the body forces (as manifested by a small Eötvös number).

7.4. 3D Static bubble

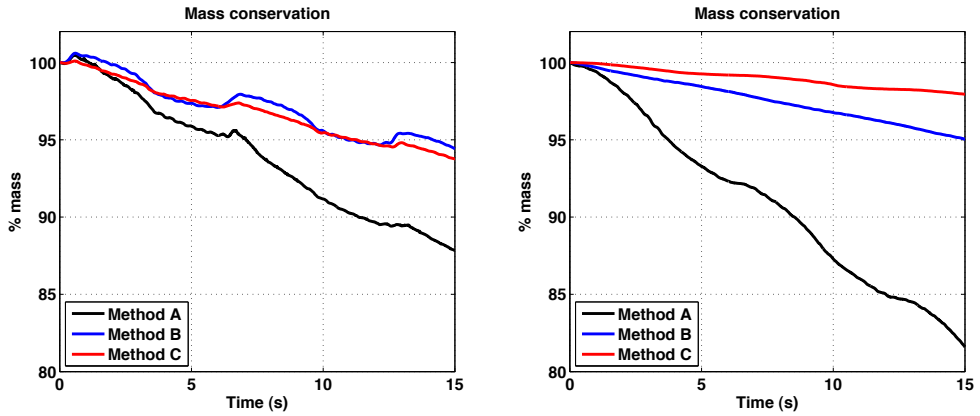
A stationary 3D spherical bubble at equilibrium is considered. The setup of the test case follows the description in [49]. The radius of the bubble is $r = 2/3$ m and is positioned at the center of the computational domain Ω which is a cubic container with side 2.0 m as shown in Figure 29. The densities and viscosities of the two fluids in Ω_1 and Ω_2 are identical with $\varrho_1 = \varrho_2 = 1$ kg/m^3 and $\mu_1 = \mu_2 = 1$ kg/s/m. No externally applied forces are considered and no-slip conditions are assumed along the walls of the container. The pressure $p = 0$ N/m^2 is fixed at the corner ($x = 0$ m, $y = 2.0$ m, $z = 2.0$ m) of the domain. The surface tension coefficient is set as $\gamma = 1$ kg/s^2.

Similar to the 2D version, this test case is characterized by the non-dimensional numbers: the density ratio $\varrho_2/\varrho_1 = 1$, the viscosity ratio $\mu_2/\mu_1 = 1$ and the Laplace number $La = 4/3$. The much larger Laplace number (compared to the 2D version) means that surface tension forces dominate the viscous forces. Due to surface tension effects, the numerical capillarity time-step size limit (49) is applicable, where $\Delta t^{(\mathrm{ca})}_{\mathrm{num}} = 0.04$ s. Nevertheless, a much smaller time step size in the Crank-Nicholson scheme is chosen, i.e. $\Delta t = 0.002$ s. The simulation is performed until $t = 0.4$ s with 200 time steps.

Since the bubble is stationary and the velocity is zero, the interfacial condition (9) reduces to

$$p_1 - p_2 = 2\frac{\gamma}{r}, \tag{71}$$

which is the 3D version of the Laplace-Young law. This implies that a pressure jump of $p_1 - p_2 = 3$ N/m^2 across the two-fluid interface is to be expected. The zero velocity field and the theoretical pressure jump computed above will be used as the reference solutions upon which the errors are computed. For the spatial discretization, a fixed coarse mesh size of $4 \times 4 \times 4$ elements with refinement levels $n_{\mathrm{ref}} = 1, 2, 3, 4$ are employed. This translates to resolutions of $1/4, 1/8, 1/16, 1/32$ in the vicinity of the interface.



(a) Interpolation-reinitialized (b) PDE-reinitialized

Figure 28: Comparison of mass conservation for the rising bubble test case using (a) an interpolation-reinitialized level-set and (b) a PDE-reinitialized level-set.

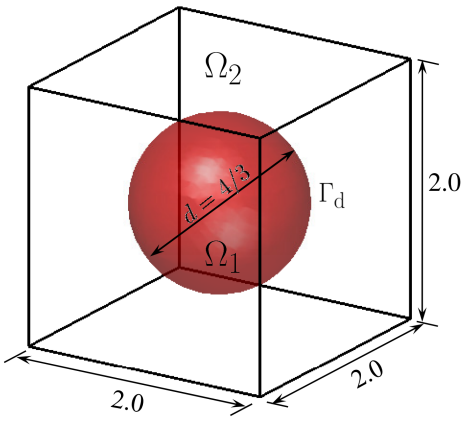
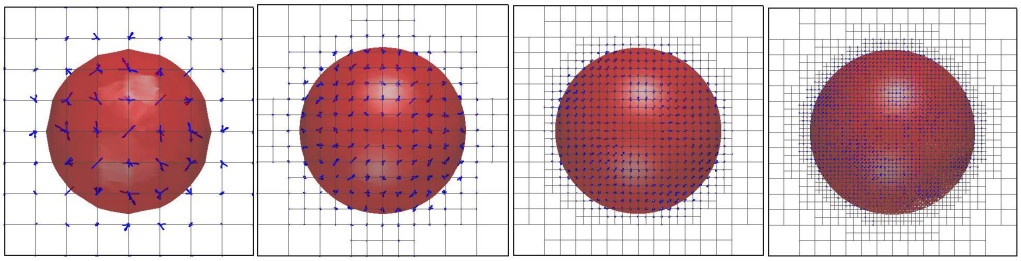
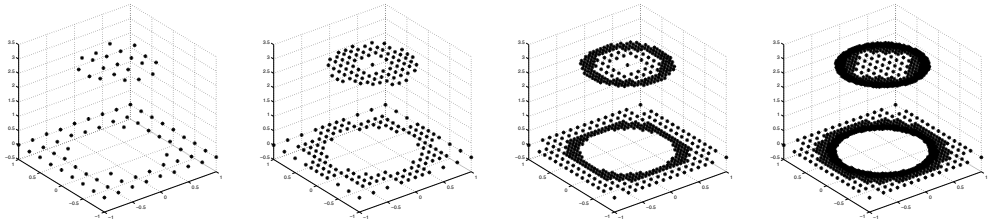


Figure 29: Problem statement for the 3D static bubble test case.



(a) $n_{\text{ref}} = 1$. (b) $n_{\text{ref}} = 2$. (c) $n_{\text{ref}} = 3$. (d) $n_{\text{ref}} = 4$.

Figure 30: Spurious velocities for the 3D static bubble at $t = 0.4$ s computed for 4 different mesh resolutions.



(a) $n_{\text{ref}} = 1$. (b) $n_{\text{ref}} = 2$. (c) $n_{\text{ref}} = 3$. (d) $n_{\text{ref}} = 4$.

Figure 31: Pressure field on the cross-section $z = 0$ m of the 3D static bubble at $t = 4$ s computed for different mesh resolutions.

The pressure approximation is sign-enriched according to (26). The criterion (33) is employed where all elements within a certain distance $d = 0.05$ from the interface are refined. The level-set is kept fixed throughout the simulation and no reinitialization is required. The errors in the pressure and velocity fields for this test case come from three sources: (i) approximation error for the discontinuous pressure, (ii) geometrical error of the interpolated interface and (iii) discretization error of the surface tension force. Due to these error sources, the computed velocity field is not exactly zero. Figure 30 displays the (non-zero) velocity field for the four different mesh resolutions at the final time $t = 4$ s. Visual inspection reveals that the spurious velocities decrease in magnitude as mesh resolution is increased. Figure 31 shows the variation of the pressure field on the cross-section $z = 0$ m. The jump in the pressure field across the two-fluid interface is apparent.

The results of this study are compared with the results of Groß [49] (abbreviated 'Gross (2008)') which employs a linear approximation of the pressure field and a quadratic approximation for the velocity field and level-set. Further, similar to this study, the pressure field is sign-enriched. The normal vectors are computed using a procedure similar to method B (Section 7.1) which the author dubbed the 'improved Laplace-Beltrami discretization'. A detailed comparison between the methodologies employed in

'Gross (2008)' and in this study appears in Table 1.

Table 1: Comparison between Gross (2008) and this study.

	Study	
	Gross (2008)	**This study**
1. **Mesh**	unstructured tetrahedral mesh	structured hexahedral mesh
2. **Refinement**	hierarchy of tetrahedral elements, conforming mesh	hierarchy of hexahedral elements via hanging nodes, 1-irregular mesh
3. **Approximation**	linear for pressure, quadratic for velocity, level-set	trilinear for pressure, velocity and level-set
4. **Enrichment**	sign-enriched for pressure	signed-enriched for pressure
5. **Interface description**	level-set method	level-set method
6. **Reinitialization**	fast-marching method	PDE-reinitialized
7. **Surface tension discretization**	Laplace-Beltrami	Laplace-Beltrami
8. **Normal vector**	quadratic interpolation (method B)†	trilinear interpolation (method C)†

† Refer to Section 7.1.

We compute the L_2-norm of the velocity and pressure errors defined as in (67) and (68), respectively. A side-by-side comparison of the convergence plots between the two studies is not meaningful since the definition of an element size is not available for Gross (2008) due to the use of an unstructured mesh. Therefore, only convergence rates are compared. The study of Gross (2008) uses a uniform initial triangulation where the vertices form a $5 \times 5 \times 5$ lattice. Four levels of refinement are subsequently applied $n_{\text{ref}} = 1, 2, 3, 4$ to extract the successive convergence rates.

The pressure and velocity errors for Gross (2008) and this study are displayed in Table 2 and Table 3, respectively. We reiterate that one should *not* compare the error norms corresponding to the different refinement levels since the corresponding mesh resolutions are not equivalent in the two separate studies. The only meaningful comparison is in the convergence order. For the pressure errors, it is observed that the two studies yield very close convergence orders averaging around 1.5. This number is also consistent with the result of the 2D version (Figure 18(a)). Results for velocity errors show that Gross (2008) yields an average convergence rate of 2.0 compared to the lower rate of 1.5 for this study. This is not unexpected since Gross (2008) employs a quadratic approximation for both the velocity field and the level-set (which also leads to a more accurate discretization of the surface tension) compared to a trilinear approximation used in this study.

7.5. Rising n-butanol droplet

Finally, a rising n-butanol droplet is considered. The setup of the test case follows the description in [49] and is shown in Figure 32. A single n-butanol droplet is located in a tank Ω of dimensions [0.02 m $\times$ 0.02 m $\times$ 0.03 m]. The densities of the two fluids in Ω_1 and Ω_2 are $\varrho_1 = 845.4$ kg/m^3 and $\varrho_2 = 986.5$ kg/m^3 and the viscosities are

Table 2: Errors and convergence rates for pressure and velocity for Gross (2008).

Gross (2008)				
$\mathbf{n}_{\text{ref}}$	$\| p^{ex} - p^h \|_{L_2}$	**Order**	$\| \boldsymbol{u}^{ex} - \boldsymbol{u}^h \|_{L_2}$	**Order**
0	$1.64e^{-1}$	-	$7.16e^{-3}$	-
1	$4.97e^{-2}$	1.73	$1.57e^{-3}$	2.19
2	$1.66e^{-2}$	1.58	$3.25e^{-4}$	2.28
3	$7.16e^{-3}$	1.22	$8.57e^{-5}$	1.92
4	$2.83e^{-3}$	1.34	$1.75e^{-5}$	2.29

Table 3: Errors and convergence rates for pressure and velocity for this study.

This study				
$\mathbf{n}_{\text{ref}}$	$\| p^{ex} - p^h \|_{L_2}$	**Order**	$\| \boldsymbol{u}^{ex} - \boldsymbol{u}^h \|_{L_2}$	**Order**
1	$7.90e^{-1}$	-	$3.73e^{-3}$	-
2	$2.54e^{-1}$	1.64	$1.58e^{-3}$	1.24
3	$1.05e^{-1}$	1.27	$5.47e^{-4}$	1.53
4	$3.89e^{-2}$	1.48	$1.75e^{-4}$	1.65

$\mu_1 = 1.388 \cdot 10^{-3}$ kg/s/m and $\mu_2 = 3.281 \cdot 10^{-3}$ kg/s/m. A gravitational acceleration of $g_z = -9.81$ m/s^2 is considered. Slip-conditions are assumed along the walls of the tank, and $p = 0$ N/m^2 is fixed at one corner of the domain. The initial position of the droplet's center is located at $(0.01 \text{ m}, 0.01 \text{ m}, 0.003 \text{ m})$. The surface tension coefficient is set as $\gamma = 1.63 \cdot 10^{-3}$ kg/s^2 and the situation is observed for $t = (0 \text{ s}, 0.5 \text{ s})$.

Different radii of the bubble are considered: $r = 0.75$ mm, 1.0 mm, 1.25 mm, 1.50 mm, 1.75 mm, and 2.0 mm. This test case is characterized by the 4 non-dimensional numbers: the density ratio $\varrho_2/\varrho_1 = 1.167$, the viscosity ratio $\mu_2/\mu_1 = 0.423$, the Morton number $M = 8.53 \cdot 10^{-6}$ and the Eötvös number which varies from $Eo = 13.4$ to $Eo = 95.0$, see Table 4.

For the spatial discretization, a coarse mesh size of $6 \times 6 \times 9$ elements with refinement level $n_{\text{ref}} = 3$ is chosen for the bubble radii $r = 1.50$ mm, 1.75 mm, and 2.0 mm. On the other hand, a coarse mesh size of $4 \times 4 \times 6$ elements with a refinement level $n_{\text{ref}} = 4$ is chosen for the bubble radii $r = 0.75$ mm, 1.0 mm and 1.25 mm. This translates to resolutions of 0.02/48 m and 0.02/64 m in the vicinity of the interface for the two groups of radii, respectively. The use of a finer resolution for the latter group is due to the smaller size of the bubble which requires a finer mesh to capture the interfacial position accurately. Due to surface tension effects, the numerical capillarity time-step size limit (49) is applicable, where $\Delta t_{\text{num}}^{(\text{ca})} = 0.004$s. Therefore, the time step in the Crank-Nicholson scheme is chosen as $\Delta t = 0.001$ s.

The pressure approximation is sign-enriched according to (26). The criterion (33) is employed where all elements within a certain distance $d = 0.0008$ m from the interface

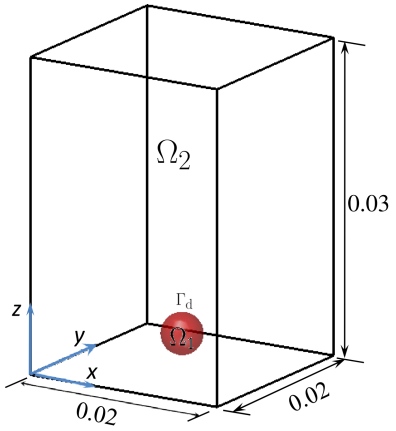
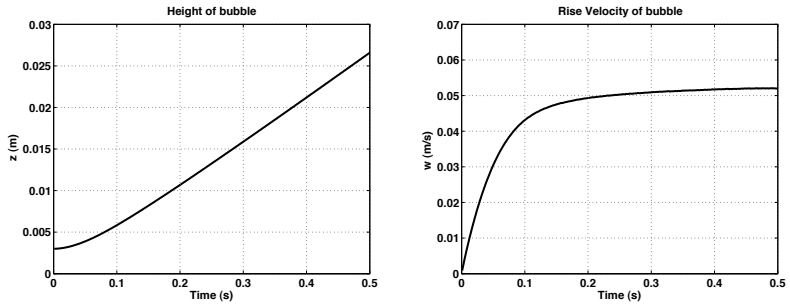


Figure 32: Problem statement for the 3D rising n-butanol droplet test case.

Table 4: Different sizes of the droplet with corresponding Eötvös numbers considered in this study.

Radius (mm)	Eötvös number
0.75	13.4
1.00	23.7
1.25	37.1
1.50	53.4
1.75	72.7
2.00	95.0



(a) (b)

Figure 33: Variation of (a) the barycenter $\bar{\boldsymbol{r}}_d(t)$ and (b) the average velocity $\bar{\boldsymbol{v}}_d(t)$ of the $r = 1.0$ mm droplet with time.

are refined. Following Groß [49], we define the droplet's barycenter $\bar{\boldsymbol{r}}_d$ as

$$\bar{\boldsymbol{r}}_d(t) = \frac{1}{V(\Omega_1)} \int_{\Omega_1} \boldsymbol{x} \, \mathrm{d}\Omega, \tag{72}$$

where $V(\Omega_1)$ is the volume of the droplet. Similarly, the average velocity of the droplet $\bar{\boldsymbol{v}}_d$ is defined as

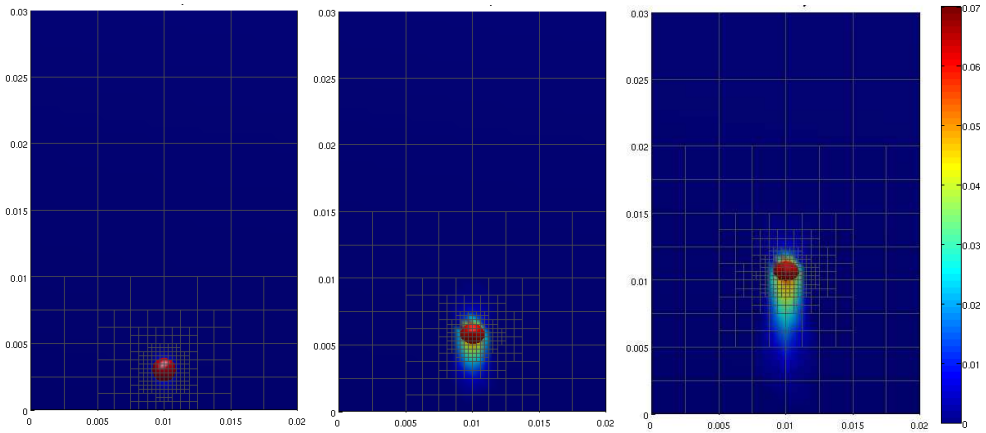
$$\bar{\boldsymbol{v}}_d(t) = \frac{1}{V(\Omega_1)} \int_{\Omega_1} \boldsymbol{v}(\boldsymbol{x}, t) \, \mathrm{d}\Omega. \tag{73}$$

The variations of $\bar{\boldsymbol{r}}_d(t)$ and $\bar{\boldsymbol{v}}_d(t)$ for the $r = 1.0$ mm droplet with time are shown in Figure 33(a) and (b), respectively. As can be seen from Figure 33(b), the average velocity $\bar{\boldsymbol{v}}_d(t)$ reaches a certain steady value after some time. We refer to this as the *terminal velocity* of the droplet. The evolution of the z-component of the velocity field for the same $r = 1.0$ mm droplet on the cross-section $y = 0.01$ m is shown in Figure 34.

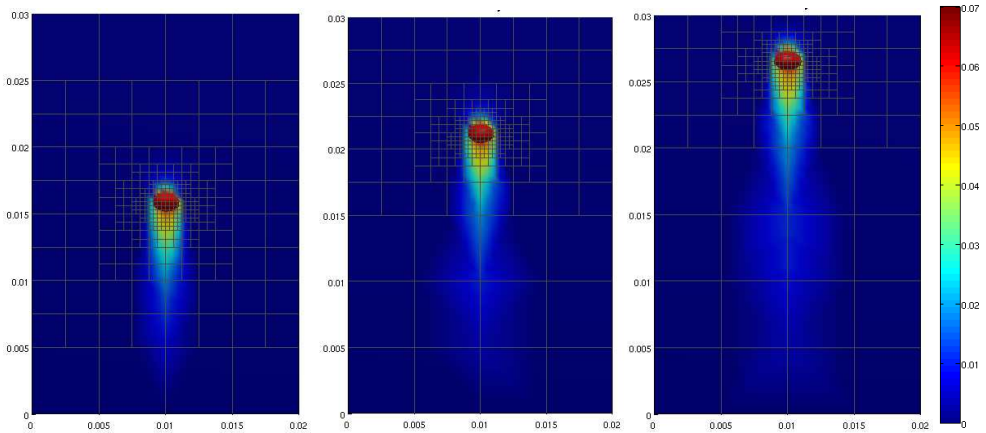
The terminal velocities of the n-butanol droplet for the various radii are computed and shown in Table 5. The numerical predictions are compared to the theoretical model predictions of Henschke [50] and the numerical predictions of Groß [49]. The same results are plotted and compared in Figure 35. It is observed that both Gross (2008) and this study yield results very close to those predicted by the model. Further, it can be seen that this study predicts terminal rise velocities which are consistently slightly higher than those predicted by Gross (2008). Figure 35(b) shows a close-up view with straight segments drawn between the data points in order to reveal the very similar trends in both numerical studies. That is, both studies exhibit a convex shape beyond $r_d = 1.5$ mm instead of a concave shape predicted by the theoretical model. Nevertheless, the theoretical prediction is sandwiched between the two numerical predictions which are very close.

8. SUMMARY AND CONCLUSIONS

The current two-fluid flow solver employs the XFEM for the enrichment of the pressure space so that the jump across the two-fluid interface (in the presence of surface



(a) $t = 0.01$ s (b) $t = 0.1$ s (c) $t = 0.2$ s

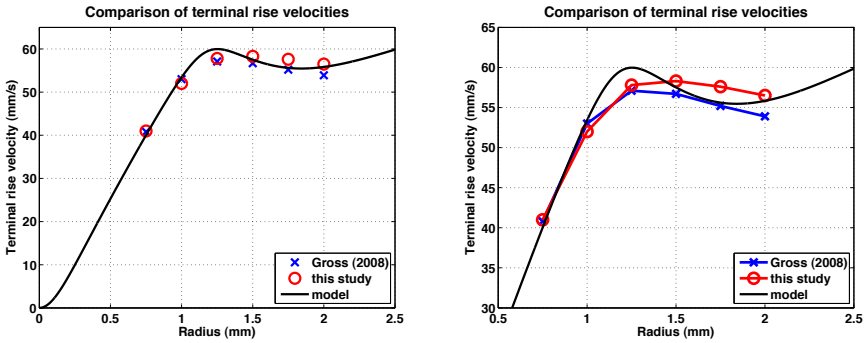


(e) $t = 0.3$ s (f) $t = 0.4$ s (g) $t = 0.5$ s

Figure 34: Evolution of the z-component of the velocity field with time for the $r = 1.0$ mm droplet on the cross-section $y = 0.01$ m.

Table 5: Comparison of terminal rise velocities.

Radius	**Terminal rise velocity** $u_r(mm/s)$		
r_d (mm)	Model	Gross (2008)	This study
0.75	40.3	40.8	41.0
1.00	53.7	53.0	52.0
1.25	60.0	57.1	57.8
1.50	57.5	56.7	58.3
1.75	55.6	55.2	57.6
2.00	55.8	53.9	56.5



(a) (b)

Figure 35: Comparison of terminal rise velocities for different droplet radii between results from (i) model prediction, (ii) Gross (2008) and (iii) this study. (b) shows a close-up view which reveals the very similar trends between (ii) and (iii).

tension) can be reproduced accurately. An important feature of the flow solver is the use of a multi-level mesh refinement realized via hanging nodes, resulting in 1-irregular non-conforming meshes. Such a refinement procedure is considerably more straightforward compared to refinements on conforming meshes. The h-adaptivity feature of the flow solver is essential to ensuring affordable computational costs especially when applied to three-dimensional simulations.

In this study, the reinitialization of the level-set is realized by solving a PDE to steady state where reinitialized level-set once again possesses the sign-distance property. This method has proven to be superior to an interpolation reinitialization as far as mass conservation is concerned. For the computation of the surface tension, different methods of computing the normal vectors are proposed. The method which leads to the best results utilizes the level-set definition of the normal vector and employs a finite difference method to first compute the gradient of the level-set at the nodes. A standard FE interpolation is then used to interpolate the gradients at points on the interface within the cut element. A drawback of this method is the assumption of a Cartesian mesh (at least in the vicinity of the interface). For general unstructured meshes, it is suggested that a higher-order representation of the level-set can be used to improve the results.

The numerical results demonstrate the ability of the flow solver to handle situations where large surface tensions (and therefore large jumps in the pressure field) exist across the two-fluid moving interface. The accuracy of the solver is demonstrated by the convergence studies in the L_2 error norms of the pressure and velocity fields in both two and three dimensions for the case of a static bubble. The accuracy of the solver is further testified by the case of the rising n-butanol droplet with good agreement between the numerical and model predictions of the terminal velocities. Overall, we have achieved the objective of developing a robust two-fluid flow solver for the simulation of two-phase incompressible flow capable of handling a variety of physical flow scenarios in both two and three dimensions.

References

[1] S. Ganesan, G. Matthies, and L. Tobiska. On spurious velocities in incompressible flow problems with interfaces. *Comp. Methods Appl. Mech. Engrg.*, 196:1193 – 1202, 2007.

[2] J.F. Gerbeau, C. le Bris, and M. Bercovier. Spurious velocities in the steady flow of an incompressible fluid subjected to external forces. *Int. J. Numer. Methods Fluids*, 25:679 – 695, 1997.

[3] A. Smolianski. *Numerical modelling of two-fluid interfacial flows*. Dissertation, University of Jyväskylä, 2001.

[4] A.-K. Tornberg. *Interface Tracking Methods with Application to Multiphase Flows*. Phd thesis, Royal Institute of Technology, 2000.

[5] A.-K. Tornberg and B. Engquist. A finite element based level-set method for multiphase flow applications. *Comput. Visual Sci.*, 3:93–101, 2000.

[6] S. Hysing. A new implicit surface tension implementation for interfacial flows. *Int. J. Numer. Methods Fluids*, 51:659 – 672, 2005.

[7] T. Belytschko, N. Moës, S. Usui, and C. Parimi. Arbitrary discontinuities in finite elements. *Internat. J. Numer. Methods Engrg.*, 50:993 – 1013, 2001.

[8] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *Internat. J. Numer. Methods Engrg.*, 46:131 – 150, 1999.

[9] J. Chessa and T. Belytschko. An extended finite element method for two-phase fluids. *ASME J. Appl. Mech.*, 70:10 – 17, 2003.

[10] J. Chessa and T. Belytschko. Arbitrary discontinuities in space-time finite elements by level-sets and X-FEM. *Internat. J. Numer. Methods Engrg.*, 61:2595 – 2614, 2004.

[11] J. Chessa and T. Belytschko. A local space-time discontinuous finite element method. *Comp. Methods Appl. Mech. Engrg.*, 195:1325 – 1343, 2006.

[12] S. Groß and A. Reusken. An extended pressure finite element space for two-phase incompressible flows with surface tension. *J. Comput. Phys.*, 224:40 – 58, 2007.

[13] T.P. Fries. The intrinsic XFEM for two-fluid flows. *Int. J. Numer. Methods Fluids*, page DOI: 10.1002/fld.1901, 2008.

[14] S. Zlotnik and P. Díez. Hierarchical x-fem for n-phase flow ($n > 2$). *Comp. Methods Appl. Mech. Engrg.*, 198:2329 – 2338, 2009.

[15] T.P. Fries, A. Byfut, A. Alizada, K.W. Cheng, and A. Schroeder. Hanging nodes and xfem. *Internat. J. Numer. Methods Engrg.*, doi: 10.1002/nme.3024, 2010.

[16] S. Osher and R.P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, Berlin, 2003.

[17] J. Chessa and T. Belytschko. An enriched finite element method and level sets for axisymmetric two-phase flow with surface tension. *Internat. J. Numer. Methods Engrg.*, 58:2041 – 2064, 2003.

[18] K. Ruschak. A method for incorporating free boundaries with surface level solvers for finite element discretisations of mixed problems. *Internat. J. Numer. Methods Engrg.*, 15:639648, 1980.

[19] G. Dziuk. An algorithm for evolutionary surfaces. *Numer. Math.*, 58:603–611, 1991.

[20] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Methods Appl. Mech. Engrg.*, 32:199 – 259, 1982.

[21] T.E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. In J.W. Hutchinson and T.Y. Wu, editors, *Advances in Applied Mechanics*, volume 28. Academic Press, New York, NY, 1992.

[22] T.E. Tezduyar, S. Mittal, S.E. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Comp. Methods Appl. Mech. Engrg.*, 95:221 – 242, 1992.

[23] T.P. Fries and A. Zilian. On time integration in the XFEM. *Internat. J. Numer. Methods Engrg.*, page DOI: 10.1002/nme.2558, 2009.

[24] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146 – 159, 1994.

[25] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley & Sons, Chichester, 2003.

[26] L. D. Landau and E. M. Lifshitz. *Course of Theoretical Physics: Volume 3, Fluid Mechanics*. Pergamon Press, New York, 1959.

[27] S. Osher. and J. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

[28] P.M. Gresho and R.L. Sani. *Incompressible Flow and the Finite Element Method*, volume 1+2. John Wiley & Sons, Chichester, 2000.

[29] K.W. Cheng. *h- and p-XFEM with application to two-phase incompressible flow*. Ph.D. Dissertation, Aachen Institute for Advanced Study in Computational Engineering Science, RWTH Aachen University, Germany, 2010.

[30] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high- order accurate essentially non-oscillatory schemes iii. *J. Comput. Phys.*, 71:231 – 303, 1987.

[31] M. Sussmann, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flows. *J. Comput. Phys.*, 114:146 – 159, 1994.

[32] B. Merriman Y.C. Chang, T.Y. Hou and S. Osher. A level set formulation of eulerian interface capturing methods for incompressible fluid flows. *J. Comput. Phys.*, 124:449 – 464, 1996.

[33] M. Sussmann and E. Fatemi. An efficient interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comput.*, 20:1165 – 1191, 1999.

[34] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155:410–438, 1999.

[35] M. Sussmann, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.*, 148, 1994.

[36] I. Babuška and J.M. Melenk. The partition of unity finite element method. Technical Report BN-1185, Institute for Physical Science and Technology, University of Maryland, 1995.

[37] N. Sukumar, N. Moës, B. Moran, and T. Belytschko. Extended finite element method for three-dimensional crack modelling. *Internat. J. Numer. Methods Engrg.*, 48:1549 – 1570, 2000.

[38] K.W. Cheng and T.P. Fries. Higher-order XFEM for arbitrary curved strong and weak discontinuities. *Internat. J. Numer. Methods Engrg.*, 82(5):564–590, 2009.

[39] R.E. Bank, A.H. Sherman, and A. Weiser. Some refinement algrorithms and data structures for regular local mesh refinement. In Stepleman et al., editor, *Transactions on Scientific Computing*, volume 1. IMACS, Amsterdam, North Holland, 1983.

[40] A.K. Gupta. A finite element for transition from a fine grid to a coarse grid. *Internat. J. Numer. Methods Engrg.*, 12:35 – 45, 1978.

[41] D.J. Morton, J.M. Tyler, and J.R. Dorroh. A new 3d finite element for adaptive h-refinement in 1-irregular meshes. *Internat. J. Numer. Methods Engrg.*, 38:3989 – 4008, 1995.

[42] I. Babuška. Error-bounds for finite element method. *Numer. Math.*, 16:322 – 333, 1971.

[43] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrange multipliers. *RAIRO Anal. Numér.*, R-2:129 – 151, 1974.

[44] F. Shakib, T.J.R. Hughes, and Z. Johan. A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations. *Comp. Methods Appl. Mech. Engrg.*, 89:141 – 219, 1991.

[45] S. Mittal. On the performance of high aspect ratio elements for incompressible flows. *Comp. Methods Appl. Mech. Engrg.*, 188:269 – 287, 2000.

[46] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.*, 100:335–354, 1992.

[47] M.P. do Carmo. *Differential geometry of curves and surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[48] T.P. Fries and T. Belytschko. The intrinsic XFEM: A method for arbitrary discontinuities without additional unknowns. *Internat. J. Numer. Methods Engrg.*, 68:1358 – 1385, 2006.

[49] S. Groß. *Numerical methods for three-dimendional incompressible two-phase flow problems*. Dissertation, RWTH Aachen University, 2008.

[50] M. Henschke. *Auslegung pulsierter Siebboden-Extraktionskolonnen. Habilitation thesis (in German)*. Dissertation, RWTH Aachen University, 2003.