

A Cartesian Method with Second-Order Pressure Resolution for Incompressible Flows with Large Density Ratios

Michel Bergmann, Lisl Weynans

▶ **To cite this version:**

Michel Bergmann, Lisl Weynans. A Cartesian Method with Second-Order Pressure Resolution for Incompressible Flows with Large Density Ratios. Fluids, 2021, 6 (11), pp.402. 10.3390/fluids6110402 . hal-03506971

HAL Id: hal-03506971
https://inria.hal.science/hal-03506971

Submitted on 3 Jan 2022

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

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

fluids

MDPI

Article

A Cartesian Method with Second-Order Pressure Resolution for Incompressible Flows with Large Density Ratios

Michel Bergmann [1,†] and Lisl Weynans [2,*,†]

[1] Memphis Team, INRIA, University Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France; michel.bergmann@inria.fr
[2] Carmen Team, INRIA, University Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France
* Correspondence: lisl.weynans@math.u-bordeaux.fr
† These authors contributed equally to this work.

Abstract: An Eulerian method to numerically solve incompressible bifluid problems with high density ratio is presented. This method can be considered as an improvement of the Ghost Fluid method, with the specificity of a sharp second-order numerical scheme for the spatial resolution of the discontinuous elliptic problem for the pressure. The Navier–Stokes equations are integrated in time with a fractional step method based on the Chorin scheme and discretized in space on a Cartesian mesh. The bifluid interface is implicitly represented using a level-set function. The advantage of this method is its simplicity to implement in a standard monofluid Navier–Stokes solver while being more accurate and conservative than other simple classical bifluid methods. The numerical tests highlight the improvements obtained with this sharp method compared to the reference standard first-order methods.

Keywords: incompressible Navier–Stokes equations; projection method; finite differences; Cartesian grid; immersed interfaces; level-set; interface unknowns

check for updates

Citation: Bergmann, M.; Weynans, L. A Cartesian Method with Second-Order Pressure Resolution for Incompressible Flows with Large Density Ratios. *Fluids* **2021**, *6*, 402. https://doi.org/10.3390/fluids6110402

Academic Editor: Sourabh V. Apte

Received: 17 September 2021
Accepted: 9 October 2021
Published: 6 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Bifluid problems are ubiquitous in nature and in many industrial applications such as combustion in engines, water waves energy converters, and jet printers to cite only a few. In such applications, the density ratio between the two fluids can be large, for instance the ratio is equal to 1000 between water and air. Accurate numerical modeling and numerical simulations of these kind of phenomena are then necessary, in particular to optimize such devices.

In this paper we are thus concerned with the numerical modeling of incompressible bifluid flows with large density ratios, like air and water, and by the accurate description of the phenomena occurring at their interface. We present a sharp Cartesian method for the simulation of incompressible flows with high density and viscosity ratios. This method is an extension of the second-order Cartesian method for elliptic problems with immersed interfaces developed in [1].

Cartesian grids are an attractive alternative to body fitted meshes. Indeed, they avoid complex mesh generation as well as mesh adaptation when unsteady interfaces are considered. Moreover, the numerical resolution of the governing equations can be simplified with an easy parallelization and the use of standard linear algebra libraries. Generally speaking, numerical schemes are easy to implement on a Cartesian mesh because a dimensional splitting is often possible; however, some numerical modeling is necessary near a complex interface that does not fit the background Cartesian grid. This is the case for fluid structure interface and moreover for bifluid interface where the properties of the flow are discontinuous. Indeed, applying naively a numerical scheme originally devised for a flow with constant or continuously varying density will lead to a non-consistent treatment of the interface. Most of the time, it will result in severe stability issues if the density ratio is

large as highlighted in [2] and references therein; therefore, as already mentioned, one has to devise specific numerical schemes at the vicinity of the interface. This region is called narrow band and is the set of numerical points that have at least one neighbor on the other side of the interface.

Conservative or non-conservative approaches can both be used to face this issue. Among the non-conservative approaches, one solution is to regularize the properties of the fluids in the vicinity of the interface, so that the density, viscosity, and their derivatives are continuous in the whole computational domain. This idea leads to the well-known "Continuous Surface Force" (CSF) method [3], where the discontinuous quantities are smoothed near the interface, and in case of a fluid with surface tension, this surface tension is taken into account as a smooth volume force instead of a surface force. This method is widely used (see for instance [4,5]) because it offers a straightforward way to implement the presence of two fluids in an already existing monofluid Navier–Stokes code. However, the exact way that the regularization should be performed is not always clear, and spurious oscillations at the bifluid interface can appear due to errors in the pressure gradient computations. Another non-conservative method introduced by Kang, Fedkiw, and Liu [6] after the CSF is the Ghost Fluid Method (GFM). It is based on a first-order method developed in [7] to solve an immersed interface elliptic problem, with a dimensional splitting making the method easy to implement. The resulting linear system is symmetric and has the same structure as the usual matrix to discretize a Poisson equation with variable coefficient on a Cartesian grid. This method has been used successfully in a lot of works, for instance [8,9]. One drawback is that the method is only first-order accurate near the interface [2] and a loss of conservativity of the momentum of each fluid near the interface can occur leading to erroneous velocities.

Non-conservative methods are often associated with a level-set representation of the interface [4] because the level-set method is itself intrinsically non-conservative at the discrete level, and convenient to use on a Cartesian grid.

The other family of methods is based on the conservative form of the Navier–Stokes equations, where mass and momentum fluxes of each fluid are explicitly computed, see for instance [2,10–13]. An explicit interface representation is necessary even if the interface do not coincide with grid points. Conservative methods are generally more stable than non-conservative methods. The price for this increased stability is an additional amount of numerical developments due to the interface reconstruction, which can be performed from information carried by Lagrangian markers or by cell quantities such as volume fractions.

Another approach has been developed recently [14,15] to deal with large density ratio. In this approach, a fully second-order method is obtained at the interface (for both velocity and pressure) with several physical boundary treatments, including velocity and traction boundary conditions.

In this paper we aim to preserve as much as possible the simplicity of the Ghost Fluid method of [6], avoiding an explicit identification of the volume fractions near the interface, while improving the accuracy and stability of the pressure computation. We thus propose a method, mainly based on the improvement of the Ghost Fluid method, where the discontinuities across the interface are taken into account in a sharp way with a second-order scheme inspired from [1]. This second-order treatment improves the conservativity of the method, as it will be proved numerically in the section devoted to numerical validations.

After having described the governing equations for the incompressible bifluid flows that we consider (Section 2), the discretization of these equations in each fluid and at the interface are presented (Sections 3 and 4). The second-order numerical resolution of the elliptic problem arising from the computation of the pressure is introduced (Section 5), and the overall is validated on several two-dimensional numerical test cases (Section 6).

2. Governing Equations

2.1. Flow Equations

We consider a rectangular domain Ω filled with two viscous incompressible fluids with different densities and viscosities. The subdomains Ω^- and Ω^+ corresponding to the two fluids are separated by an interface Γ as depicted in Figure 1.

In this work, these domains are implicitly defined with a scalar function ϕ, usually called the level-set function, see Section 2.2, that takes different values in each subdomain with a fixed value on the interface. For instance we chose $\phi = 0$ on Γ, $\phi > 0$ in Ω^+ and $\phi < 0$ in Ω^-. The unit normal to the interface is denoted $\boldsymbol{n}$ and the unit tangent vector is denoted $\boldsymbol{\eta}$. The density is denoted

$$\rho = \rho^- + H(\phi)(\rho^+ - \rho^-), \tag{1}$$

and the viscosity is denoted

$$\mu = \mu^- + H(\phi)(\mu^+ - \mu^-), \tag{2}$$

where H is the Heavyside function, i.e., $H(\phi) = 1$ if $\phi > 0$ and $H(\phi) = 0$ if $\phi < 0$. Finally, the two-dimensional velocity vector is denoted $\boldsymbol{u} = (u, v)$.

The flow is modeled in the whole domain with the incompressible Navier–Stokes equations:

$$\begin{aligned} \rho(\boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}) &= -\nabla p + \nabla \cdot \tau + \rho \boldsymbol{g} - \sigma\kappa\nabla H, \\ \nabla \cdot \boldsymbol{u} &= 0, \end{aligned}$$

with $\boldsymbol{g}$ the gravitational acceleration vector, τ the viscous stress tensor:

$$\tau = \mu(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T), \tag{3}$$

and the term $\sigma\kappa\nabla H$ accounting for the surface tension effects, with σ the surface tension itself and κ the local curvature of the interface between the fluids. This formulation of the bifluid incompressible Navier–Stokes equations contains a singular term which is not trivial to handle.

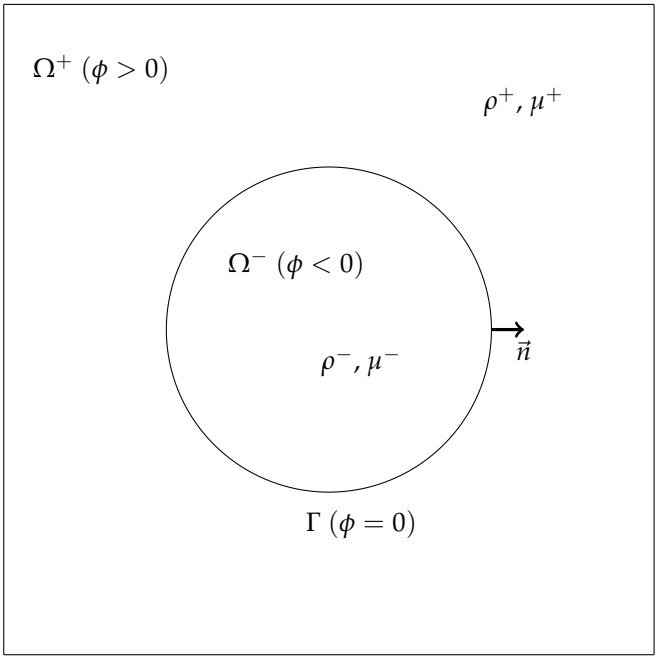


Figure 1. Sketch of the computational domain.

Alternatively, the flow can also be modeled in each subdomain with the incompressible Navier–Stokes equations:

$$\begin{aligned} \rho(\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}) &= -\nabla p + \nabla\cdot\tau + \rho\boldsymbol{g}, \\ \nabla\cdot\boldsymbol{u} &= 0. \end{aligned}$$

The above equations are completed by jump conditions at the interface Γ between the two fluids. In what follows, jumps are defined by the notation $[\psi] = \psi^+ - \psi^-$. The first jump conditions describe the balance between the normal stresses at the interface and the surface tension σ, with κ the local curvature of the interface Γ,

$$[p - 2\mu(\nabla u\cdot\boldsymbol{n}, \nabla v\cdot\boldsymbol{n})\cdot\boldsymbol{n}] = \sigma\kappa, \tag{4}$$

$$[\mu(\nabla u\cdot\boldsymbol{n}, \nabla v\cdot\boldsymbol{n})\cdot\boldsymbol{\eta} + (\nabla u\cdot\boldsymbol{\eta}, \nabla v\cdot\boldsymbol{\eta})\cdot\boldsymbol{n}] = 0. \tag{5}$$

Others jump conditions can be derived from continuity properties across the interface. For instance, for a viscous fluid, the velocity field is continuous across the interface

$$[u] = 0, \tag{6}$$

$$[v] = 0. \tag{7}$$

Since the material derivative of (6) and (7) is zero, one can write

$$0 = \frac{\partial[\boldsymbol{u}]}{\partial t} + (\boldsymbol{u}\cdot\nabla)[\boldsymbol{u}] = \left[-\frac{\nabla p}{\rho} + \frac{\nabla\cdot\tau}{\rho} + \boldsymbol{g}\right], \tag{8}$$

which leads to

$$\left[\frac{\nabla p}{\rho}\right] = \left[\frac{\nabla\cdot\tau}{\rho}\right]. \tag{9}$$

The jump condition for the pressure p can be simplified. We differentiate the jump on the velocity in the tangential direction:

$$\begin{aligned} \left[\frac{\partial u}{\partial\eta}\right] &= 0, \\ \left[\frac{\partial v}{\partial\eta}\right] &= 0. \end{aligned}$$

Moreover, because the velocity is divergent-free on each side of the interface,

$$0 = [\nabla\cdot\boldsymbol{u}] = [(\nabla u\cdot\boldsymbol{n}, \nabla v\cdot\boldsymbol{n})\cdot\boldsymbol{n} + (\nabla u\cdot\boldsymbol{\eta}, \nabla v\cdot\boldsymbol{\eta})\cdot\boldsymbol{\eta}]. \tag{10}$$

Combining the two last relationships, we obtain

$$[(\nabla u\cdot\boldsymbol{n}, \nabla v\cdot\boldsymbol{n})\cdot\boldsymbol{n}] = 0. \tag{11}$$

Consequently

$$[p] = 2[\mu](\nabla u\cdot\boldsymbol{n}, \nabla v\cdot\boldsymbol{n})\cdot\boldsymbol{n} + \sigma\kappa. \tag{12}$$

Finally, we will use Equations (9) and (12) to compute the pressure jump at the interface.

2.2. Interface Description

The interface between the two fluid subdomains is implicitly defined by a scalar function ϕ. Local geometrical information on the function ϕ is needed to obtain an accurate discretization of the interface. To this purpose we use the level set method, introduced

by Osher and Sethian [16] and described in [17–19]. A common choice for the level-set function ϕ is the signed distance function to the interface:

$$\phi(x) = \begin{cases} dist_\Gamma(x) & \text{if } x \in \Omega^+, \\ -dist_\Gamma(x) & \text{if } x \in \Omega^-, \\ 0 & \text{if } x \in \Gamma. \end{cases} \tag{13}$$

The zero isoline of ϕ thus represents implicitly the interface Γ immersed in the computational domain.

We assume that the interface is smooth enough, so that the derivatives of the level-set function in the vicinity of the interface are well-defined. A useful property of the level-set function is a straightforward computation of its normal with the formula

$$\boldsymbol{n}(x) = \frac{\nabla\phi(x)}{|\nabla\phi(x)|}. \tag{14}$$

In the same way, the curvature of the interface can be computed with the formula

$$\kappa = \nabla \cdot \boldsymbol{n}. \tag{15}$$

We impose a curvature threshold $1/h$, with h the grid spacing, corresponding to the minimum size of a bubble for a given spatial discretization.

For a moving interface as it is usually the case for bifluid problems, the flow density and viscosity are updated with ϕ tracking the interface thanks to the transport equation

$$\phi_t + \hat{\boldsymbol{u}} \cdot \nabla\phi = 0, \tag{16}$$

where the velocity fields $\hat{\boldsymbol{u}}$ coincides with the flow velocity field $\boldsymbol{u}$ on the interface Γ. Different choices for the value of $\hat{\boldsymbol{u}}$ in Ω^+ and Ω^- can be a priori used. A natural choice we have considered is $\hat{\boldsymbol{u}} = \boldsymbol{u}$ in the whole domain:

$$\phi_t + \boldsymbol{u} \cdot \nabla\phi = 0. \tag{17}$$

Another possible choice is the extension velocity introduced in [20].

3. Navier–Stokes Monofluid Solver and Numerical Method for Interface Evolution

The computational domain is discretized on a two-dimensional uniform Cartesian grid with a grid spacing $\Delta x = \Delta y = h$. However, the following approach stands for nonuniform Cartesian meshes. The points on the Cartesian grid are defined as $M_{i,j} = (x_i, y_j)$. Similarly, we denote by u_{ij} the approximation of u at the point (x_i, y_j). In what follows, all the unknowns are collocated in space on Cartesian meshes. An odd–even coupling can sometimes be observed when one uses collocated unknowns on a Cartesian grid. This problem can be fixed using some corrections such as [21,22]. However, in all the applications considered in this paper we have not observed any odd-even coupling.

In this section we present the numerical solver devised for monofluid incompressible Navier–Stokes equations that is used in each fluid when the interface is not taken into account. We also provide information about the numerical methods used to compute the evolution of the level-set function.

3.1. Flow Computation

We use a classical projection method [23,24] to solve the Navier–Stokes equations in each fluid outside the narrow band. In what follows, a non-incremental projection method is used, i.e., the guess value for the pressure in the prediction step is zero.

We thus compute successively:

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} = -(\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n + \frac{1}{\rho}(\nabla \cdot \tau)^n + \boldsymbol{g} \quad \text{(prediction step)}, \tag{18}$$

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^*}{\Delta t} = -\frac{\nabla p}{\rho} \quad \text{(correction step)}. \tag{19}$$

The convective terms are computed with a fifth-order WENO scheme, and the viscous terms with an explicit second-order centered finite-difference scheme. The time integration is performed with a first-order explicit Euler scheme, which is consistant with the use of a non-incremental version of the projection scheme.

The pressure appearing in Equation (19) is computed through the resolution of a Poisson equation in order to enforce the divergence-free condition. At each point in one subdomain, the following relationship is satisfied:

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p\right) = \frac{\nabla \cdot \boldsymbol{u}^*}{\Delta t}. \tag{20}$$

We will provide additional details about the jump conditions that have to be satisfied across the interface for this problem in Section 4. On the exterior boundary of the domain Ω, Neumann boundary conditions are satisfied:

$$\frac{\nabla p}{\rho} = \frac{\boldsymbol{u}^b - \boldsymbol{u}^*}{\Delta t}, \tag{21}$$

where $\boldsymbol{u}^b$ is the value of the velocity to be imposed on the external boundaries.

We compute at each iteration an adaptive time step taking into account the restrictions due to convection, viscosity, and surface tension. The convective time step restriction is given by

$$\Delta t\left(\frac{|u|_{\max}}{\Delta x} + \frac{|v|_{\max}}{\Delta y}\right) \leq 1. \tag{22}$$

with $|u|_{\max}$ and $|v|_{\max}$ the maximum magnitudes of the horizontal and vertical velocities. The viscous time step restriction is given by

$$\Delta t\left(\max\left(\frac{\mu^-}{\rho^-}, \frac{\mu^+}{\rho^+}\right)\left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}\right)\right) \leq 1 \tag{23}$$

We also apply a time step restriction associated with the surface tension evaluated only in the narrow band. This time step restriction is similar to the one in [6] and in [8], and is in this context usually the most restrictive:

$$\Delta t\sqrt{\frac{\sigma|\kappa|}{\min(\rho^+, \rho^-)\min(\Delta x^2, \Delta y^2)}} \leq 1. \tag{24}$$

Finally, at each time step, the overall algorithm is the following:

- Prediction: evaluate convective and diffusive fluxes and compute $\boldsymbol{u}^*$,
- Interface evolution: convect the level-set with velocity $\boldsymbol{u}$ and re-initialize if necessary,
- Construction and resolution of the linear system for the pressure,
- Correction step: update velocity with pressure gradient.

3.2. Numerical Method for the Level-Set Evolution

The computation of the level-set function should be performed very accurately when one deals with moving interfaces. Indeed, as the level-set method is not intrinsically conservative, a lack of accuracy in the computation of the level set evolution results often in a substantial loss of mass for one of the fluids. It can also increase the problem of transfer

of momentum between both fluids and generate spurious velocity oscillations. Moreover, if one wants to compute the curvature of the interface from (15), the level-set function needs to be accurate enough (at least third-order) so that the finite difference formulas used to discretize (15) are consistent.

Unfortunately, the property of the signed distance function is usually lost when the interface evolves with the flow velocity. The norm of the level set gradient can be far from unity. These gradients variations of the level-set are harmful to the accuracy of the numerical evaluation of the normal to the interface and the curvature. To circumvent the problem, Sussman et al. [4] introduced a reinitialization algorithm to recover the signed distance function through the resolution of the eikonal equation

$$|\nabla \phi| = 1. \tag{25}$$

Several methods have been developed over the years to perform this reinitialization step, either by using a relaxation method and searching a stationary solution to a time dependent Hamilton–Jacobi equation [4] or using a Gauss–Seidel-based method as in fast marching methods [25,26] or fast sweeping methods [27].

The usual level set strategy for an evolving interface is thus the following:

- Use a transport equation to update ϕ.
- From time to time, reinitialize ϕ with the signed distance function.

One of the most widespread option in the literature is to combine a fifth-order WENO scheme [28] with a RK3 scheme, for the transport and for the reinitialization through a relaxation equation. It provides a high-order yet stable resolution; however, although the reinitialization procedure, performed with such a numerical scheme, may improve mass conservation, it also introduces some error by slightly moving the interface as shown in [29]. For this reason, Russo and Smereka [29] introduced a subcell fix taking into account the interface location in the reinitialization procedure. This technique was extended to a higher order accuracy in [30] through the use of third-order ENO schemes near the interface.

Usually, this reinitialization steps are performed uniformly every each n iterations. A recent study [31] proposes a strategy to sample the reinitialization steps based on interface deformation criteria. This should also be coupled with the high-order decentered reinitialization scheme of [30] near the interface.

In what follows, we use the classical option of the fifth-order WENO scheme for the spatial discretization. Since it is the most commonly used technique in the literature, it will allow us to distinguish the effects of the new scheme for the pressure computation from the effects of the reinitialization technique.

4. Navier–Stokes Solver near the Interface

4.1. Notations

Let us introduce some definitions and notations. A grid point is defined to be irregular if at least one of its neighbors is on the other side of the interface, i.e., if the sign of ϕ changes between this point and at least one of its neighbors, see Figure 2. The set of irregular points is called the narrow band. All the other points are called regular grid points.

We define the interface point $I_{i,j,E} = (\tilde{x}_{i,j,E}, y_j)$ as the intersection of the interface Γ and the *East* segment $[M_{ij}M_{i+1j}]$, if it exists. Similarly, the interface points $I_{i,j,W} = (\tilde{x}_{i,j,W}, y_j)$, $I_{i,j,N} = (x_i, \tilde{y}_{i,j,N})$ and $I_{i,j,S} = (x_i, \tilde{y}_{i,j,S})$ are defined as the intersection of the interface and the *West* $[M_{i-1j}M_{ij}]$, *North* $[M_{ij}M_{ij+1}]$ and *South* $[M_{ij-1}M_{ij}]$ segments, respectively. With this notation the same interface point can be described in two different ways

$$I_{i,j,S} = I_{i,j-1,N} \text{ or } I_{i,j,E} = I_{i+1,j,W}. \tag{26}$$

The set of interface points is denoted Γ_h, see Figure 2 for an illustration.

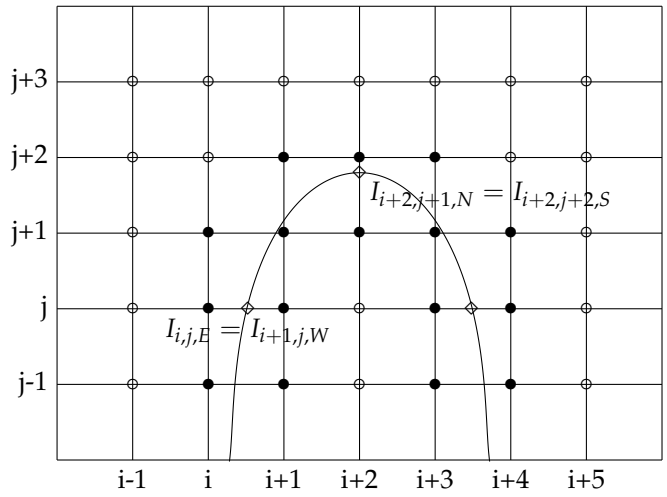


Figure 2. Example of geometrical configuration, with regular grid points (white circles), irregular points (black circles), and interface points (diamonds) with the two possible notations.

4.2. Modeling Choices for the Discontinuities across the Interface

In this study, the values of the viscosity and the densities are discontinuous across the interface; therefore, if the numerical scheme for solving incompressible Navier–Stokes equations described previously in Section 3.1 was applied on the irregular grid points of the narrow band, the approximations of the following terms would not be consistent:

- Prediction step: viscous terms.
- Correction step: divergence of the predicted velocity, elliptic operator, and gradient of the pressure.

This lack of consistency could eventually lead to stability problems.

The computation of the convective terms is not mentioned in the above enumeration because it is performed with a fifth-order WENO scheme, which automatically provides spatial adaptivity. Therefore, we assume that the gradients computed with the WENO scheme are decentered near the interface, and consequently, consistent. Moreover, the level-set function is classically evolved with such a scheme, because it is crucial to have a good accuracy in the computation of the interface evolution; thus, it seems coherent to have the same numerical scheme for the convection of the interface and the convection of the fluids.

In this work, we use two different strategies to handle the lack of consistency near the interface, one for the viscous terms and another for the pressure computation. For the viscous term in the prediction step, we follow a continuous approach and regularize the quantities used for the computation of the viscous terms. It has been proven in [32,33] that this continuous approach provides correct accuracy for high Reynolds number flows. It has also been used successfully in [2,3]. A sharp approach for the viscous terms could probably improve the accuracy of the simulations; however, the complexity of the computations would be increased due to the treatment of the jump conditions for the viscous terms (5) implying derivatives of the velocity components in both normal and tangential directions. Moreover, if one needs to use an implicit treatment of the viscous terms, such a sharp treatment would become more complex to handle.

The discretization of the prediction step becomes:

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} = -(\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n + \frac{1}{\tilde{\rho}}\Big(\nabla \cdot \tilde{\mu}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)\Big)^n + \boldsymbol{g}. \tag{27}$$

In practice the viscosity and the inverse of the density were regularized by a discrete convolution [2]:

$$\begin{aligned} 16\,\tilde{\mu}_{i,j} &= 4\,\mu_{i,j} + 2\,\mu_{i+1,j} + 2\,\mu_{i-1,j} + 2\,\mu_{i,j+1} + 2\,\mu_{i,j-1} \\ &\quad + \mu_{i+1,j+1} + \mu_{i+1,j-1} + \mu_{i-1,j+1} + \mu_{i-1,j-1}, \\ \frac{16}{\tilde{\rho}_{i,j}} &= \frac{4}{\rho_{i,j}} + \frac{2}{\rho_{i+1,j}} + \frac{2}{\rho_{i-1,j}} + \frac{2}{\rho_{i,j+1}} + \frac{2}{\rho_{i,j-1}} \\ &\quad + \frac{1}{\rho_{i+1,j+1}} + \frac{1}{\rho_{i+1,j-1}} + \frac{1}{\rho_{i-1,j+1}} + \frac{1}{\rho_{i-1,j-1}}. \end{aligned}$$

Then, we discretize the viscous terms with a classical second-order centered scheme as in Section 3.1.

In the correction step, according to the jump conditions (4)–(12) presented in Section 2, the pressure satisfies an elliptic problem with discontinuous values of the solution and its derivative across the interface:

$$\begin{aligned} \nabla \cdot \left(\frac{1}{\rho} \nabla p \right) &= \frac{\nabla \cdot \boldsymbol{u}^*}{\Delta t} \text{ in } \Omega^+ \cup \Omega^-, \\ [p] &= \sigma\,\kappa + 2[\mu]\,(u_n, v_n) \cdot \boldsymbol{n} \text{ on } \Gamma, \\ \left[\frac{\nabla p}{\rho} \right] &= \left[\frac{\nabla \cdot \tau}{\rho} \right] \text{ on } \Gamma. \end{aligned}$$

Because the viscous terms in the prediction step are handled with a regularization approach, we have $[\mu] = 0$ and $\left[\frac{\nabla \cdot \tau}{\rho}\right] = 0$. Therefore, the pressure computed for the correction step satisfies rather:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = \frac{\nabla \cdot \boldsymbol{u}^*}{\Delta t} \text{ in } \Omega^+ \cup \Omega^-, \tag{28}$$

$$[p] = \sigma\,\kappa \text{ on } \Gamma, \tag{29}$$

$$\left[\frac{\nabla p}{\rho} \right] = 0 \text{ on } \Gamma. \tag{30}$$

The details of the resolution of this elliptic problem will be provided in Section 5. Let us remark however that as the values of p are discontinuous across the interface, two values of p will be created at each point of the interface considered in the numerical scheme.

4.3. Gradient and Divergence for Correction Step

The predicted velocity $\boldsymbol{u}^*$ obtained after the prediction step (18) is defined only on grid points. We need to compute the divergence of this predicted velocity in order to solve the elliptic Equation (20). However, since the two fluids have different properties across the interface and the derivatives of the velocity are not necessarily continuous, we need to use a decentered stencil on each side of the interface. Consequently, we have to compute two values for $\boldsymbol{u}^*$ on each interface point, one for each side of the interface. In practice, as jump conditions for $\boldsymbol{u}^*$ are not available, we perform simply linear extrapolations from the grid values on the interface points. Then, to compute the divergence of $\boldsymbol{u}^*$ on an irregular grid point $M_{i,j}$, we use a standard five point stencil, see Figure 3. Formally this is equivalent to a standard first-order decentered scheme.

More precisely, we denote $\boldsymbol{u}^*_S$ the value of the predicted velocity $\boldsymbol{u}^*$ on the nearest point to $M_{i,j}$ in the south direction (possibly an interface point), with coordinates (x_S, y_S). Similarly, we define $\boldsymbol{u}^*_N$, $\boldsymbol{u}^*_W$ and $\boldsymbol{u}^*_E$ and the associated coordinates (x_N, y_N), (x_W, y_W) and (x_E, y_E). The discretization reads

$$\left(\nabla \cdot \boldsymbol{u}^* \right)_{i,j} = \frac{u^*_E - u^*_W}{x_E - x_W} + \frac{v^*_N - v^*_S}{y_N - y_S}.$$

Similarly, in order to keep a consistent discretization, the gradient of the pressure p appearing in Equation (19) is also computed with an adapted decentered stencil near the interface, see Figure 3. More precisely, with the same notations S, N, W and E, as before, the discretization reads

$$(\nabla p)_{i,j} = \begin{pmatrix} \dfrac{p_E - p_W}{x_E - x_W} \\ \dfrac{p_N - p_S}{y_N - y_S} \end{pmatrix}. \tag{31}$$

If one of the discretization point is an interface point, we consider the value of the pressure on this point corresponding to the same subdomain than point $M_{i,j}$. Indeed we recall that, since the pressure is discontinuous across the interface, two pressure unknowns (one for each subdomain) are computed at an interface point. If no interface point is involved, the numerical scheme reduces to the classical second-order central finite differences scheme.

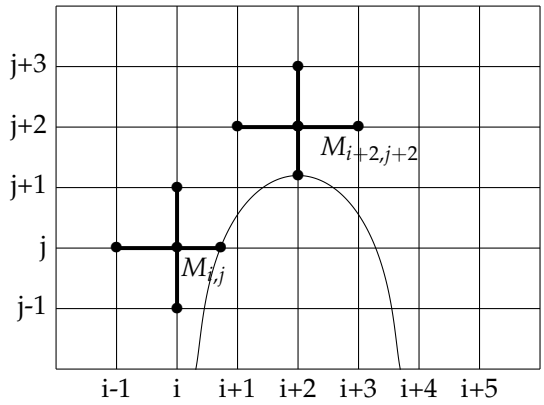


Figure 3. Example of geometrical configuration, with points involved in the discretization of the divergence of the predicted velocity and pressure gradient at grid points $M_{i,j}$ and $M_{i+2,j+2}$ in black.

5. Numerical Resolution of Elliptic Problems with Immersed Interfaces

The elliptic problem with discontinuous values across an interface (28)–(30) is solved with the second-order method developed in [1]. The interface points are used to impose the jump conditions across the interface in the numerical scheme. Since the pressure is discontinuous across the interface, two unknowns are created, one for each side of the interface. The accuracy of this method is based on the use of unknowns located at the interface. The size of this linear system is thus augmented with two unknowns for each interface point. These interface unknowns are used to discretize the flux jump conditions and the elliptic operator accurately enough to get a second-order convergence in maximum norm. Actually, to this purpose, near the interface the elliptic operator needs to be discretized with a first-order truncation error, and the fluxes with a second-order truncation error. For a visual explanation of the discretization we refer to Figure 4. The advantage of using this method, compared to the reference work of [6] is that the jump conditions in the correction step are solved with second-order accuracy instead of first-order. The drawback is that the linear system is not symmetric anymore and it is solved with the preconditioned GMRES method.

5.1. Discrete Elliptic Operator

We use a standard five-point stencil including the grid point $M_{i,j}$ and its nearest neighbors in each direction: interface or grid points. More precisely, we denote p_S the value of the solution on the nearest point in the south direction, with coordinates (x_S, y_S).

Similarly, we define p_N, p_W, and p_E and the associated coordinates (x_N, y_N), (x_W, y_W), and (x_E, y_E). Since the density is piecewise constant, the discretization reads

$$\left(\nabla.(\frac{1}{\rho^{\pm}}\nabla p)\right)_{i,j} = \frac{1}{\rho^{\pm}}\Delta p = \frac{1}{\rho^{\pm}}\frac{\frac{p_E - p_{ij}}{x_E - x_i} - \frac{p_{ij} - p_W}{x_i - x_W}}{\frac{x_E - x_W}{2}} + \frac{1}{\rho^{\pm}}\frac{\frac{p_N - p_{ij}}{y_N - y_j} - \frac{p_{ij} - p_S}{y_j - y_S}}{\frac{y_N - y_S}{2}}, \quad (32)$$

where $\rho^{\pm}$ stands for ρ^+ or ρ^-.

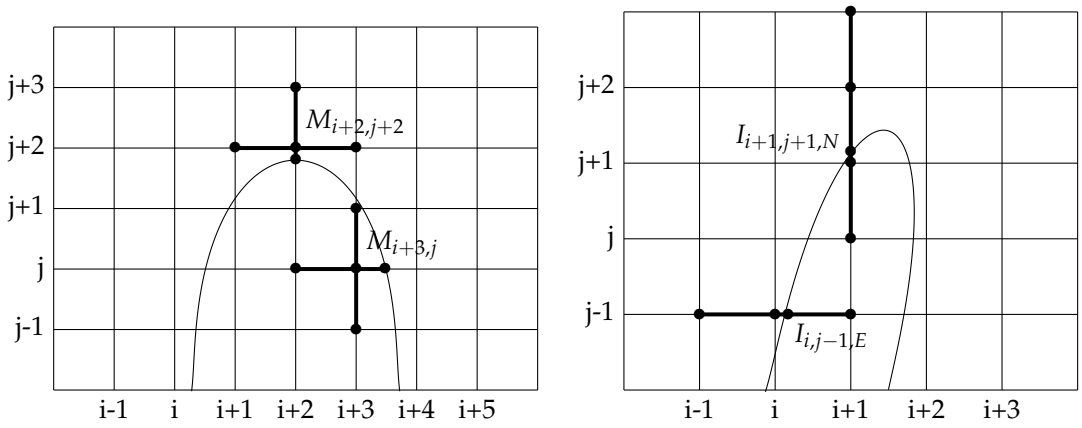


Figure 4. Left: points involved in the discretization of the elliptic operator at grid nodes $M_{i+2,j+2}$ and $M_{i+3,j}$ in black, right: Example of stencils for the discretization of jump conditions. Points involved in the discretization of the x-derivative of the pressure at interface point $I_{i,j-1,E}$ and of the y-derivative of the pressure at interface point $I_{i+1,j+1,N}$ in black. For $I_{i,j-1,E}$ both derivatives are expressed with second-order accuracy while for $I_{i+1,j+1,N}$ the left derivative is expressed with second-order and the right derivative with first-order accuracy.

5.2. Discrete Flux Transmission Conditions

As written before, at each interface point we create two additional unknowns, called interface unknowns. We denote them by $p^{\pm}_{i,j,\gamma}$ with $\gamma = E, W, N$ or S. The interface unknowns carry the values of pressure on each side of the interface.

Contrarily to [1], we do not have a jump condition on the normal derivative, but on the whole gradient, as expressed in formula (30). To obtain the same number of equations and unknowns we have to chose in which direction we want to project this gradient equality. As we use a Cartesian grid, it is easier to discretize the $x-$ and $y-$derivatives than derivatives in other directions. Consequently, we discretize the following jump conditions at each interface point $I_{i,j,\gamma}$, with $\gamma = N, S, W, E$.

$$p^+_{i,j,\gamma} - p^-_{i,j,\gamma} = \sigma\,\kappa_{i,j,\gamma}, \quad (33)$$

$$\frac{1}{\rho^+}(\partial_x p^+)_{i,j,\gamma} - \frac{1}{\rho^-}(\partial_x p^-)_{i,j,\gamma} = 0 \text{ if } \gamma = E, W. \quad (34)$$

$$\frac{1}{\rho^+}(\partial_y p^+)_{i,j,\gamma} - \frac{1}{\rho^-}(\partial_y p^-)_{i,j,\gamma} = 0 \text{ if } \gamma = N, S. \quad (35)$$

We want the truncation error of the discretization of flux equality (34) and (35) to be second-order accurate in order to solve the problem with a second-order accuracy. A possible configuration of the interface is illustrated in Figure 4. In the x-direction, it is straightforward to compute a second-order approximation of the x-derivative with three a priori nonequidistant points. For example we approximate the flux on the left side of

interface point $I_{i,j,E}$, if it exists, with the values of p on the points $M_{i-1,j}$, $M_{i,j}$ and $I_{i,j,E}$ with the formula:

$$(\partial_x p^{\pm})_{i,j,E} \approx \frac{(p_{i-1,j} - p^{\pm}_{i,j,E})(x_i - \tilde{x}_{i,j,E})}{\Delta x(x_{i-1} - \tilde{x}_{i,j,E})} - \frac{(p_{i,j} - p^{\pm}_{i,,Ej})(x_{i-1} - \tilde{x}_{i,j,E})}{\Delta x(x_i - \tilde{x}_{i,j,E})}. \tag{36}$$

The right x-derivative is approximated in the same way.

$$(\partial_x p^{\pm})_{i,j,E} \approx -\frac{(p_{i+2,j} - p^{\pm}_{i,j,E})(x_{i+1} - \tilde{x}_{i,j,E})}{\Delta x(x_{i+2} - \tilde{x}_{i,j,E})} + \frac{(p_{i+1,j} - p^{\pm}_{i,,Ej})(x_{i+2} - \tilde{x}_{i,j,E})}{\Delta x(x_{i+1} - \tilde{x}_{i,j,E})}. \tag{37}$$

The same discretization holds for the y-derivative. The formulas (36) and (37) are consistent if both grid points involved in the formula, for instance $M_{i-1,j}$ and $M_{i,j}$, belong to the same domain. If on one side of the interface the two closest grid points aligned with the intersection point do not belong to the same subdomain, then the second-order discretization is not possible anymore. In this case, we use a first-order discretization involving only two points: the interface point and the closest grid point on the same side of the interface. Such a case is illustrated on Figure 4. In fact, this first-order discretization is equivalent to the Ghost Fluid method [6].

Let us notice that, because we use a dimensional splitting for the jump conditions across the interface, it is quite straightforward to eliminate the interface unknowns from the linear system. We simply inject expressions (36) and (37) in the jump condition (34), and use the resulting equality to express $p^{\pm}_{i,j,E}$ as a function of $p_{i-1,j}$, $p_{i,j}$, $p_{i+1,j}$ and $p_{i+2,j}$. This expression for $p^{\pm}_{i,j,E}$ can then be used in the discretization of the elliptic operator (32).

The local curvature $\kappa_{i,j,\gamma}$ at the interface point $I_{i,j,\gamma}$ is computed in the following way. We first compute on all irregular grid points the value

$$\kappa = \frac{\phi_x^2\phi_{yy} + \phi_y^2\phi_{xx} - 2\phi_x\phi_y\phi_{xy}}{(\phi_x^2 + \phi_y^2)^{3/2}} \tag{38}$$

with centered second-order finite-difference formulas. Then we perform a one-dimensional linear interpolation of these values on the interface points.

6. Numerical Results and Validations

This section is devoted to the numerical simulations and validations.

After having studied the well-balanced behavior of the proposed approach, i.e., the capacity of the method to preserve equilibria state for a bubble, in Section 6.1, we perform a quantitative validation for the well-known dam break problem from Section 6.2. We then present other numerical simulations for the rising of bubbles with different sizes that can qualitatively been compared to some reference results in Section 6.3.

6.1. Equilibria Preservation for a Bubble: the Parasitic Oscillations

This first test case aims to assess the influence of the interface curvature error on the stability of the numerical scheme. A bubble is located at the center of the computational domain in Sections 6.1.1 and 6.1.2. Due to the Laplace law and the concavity of the interface, the pressure inside the bubble is larger than the pressure outside. If the curvature of the interface is computed numerically, the errors due to the numerical approximation in the right-hand side of Equation (33) cause small errors in the resolution of the pressure equation and the system is thus not well balanced. These errors create artificial values of the velocity near the interface which should theoretically be zero. These artificial velocities are often called parasitic currents. The amplitude of the parasitic currents is an indication of the stability and the accuracy of the numerical method, and especially of the pressure computational step. Indeed, they are the only source of numerical errors.

In what follows several comparisons with other references methods, the Ghost Fluid and the CSF methods in Section 6.1.1 and the Volume of Fluid method in Section 6.1.2, are presented on slightly different test cases.

6.1.1. Comparison with the Ghost Fluid and the CSF Methods

We use the same parameters as in [8], where a Ghost Fluid and a CSF method were implemented. The amplitude of the parasitic currents, compared to the results in [8], are reported in Table 1. The L^∞ and L^2 norms are computed over the whole domain Ω. The initial configuration is described in Figure 5. As it can be observed, the amplitude of the parasitic currents generated by our method is several orders of magnitude smaller than those of the CSF method, and significantly lower than those of the Ghost Fluid method when the grid is refined.

$$\begin{cases} L = 2 \text{ cm}, \\ R = 1 \text{ cm}, \\ \rho_{int} = 1000 \text{ kg} \cdot \text{m}^{-3}, \\ \mu_{int} = 10^{-3} \text{ Pa} \cdot \text{s}, \\ \rho_{ext} = 1 \text{ kg} \cdot \text{m}^{-3}, \\ \mu_{ext} = 10^{-5} \text{ Pa} \cdot \text{s}, \\ \sigma = 0.1 \text{ N} \cdot \text{m}^{-1} \end{cases} \tag{39}$$

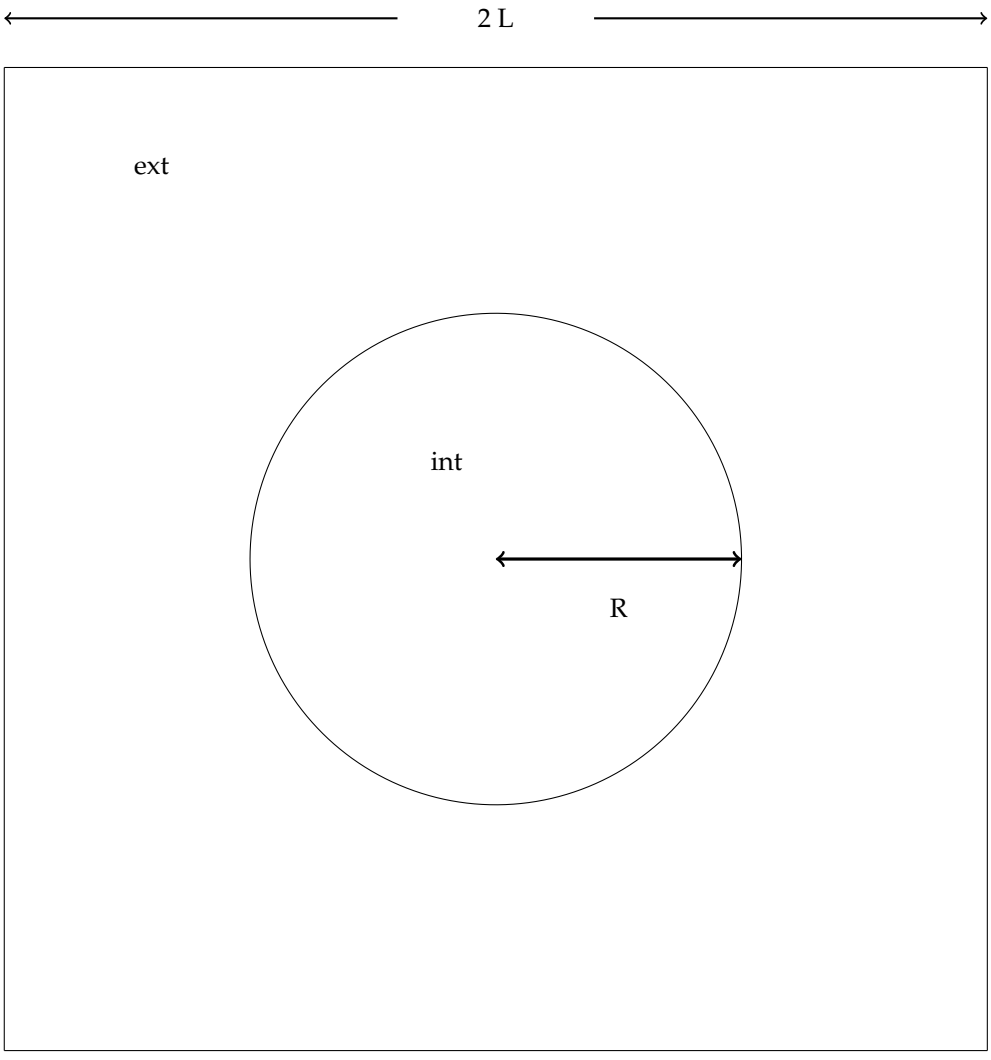


Figure 5. Test case of the static bubble with parasitic oscillations.

Table 1. Comparison between the new method and the numerical results obtained in [8] for the Ghost Fluid method and the CSF method for parasitic oscillations, at time $t = 1$.

	Ghost Fluid Method		CSF		New Method	
N	L^∞ **Error**	L^2 **Error**	L^∞ **Error**	L^2 **Error**	L^∞ **Error**	L^2 **Error**
16	8.08×10^{-3}	1.88×10^{-3}	3.55×10^{-2}	1.94×10^{-2}	5.21×10^{-3}	7.31×10^{-5}
32	3.42×10^{-4}	7.50×10^{-5}	3.12×10^{-2}	1.18×10^{-2}	9.26×10^{-5}	1.42×10^{-6}
64	5.13×10^{-5}	7.97×10^{-6}	2.12×10^{-2}	5.44×10^{-3}	1.36×10^{-5}	1.47×10^{-7}
128	2.79×10^{-5}	4.74×10^{-6}	6.44×10^{-3}	1.38×10^{-3}	2.22×10^{-6}	1.92×10^{-8}

6.1.2. Comparison with a Volume of Fluid Method

We now compare the behavior of our method to the Volume of Fluid method developed in [12]. The density and viscosity ratio are both chosen to be one for this test case. The coefficient σ is chosen so as to obtain an Ohnesorge number $Oh = \dfrac{\mu}{\sqrt{\sigma \rho D}}$ satisfying $Oh^2 = \dfrac{1}{12000}$. The maximum velocity is computed for varying grids at non-dimensional time $t^* = \dfrac{t}{T} = 250$, with $T = \dfrac{D\mu}{\sigma}$.

$$\begin{cases} L = 1.25 \text{ m}, \\ R = 1 \text{ m}, \\ \rho_{int} = 1 \text{ kg} \cdot \text{m}^{-3}, \\ \mu_{int} = 10^{-3} \text{ Pa} \cdot \text{s}, \\ \rho_{ext} = 1 \text{ kg} \cdot \text{m}^{-3}, \\ \mu_{ext} = 10^{-3} \text{ Pa} \cdot \text{s}, \\ \sigma = 0.00012 \text{ N} \cdot \text{m}^{-1} \end{cases} \tag{40}$$

A comparison between our method and the Volume of Fluid method is presented in Table 2. The new method provides a better accuracy than the Volume of Fluid method for the coarsest grid. As expected, the Volume of Fluid method outperforms our new approach for finer meshes due to more sophisticated schemes near the interface. Nonetheless, Table 2 show a second-order accuracy for our new method.

Table 2. Numerical results for parasitic oscillations at non-dimensional time $t^* = 250$ for [12] and our method.

Δx	Error L^∞ for [12]	Error L^∞ for Our Method
2.5/16	7.3×10^{-4}	7.48×10^{-5}
2.5/32	4.5×10^{-6}	4.7×10^{-6}
2.5/64	5.5×10^{-8}	1.26×10^{-6}

6.2. Collapse of a Water Column: the Dam Break Problem

This test case is studied in [2,34], and based on experiments conducted in [35]. The initial configuration is a water column at rest in air. The initial height and width of the column are both 5.715 cm. The domain size is 40 cm×10 cm. The physical constants are the same than for the rising bubble (Section 6.1.1). For more details, we refer the reader to [2]. We present in Figure 6 the interface evolution at non-dimensional times $T = t\sqrt{g/H} = 0, 1, 2, 3, 4$, with H the initial height of the water column. The computations are performed with 256×64 points.

Figure 7 present the temporal evolution of the water front, compared to the experimental results [35], to results with the Ghost Fluid method used for pressure resolution, and to the conservative method of Raessi and Pitsch [2]. We observe that the front propagation is in agreement with the experimental results and the results of the conservative method [2]. This means that, though the method is not strictly conservative, the numerical errors due to

momentum transfer across the interface are not large enough to slow down the propagation of the front. It is not the case for instance for the Ghost Fluid method, as it can be noticed in Figure 7 and has been reported in [2].

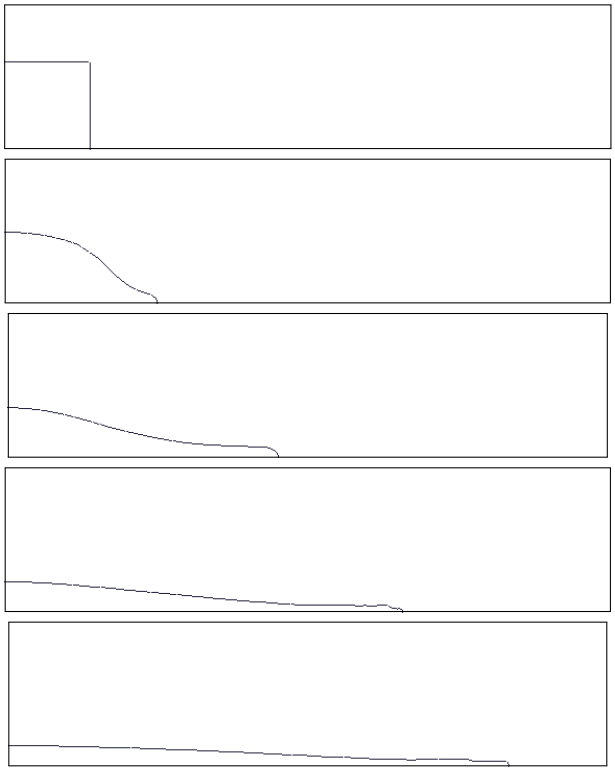


Figure 6. Evolution of the interface for the dam break problem at non-dimensional times $T = t\sqrt{g/H} = 0, 1, 2, 3, 4$.

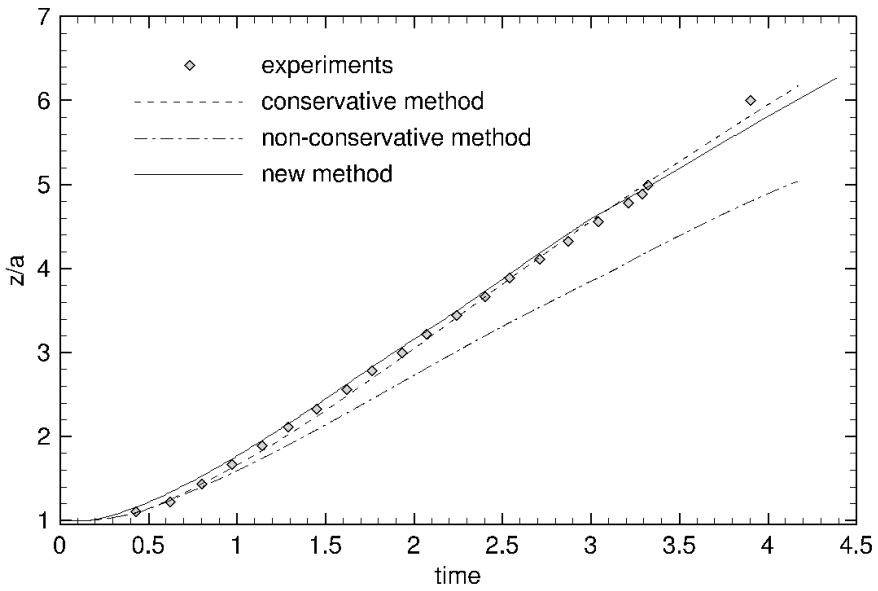


Figure 7. Evolution of the front of propagation: comparison between experimental data and several numerical methods: the Ghost Fluid method (non-conservative method), the conservative method of Raessi and Pitsch and our new method, The dimensionless location of the front $\frac{z}{a}$ is plotted as a function of the dimensionless time $T = t\sqrt{g/H}$.

6.3. Rising of Air Bubble in Water

We study the evolution of fluid bubbles rising in an heavier fluid, and compare our results to several methods in the literature. The initial configuration is described in Figure 8.

6.3.1. Comparison with the Ghost Fluid Method

We consider air bubbles rising in water, as in the test case proposed for the Ghost Fluid method in [6]. The value of the physical parameters are

$$\begin{cases} R = 1/300 \text{ m (small bubble)} \\ R = 1/3 \text{ m (large bubble)} \\ \rho_{water} = 1000 \text{ kg/m}^3, \\ \mu_{water} = 1.137 \times 10^{-3} \text{ kg/ms}, \\ \rho_{air} = 1.226 \text{ kg/m}^3, \\ \mu_{air} = 1.78 \times 10^{-5} \text{ kg/ms}, \\ \sigma = 0.0728 \text{ kg/s}^2, \\ g = -9.8 \text{ m/s}^2. \end{cases} \tag{41}$$

We consider two cases: a small bubble with $R = 1/300$ m and a large one $R = 1/3$ m. In the first case, the surface tension plays an important role in the evolution of the interface because of the high bubble curvature. In the second case, the surface tension has less influence, and larger deformations occur. The interface of the small bubble and the vorticity values are plotted at times t= 0, 0.02, 0.035, 0.05 in Figure 9. The interface of the large bubble and the vorticity values are plotted at times t= 0, 0.2, 0.35, 0.5 in Figure 10. Our numerical results are in good agreement with [6].

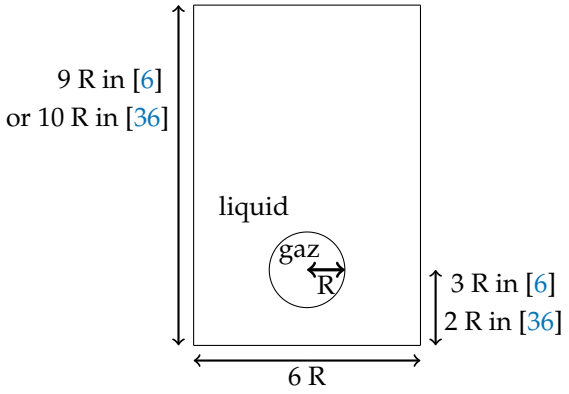


Figure 8. Initial fluid domain for the test case of the rising bubble in water in the References [6,36].

6.3.2. Comparison with SPH and the Level-Set Method

This test case is taken from [36], and inspired from a test case presented in [4]. It gives us the opportunity to compare our method to another class of methods, based on the SPH formulation. The initial configuration is described on Figure 8. The values of the physical parameters are

$$\begin{cases} R = 0.025 \text{ m}, \\ \rho_{water} = 1000 \text{ kg/m}^3, \\ \mu_{water} = 1.137 \times 10^{-3} \text{ kg/ms}, \\ \rho_{air} = 1.226 \text{ kg/m}^3, \\ \mu_{air} = 1.78 \times 10^{-5} \text{ kg/ms}, \\ \sigma = 0.0728 \text{ kg/s}^2, \\ g = -9.8 \text{m/s}^2. \end{cases} \tag{42}$$

The evolution of the interface and the vorticity values are plotted on Figure 11, for 120 $\times$ 200 grid points. We observe that the interface deforms in a way similar to the results in [36].

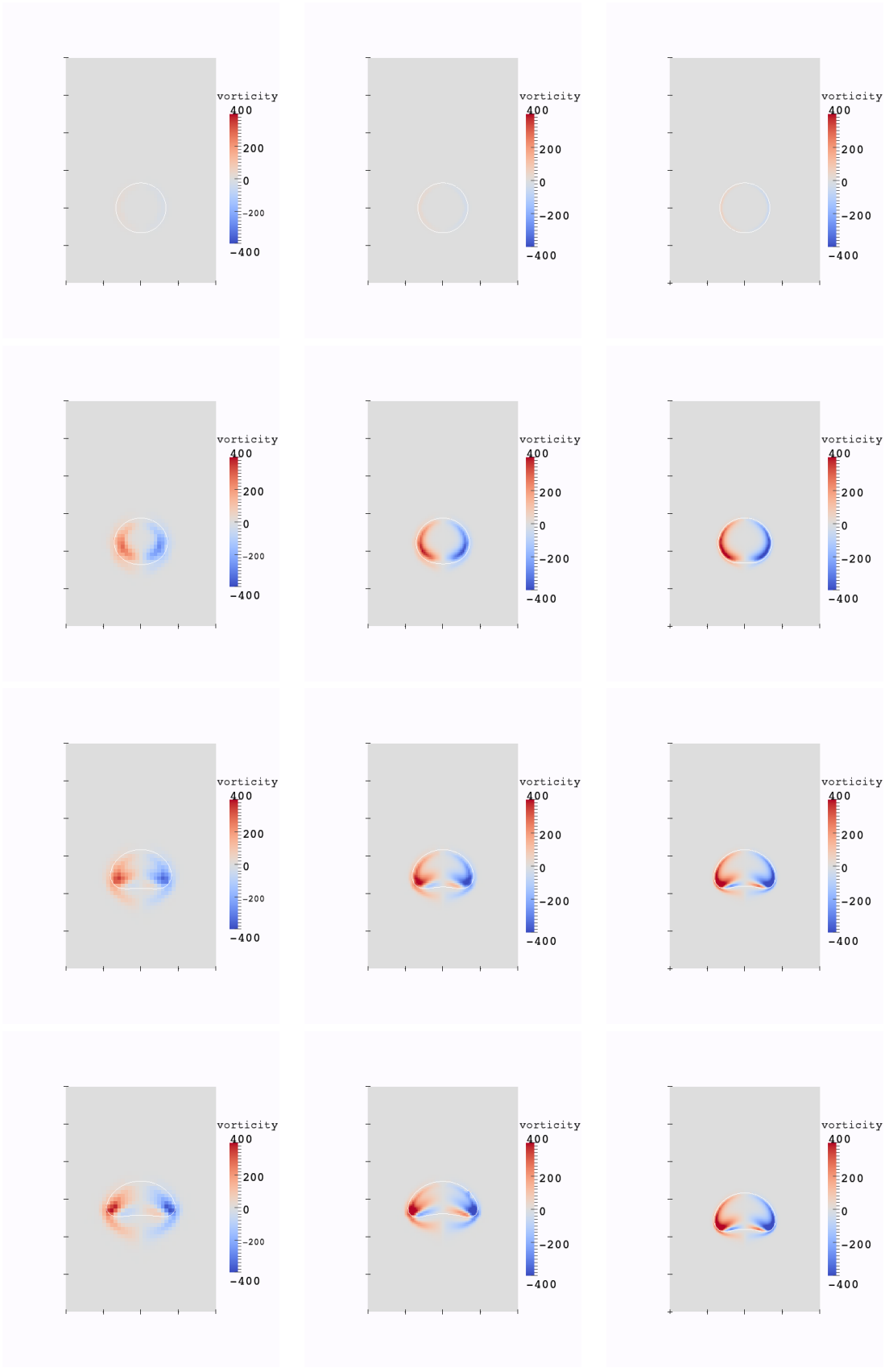


Figure 9. Evolution of the interface and vorticity values for the small bubble test case, at times 0, 0.02, 0.035, 0.05, resolution: 40 × 60 (**left**); 80 × 120 (**middle**); 160 × 240 (**right**).

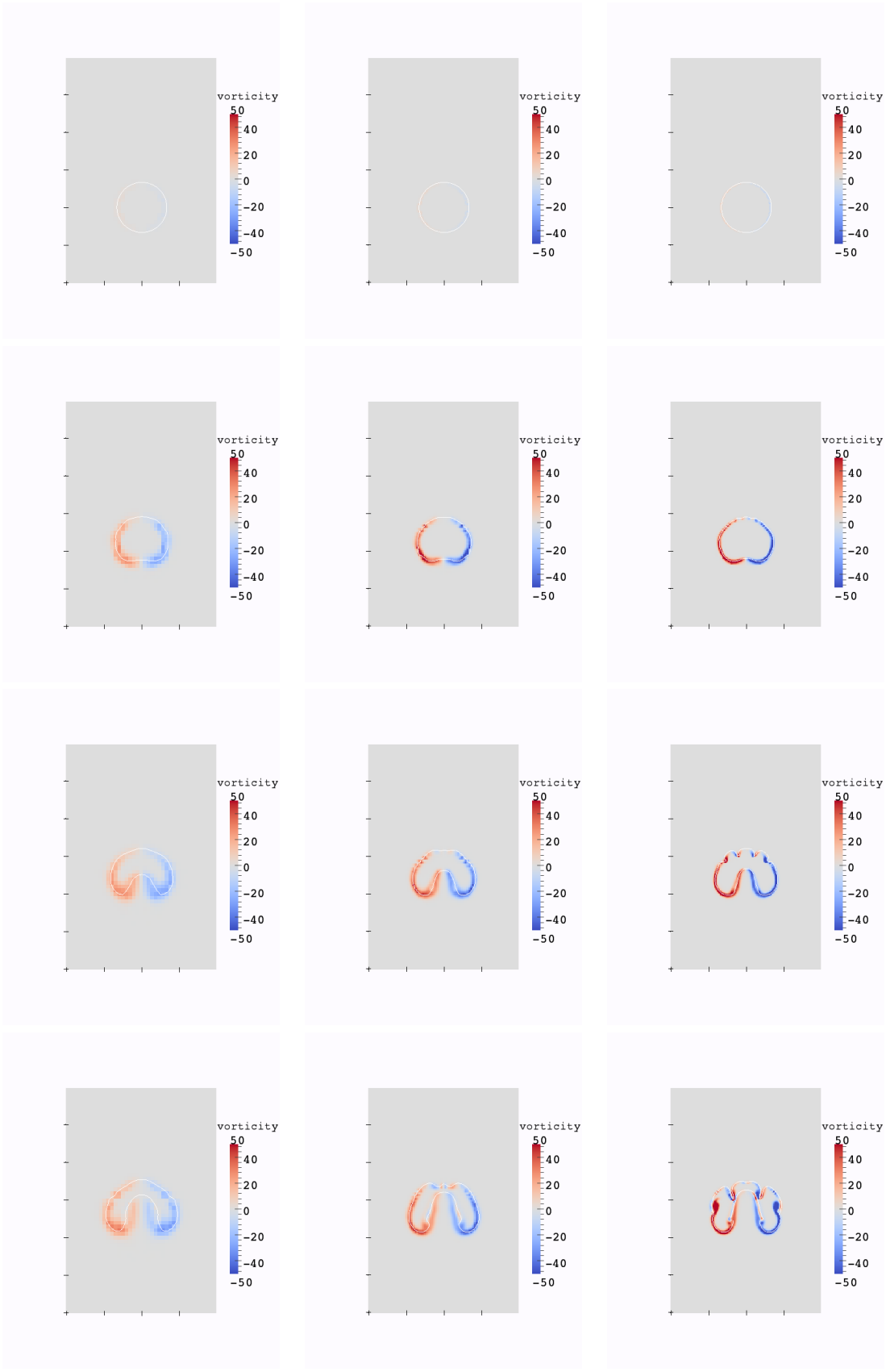


Figure 10. Evolution of the interface and vorticity values for the large bubble test case, at times 0, 0.2, 0.35, 0.5, resolution: 40 × 60 (**left**); 80 × 120 (**middle**); 160 × 240 (**right**).

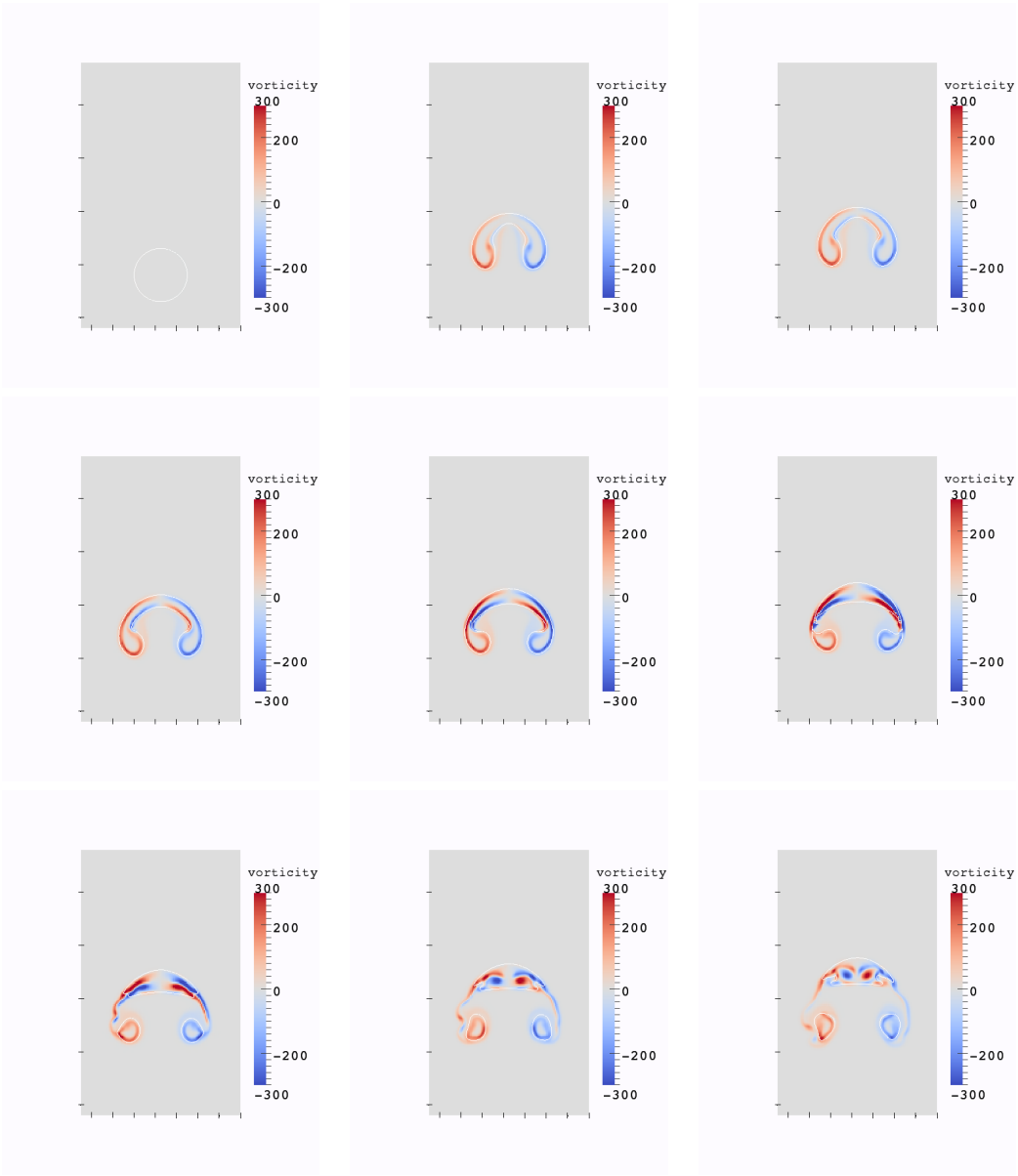


Figure 11. Evolution of the interface and vorticity values for the bubble test case from [36] at non-dimensional times $t * \sqrt{-g/0.025} = 0., 2.8, 6, 4., 4.4, 4.8, 5.2, 5.6, 6.$, resolution 120×200.

6.4. Two Air Bubbles in Water

Two air bubble are initially at rest in water, see Figure 12. At the same time that they are rising, their interaction produces larger deformations than for a single bubble. This test case is meant to assessing the conservativity of the new method, as the increase in pression resolution is meant to increase this conservativity.

The value of the physical parameters are

$$\begin{cases} R = 1/30 \text{ m}, \\ \text{center of first bubble} = (0,0), \\ \text{center of second bubble} = (-0.04, 0.08), \\ \text{final time } T_f = 0.12, \\ \rho_{water} = 1000 \text{ kg/m}^3, \\ \mu_{water} = 1.137 \times 10^{-3} \text{ kg/ms}, \\ \rho_{air} = 1.226 \text{ kg/m}^3, \\ \mu_{air} = 1.78 \times 10^{-5} \text{ kg/ms}, \\ \sigma = 0.0728 \text{ kg/s}^2, \\ g = -9.8 \text{ m/s}^2. \end{cases} \tag{43}$$

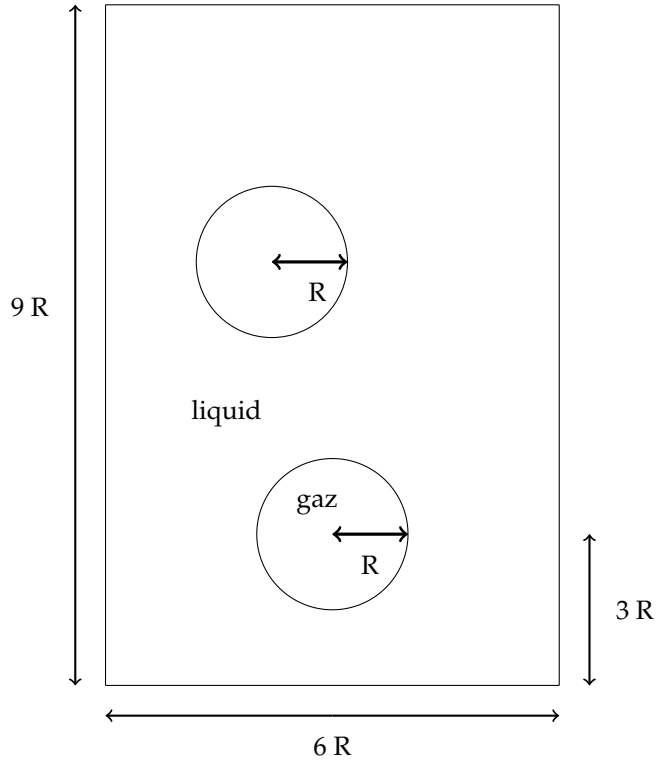


Figure 12. Initial fluid domain for the test case of the two rising bubbles in water.

In Figure 13, the interface evolution and the vorticity values for the Ghost Fluid method (used for the pressure computation) and the new method are plotted. One observes small oscillations of the vorticity near the interface between two fluids for the Ghost Fluid method, but not for the new method. The conservation of the partial mass of air and momentum of air in each dimension for the new method, and the Ghost Fluid method are depicted in Figures 14–16 respectively. As air is the less dense fluid, it is more prone to big oscillations if erroneous mass and momentum transfers between the two fluids occur. One can observe that with the new method and its second-order pressure resolution, the oscillations of these quantities are notably less present than for the Ghost Fluid. The partial mass is better conserved too.

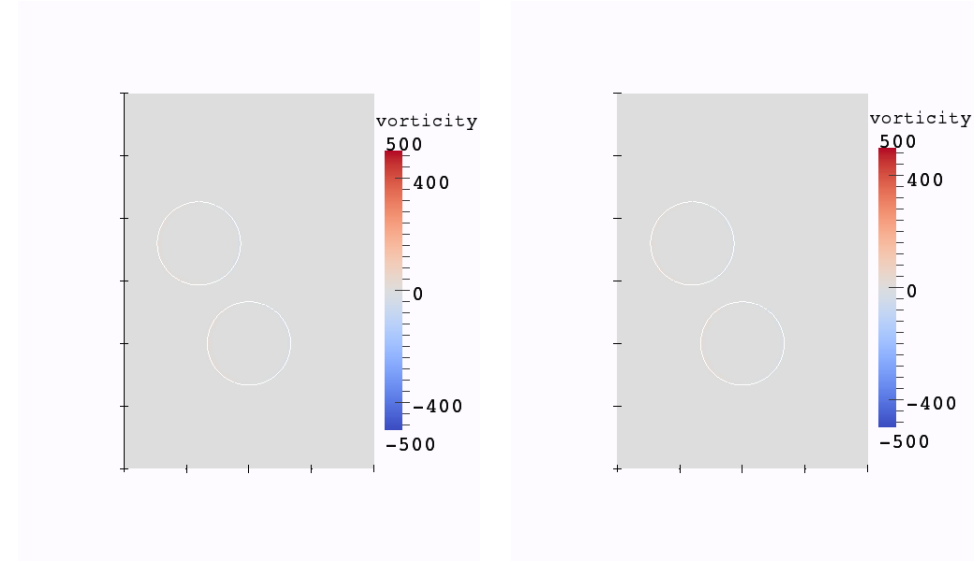


Figure 13. *Cont.*

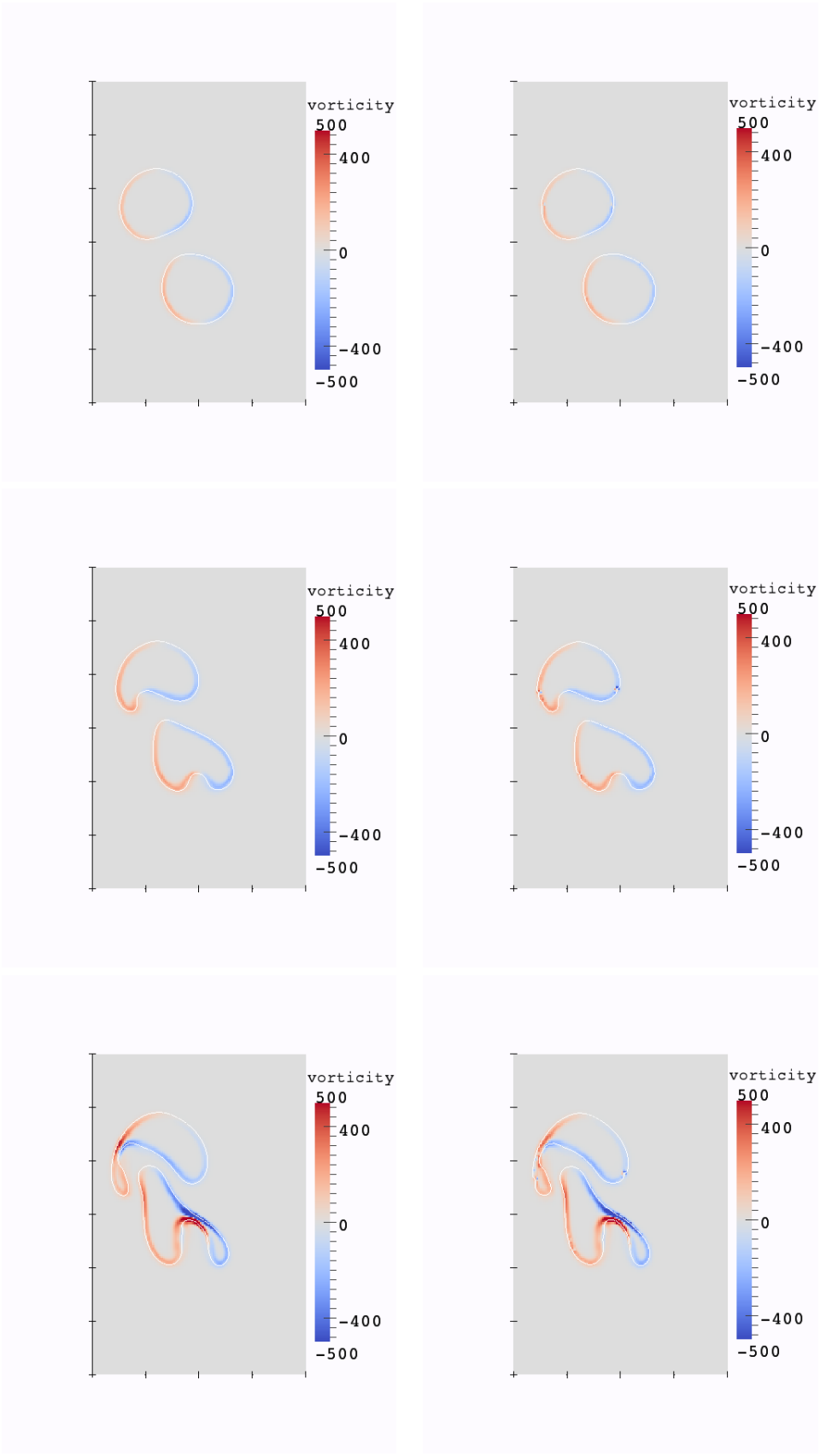


Figure 13. Evolution of the interface and vorticity values for the two bubbles test case at times $t = 0., 0.04, 0.08, 0.12$, resolution 120×180, (**left**): new method for pressure, (**right**): Ghost Fluid method for pressure.

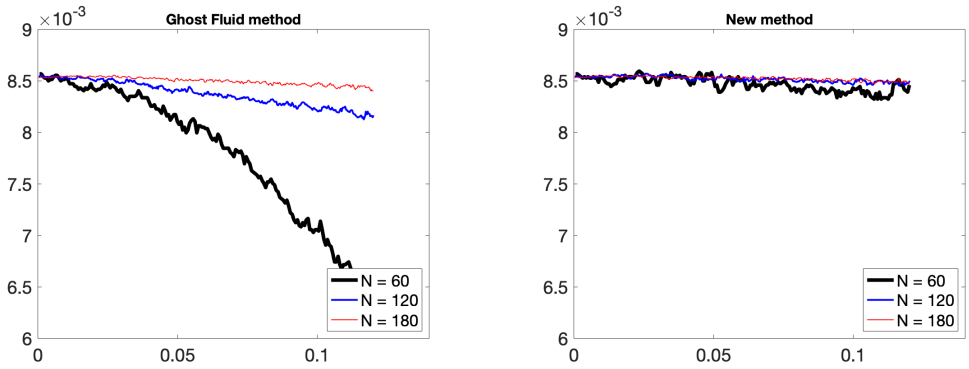


Figure 14. Evolution in time of partial mass of air with N = 60, 90, 120 points in x-direction with (**a**) Ghost Fluid method, (**b**) new method.

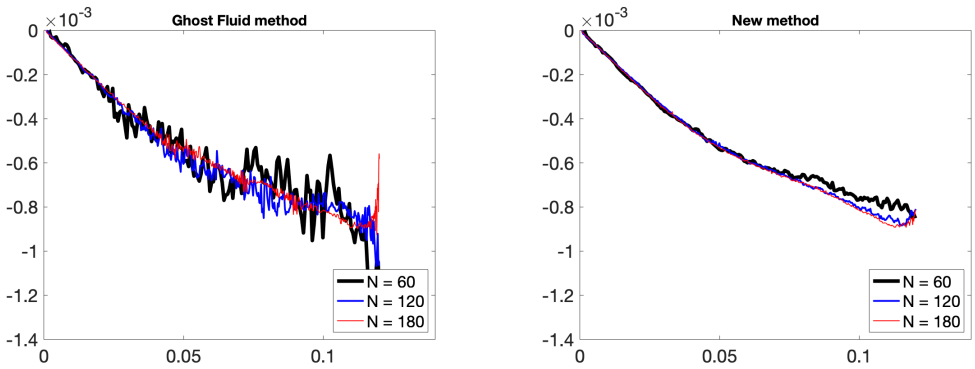


Figure 15. Evolution in time of partial x-momentum of air N = 60, 90, 120 points in x-direction with (**a**) Ghost Fluid method, (**b**) new method.

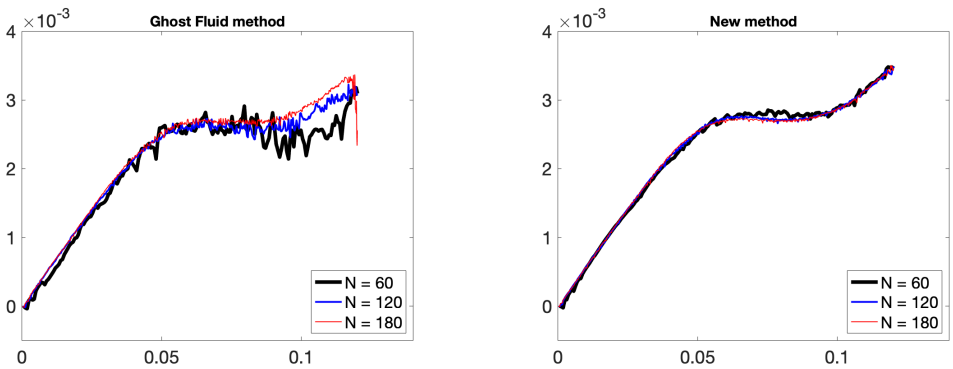


Figure 16. Evolution in time of partial y-momentum of air N = 60, 90, 120 points in x-direction with (**a**) Ghost Fluid method, (**b**) new method.

7. Conclusions

We have developed a new method on Cartesian grids for the simulation of incompressible flows with large density ratios. This method relies on a sharp resolution of the pressure term across the interface defined by a level-set function. The advantage of the proposed approach is its simplicity to implement in an existing Cartesian monofluid Navier–Stokes solver as for instance the Ghost Fluid or CSF methods. It is only necessary to modify the stencil for the pressure equation at some irregular grid points by adding one additional point. This Cartesian scheme uses thus additional unknowns located on the interface to

discretize with second-order accuracy the jump conditions across the interface. The viscous term is treated with a regularizing approach that allows us to eliminate terms in the jump conditions. Indeed, it has been shown [32,33] that the regularization has no significant impact on the accuracy of the results. Numerical results show that this method leads to more accurate and stable results than some reference methods as for instance the well known Ghost Fluid or CSF methods. The conservation of volume of each phase is also better conserved. We thus take advantage of both the regularity of the interface defined by the level-set function (curvature is numerically consistent), and the mass conservation of each phase. The CLSVOF [11] can also provide good mass conservation and precision of the interface but with a more complex numerical implementation.

Future works include an extension of the method to three-dimensional problems with interactions with solids including the numerical simulation of wave energy converters [37,38]. We also aim to study the sensibility of the reinitialization procedure of the level-set function defining the bifluid interface, as for instance the one developed in [31].

Author Contributions: M.B. and L.W. both contributed to all aspects: conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, supervision, project administration, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the Investments for the future Programme IdEx Bordeaux (ANR-10-IDEX-03-02).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cisternino, M.; Weynans, L. A parallel second order Cartesian method for elliptic interface problems. *Commun. Comput. Phys.* **2012**, *12*, 1562–1587. [CrossRef]
2. Raessi, M.; Pitsch, H. Consistent mass and momentum transport for simulating incompressible interfacial flows with large density ratios using the level set method. *Comput. Fluids* **2012**, *63*, 70–81. [CrossRef]
3. Brackbill, J.U.; Kothe, D.B.; Zemach, C. A continuum method for modeling surface tension. *J. Comput. Phys.* **1992**, *100*, 335–354. [CrossRef]
4. Sussman, M.; Smereka, P.; Osher, S. A level-set approach for computing solutions to incompressible two-phase flows. *J. Comput. Phys.* **1994**, *114*, 146–159. [CrossRef]
5. Galusinski, C.; Vigneaux, P. On stability condition for bifluid flows with surface tension: Application to microfluidics. *J. Comput. Phys.* **2008**, *227*, 6140–6164. [CrossRef]
6. Kang, M.; Fedkiw, R.P.; Liu, X.D. A Boundary Condition Capturing Method for Multiphase Incompressible Flow. *J. Sci. Comput.* **2000**, *15*, 323–360. [CrossRef]
7. Liu, X.D.; Fedkiw, R.P.; Kang, M. A boundary capturing method for Poisson's equation on irregular domains. *J. Comput. Phys.* **2000**, *160*, 151–178. [CrossRef]
8. Couderc, F. Développement d'un code de calcul pour la simulation d'écoulements de fluides non miscibles: Application à la désintégration assistée d'un jet liquide par un courant gazeux. PhD Thesis, ENSAE, Toulouse, France, 2007.
9. Tanguy, S.; Menard, T.; Berlemont, A. A Level Set Method for vaporizing two-phase flows. *J. Comput. Phys.* **2007**, *221*, 837–853. [CrossRef]
10. Rudman, M. A volume-tracking method for computing incompressible multifluid flows with large density variations. *Int. J. Numer. Meth. Fluids* **1998**, *28*, 357–378. [CrossRef]
11. Sussman, M.; Puckett, E.G. A coupled level set and volume of fluid method for computing 3D and axisymmetric incompressible two-phase flows. *J. Comput. Phys.* **2000**, *162*, 301–337. [CrossRef]
12. Sussman, M.; Smith, K.M.; Hussaini, M.Y.; Ohta, M.; Zhi-Wei, R. A sharp interface method for incompressible two-phase flows. *J. Comput. Phys.* **2007**, *221*, 469–505. [CrossRef]
13. Hu, X.; Khoo, B.C.; Adams, N.A.; Huang, F.L. A conservative interface method for compressible flows. *J. Comput. Phys.* **2006**, *219*, 553–578. [CrossRef]
14. Nangia, N.; Griffith, B.E.; Patankar, N.A.; Bhalla, A.P.S. A robust incompressible Navier-Stokes solver for high density ratio multiphase flows. *J. Comput. Phys.* **2019**, *390*, 548–594. [CrossRef]
15. Nangia, N.; Patankar, N.A.; Bhalla, A.P.S. A DLM immersed boundary method based wave-structure interaction solver for high density ratio multiphase flows. *J. Comput. Phys.* **2019**, *398*, 108804. [CrossRef]
16. Osher, S.; Sethian, J.A. Fronts propagating with curvature-dependent speed: Algorithms based on Hamiltonâ Jacobi formulations. *J. Comput. Phys.* **1988**, *79*, 12–49. [CrossRef]

17. Sethian, J.A. *Level Set Methods and Fast Marching Methods*; Cambridge University Press: Cambridge, UK, 1999.
18. Sethian, J.A. Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts. *J. Comput. Phys.* **2001**, *169*, 503–555. [CrossRef]
19. Osher, S.; Fedkiw, R. *Level Set Methods and Dynamic Implicit Surfaces*; Springer: Berlin/Heidelberg, Germany, 2003.
20. Adalsteinsson, D.; Sethian, J.A. The Fast Construction of Extension Velocities in Level Set Methods. *J. Comput. Phys.* **1999**, *148*, 2–22. [CrossRef]
21. Rhie, C.; Chow, W. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA J.* **1983**, *21*, 1525–1532. [CrossRef]
22. Mittal, R.; Dong, H.; Bozkurttas, M.; Najjar, F.; Vargas, A.; von Loebbecke, A. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *J. Comput. Phys.* **2008**, *227*, 4825 – 4852. [CrossRef]
23. Chorin, A. Numerical solution of the Navier-Stokes equations. *Math. Comp.* **1968**, *22*, 745–762. [CrossRef]
24. Temam, R. Sur l'approximation de la solution des equations de Navier-Stokes par la méthode des pas fractionnaires II. *Archiv. Rat. Mech. Anal.* **1969**, *32*, 377–385. [CrossRef]
25. Sethian, J. A fast marching level set method for monotonically advancing fronts. *Appl. Math.* **1996**, *93*, 1591–1595. [CrossRef]
26. Rouy, E.; Tourin, A. A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.* **1992**, *29*, 867–884. [CrossRef]
27. Tsai, Y.H.R.; Cheng, L.T.; Osher, S.; Zhao, H.K. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *SIAM J. Numer. Anal.* **2003**, *41*, 673–694. [CrossRef]
28. Jiang, G.S.; Shu, C.W. Efficient Implementation of Weighted ENO Schemes. *J. Comput. Phys* **1995**, *126*, 202–228. [CrossRef]
29. Russo, G.; Smereka, P. A remark on computing distance functions. *J. Comput. Phys.* **2000**, *163*, 51–67. [CrossRef]
30. duChene, A.; Min, C.; Gibou, F. Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes. *J. Sci. Comput.* **2008**, *35*, 114–131. [CrossRef]
31. Luddens, F.; Bergmann, M.; Weynans, L. Enablers for high-order level set methods in fluid mechanics. *Int. J. Numer. Meth. Fluids* **2015**, *79*, 654–675. [CrossRef]
32. Desjardins, O.; Pitsch, H. A spectrally refined interface approach for simulating multiphase flows. *J. Comput. Phys.* **2009**, *228*, 1658–1677. [CrossRef]
33. Herrmann, M. The influence of density ratio on the primary atomization of a turbulent jet in crossflow. *Proc. Combust. Inst.* **2011**, *33*, 2079–2088. [CrossRef]
34. Chenadec, V.L.; Pitsch, H. A monotonicity preserving conservative sharp interface flow solver for high density ratio two-phase flows. *J. Comput. Phys.* **2013**, *249*, 185–203. [CrossRef]
35. Martin, J.C.; Moyce, W.J. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philos. Trans. R. Soc. London, Ser. A* **1952**, *244*, 312–324.
36. Grenier, N.; Antuono, M.; Colagrossi, A.; Touze, D.L.; Alessandrini, B. An Hamiltonian interface SPH formulation for multi-fluid and free-surface flows. *J. Comput. Phys.* **2009**, *228*, 8380–8393. [CrossRef]
37. Khedkar, K.; Nangia, N.; Thirumalaisamy, R.; Bhalla, A.P.S. The inertial sea wave energy converter (ISWEC) technology: Device-physics, multiphase modeling and simulations. *Ocean Eng.* **2021**, *229*, 108879. [CrossRef]
38. Bhalla, A.P.S.; Nangia, N.; Dafnakis, P.; Bracco, G.; Mattiazzo, G. Simulating water-entry/exit problems using Eulerian–Lagrangian and fully-Eulerian fictitious domain methods within the open-source IBAMR library. *Appl. Ocean Res.* **2020**, *94*, 101932. [CrossRef]