A sharp Cartesian method for incompressible flows with large density ratios

Michel Bergmann, Lisl Weynans

A sharp Cartesian method for incompressible flows with large density ratios

M. Bergmann[1], **L. Weynans**[1]

[1]**Team Memphis, INRIA Bordeaux-Sud-Ouest & CNRS UMR 5251, Université de Bordeaux, France**

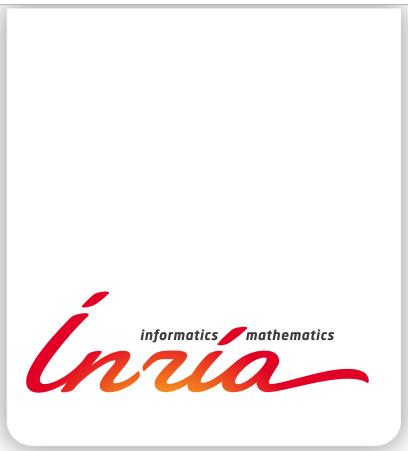
RESEARCH REPORT

N° 8926

Mars 2017

Project-Teams Memphis

ISSN 0249-6399 ISRN INRIA/RR--8926--FR+ENG



A sharp Cartesian method for incompressible flows with large density ratios

M. Bergmann[1], L. Weynans[1]*

[1]Team Memphis, INRIA Bordeaux-Sud-Ouest & CNRS UMR 5251,
Université de Bordeaux, France

Project-Teams Memphis

Research Report n° 8926 — Mars 2017 — 23 pages

Abstract: A new Cartesian method for bifluid incompressible flows with high density ratios is presented. The specificity of the method relies on 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 thanks to 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 numerical tests show the improvements due to this sharp method compared to classical first order methods.

Key-words: Incompressible flows, bifluid flows, finite-differences, projection method, cartesian grid, level-set, jump conditions across interface, interface unknowns

* Corresponding author: lisl.weynans@inria.fr

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

351, Cours de la Libération
Bâtiment A 29
33405 Talence Cedex

Une méthode cartésienne précise sur l'interface pour des écoulements incompressibles avec de grands ratios de densité

Résumé : Nous présentons une nouvelle méthode cartésienne pour des écoulements incompressibles bifluides avec de grands ratios de densité. La spécificité de la méthode repose sur l'utilisation d'un schéma numérique non régularisé et d'ordre deux pour la résolution du problème elliptique discontinu pour la pression. Les équations de Navier-Stokes sont intégrées en temps grâce à une méthode à pas fractionnaire basée sur le schéma de Chorin, et sont discrétisées en espace sur une grille cartésienne. L'interface entre les fluides est représentée implicitement par une fonction level-set. Les tests numériques montrent les améliorations apportées par cette nouvelle méthode comparée aux méthodes d'ordre un classiques dans la littérature.

Mots-clés : Ecoulements incompressibles, écoulements bifluides, différences finies, méthode de projection, grille cartésienne, level-set, conditions de saut au travers de l'interface, inconnues d'interface

Contents

1 Introduction

In this paper we are concerned with incompressible bi-fluid flows like air and water, and by the accurate description of the phenomena occurring at their interface. We thus present a sharp Cartesian method for the simulation of incompressible flows with high density and viscosity ratios. This method is inspired from the second-order Cartesian method for elliptic problems with immersed interfaces developed in [5].

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 bi-fluid 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 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 [22] 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 in the vicinity of the interface the properties of the fluids, 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 [2], 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 [17] and [9]) because it offers a straightforward way to implement the presence of two fluids in an already existing mono-fluid Navier-Stokes code. However, the exact way that the regularization should be performed is not always clear, and spurious oscillations at the bi-fluid interface can appear due to errors in the pressure gradient computations. Another non-conservative method introduced by Kang, Fedkiw and Liu [13] after the CSF is the Ghost Fluid Method (GFM). It is based on a first-order method developed in [14] 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 numerous works, for instance [6] and [33]. One drawback is that the method is only first-order accurate near the interface [22] 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 [17] 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 [25], [31], [32], [12] and more recently [22]. An explicit interface representation is necessary even if the interface do not coincide with grid points. Conservative methods are generally more stable that non-conservative methods. The price for this increased stability is an additional amount of work due to the interface reconstruction, which can be performed from informations carried by Lagrangian markers or by cell quantities such as volume fractions.

In this paper we aim to preserve as much as possible the simplicity of the Ghost-Fluid Method of [13], 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 new method where the discontinuities across

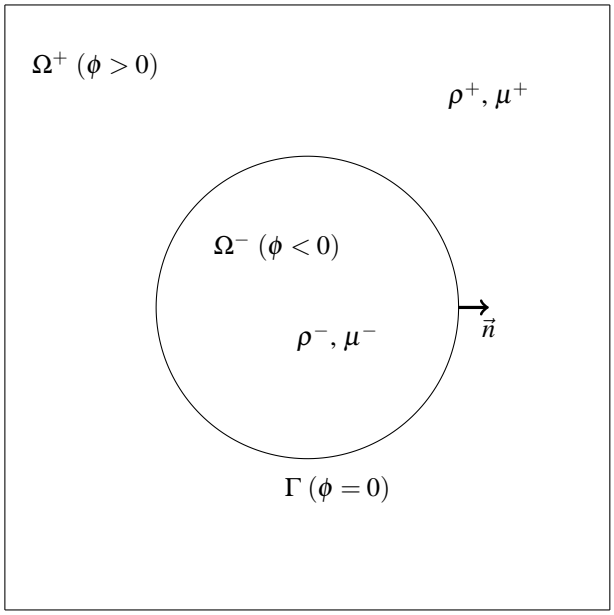


Figure 1: Sketch of the computational domain.

the interface are taken into account in a sharp way with a second-order scheme inspired from [5]. 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 bi-fluid flows that we consider (§2), the discretization of these equations in each fluid and at the interface are presented (§3-4). The second-order numerical resolution of the elliptic problem arising from the computation of the pressure is introduced (§5), and the overall is validated on several two-dimensional numerical test cases (§6).

2 Governing equations

2.1 Flow configuration

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. These domains are generally defined with a scalar function ϕ that takes different values in each subdomains 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 $\boldsymbol{n}$ and the unit tangent vector is $\boldsymbol{\eta}$. The density is

$$\rho = \rho^- + H(\phi)(\rho^+ - \rho^-),$$

and the viscosity is

$$\mu = \mu^- + H(\phi)(\mu^+ - \mu^-),$$

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

The flow is modeled in each subdomain with the incompressible Navier-Stokes equations:

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

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

$$\tau = \mu(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T).$$

2.2 Jump conditions at the interface

The above equations are completed by jump conditions at the interface Γ between the two fluids. In what follows jumps are defined by $[\psi] = \psi^+ - \psi^-$.

- The first ones describe the balance between the normal stresses at the interface and the surface tension σ, with κ the local curvature of the interface Γ,

$$[p - 2\mu(\nabla u\cdot\boldsymbol{n}, \nabla v\cdot\boldsymbol{n})\cdot\boldsymbol{n}] = \sigma\kappa, \tag{1}$$
$$[\mu(\nabla u\cdot\boldsymbol{n}, \nabla v\cdot\boldsymbol{n})\cdot\boldsymbol{\eta} + (\nabla u\cdot\boldsymbol{\eta}, \nabla v\cdot\boldsymbol{\eta})\cdot\boldsymbol{n}] = 0. \tag{2}$$

  Several others jump conditions can be derived from continuity properties across the interface.

- For a viscous fluid the velocity field is continuous across the interface

$$[u] = 0, \tag{3}$$
$$[v] = 0. \tag{4}$$

- The material derivative of (3)-(4) is zero, therefore

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

  which leads to

$$\left[\frac{\nabla p}{\rho}\right] = \left[\frac{\nabla\cdot\tau}{\rho}\right]. \tag{5}$$

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

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

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

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

  Combining the two last relationships, we obtain

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

  Consequently

$$[p] = 2[\mu](\nabla u\cdot\boldsymbol{n}, \nabla v\cdot\boldsymbol{n})\cdot\boldsymbol{n} + \sigma\kappa. \tag{6}$$

  We will use this equation to compute the pressure jump at the interface.

2.3 Moving interface

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

$$\phi_t + \hat{\boldsymbol{u}} \cdot \nabla \phi = 0, \tag{7}$$

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 is to consider $\hat{\boldsymbol{u}} = \boldsymbol{u}$ in the whole domain. Another choice is the extension velocity introduced in [1].

3 Numerical method outside the discontinuities

For simplicity reason the computational domain is discretized on a two-dimensional uniform Cartesian grid with a grid spacing $\Delta x = \Delta y = h$, but the following approach stands for non uniform Cartesian meshes. The points on the Cartesian grid are named with indices such as $M_{i,j} = (x_i, y_j)$. We denote by u_{ij} the approximation of u at the point (x_i, y_j). The narrow band is the set of points having at least one neighbor on the other side of the interface. An accurate computation of the interface is thus necessary.

3.1 Interface computation

In order to provide an accurate discretization in the vicinity of the interface we need some geometric information about the interface, and thus a good choice for ϕ is necessary. This information can be provided by the level set method, introduced by Osher and Sethian [21]. We refer the interested reader to [27], [28] and [20] for recent reviews of this method. A useful choice for the level set function is the signed distance function to the interface:

$$\phi(x) = \begin{cases} dist_\Gamma(x) & \text{if } x \in \Omega^+, \\ -dist_\Gamma(x) & \text{if } x \in \Omega^-, \\ 0 & \text{if } x \in \Gamma. \end{cases} \tag{8}$$

The zero isoline 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 an easy computational of its normal

$$\boldsymbol{n}(x) = \frac{\nabla \phi(x)}{|\nabla \phi(x)|}. \tag{9}$$

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

$$\kappa = \nabla \cdot \boldsymbol{n}. \tag{10}$$

In what follows, the level-set function is advected by the fluid velocity $\boldsymbol{u}$ in the whole domain

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

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 ones wants to compute the curvature of the interface from (10), the

level-set function needs to be accurate enough (at least third-order) so that the finite difference formulas used to discretize (10) 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. [17] introduced a reinitialization algorithm to recover the signed distance function through the resolution of the eikonal equation

$$|\nabla\phi| = 1.$$

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 [17] or using a Gauss-Seidel based method as in fast marching methods [29, 24] or fast sweeping methods [35].

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 [30] with a RK3 scheme, for the transport and for the reinitialization through a relaxation equation. It provides a high-order yet stable resolution. But, 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 noticed in [26]. For this reason, Russo and Smereka [26] introduced a subcell fix taking into account the interface location in the reinitialization procedure. This technique was extended to a higher order accuracy in [8] 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 [15] 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 [8] near the interface.

In what follows, we will 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 to distinguish the effects of the new scheme from the effects of the reinitialization technique.

3.2 Flow computation

We use a classical non-incremental projection method [4, 34], *i.e.* the guess value for the pressure in the prediction step is zero. This choice avoids instability issues due to the discontinuous pressure values when the interface moves. It avoids to wonder which jump conditions across the interface should be satisfied by the pressure correction. Indeed, these jump conditions for the pressure correction are not based on physical considerations.We thus compute successively:

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} = -(\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n + \frac{1}{\rho}(\nabla \cdot \tau)^n + \boldsymbol{g} \quad \text{(prediction step)}, \tag{11}$$

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^*}{\Delta t} = -\frac{\nabla p}{\rho} \qquad \text{(correction step)}. \tag{12}$$

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.

The pressure is computed through the resolution of a Poisson equation in order to enforce the divergence-free condition. At each grid point outside the narrow band around the interface, the following relationship is satisfied:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = \frac{\nabla \cdot \boldsymbol{u}^*}{\Delta t}, \tag{13}$$

with Neumann boundary condition:

$$\frac{\nabla p}{\rho} = \frac{\boldsymbol{u}^b - \boldsymbol{u}^*}{\Delta t}, \tag{14}$$

where $\boldsymbol{u}^b$ is the value of the velocity to be imposed on the external boundaries. We will provide additional details about the jump conditions that have to be satisfied across the interface for this problem in subsection 4.3.

The overall algorithm is the following:

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

We compute at each iteration an adaptive time step taking into account the restrictions due to convection, viscosity, surface tension and gravity.

The convective time step restriction is given by

$$\Delta t \left(\frac{|u|_{\max}}{\Delta x} + \frac{|v|_{\max}}{\Delta y} \right) \leq 1. \tag{15}$$

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

$$\Delta t \left(\max\left(\frac{\mu^-}{\rho^-}, \frac{\mu^+}{\rho^+}\right)\left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}\right) \right) \leq 1 \tag{16}$$

The time step restriction associated with the surface tension is simiar to the one in [13] and in [6]:

$$\Delta t \sqrt{\frac{\sigma |\kappa|}{\min(\rho^+, \rho^-) \min(\Delta x^2, \Delta y^2)}} \leq 1. \tag{17}$$

We only apply it for grid points in the narrow band.

In what follows, all the unknowns are collocated in space on Cartesian meshes. In all the applications considered we have not observed any odd-even coupling. However, this point can be fixed using some corrections [23, 19].

4 Specific sharp method for the discontinuities across the interface

The values of the viscosity and the densities are discontinuous across the interface. Therefore, if the numerical scheme described above was applied on the grid points in the narrow band, stability problems could occur because the approximations of the following terms are not consistent:

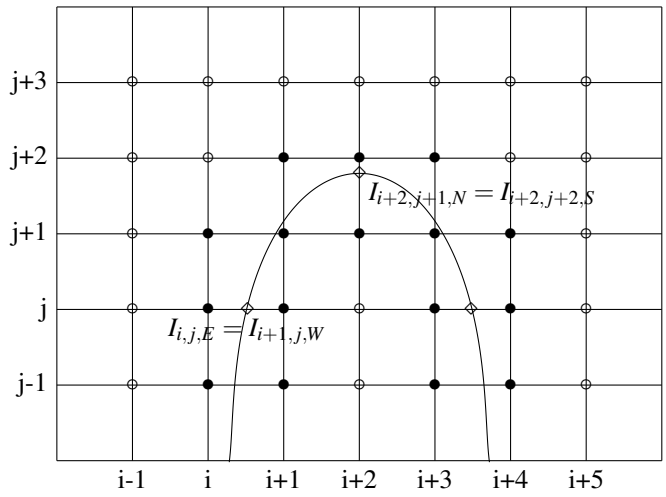


Figure 2: Example of geometrical configuration, with regular grid points (white circles), irregular points (black circles), and interface points (diamonds) with the two possible notations.

- viscous terms,
- divergence of the predicted velocity,
- elliptic operator in the correction step,
- gradient of the pressure.

We have to devise a specific treatment for the points near the interface, for each of these steps. The computation of the convective terms is not mentioned in the above enumeration because it is performed with a fifth-order WENO scheme, which provides automatically 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. Therefore, it seems coherent to have the same numerical scheme for the convection of the interface and the convection of the fluids.

Let us introduce some 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. 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 respectively 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 . With this notation the same interface point can be described in two different ways

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

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

4.1 Viscous terms

We follow a continuous approach and regularize the quantities used for the computation of the viscous terms. It has been proven in [7] and [11] that this continuous approach provides correct accuracy for high Reynolds numbers flows. It has also been used successfully in [22] and [2]. 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 (2) 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 viscosity and the inverse of the density are regularized in this step by a discrete convolution [22]:

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

Then we discretize the viscous terms with a classical second-order centered scheme.

Due to this continuous approach for the viscous terms, the jump condition for the pressure (6) writes

$$[p] = \sigma\kappa. \tag{18}$$

and we avoid the computation of the term $2[\mu](\nabla u \cdot \boldsymbol{n}, \nabla v \cdot \boldsymbol{n}) \cdot \boldsymbol{n}$ across the interface.

4.2 Divergence of predicted velocity

The predicted velocity $\boldsymbol{u}^*$ obtained after the prediction step (11) is defined only on grid points. We need to compute the divergence of this predicted velocity to solve the elliptic equation (13). 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. Formally this is equivalent to a standard first-order decentered scheme.

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. More precisely, we denote $\boldsymbol{u}^*_S$ the value of the solution 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

$$\left(\nabla \cdot \boldsymbol{u}^*\right)_{i,j} = \frac{u^*_N - u^*_S}{x_N - x_S} + \frac{v^*_E - v^*_W}{y_E - y_W}.$$

4.3 Elliptic problem near the interface

To compute the pressure, according to the jump conditions (1) - (6) presented in section 2, it is necessary to solve an elliptic problem with discontinuous values of the solution and its derivative across the interface:

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

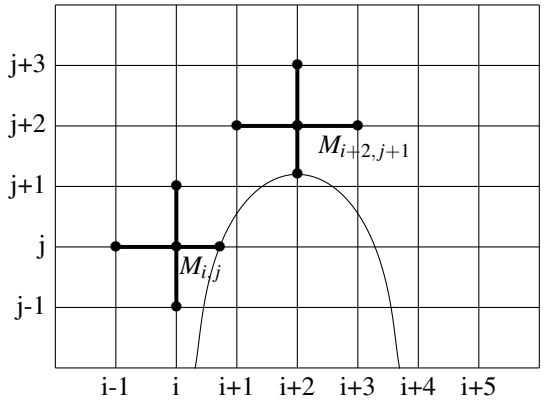


Figure 3: Example of geometrical configuration, with points involved in the discretization of the divergence of the predicted velocity and pressure gradient at grid points $M_{i,j}$ and $M_{i+2,j+2}$ in black.

Because the viscous terms are handled with a regularization approach, they can not be considered as discontinuous anymore. Therefore, the elliptic problem to solve becomes

$$\nabla \cdot (\frac{1}{\rho}\nabla p) = \frac{\nabla \cdot \boldsymbol{u}^*}{\Delta t}, \text{ in } \Omega^+ \cup \Omega^- \tag{19}$$

$$[p] = \sigma\,\kappa \text{ on } \Gamma, \tag{20}$$

$$\left[\frac{\nabla p}{\rho}\right] = 0 \text{ on } \Gamma, \tag{21}$$

avoiding the computation of the terms $(u_n, v_n).\boldsymbol{n}$ and $\left[\frac{\nabla \cdot \tau}{\rho}\right]$. The details of the resolution of this elliptic problem will be provided in section 5.

4.4 Pressure gradient near the interface and correction step

In order to keep a consistent discretization, the gradient of the pressure p computed through the resolution of the elliptic problem in the last subsection, is also computed with an adapted decentered stencil near the interface, see Figure 3. More precisely, with the same notations as before, the discretization reads

$$(\nabla p)_{i,j} = \begin{pmatrix} \frac{p_N - p_S}{x_N - x_S} \\ \frac{p_E - p_W}{y_E - y_W} \end{pmatrix}. \tag{22}$$

If one of the discretization point is an interface point, we consider the value of the interface 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.

5 Numerical resolution of elliptic problems with immersed interfaces

The elliptic problem with discontinuous values across an interface (19) - (21) is solved with the second-order method developed in [5]. 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 [13] 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

$$\left(\nabla.(\frac{1}{\rho^{\pm}}\nabla p)\right)_{i,j} = \frac{1}{\rho^{\pm}}\Delta p = \frac{1}{\rho^{\pm}}\frac{\frac{p_N - p_{ij}}{x_N - x_i} - \frac{p_{ij} - p_S}{x_i - x_S}}{\frac{x_N - x_S}{2}} + \frac{1}{\rho^{\pm}}\frac{\frac{p_E - p_{ij}}{y_E - y_j} - \frac{p_{ij} - p_W}{y_j - y_W}}{\frac{y_E - y_W}{2}}, \tag{23}$$

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

5.2 Discrete flux transmission conditions

At each interface point we create two additional unknowns, called interface unknowns, and denoted 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 [5], we do not have a jump condition on the normal derivative, but on the whole gradient, as expressed in formula (21). 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 $\gamma = N, S, W, E$.

$$p^+_{i,j,\gamma} - p^-_{i,j,\gamma} = \sigma\,\kappa_{i,j,\gamma}, \tag{24}$$

$$\frac{1}{\rho^+}(\partial_x p^+)_{i,j,\gamma} - \frac{1}{\rho^-}(\partial_x p^-)_{i,j,\gamma} = 0 \text{ if } \gamma = E, W. \tag{25}$$

$$\frac{1}{\rho^+}(\partial_y p^+)_{i,j,\gamma} - \frac{1}{\rho^-}(\partial_y p^-)_{i,j,\gamma} = 0 \text{ if } \gamma = N, S. \tag{26}$$

We want the truncation error of the discretization of flux equality (25-26) 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* non equidistant points. For example we approximate the flux on the

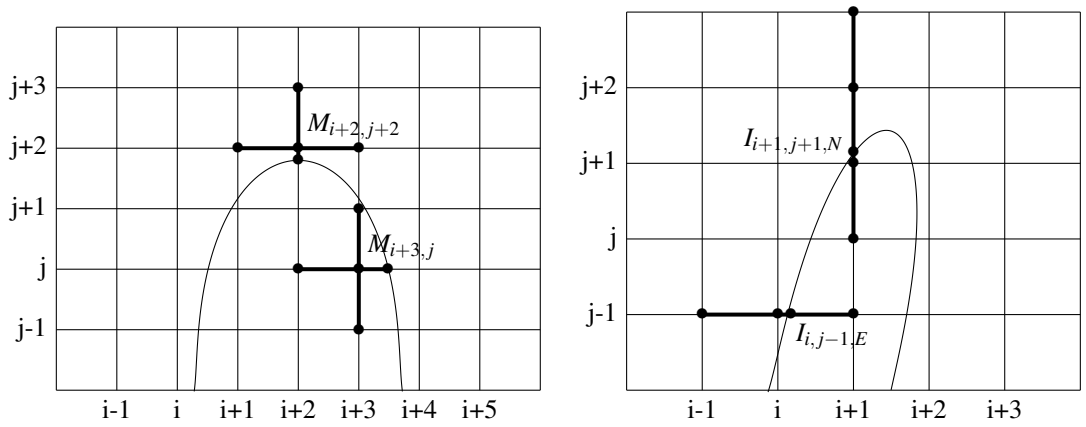


Figure 4: Left: points involved in the discretization of the elliptic operator at grid nodes $M_{i+2,j+2}$ and $M_{i+3,j}$ in black, right: Example of stencils for the discretization of jump conditions. Points involved in the discretization of the x-derivative of the pressure at interface point $I_{i,j-1,E}$ and 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.

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:

$$(\partial_x p^{\pm})_{i,j,E} \approx \frac{(p_{i-1,j} - p^{\pm}_{i,j,E})(x_i - \tilde{x}_{i,j,E})}{\Delta x(x_{i-1} - \tilde{x}_{i,j,E})} - \frac{(p_{i,j} - p^{\pm}_{i,,Ej})(x_{i-1} - \tilde{x}_{i,j,E})}{\Delta x(x_i - \tilde{x}_{i,j,E})}. \tag{27}$$

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

$$(\partial_x p^{\pm})_{i,j,E} \approx -\frac{(p_{i+2,j} - p^{\pm}_{i,j,E})(x_{i+1} - \tilde{x}_{i,j,E})}{\Delta x(x_{i+2} - \tilde{x}_{i,j,E})} + \frac{(p_{i+1,j} - p^{\pm}_{i,,Ej})(x_{i+2} - \tilde{x}_{i,j,E})}{\Delta x(x_{i+1} - \tilde{x}_{i,j,E})}. \tag{28}$$

The same discretization holds for the y-derivative. The formulas (27) and (28) 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 [13].

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 (27) and (28) in the jump condition (25), 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 (23).

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

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

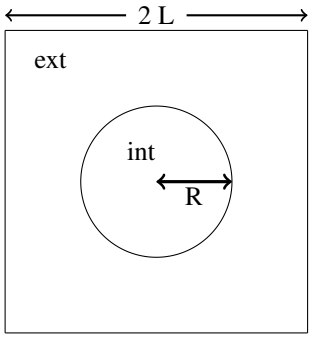


Figure 5: Test case of the static bubble with parasitic oscillations.

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

6 Numerical validation

6.1 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. Due to 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 (24) cause small errors in the resolution of the pressure equation. 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.

6.1.1 Comparison with the Ghost Fluid and the CSF methods

We use the same parameters as in [6], where a Ghost-Fluid and a CSF method where implemented. The amplitude of the parasitic currents, compared to the results in [6], 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 new 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.

$$\left\{\begin{array}{c} L = 2\ cm, \\ R = 1\ cm, \\ \rho_{int} = 1000\ kg.m^{-3}, \\ \mu_{int} = 10^{-3}\ Pa.s, \\ \rho_{ext} = 1\ kg.m^{-3}, \\ \mu_{ext} = 10^{-5}\ Pa.s, \\ \sigma = 0.1\ N.m^{-1} \end{array}\right. \tag{29}$$

	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}

Table 1: Comparison between the new method and the numerical results obtained in [6] for the ghost-fluid method and the CSF method for parasitic oscillations, at time $t = 1$.

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 [32]. 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 $T = \dfrac{D\mu}{\sigma}$.

$$\begin{cases} L = 1.25\ m, \\ R = 1\ m, \\ \rho_{int} = 1\ kg.m^{-3}, \\ \mu_{int} = 10^{-3}\ Pa.s, \\ \rho_{ext} = 1\ kg.m^{-3}, \\ \mu_{ext} = 10^{-3}\ Pa.s, \\ \sigma = 0.00012\ N.m^{-1} \end{cases} \tag{30}$$

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.

Δx	error L^∞ for [32]	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}

Table 2: Numerical results for parasitic oscillations at non-dimensional time $t^* = 250$ for [32] and our method.

6.2 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 6.

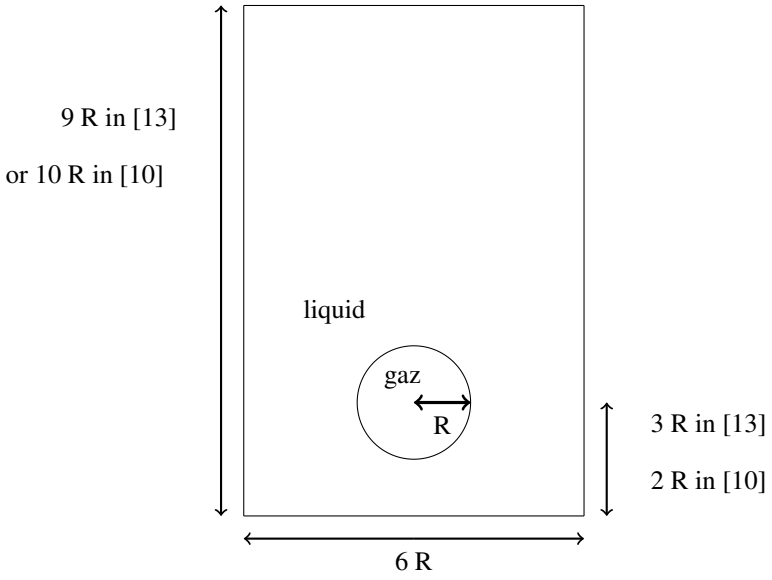


Figure 6: Initial fluid domain for the test case of the rising bubble in water in the references [13] and [10]

6.2.1 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 [13]. The value of the physical parameters are

$$\begin{cases} R = 1/300\ m \text{ (small bubble)} \\ R = 1/3m \text{ (large bubble)} \\ \rho_{water} = 1000\ kg/m^3, \\ \mu_{water} = 1.137 \times 10^{-3}\ kg/ms, \\ \rho_{air} = 1.226\ kg/m^3, \\ \mu_{air} = 1.78 \times 10^{-5}\ kg/ms, \\ \sigma = 0.0728\ kg/s^2 \\ g = -9.8m/s^2 \end{cases} \tag{31}$$

We consider two cases: a small bubble with $R = 1/300m$ and a large one $R = 1/3m$. 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 is plotted at times t= 0.,0.02,0.035, 0.05 in Figure 7. The interface of the large bubble is plotted at times t= 0.,0.2,0.35, 0.5 in Figure 8. Our numerical results are in good agreements with [13].

6.2.2 Comparison with SPH [10] and the level-set method [17]

This test case is taken from [10], and inspired from a test case presented in [17]. 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 6.

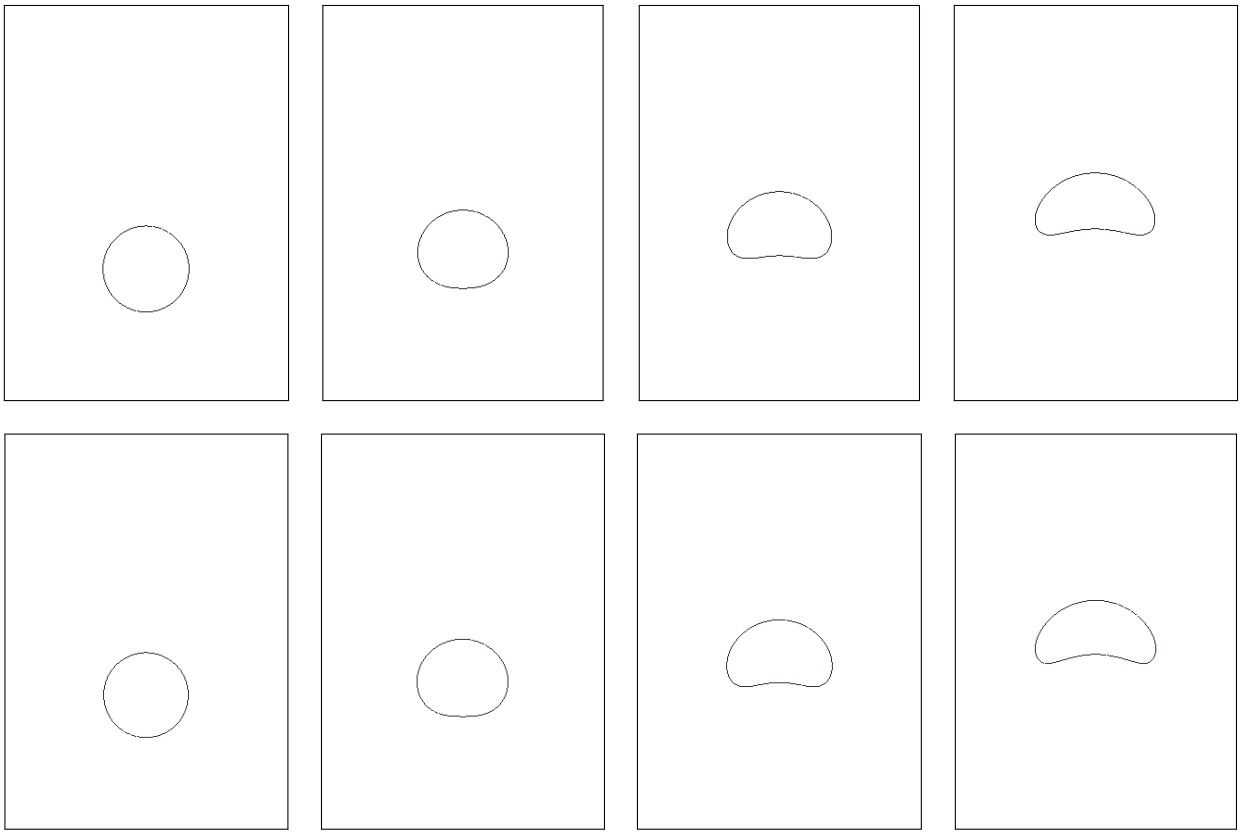


Figure 7: Evolution of the interface for the small bubble test case, resolution 80×120 (top) and 160×240 (bottom) .

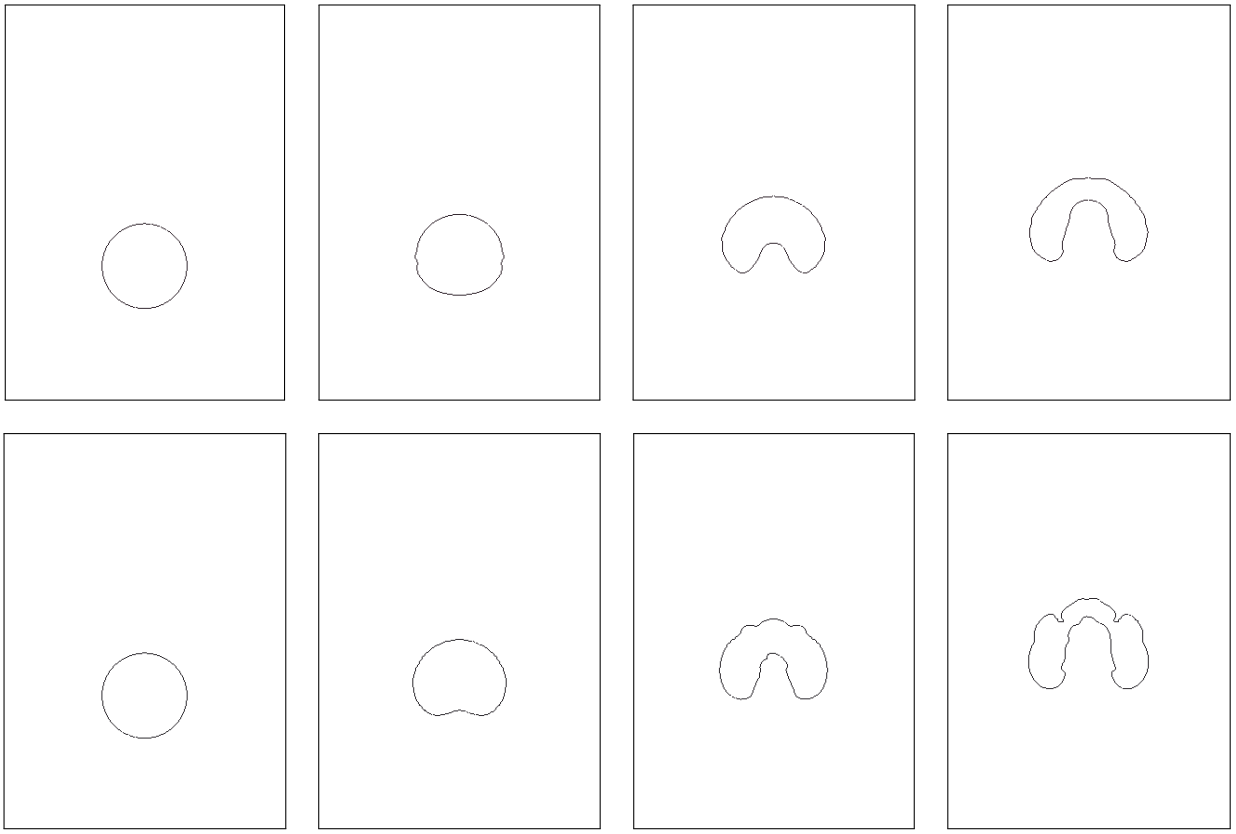


Figure 8: Evolution of the interface for the large bubble test case, resolution 80×120 (top) and 160×240 (bottom) .

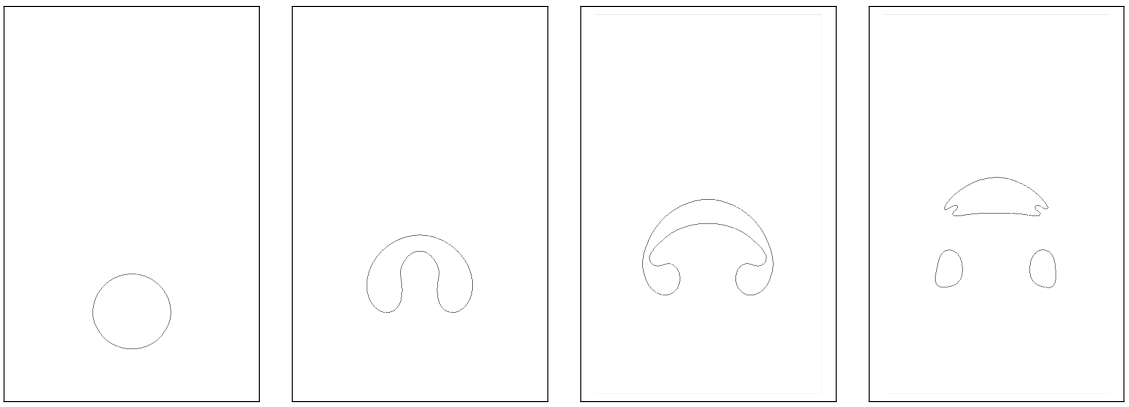


Figure 9: Evolution of the interface for the bubble test case from [10], resolution 120×200.

The value of the physical parameters are

$$\begin{cases} R = 0.025\ m \\ \rho_{water} = 1000\ kg/m^3, \\ \mu_{water} = 1.137 \times 10^{-3}\ kg/ms, \\ \rho_{air} = 1.226\ kg/m^3, \\ \mu_{air} = 1.78 \times 10^{-5}\ kg/ms, \\ \sigma = 0.0728\ kg/s^2 \\ g = -9.8 m/s^2 \end{cases} \tag{32}$$

The evolution of the interface is plotted on Figure 9, for 120×200 grid points. We observe that the interface deforms in a way similar to the results in [10]. We impose a curvature threshold $1/h$ to allow the bubble break-up.

6.3 Collapse of a water column (dam break)

This test case is studied in [22] and [3], and based on experiments conducted in [18]. 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$\times$10 cm. The physical constants are the same than for the rising bubble (§6.1.1). For more details, we refer the reader to [22]. We present in Figure 10 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 11 present the temporal evolution of the water front, compared to the experimental results [18], to Ghost-Fluid results, and to the conservative method of Raessi and Pitsch [22]. We observe that the front propagation is in agreements with the experimental results and the results of the conservative method [22]. It 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 11 and has been reported in [22].

7 Conclusion

We have developed a new method on Cartesian grids for the simulation of incompressible flows with large density ratios, based on a sharp resolution of the pressure term. This Cartesian scheme uses additional unknowns located on the interface to discretize with second-order accuracy the jump conditions across the interface. The viscous terms are treated with a regularizing approach which allows to eliminate terms in the jump conditions without damaging the accuracy of the results. Numerical results show that this new method leads to more accurate and stable results than reference methods. In the same time, the resulting numerical scheme remains simple to implement: it simply amounts to modifying the stencil of the pressure equation for irregular grid points by adding one additional point. Future works include an extension of the method to three-dimensional problems, possibly with interactions with solids, as for instance in [16]. We also aim to study in details the effects of the reinitialization procedure presented in [15].

References

[1] D. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *J. Comput. Phys.*, 148:2–22, 1999.

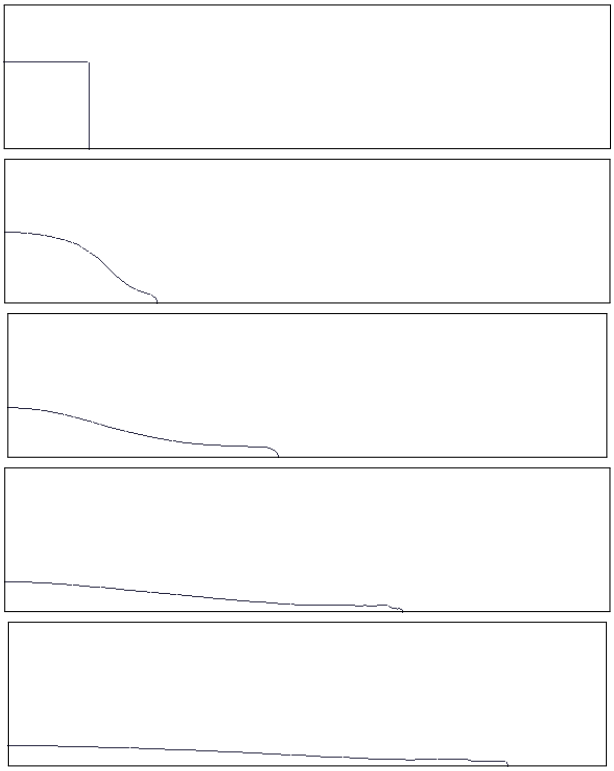


Figure 10: Evolution of the interface for the dam break problem at non-dimensional times $T = t\sqrt{g/H} = 0, 1, 2, 3, 4$.

[2] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.*, 100(2):335–354, 1992.

[3] V. Le Chenadec and H. Pitsch. A monotonicity preserving conservative sharp interface flow solver for high density ratio two-phase flows. *J. Comput. Phys.*, 249:185–203, 2013.

[4] A.J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.

[5] M. Cisternino and L. Weynans. A parallel second order Cartesian method for elliptic interface problems. *Commun. Comput. Phys.*, 12(5):1562–1587, 2012.

[6] F. Couderc. *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, 2007.

[7] O. Desjardins and H. Pitsch. A spectrally refined interface approach for simulating multiphase flows. *J. Comput. Phys.*, 228:1658–1677, 2009.

[8] A. duChene, C. Min, and F. Gibou. Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes. *J. Sci. Comput.*, 35:114–131, 2008.

[9] C. Galusinski and P. Vigneaux. On stability condition for bifluid flows with surface tension: Application to microfluidics. *J. Comput. Phys.*, 227:6140–6164, 2008.

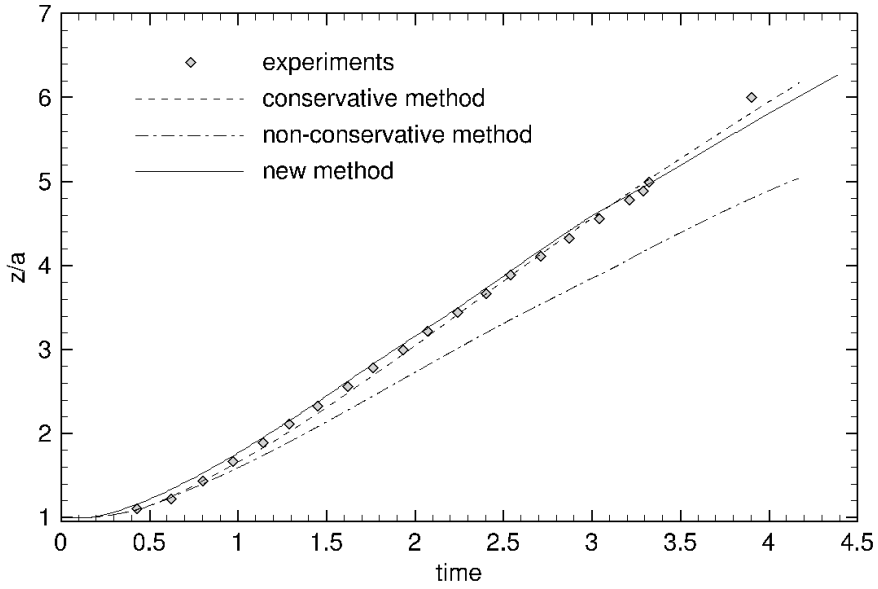


Figure 11: Evolution of the front of propagation: comparison between experimental date and several numerical methods: the Ghost Fluid method (non-conservative method), the conservative method of Raessi and Pitsch and our new method, The dimensionless location of the front $\frac{z}{a}$ is plotted as a function of the dimensionless time $T = t\sqrt{g/H}$.

[10] N. Grenier, M. Antuono, A. Colagrossi, D. Le Touze, and B. Alessandrini. An hamiltonian interface sph formulation for multi-fluid and free-surface flows. *J. Comput. Phys.*, 228:8380–8393, 2009.

[11] M. Herrmann. The influence of density ratio on the primary atomization of a turbulent jet in crossflow. *Proc. Combust. Inst.*, 33:2079–2088, 2011.

[12] X.Y. Hu, B. C. Khoo, N. A. Adams, and F. L. Huang. A conservative interface method for compressible flows. *J. Comput. Phys.*, 219:553–578, 2006.

[13] Myungjoo Kang, Ronald P. Fedkiw, and Xu-Dong Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15(3):323–360, 2000.

[14] X.-D. Liu, R. P. Fedkiw, and M. Kang. A boundary capturing method for poisson's equation on irregular domains. *J. Comput. Phys.*, 160:151–178, 2000.

[15] F. Luddens, M. Bergmann, and L. Weynans. Enablers for high-order level set methods in fluid mechanics. *Int. J. Numer. Meth. Fluids*, 79:654–675, 2015.

[16] Bergmann M., Hovnanian J., and Iollo A. An accurate cartesian method for incompressible flows with moving boundaries. *Comm. Comput. Phys.*, 15:1266–1290, 2014.

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

[18] J. C. Martin and W. J. Moyce. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philos. Trans. R. Soc. London, Ser. A*, 244:312–324, 1952.

[19] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, and A. von Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227(10):4825 – 4852, 2008.

[20] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[21] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamiltonâ jacobi formulations. *J. Comput. Phys.*, 79(12), 1988.

[22] M. Raessi and H. Pitsch. Consistent mass and momentum transport for simulating incompressible interfacial flows with large density ratios using the level set method. *Computers and Fluids*, 63:70–81, 2012.

[23] C.M. Rhie and W.L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21(11):1525–1532, November 1983.

[24] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.*, 29:867–884, 1992.

[25] M. Rudman. A volume-tracking method for computing incompressible multifluid flows with large density variations. *Int. J. Numer. Meth. Fluids*, 28:357–378, 1998.

[26] G. Russo and P. Smereka. A remark on computing distance functions. *J. Comput. Phys.*, 163:51–67, 2000.

[27] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge, UK, 1999.

[28] J. A. Sethian. Evolution, implementation, and application of level set and fast marching methods for advancing fronts. *J. Comput. Phys.*, 169:503–555, 2001.

[29] J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Applied Mathematics*, 93:1591–1595, 1996.

[30] Guang shan Jiang, Chi-Wang Shu, and In L. Efficient implementation of weighted eno schemes. *J. Comput. Phys*, 126:202–228, 1995.

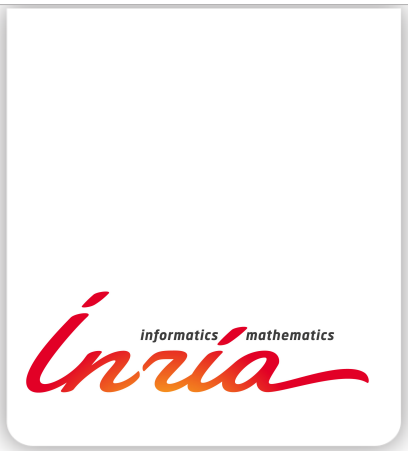
[31] M. Sussman and E. G. Puckett. A coupled level set and volume of fluid method for computing 3d and axisymmetric incompressible two-phase flows. *J. Comput. Phys.*, 162:301–337, 2000.

[32] M. Sussman, K. M. Smith, M. Y. Hussaini, M. Ohta, and R. Zhi-Wei. A sharp interface method for incompressible two-phase flows. *J. Comput. Phys.*, 221:469–505, 2007.

[33] S. Tanguy, T. Menard, and A. Berlemont. A level set method for vaporizing two-phase flows. *J. Comput. Phys.*, 221:837–853, 2007.

[34] R. Temam. Sur l'approximation de la solution des equations de navier-stokes par la méthode des pas fractionnaires ii. *Archiv. Rat. Mech. Anal.*, 32:377–385, 1969.

[35] Y.-H., L.-T. Cheng, S. Osher, and H.-K. Zhao. Fast sweeping algorithms for a class of hamilton-jacobi equations. *SIAM J. Numer. Anal.*, 41:673–694, 2003.



RESEARCH CENTRE
BORDEAUX – SUD-OUEST

351, Cours de la Libération
Bâtiment A 29
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399