



HAL
open science

On the computation of viscous terms for incompressible two-phase flows with Level Set/Ghost Fluid Method

Benjamin Lalanne, Lucia Rueda Villegas, Sébastien Tanguy, Frédéric Risso

► To cite this version:

Benjamin Lalanne, Lucia Rueda Villegas, Sébastien Tanguy, Frédéric Risso. On the computation of viscous terms for incompressible two-phase flows with Level Set/Ghost Fluid Method. *Journal of Computational Physics*, 2015, vol. 301, pp.289-307. 10.1016/j.jcp.2015.08.036 . hal-01308135

HAL Id: hal-01308135

<https://hal.science/hal-01308135v1>

Submitted on 27 Apr 2016

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.



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 14504

To link to this article : DOI : 10.1016/j.jcp.2015.08.036
URL : <http://dx.doi.org/10.1016/j.jcp.2015.08.036>

To cite this version : Lalanne, Benjamin and Rueda Villegas, Lucia and Tanguy, Sébastien and Risso, Frédéric *On the computation of viscous terms for incompressible two-phase flows with Level Set/Ghost Fluid Method*. (2015) Journal of Computational Physics, vol. 301. pp.289-307. ISSN 0021-9991

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

On the computation of viscous terms for incompressible two-phase flows with Level Set/Ghost Fluid Method

Benjamin Lalanne, Lucia Rueda Villegas, Sébastien Tanguy*, Frédéric Risso

Institut de Mécanique des Fluides de Toulouse, Université de Toulouse and CNRS, Allée Camille Soula, 31400 Toulouse, France

A B S T R A C T

In this paper, we present a detailed analysis of the computation of the viscous terms for the simulation of incompressible two-phase flows in the framework of Level Set/Ghost Fluid Method when viscosity is discontinuous across the interface. Two pioneering papers on the topic, Kang et al. [10] and Sussman et al. [26], proposed two different approaches to deal with viscous terms. However, a definitive assessment of their respective efficiency is currently not available. In this paper, we demonstrate from theoretical arguments and confirm from numerical simulations that these two approaches are equivalent from a continuous point of view and we compare their accuracies in relevant test-cases. We also propose a new intermediate method which uses the properties of the two previous methods. This new method enables a simple implementation for an implicit temporal discretization of the viscous terms. In addition, the efficiency of the Delta Function method [24] is also assessed and compared to the three previous ones, which allow us to propose a general overview of the accuracy of all available methods. The selected test-cases involve configurations wherein viscosity plays a major role and for which either theoretical results or experimental data are available as reference solutions: simulations of spherical rising bubbles, shape-oscillating bubbles and deformed rising bubbles at low Reynolds numbers.

Keywords:

Level Set
Ghost Fluid Method
Two-phase flows
Viscosity jump condition

1. Introduction

In pioneering works on simulation of incompressible two-phase flows with interface tracking or interface capturing algorithms, discontinuities of pressure field, density field or viscosity field were tackled by smoothing the singular terms across the interface on a finite layer of about 2 or 3 grid cells thick. This kind of approaches has been proposed in the framework of *Front-Tracking* algorithms [1,30], for *Volume-of-Fluid* methods [6,12], and for *Level-Set* methods [24]. In the literature, they are referred to as *Continuum Surface Force methods* or *Delta Function methods*. Such methods still provide satisfactory results in many situations. Nevertheless, they suffer from some drawbacks such as the presence of intense spurious currents at high capillary-numbers, which can induce strong numerical instabilities. Also, they have to face the tricky problem of describing a surface as a volume of small thickness. A few years ago, a new class of method appeared in the literature, where singular terms across an interface are treated as jump conditions, which prevents from interface thickening as well as it reduces spurious currents. Significant contributions to this new approach, well known as *Sharp Interface method*, can be found in [21] for *Front-Tracking* methods, in [10,15,26] for *Level-Set* methods, and [5] *Volume-of-Fluid* methods. In [26], a *Level-Set* method was coupled with a *Volume-of-Fluid* method, but the way to impose the jump conditions depended only on the

* Corresponding author. Tel.: +33 5 34 32 28 08; fax: +33 5 34 32 28 99.
E-mail address: tanguy@imft.fr (S. Tanguy).

Level-Set function – the volume fraction being just used to improve interface advection and to compute interface curvature. Most of these papers emphasize on improvements resulting from a sharp description of the surface tension, but they are not conclusive about the effectiveness of Sharp Interface methods for the discretization of viscous terms. However, Kang et al. [10] proposed a new discretization of the viscous terms based on the general principle of the Ghost Fluid Method [4,15]. The main feature of this new approach consists in computing the contraction of the divergence of the viscous stress tensor in a Laplacian operator of the velocity components $\mu\Delta\vec{V}$ rather than its full expression (with μ the dynamic viscosity and \vec{V} the velocity field). This contraction is only possible for incompressible flow when viscosity is uniform; it is hence apparently not suited to compute the viscous terms on grid cells that are crossed by an interface between two different fluids. To solve this issue for two-phase flows, once $\mu\Delta\vec{V}$ has been computed, the authors propose to add explicitly the jump of the normal viscous stresses at the interface, resulting from the jump of viscosity, to the pressure jump while solving the Poisson equation for the pressure. Another approach to compute the viscous terms in the framework of *Sharp Interface* methods, inspired of a previous work [15], was proposed by Sussman et al. [26]. In a similar way of what it is usually done in *Continuum Surface Force* methods, the full expression of the divergence of the viscous stress tensor $\nabla\cdot(2\mu\mathbf{D})$ is discretized, the viscous stress tensor being defined by $2\mu\mathbf{D}$ for a Newtonian fluid, with \mathbf{D} the rate-of-deformation tensor. In this approach, contrary to previous work [10], it seems that the contribution of the normal viscous stresses has not to be added explicitly to the pressure jump condition at the interface. Valuable numerical validations of this method are proposed in [26]. However, whether these two approaches are formally equivalent or not remained an open question. In particular, the fact that jump of the normal viscous stresses has to be explicitly added to the pressure jump in [10] and not in [26] still required to be justified.

In this paper, we focus on the computation of viscous terms for incompressible two-phase flows using *Level-Set* methods.

In the framework of *Sharp-Interface* methods, we shall demonstrate that the two approaches introduced in [10] and [26] are equivalent from a theoretical point of view, even if their numerical implementation is quite different.

From this original demonstration, a new intermediate method is proposed, which allows to make more easily the implicit temporal discretization of the viscous terms as the two methods proposed in [10] and [26].

To complete this survey of numerical methods, we also discuss the consistency of these three *Sharp-Interface* methods with the *Delta-Function* method that is based on a *Continuum Surface Force* approach.

Section 2 describes the formalism of this set of four methods and proves their equivalence. Then, Section 3 proposes three numerical benchmarks that are particularly relevant to assess the reliability of these methods and compare their accuracy: spherical rising bubbles, damped shape oscillations of a bubble, and deformed rising bubbles at low Reynolds number.

2. Equations & numerical methods

2.1. Equation of motion for two-phase flows, jump condition and singular source term

Let us consider the Navier–Stokes equations for a single-phase incompressible flow; the following two equations express respectively momentum and mass conservation:

$$\rho \frac{D\vec{V}}{Dt} = -\nabla p + \nabla \cdot (2\mu\mathbf{D}) + \rho\vec{g}, \quad (1)$$

$$\nabla \cdot \vec{V} = 0, \quad (2)$$

where \vec{V} is the velocity field, $\frac{D}{Dt}$ is the Lagrangian derivative, ρ is the fluid density, μ is the fluid viscosity, p is the pressure field, \vec{g} is the gravity acceleration, and \mathbf{D} is the rate-of-deformation tensor defined as:

$$\mathbf{D} = \frac{\nabla\vec{V} + \nabla\vec{V}^T}{2}. \quad (3)$$

For incompressible flows with constant viscosity, it can be easily shown that:

$$\nabla \cdot (2\mu\mathbf{D}) = \mu\Delta\vec{V}, \quad (4)$$

which leads to the following simplified expression for Navier–Stokes equation:

$$\rho \frac{D\vec{V}}{Dt} = -\nabla p + \mu\Delta\vec{V} + \rho\vec{g}. \quad (5)$$

Let us now consider an incompressible two-phase flow, the *whole-domain-formulation* [1,23] leads to the following expression that includes the capillary forces:

$$\rho \frac{D\vec{V}}{Dt} = -\nabla p + \nabla \cdot (2\mu\mathbf{D}) + \sigma\kappa\vec{n}\delta_\Gamma + \rho\vec{g}, \quad (6)$$

where σ is the surface tension, κ is the interface curvature, \vec{n} is the normal vector at the interface and δ_Γ is a multidimensional Dirac distribution localized at the interface. If no phase change occurs, eq. (2) stays valid. As density and viscosity can vary across interface, they can be defined as:

$$\rho = \rho_{gas} + (\rho_{liq} - \rho_{gas})H_\Gamma = \rho_{gas} + [\rho]_\Gamma H_\Gamma, \quad (7)$$

$$\mu = \mu_{gas} + (\mu_{liq} - \mu_{gas})H_\Gamma = \mu_{gas} + [\mu]_\Gamma H_\Gamma, \quad (8)$$

where H_Γ is a Heaviside distribution, equal to 1 in the liquid phase and equal to 0 in the gas phase, with the following definitions for density and viscosity jump conditions:

$$[\rho]_\Gamma = \rho_{liq} - \rho_{gas}, \quad (9)$$

$$[\mu]_\Gamma = \mu_{liq} - \mu_{gas}. \quad (10)$$

Moreover, considering a divergence-free condition on the velocity field, the following unsteady advection equation for the density field is obtained from mass conservation:

$$\frac{D\rho}{Dt} = 0. \quad (11)$$

From that equation, we can deduce the usual propagation equation for interface motion that can be used with Level Set methods:

$$\frac{D\phi}{Dt} = 0, \quad (12)$$

where ϕ is a signed distance function from the interface. Note that another similar equation can be derived for Volume Of Fluid methods:

$$\frac{DF}{Dt} = 0, \quad (13)$$

where F is the local volume fraction of the considered phase in a computational cell.

The balance of normal stresses at the interface involves a jump of pressure and a jump of normal viscous stresses:

$$[p]_\Gamma = \sigma\kappa + 2[\mu]_\Gamma \frac{\partial V_n}{\partial n}. \quad (14)$$

Let us remark now that the viscous-stress tensor can be split into two parts:

$$\nabla \cdot (2\mu\mathbf{D}) = \mu\nabla \cdot (2\mathbf{D}) + 2\mathbf{D} \cdot \nabla\mu. \quad (15)$$

If viscosity is a piecewise constant, by using the definition of the Heaviside derivative, it can be shown that:

$$\nabla\mu = [\mu]_\Gamma \vec{n} \delta_\Gamma. \quad (16)$$

Finally, we observe that the previous splitting (15) is a decomposition of the global viscous term in a continuous part and a singular part (zero everywhere except at the interface):

$$\nabla \cdot (2\mu\mathbf{D}) = \mu\Delta\vec{V} + 2[\mu]_\Gamma \mathbf{D} \cdot \vec{n} \delta_\Gamma \quad (17)$$

This highlights that the contribution of the viscous terms on the pressure jump is already included in the left-hand side of eq. (17). Thus, we can suppose that computing the divergence of the overall viscous stress tensor allows including the contribution of viscous effects in the pressure jump. A more rigorous proof of that assertion is provided in Section 2.4 of this paper.

To summarize, the momentum equation can be defined in a *whole-domain formulation* leading to eq. (6) with a density and a viscosity field defined respectively by eqs. (7) and (8), or in a *jump-condition formulation* leading to eq. (5) with additional jump conditions (9), (10), (14).

2.2. About the link between jump conditions and singular source terms

Let us consider a volume Ω , bounded by a surface Σ , and defined as $\Omega_1 \cup \Omega_2$, where the two subdomains Ω_1 and Ω_2 are separated by a closed surface Γ with $\Gamma \cap \Sigma = \emptyset$, as illustrated in Fig. 1. We define \vec{n}_1 and \vec{n}_2 as the normal vectors at the surface Γ pointing respectively towards Ω_1 and Ω_2 .

Let us now consider a piecewise continuous function ψ which respects the Poisson equation,

$$\Delta\psi = 0 \quad (18)$$

inside both subdomains Ω_1 and Ω_2 , with the following jump condition at the interface:

$$[\psi]_\Gamma = a(\vec{x}_\Gamma), \quad (19)$$

where $a(\vec{x}_\Gamma)$ is a function of the interface coordinate \vec{x}_Γ . The jump condition (19) involves that the following surface integral is equal to zero

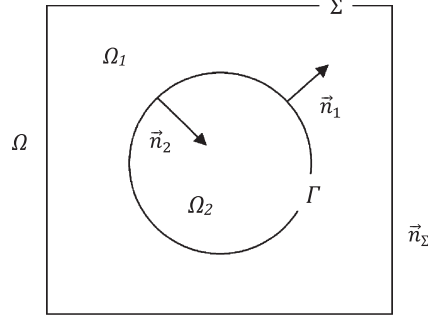


Fig. 1. Sketch of a two-phase flow domain $\Omega_1 \cup \Omega_2$ with an immersed interface Γ .

$$\iint_{\Gamma} ([\psi]_{\Gamma} - a(\bar{x}_{\Gamma})) \bar{n}_2 dS = \vec{0} \quad (20)$$

Considering a homogeneous Dirichlet boundary condition on the external boundary Σ of the computational field, it can be easily demonstrated that

$$\iint_{\Sigma} \psi \bar{n}_{\Sigma} dS = 0, \quad (21)$$

where \bar{n}_{Σ} is the normal vector pointing outwards the external boundary Σ . Thus

$$\iint_{\Sigma} \psi \bar{n}_{\Sigma} dS + \iint_{\Gamma} ([\psi]_{\Gamma} - a(\bar{x}_{\Gamma})) \bar{n}_2 dS = 0. \quad (22)$$

Moreover, by remarking that

$$\iint_{\Gamma} ([\psi]_{\Gamma} - a(\bar{x}_{\Gamma})) \bar{n}_2 dS = \iint_{\Gamma} \psi \bar{n}_2 dS + \iint_{\Gamma} \psi \bar{n}_1 dS - \iint_{\Gamma} a(\bar{x}_{\Gamma}) \bar{n}_2 dS \quad (23)$$

we obtain the following equality:

$$\iint_{\Sigma} \psi \bar{n}_{\Sigma} dS + \iint_{\Gamma} \psi \bar{n}_2 dS + \iint_{\Gamma} \psi \bar{n}_1 dS = \iint_{\Gamma} a(\bar{x}_{\Gamma}) \bar{n}_2 dS. \quad (24)$$

Eq. (24) can also be written as

$$\iiint_{\Omega_1} \nabla \psi d\Omega + \iiint_{\Omega_2} \nabla \psi d\Omega = \iiint_{\Omega} \nabla \psi d\Omega = \iiint_{\Omega} a(\bar{x}_{\Gamma}) \bar{n}_2 \delta_{\Gamma} d\Omega, \quad (25)$$

which leads to the following local formulation:

$$\nabla \psi = a(\bar{x}_{\Gamma}) \bar{n}_2 \delta_{\Gamma}. \quad (26)$$

Finally, after differentiation, we obtain:

$$\Delta \psi = \nabla \cdot (a(\bar{x}_{\Gamma}) \bar{n}_2 \delta_{\Gamma}). \quad (27)$$

We have therefore shown that adding the following source term $\nabla \cdot (a(\bar{x}_{\Gamma}) \bar{n}_2 \delta_{\Gamma})$ to the right-hand side of a Poisson equation imposes a jump condition $[\psi]_{\Gamma} = a(\bar{x}_{\Gamma})$ on its solution ψ . This is also valid for a Poisson equation like eq. (18) with a right-hand side different from zero.

Moreover, following a similar approach, it can also be demonstrated that a scalar field that is solution of the following equation

$$\Delta \psi = b(\bar{x}_{\Gamma}) \delta_{\Gamma}, \quad (28)$$

with $b(\bar{x}_{\Gamma})$ another function depending on interface coordinate, will also fulfill the following jump condition:

$$[\bar{n}_2 \cdot \nabla \psi] = b(\bar{x}_{\Gamma}). \quad (29)$$

More accurate demonstrations on the derivation of jump conditions can be found in [14].

2.3. Projection method for whole-domain formulation and jump-condition formulation

Let us consider now a first-order explicit projection method for the single-phase incompressible Navier–Stokes equation. It consists in prescribing a first intermediate velocity field which does not respect the divergence-free condition (predictor step):

$$\vec{V}_1^* = \vec{V}^n - \Delta t \left((\vec{V}^n \cdot \nabla) \vec{V}^n - \nu \Delta \vec{V}^n - \vec{g} \right), \quad (30)$$

with ν the kinematic viscosity.

Next, a Poisson equation must be solved to determine the pressure field:

$$\Delta p^{n+1} = \rho^{n+1} \frac{\nabla \cdot \vec{V}_1^*}{\Delta t}. \quad (31)$$

Then, in a last step, the final divergence-free velocity field can be computed using the pressure field determined at the previous step:

$$\vec{V}^{n+1} = \vec{V}_1^* - \Delta t \frac{\nabla p^{n+1}}{\rho^{n+1}}. \quad (32)$$

Applying the projection method to an incompressible two-phase flow is less straightforward. Depending on whether the *whole-domain formulation* or the *jump-condition formulation* is used, the intermediate velocity in the first step of the projection method has to be different.

Let us first consider a *whole-domain formulation*. An explicit temporal discretization of the capillary and viscous terms leads quite naturally to the following predictor step

$$\vec{V}_2^* = \vec{V}^n - \Delta t \left((\vec{V}^n \cdot \nabla) \vec{V}^n - \frac{\nabla \cdot (2\mu \mathbf{D}^n)}{\rho^{n+1}} - \frac{\sigma \kappa \vec{n} \delta_\Gamma}{\rho^{n+1}} - \vec{g} \right), \quad (33)$$

with the corresponding Poisson equation

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_2^*}{\Delta t} = \nabla \cdot \left(\vec{V}^n - \Delta t \left((\vec{V}^n \cdot \nabla) \vec{V}^n - \frac{\nabla \cdot (2\mu \mathbf{D}^n)}{\rho^{n+1}} - \frac{\sigma \kappa \vec{n} \delta_\Gamma}{\rho^{n+1}} - \vec{g} \right) \right). \quad (34)$$

Next, the correction step gives:

$$\vec{V}^{n+1} = \vec{V}_2^* - \Delta t \frac{\nabla p^{n+1}}{\rho^{n+1}}. \quad (35)$$

Let us now consider the application of the projection method to a *jump-condition formulation* of the Navier–Stokes equation. We have to solve eq. (5), so the first step of the projection method is similar to eq. (30) for a single-phase flow. As we want to impose the appropriate jump condition at the interface on the pressure field (eq. (14)), we can show, using the results presented in Section 2.2, that it can be done by solving the following Poisson equation for the pressure with a right-hand side split into a continuous part and a singular part:

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_1^*}{\Delta t} + \nabla \cdot \left(\frac{(\sigma \kappa + 2[\mu]_\Gamma \frac{\partial V_n}{\partial n}) \vec{n} \delta_\Gamma}{\rho^{n+1}} \right). \quad (36)$$

Using the splitting of the viscous term given in eq. (17) and noting that $\vec{n} \cdot \mathbf{D} \cdot \vec{n} = \frac{\partial V_n}{\partial n}$, we find the following relation between \vec{V}_2^* and \vec{V}_1^* :

$$\vec{V}_2^* = \vec{V}_1^* + \Delta t \frac{(\sigma \kappa + 2[\mu]_\Gamma \frac{\partial V_n}{\partial n}) \vec{n} \delta_\Gamma}{\rho^{n+1}}. \quad (37)$$

The use of eq. (37) allows demonstrating that eq. (34) and eq. (36) are formally identical:

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_2^*}{\Delta t} = \frac{\nabla \cdot \vec{V}_1^*}{\Delta t} + \nabla \cdot \left(\frac{(\sigma \kappa + 2[\mu]_\Gamma \frac{\partial V_n}{\partial n}) \vec{n} \delta_\Gamma}{\rho^{n+1}} \right). \quad (38)$$

Then, by considering eq. (37), we can deduce the following correction step for a *jump-condition formulation*, leading to the same field \vec{V}^{n+1} than in eq. (35):

$$\vec{V}^{n+1} = \vec{V}_1^* - \frac{\Delta t}{\rho^{n+1}} \left(\nabla p^{n+1} - \left(\sigma \kappa + 2[\mu]_\Gamma \frac{\partial V_n}{\partial n} \right) \vec{n} \delta_\Gamma \right). \quad (39)$$

To summarize, we have shown that both a *whole-domain formulation* and a *jump-condition formulation* can be used when applying a projection method to an incompressible two-phase flow. These two methods are formally equivalent, but they

lead to a different definition of the predicted velocity field in the first step of the projection method. In [10], the intermediate velocity field \vec{V}_1^* is used, whereas in [1,3,6,24] it is the intermediate velocity field \vec{V}_2^* . These two methods will be respectively named in the rest of this paper the *Ghost-Fluid Primitive viscous Method* (GFPM) and the *Delta-Function Method* (DFM).

2.4. Projection method for intermediate formulations

Other definitions of the predicted velocity field are possible. For example, we propose the following two definitions, the respective interests of which will be discussed later:

$$\vec{V}_3^* = \vec{V}^n - \Delta t \left((\vec{V}^n \cdot \nabla) \vec{V}^n - \frac{\nabla \cdot (2\mu \mathbf{D}^n)}{\rho^{n+1}} - \vec{g} \right), \quad (40)$$

$$\vec{V}_4^* = \vec{V}^n - \Delta t \left((\vec{V}^n \cdot \nabla) \vec{V}^n - \frac{\nabla \cdot (\mu \nabla \vec{V}^n)}{\rho^{n+1}} - \vec{g} \right). \quad (41)$$

In the same way as in the previous section, they lead to the definition of these two Poisson equations, formally equivalent to eq. (34) and eq. (36):

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_3^*}{\Delta t} + \nabla \cdot \left(\frac{\sigma \kappa \vec{n} \delta_\Gamma}{\rho^{n+1}} \right), \quad (42)$$

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_4^*}{\Delta t} + \nabla \cdot \left(\frac{(\sigma \kappa + [\mu]_\Gamma \frac{\partial V_n}{\partial n}) \vec{n} \delta_\Gamma}{\rho^{n+1}} \right). \quad (43)$$

The following relations have been used to determine eq. (43):

$$\nabla \cdot (2\mu \mathbf{D}) = \nabla \cdot (\mu (\nabla \vec{V} + \nabla^T \vec{V})) = \nabla \cdot (\mu \nabla \vec{V}) + \mu \nabla \cdot \nabla^T \vec{V} + \nabla \mu \cdot \nabla^T \vec{V}, \quad (44)$$

$$\nabla \cdot (2\mu \mathbf{D}) = \nabla \cdot (\mu \nabla \vec{V}) + [\mu]_\Gamma \nabla^T \vec{V} \cdot \vec{n} \delta_\Gamma, \quad (45)$$

since $\nabla \cdot \nabla^T \vec{V} = 0$ for an incompressible flow, and

$$\vec{n} \cdot \nabla^T \vec{V} \cdot \vec{n} = \frac{\partial V_n}{\partial n}. \quad (46)$$

Then, the two final corrector steps, respectively suitable with eq. (40) and eq. (41), are:

$$\vec{V}^{n+1} = \vec{V}_3^* - \frac{\Delta t}{\rho^{n+1}} (\nabla p^{n+1} - \sigma \kappa \vec{n} \delta_\Gamma), \quad (47)$$

$$\vec{V}^{n+1} = \vec{V}_4^* - \frac{\Delta t}{\rho^{n+1}} \left(\nabla p^{n+1} - (\sigma \kappa + [\mu]_\Gamma \frac{\partial V_n}{\partial n}) \vec{n} \delta_\Gamma \right). \quad (48)$$

The intermediate velocity field \vec{V}_3^* has already been used in Refs. [5,21,26] in the framework of *Sharp Interface methods*. In the rest of this paper, the *Ghost-Fluid Conservative viscous Method* (GFCM) will refer to the numerical method proposed in [26]. Finally, the use of \vec{V}_4^* leads to an original numerical method to deal with the viscous terms, that we name the *Ghost-Fluid Semi-Conservative viscous Method* (GFSCM).

The four methods DFM, GFPM, GFCM and GFSCM described above are therefore theoretically equivalent, but they lead to different numerical approximations. The next objective of this paper is to discuss about their respective interests and numerical accuracy.

2.5. Spatial discretization of surface tension

When a *whole-domain formulation* [3,6,12,24,30] is used, the surface-tension singular term is accounted for when \vec{V}_2^* is computed. The intermediate velocity field is next differentiated to compute the right-hand side of the pressure Poisson equation. Therefore, it is well suited to a smooth discretization of the surface-tension term that involves a fictitious interface thickness. Its application to a sharp-interface description would be trickier. This approach is that of the *Delta-Function Method*, known as *Continuum Surface Force* method. In this framework, the numerical discretization of discontinuous terms makes use of the following regularized smoothed Heaviside function defined from the Level-Set function ϕ :

$$H_\varepsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\varepsilon \\ \frac{1}{2} \left(1 + \frac{\phi}{\varepsilon} + \frac{\sin(\frac{\pi\phi}{\varepsilon})}{\pi} \right) & \text{if } |\phi| \leq \varepsilon \\ 1 & \text{if } \phi > +\varepsilon, \end{cases} \quad (49)$$

where ε is the fictitious interface thickness, which is equal to 2 or 3 cell sizes. The surface-tension term can then be computed using the following approximation:

$$\sigma \kappa \bar{n} \delta_\Gamma \cong \sigma \kappa \nabla H_\varepsilon. \quad (50)$$

At the opposite, a projection method using a *jump-condition formulation* with \bar{V}_1^* as intermediate velocity field is well suited for *Sharp-Interface methods*. Thanks to this formulation, artificial smoothing of interface singularities can be avoided, allowing a sharp-interface description. It has been used in the framework of *Ghost-Fluid Method* in [15], where authors propose a first-order discretization to impose a jump condition to the solution as well as to the normal derivative of the solution when solving the Poisson equation

$$\nabla \cdot (\beta \nabla u) = f, \quad (51)$$

i.e. with the following interface jump conditions

$$[u] = a(\bar{x}_\Gamma) \quad \text{and} \quad \left[\beta \frac{\partial u}{\partial n} \right] = b(\bar{x}_\Gamma).$$

Considering that capillary effects can be computed as a jump condition on pressure, an original *Sharp-Interface method* for solving the Navier–Stokes equation in the framework of *Level-Set methods* was introduced in [10]. This method has been widely used to perform direct numerical simulations of two-phase flows [16,26–29]. Let us now briefly summarize this approach.

The discretization of a 2D Poisson equation as eq. (51), which involves a discontinuity of the solution and of its normal derivative at the interface, writes:

$$\frac{\beta_{i+1/2j} \left(\frac{u_{i+1j} - u_{ij}}{\Delta x} \right) - \beta_{i-1/2j} \left(\frac{u_{ij} - u_{i-1j}}{\Delta x} \right)}{\Delta x} + \frac{\beta_{ij+1/2} \left(\frac{u_{ij+1} - u_{ij}}{\Delta y} \right) - \beta_{ij-1/2} \left(\frac{u_{ij} - u_{ij-1}}{\Delta y} \right)}{\Delta y} = f_{ij} + g_{ij} + h_{ij}, \quad (52)$$

where interpolations of the diffusion coefficient β on the cell borders are computed as the harmonic average of its value β^+ in the region where the Level Set function is positive and its value β^- in the region where the Level Set function is negative; for example, if the interface crosses a mesh segment between x_i and x_{i+1} :

$$\beta_{i+1/2j} = \frac{\beta^+ \beta^-}{\beta^+ \theta + \beta^- (1 - \theta)}, \quad (53)$$

with

$$\theta = \frac{|\phi_{i+1j}|}{|\phi_{i+1j}| + |\phi_{ij}|}. \quad (54)$$

The source terms g_{ij} and h_{ij} in the right-hand side of eq. (52) match with the discontinuities of u and of $\beta \frac{\partial u}{\partial n}$; they can be computed as

$$g_{ij} = g_{ij}^R + g_{ij}^L + g_{ij}^T + g_{ij}^B, \quad (55)$$

$$h_{ij} = h_{ij}^R + h_{ij}^L + h_{ij}^T + h_{ij}^B, \quad (56)$$

where each of these terms are non-zero when an interface crosses the corresponding grid segment, and superscripts R , L , T , B denote respectively right, left, top and bottom sides. Their expressions are:

$$g_{ij}^R = \pm \frac{\beta_{i+1/2j} a_\Gamma}{\Delta x^2} \quad g_{ij}^L = \pm \frac{\beta_{i-1/2j} a_\Gamma}{\Delta x^2} \quad g_{ij}^T = \pm \frac{\beta_{ij+1/2} a_\Gamma}{\Delta y^2} \quad g_{ij}^B = \pm \frac{\beta_{ij-1/2} a_\Gamma}{\Delta y^2}, \quad (57)$$

$$h_{ij}^R = \pm \frac{\beta_{i+1/2j} b_\Gamma \theta}{\beta^\pm \Delta x} \quad h_{ij}^L = \mp \frac{\beta_{i-1/2j} b_\Gamma \theta}{\beta^\pm \Delta x} \quad h_{ij}^T = \pm \frac{\beta_{ij+1/2} b_\Gamma \theta}{\beta^\pm \Delta y} \quad h_{ij}^B = \mp \frac{\beta_{ij-1/2} b_\Gamma \theta}{\beta^\pm \Delta y}. \quad (58)$$

It is noteworthy that an analogy can be established by using the results demonstrated in Section 2.2 and the discretization described in this section. For sake of simplicity, let us take a diffusion coefficient β equal to unity. In Section 2.2, it has been shown that solving

$$\Delta u = f \quad \text{with} \quad [u] = a(\bar{x}_\Gamma) \quad \text{and} \quad \left[\frac{\partial u}{\partial n} \right] = b(\bar{x}_\Gamma)$$

is equivalent to solve

$$\Delta u = f + \nabla \cdot (a(\bar{x}_\Gamma) \bar{n} \delta_\Gamma) + b(\bar{x}_\Gamma) \delta_\Gamma.$$

Comparing this expression with discrete equation (52), we deduce that source terms g_{ij} and h_{ij} , that appear in the framework of the *Ghost-Fluid Method*, are respectively numerical approximations of $\nabla \cdot (a(\bar{x}_\Gamma) \bar{n} \delta_\Gamma)|_{ij}$ and $b(\bar{x}_\Gamma) \delta_\Gamma|_{ij}$:

$$\nabla \cdot (a \bar{n} \delta)|_{ij} = g_{ij} + O(\Delta x) \quad (59)$$

$$b \delta|_{ij} = h_{ij} + O(\Delta x) \quad (60)$$

Such an analogy establishes the link between the *Continuum Surface Force* approach based on numerical approximations of δ , and the *Ghost-Fluid Method* that accounts for source terms in the discretization of Poisson equations.

2.6. Spatial discretization of viscous terms

More details are provided in this section about the spatial discretization of viscous terms for the different methods described in the previous sections. In the following, for sake of simplicity, all developments are presented in 2D with Cartesian coordinates and a uniform grid. The generalization to axisymmetric coordinates or 3D geometry is straightforward.

2.6.1. Conservative viscous formulation

If a conservative viscous formulation $\nabla \cdot (2\mu \mathbf{D})$ is considered, as it is the case in the GFCM or in the DFM, the projection of the divergence of the viscous-stress tensor leads to the following relations:

$$\nabla \cdot (2\mu \mathbf{D})|_x = \frac{\partial}{\partial x} \left(2\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right) \quad (61)$$

$$\nabla \cdot (2\mu \mathbf{D})|_y = \frac{\partial}{\partial x} \left(\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right) + \frac{\partial}{\partial y} \left(2\mu \left(\frac{\partial v}{\partial y} \right) \right) \quad (62)$$

Let us remark that these two relations depend on both velocity components. This is why a fully implicit temporal discretization of viscous terms is more complex when this formulation is used. Suitable approximations of eq. (61) lead to the following spatial discretizations, D_x and D_y denoting the differential operators:

$$(\nabla \cdot (2\mu \mathbf{D})) \cdot \vec{e}_x|_{i+1/2j} \cong D_x(2\mu D_x u)|_{i+1/2j} + D_y(\mu D_y u)|_{i+1/2j} + D_y(\mu D_x v)|_{i+1/2j} \quad (63)$$

$$D_x(2\mu D_x u)|_{i+1/2j} = \frac{2\mu_{i+1j} \left(\frac{u_{i+3/2j} - u_{i+1/2j}}{\Delta x} \right) - 2\mu_{ij} \left(\frac{u_{i+1/2j} - u_{i-1/2j}}{\Delta x} \right)}{\Delta x} \quad (64)$$

$$D_y(\mu D_y u)|_{i+1/2j} = \frac{\mu_{i+1/2j+1/2} \left(\frac{u_{i+1/2j+1} - u_{i+1/2j}}{\Delta y} \right) - \mu_{i+1/2j-1/2} \left(\frac{u_{i+1/2j} - u_{i+1/2j-1}}{\Delta y} \right)}{\Delta y} \quad (65)$$

$$D_y(\mu D_x v)|_{i+1/2j} = \frac{\mu_{i+1/2j+1/2} \left(\frac{v_{i+1j+1/2} - v_{ij+1/2}}{\Delta x} \right) - \mu_{i+1/2j-1/2} \left(\frac{v_{i+1j-1/2} - v_{ij-1/2}}{\Delta x} \right)}{\Delta y} \quad (66)$$

The following spatial discretization for eq. (62) is

$$(\nabla \cdot (2\mu \mathbf{D})) \cdot \vec{e}_y|_{ij+1/2} \cong D_x(\mu D_y u)|_{ij+1/2} + D_x(\mu D_x v)|_{ij+1/2} + D_y(2\mu D_y v)|_{ij+1/2} \quad (67)$$

$$D_x(\mu D_y u)|_{ij+1/2} = \frac{\mu_{i+1/2j+1/2} \left(\frac{u_{i+1/2j+1} - u_{i+1/2j}}{\Delta y} \right) - \mu_{i-1/2j+1/2} \left(\frac{u_{i-1/2j+1} - u_{i-1/2j}}{\Delta y} \right)}{\Delta x} \quad (68)$$

$$D_x(\mu D_x v)|_{ij+1/2} = \frac{\mu_{i+1/2j+1/2} \left(\frac{v_{i+1j+1/2} - v_{ij+1/2}}{\Delta x} \right) - \mu_{i-1/2j+1/2} \left(\frac{v_{i+1/2j} - v_{i-1j+1/2}}{\Delta x} \right)}{\Delta x} \quad (69)$$

$$D_y(2\mu D_y v)|_{ij+1/2} = \frac{2\mu_{ij+1} \left(\frac{v_{ij+3/2} - v_{ij+1/2}}{\Delta y} \right) - 2\mu_{ij} \left(\frac{v_{ij+1/2} - v_{ij-1/2}}{\Delta y} \right)}{\Delta y} \quad (70)$$

This spatial discretization for the conservative viscous approach can be used both in the framework of *Smoothed-Interface methods* (e.g. in the DFM) [3,24] or *Sharp-Interface methods* (e.g. in the GFCM) [26]. The difference lies in the way viscosity is approximated on the border of a cell that is crossed by an interface.

In the DFM, eq. (8) is used with the approximation of the Heaviside function given by eq. (49).

In the GFCM proposed by Sussman et al. [26], the interpolation used to compute viscosity on the cell border is quite similar to that proposed by Liu et al. [15] to interpolate the diffusion coefficient of a Poisson equation (given here by eq. (53)). Thereafter, this kind of interpolation will be used for all the computations that involve the *Ghost-Fluid Method* (i.e. GFCM, GFPM and GFSCM).

2.6.2. Ghost-Fluid Primitive viscous Method

The GFPM of Kang et al. [10] is now briefly described. When this approach is used, the jump condition on the normal viscous stresses is added to the capillary forces when solving the Poisson equation for pressure as shown in eq. (36).

This approach for the viscous term can be summarized by considering the following projections for the Laplacian of the velocity:

$$\mu \Delta \vec{V}|_x = \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \quad (71)$$

$$\mu \Delta \vec{V}|_y = \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right). \quad (72)$$

The corresponding discretizations can be expressed by using operators D_x and D_y :

$$(\mu \Delta \vec{V}) \cdot \vec{e}_x|_{i+1/2j} \cong \mu(D_x^2 u|_{i+1/2j} + D_y^2 u|_{i+1/2j}), \quad (73)$$

$$(\mu \Delta \vec{V}) \cdot \vec{e}_y|_{ij+1/2} \cong \mu(D_x^2 v|_{ij+1/2} + D_y^2 v|_{ij+1/2}). \quad (74)$$

In [10], remarking that the velocity gradients are strongly discontinuous at the interface because of viscosity jumps, it was proposed to compute each component of the Laplacian of the velocity $\mu \Delta \vec{V}$ (eqs. (73) and (74)) by computing the different terms of the vector $\nabla \cdot (\mu \nabla \vec{V})$ and subtracting their jump at the interface; on the x -axis,

$$\mu D_x^2 u|_{i+1/2j} = D_x(\mu D_x u)|_{i+1/2j} - [\mu D_x u] \delta \Gamma|_{i+1/2j}, \quad (75)$$

$$\mu D_y^2 u|_{i+1/2j} = D_y(\mu D_y u)|_{i+1/2j} - [\mu D_y u] \delta \Gamma|_{i+1/2j}, \quad (76)$$

and on the y -axis,

$$\mu D_x^2 v|_{ij+1/2} = D_x(\mu D_x v)|_{ij+1/2} - [\mu D_x v] \delta \Gamma|_{ij+1/2}, \quad (77)$$

$$\mu D_y^2 v|_{ij+1/2} = D_y(\mu D_y v)|_{ij+1/2} - [\mu D_y v] \delta \Gamma|_{ij+1/2}. \quad (78)$$

The appropriate jump conditions for eqs. (75), (76), (77) and (78) are given in the following matrix by:

$$\begin{pmatrix} [\mu \frac{\partial u}{\partial x}] & [\mu \frac{\partial u}{\partial y}] \\ [\mu \frac{\partial v}{\partial x}] & [\mu \frac{\partial v}{\partial y}] \end{pmatrix} = [\mu] \nabla \vec{V} \begin{pmatrix} \vec{0} \\ \vec{t} \end{pmatrix}^T \begin{pmatrix} \vec{0} \\ \vec{t} \end{pmatrix} + [\mu] \vec{n}^T \vec{n} \nabla \vec{V} \vec{n}^T \vec{n} - [\mu] \begin{pmatrix} \vec{0} \\ \vec{t} \end{pmatrix}^T \begin{pmatrix} \vec{0} \\ \vec{t} \end{pmatrix} \nabla \vec{V} \vec{n}^T \vec{n}, \quad (79)$$

where \vec{t} is the unit vector tangent to the interface.

To compute accurately eqs. (75), (76), (77), (78) in the framework of the *Ghost-Fluid Method* [10], suitable sharp approximations, eq. (58), of the Dirac distribution are required.

Note that the jump conditions in eq. (79) are only valid for a continuous velocity field between the two phases: this approach cannot be used when phase changes or mass transfers occur.

2.6.3. Ghost Fluid Semi-Conservative viscous Method

In the GFSCM, updating \vec{V}_4^* requires computing the following viscous terms:

$$\nabla \cdot (\mu \nabla \vec{V})|_x = \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right), \quad (80)$$

$$\nabla \cdot (\mu \nabla \vec{V})|_y = \frac{\partial}{\partial x} \left(\mu \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial y} \right). \quad (81)$$

That can be done in a similar way as for the conservative viscous approach, except for the second-order cross-derivatives, which are non-zero only at the interface and are therefore included in the pressure jump when the Poisson equation is solved. As pointed out previously, this approach can be used with an implicit temporal discretization of the viscous terms and also when a phase change occurs.

2.7. Implicit temporal discretization of viscous terms

Further elements are provided now to deal with an implicit temporal discretization of viscous terms.

When the incompressible Navier–Stokes equations for a single-phase flow are considered, the projection method leads to the following semi-implicit algorithm for the predictor step

$$\vec{V}_1^* - \Delta t \nu \Delta \vec{V}_1^* = \vec{V}^n - \Delta t (\vec{V}^n \cdot \nabla) \vec{V}^n, \quad (82)$$

which requires solving an independent linear system for each velocity component.

When a two-phase flow with a discontinuous viscosity is considered, the use of a semi-implicit projection method to solve the incompressible Navier–Stokes equations is more complex. Indeed, in 2D, the following two coupled equations must be solved simultaneously, leading to a more complicated linear system as it is carried out in [25,26]:

$$\begin{aligned} \rho_{i+1/2j}^{n+1} u_{i+1/2j}^* - \Delta t (D_x(2\mu D_x u^*) + D_y(\mu D_y u^*) + D_y(\mu D_x v^*))|_{i+1/2j} \\ = \rho_{i+1/2j}^{n+1} (u_{i+1/2j}^n - \Delta t (\vec{V}^n \cdot \nabla) u_{i+1/2j}^n), \end{aligned} \quad (83)$$

$$\begin{aligned} \rho_{ij+1/2}^{n+1} v_{ij+1/2}^* - \Delta t (D_x(\mu D_x v^*) + D_x(\mu D_y u^*) + D_y(2\mu D_y v^*))|_{ij+1/2} \\ = \rho_{ij+1/2}^{n+1} (v_{ij+1/2}^n - \Delta t (\vec{V}^n \cdot \nabla) v_{ij+1/2}^n). \end{aligned} \quad (84)$$

Some previous works report the following temporal discretization to split the resolution of this system:

$$\begin{aligned} & \rho_{i+1/2j}^{n+1} u_{i+1/2j}^* - \Delta t (D_x(2\mu D_x u^*) + D_y(\mu D_y u^*)) \Big|_{i+1/2j} \\ &= \rho_{i+1/2j}^{n+1} (u_{i+1/2j}^n - \Delta t (\vec{V}^n \cdot \nabla) u^n \Big|_{i+1/2j}) + \Delta t D_y(\mu D_x v^n) \Big|_{i+1/2j}, \end{aligned} \quad (85)$$

$$\begin{aligned} & \rho_{ij+1/2}^{n+1} v_{ij+1/2}^* - \Delta t (D_x(\mu D_x v^*) + D_y(2\mu D_y v^*)) \Big|_{ij+1/2} \\ &= \rho_{ij+1/2}^{n+1} (v_{ij+1/2}^n - \Delta t (\vec{V}^n \cdot \nabla) v^n \Big|_{ij+1/2}) + \Delta t D_x(\mu D_y u^n) \Big|_{ij+1/2}. \end{aligned} \quad (86)$$

Nevertheless, it is doubtful that the numerical scheme composed of eqs. (85), (86) really alleviates the time step constraint imposed by viscosity since the last terms on the right is treated explicitly.

Another way to carry out an implicit treatment of the viscous terms for incompressible flows is proposed in [11]. This is based on an approximate factorization which requires solving tridiagonal matrices rather than inversion of a large sparse matrix. This alternative method has not been tested in this work.

At the opposite, as it has been previously shown, for incompressible flows the second-order cross-derivatives are perfectly compensated everywhere except at the interface where viscosity is discontinuous. Therefore, the GFSCM can be used with benefits with an implicit treatment of the viscous terms, which leads to the following discretization:

$$\begin{aligned} & \rho_{i+1/2j}^{n+1} u_{i+1/2j}^* - \Delta t (D_x(\mu D_x u^*) + D_y(\mu D_y u^*)) \Big|_{i+1/2j} \\ &= \rho_{i+1/2j}^{n+1} (u_{i+1/2j}^n - \Delta t (\vec{V}^n \cdot \nabla) u^n \Big|_{i+1/2j}), \end{aligned} \quad (87)$$

$$\begin{aligned} & \rho_{ij+1/2}^{n+1} v_{ij+1/2}^* - \Delta t (D_x(\mu D_x v^*) + D_y(\mu D_y v^*)) \Big|_{ij+1/2} \\ &= \rho_{ij+1/2}^{n+1} (v_{ij+1/2}^n - \Delta t (\vec{V}^n \cdot \nabla) v^n \Big|_{ij+1/2}). \end{aligned} \quad (88)$$

The resulting linear systems can be solved as easily as in the single-phase case: the matrices of discretization are symmetric, definite and positive, which allows the use of an efficient and simple black-box solver. Note that an implicit temporal discretization of the viscous terms can also be used with the GFPM [8].

We conclude by some comments on the way to define the density: $\rho_{i+1/2j}^{n+1}$ and $\rho_{ij+1/2}^{n+1}$. The definition must be compatible with the computation of the correction step, which itself relies on the manner density is computed when the Poisson equation on pressure is solved. Accurate results are obtained by using the following general expression for $\rho_{i+1/2j}^{n+1}$ and $\rho_{ij+1/2}^{n+1}$ in the grid cells that are crossed by an interface:

$$\frac{1}{\rho^{n+1}} = \frac{\frac{1}{\rho^+ \rho^-}}{\frac{1}{\rho^+} \theta + \frac{1}{\rho^-} (1 - \theta)} = \frac{1}{\rho^- \theta + \rho^+ (1 - \theta)}. \quad (89)$$

2.8. Summary about the four different methods to deal with viscosity

Table 1 gives an overview of the characteristics of the four different methods to deal with viscous terms in the simulations of two-phase flows with *Level-Set methods*, and describes the main steps for implementing it.

2.9. Level-Set methods for interface capturing

Interface motion is captured with a *Level-Set method* [20,24], which consists in solving a convection equation for the Level-Set function ϕ :

$$\frac{\partial \phi}{\partial t} + (\vec{V} \cdot \nabla) \phi = 0, \quad (90)$$

with \vec{V} the velocity at the interface. A reinitialization step [24] can next be performed to insure that the ϕ function remains a signed distance in the computation domain. This can be done by solving iteratively the specific PDE:

$$\frac{\partial d}{\partial \tau} = \text{sign}(\phi) (1 - |\nabla d|), \quad (91)$$

where d is the reinitialized distance function, τ a fictitious reinitialization time and $\text{sign}(\phi)$ a smoothed signed function defined in [4,24]. At the end of every physical time step, two temporal iterations of the redistancing equation (91) are solved using a second-order TVD–Runge–Kutta scheme.

The signed distance function allows preserving a good accuracy when computing interface geometrical properties as the normal vector or the curvature with the following simple relations:

$$\vec{n} = \frac{\nabla \phi}{|\nabla \phi|}, \quad (92)$$

$$\kappa(\phi) = -\nabla \cdot \vec{n}. \quad (93)$$

Table 1Summary of four different methods to deal with viscous terms with *Level-Set methods*.

Framework	Method and <i>reference papers</i>	Steps of the explicit projection method and <i>number</i> of the corresponding equations in the <i>reference paper</i>	Implementation of an implicit method
“Whole domain formulation”	<i>Delta-Function Method</i> DFM Sussman et al. [3,24,25]	$\vec{V}_2^* = \vec{V}^n - \Delta t \left((\vec{V}^n \cdot \nabla) \vec{V}^n - \frac{\nabla \cdot (2\mu \mathbf{D}^n)}{\rho^{n+1}} - \frac{\sigma \kappa \vec{n} \delta_\Gamma}{\rho^{n+1}} - \vec{g} \right) \quad (33)$ $\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_2^*}{\Delta t} \quad (34)$ $\vec{V}^{n+1} = \vec{V}_2^* - \Delta t \frac{\nabla p^{n+1}}{\rho^{n+1}} \quad (35)$	One complex coupled parabolic linear system for all the velocity components
“Jump condition formulation”	<i>Ghost-Fluid Primitive viscous Method</i> GFPM Kang et al. [10] Hong et al. [8]	$\vec{V}_1^* = \vec{V}^n - \Delta t \left((\vec{V}^n \cdot \nabla) \vec{V}^n - \frac{\mu \Delta \vec{V}^n}{\rho^{n+1}} - \vec{g} \right) \quad (30)$ $\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_1^*}{\Delta t} + \nabla \cdot \left(\frac{(\sigma \kappa + 2[\mu]_\Gamma \frac{\partial V_n}{\partial n}) \vec{n} \delta_\Gamma}{\rho^{n+1}} \right) \quad (36)$ $\Leftrightarrow \begin{cases} \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_1^*}{\Delta t} \\ [p^{n+1}]_\Gamma = \sigma \kappa + 2[\mu]_\Gamma \frac{\partial V_n}{\partial n} \end{cases}$ $\vec{V}^{n+1} = \vec{V}_1^* - \frac{\Delta t}{\rho^{n+1}} (\nabla p^{n+1} - (\sigma \kappa + 2[\mu]_\Gamma \frac{\partial V_n}{\partial n}) \vec{n} \delta_\Gamma) \quad (39)$	Several simple independent linear systems with explicit jump conditions on the components of the viscous tensor
	<i>Ghost-Fluid Conservative viscous Method</i> GFCM Sussman et al. [26]	$\vec{V}_3^* = \vec{V}^n - \Delta t \left((\vec{V}^n \cdot \nabla) \vec{V}^n - \frac{\nabla \cdot (2\mu \mathbf{D}^n)}{\rho^{n+1}} - \vec{g} \right) \quad (40)$ $\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_3^*}{\Delta t} + \nabla \cdot \left(\frac{(\sigma \kappa \vec{n} \delta_\Gamma)}{\rho^{n+1}} \right) \quad (42)$ $\Leftrightarrow \begin{cases} \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_3^*}{\Delta t} \\ [p^{n+1}]_\Gamma = \sigma \kappa \end{cases}$ $\vec{V}^{n+1} = \vec{V}_3^* - \frac{\Delta t}{\rho^{n+1}} (\nabla p^{n+1} - \sigma \kappa \vec{n} \delta_\Gamma) \quad (47)$	One complex coupled parabolic linear system for all the velocity components
	<i>Ghost-Fluid Semi-Conservative viscous Method</i> GFSCM Present paper	$\vec{V}_4^* = \vec{V}^n - \Delta t \left((\vec{V}^n \cdot \nabla) \vec{V}^n - \frac{\nabla \cdot (\mu \nabla \vec{V}^n)}{\rho^{n+1}} - \vec{g} \right) \quad (41)$ $\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_4^*}{\Delta t} + \nabla \cdot \left(\frac{(\sigma \kappa + [\mu]_\Gamma \frac{\partial V_n}{\partial n}) \vec{n} \delta_\Gamma}{\rho^{n+1}} \right) \quad (43)$ $\Leftrightarrow \begin{cases} \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \vec{V}_4^*}{\Delta t} \\ [p^{n+1}]_\Gamma = \sigma \kappa + [\mu]_\Gamma \frac{\partial V_n}{\partial n} \end{cases}$ $\vec{V}^{n+1} = \vec{V}_4^* - \frac{\Delta t}{\rho^{n+1}} (\nabla p^{n+1} - (\sigma \kappa + [\mu]_\Gamma \frac{\partial V_n}{\partial n}) \vec{n} \delta_\Gamma) \quad (48)$	Several simple independent linear systems without any jump conditions on the components of the viscous tensor

The curvature is firstly computed on all the grid nodes with simple finite difference schemes. Next, if the Ghost Fluid Method is used, the curvature is interpolated on the interface. For example, if it crosses the segment $[x_i, x_{i+1}]$ the following interpolation is used:

$$\kappa_I = \kappa_{i+1} \theta + \kappa_i (1 - \theta) \quad (94)$$

2.10. Temporal and spatial discretizations

All convective terms and spatial derivatives of the resistance equation are discretized by means of a fifth-order WENO 5 scheme [9]. First-order explicit or semi-implicit temporal integration methods have been described in previous sections. These elementary steps are finally included in a second-order TVD–Runge–Kutta scheme for the overall temporal integration of the PDE system in order to improve the computation stability. A classical time-step constraint [10,23,24] accounting for the stability conditions on convection, viscosity (in case of an explicit temporal discretization) and surface tension is imposed.

$$\Delta t_{conv} = \frac{\Delta x}{\max \|\vec{V}\|} \quad (95)$$

$$\Delta t_{visc} = \frac{1}{4} \frac{\Delta x^2}{\max(\nu_{liq}, \nu_{gaz})} \quad (96)$$

$$\Delta t_{surf_tens} = \frac{1}{2} \sqrt{\frac{\rho_{liq} \Delta x^3}{\sigma}} \quad (97)$$

If an implicit temporal discretization is used with the GFSCM, an explicit part depending on the viscosity jump condition remains in Eqs. (43), (48). Therefore, the corresponding term involves another temporal constraint which is different from Eq. (96), since it depends on a viscosity jump condition instead of a maximum kinematic viscosity value. By performing several numerical test-cases with zero surface tension and very low gravity (in order to remove the stability constraints due to surface tension and convection), we have observed that the following time step restriction is sufficient to ensure the numerical stability:

$$\Delta t_{visc} = \frac{\bar{\rho} \Delta x^2}{[\mu]} \quad (98)$$

with $\bar{\rho}$ the average density of the two phases. Whereas this constraint depends also on Δx^2 , it can be much more lenient than the classical one, Eq. (96), because $\frac{\bar{\rho}}{[\mu]} \sim \frac{1}{\nu_{liq}}$ is much higher than $\frac{1}{\nu_{gas}}$ for many typical gas–liquid systems such as air–water or the one used in Section 3.3. For example, it is shown in Section 3.3 that the implicit GFSCM allows performing the computation with a much larger time step in comparison to its explicit counterparts.

Finally, the global time step restriction can be computed with the following relation:

$$\frac{1}{\Delta t} = \frac{1}{\Delta t_{conv}} + \frac{1}{\Delta t_{visc}} + \frac{1}{\Delta t_{surf_tens}} \quad (99)$$

3. Numerical results

Relevant benchmarks have been designed to assess and compare the four different numerical methods discussed in this paper: DFM, GFPM, GFCM and GFSCM.

In the first benchmark, computations of a spherical bubble rising under gravity are performed for a range of Reynolds number varying from 20 to 100.

The second benchmark consists of numerical simulations of bubble shape-oscillations in the absence of gravity. Such a physical problem is characterized by an oscillation Reynolds number which compares the oscillations frequency to the damping rate. The accuracy of the simulations is evaluated from comparisons with the theoretical prediction of the frequency and the damping rate of the oscillations for Reynolds numbers varying from 20 to 300. In addition, a study of the spatial convergence of the simulations is carried out in order to determine how mesh-grid requirements evolve when the Reynolds number is varied.

Finally, the third benchmark focuses on the rise of a non-spherical bubble in a very viscous liquid as in Refs. [2,6,26].

These three benchmarks consist of axisymmetric computations.

3.1. A spherical bubble rising under gravity

Currently, performing well-resolved simulations of bubbly flows or droplet sprays is a challenge for the direct numerical simulation of two-phase flows. Many issues must be overcome to achieve such ambitious targets. One can expect that the development of High Performance Computing hardware and software will allow performing simulations involving a growing number of bubbles or drops. However well-resolved 3D simulations remain costly and a particular attention must be paid to the development and the assessment of accurate numerical methods to solve the fluid dynamics equations. In particular, a computation which is efficient to describe accurately the dynamics with half the number of grid cells on a bubble diameter makes possible to compute eight times more bubbles with the same computational resources. Actually, the gain would even be significantly greater since, for explicit methods, a larger time step can be used on a coarser grid. The test-case proposed in this section allows us to assess the different discretizations of the viscous terms. Simulations are performed with an axisymmetric coordinate system. The radial and axial sizes of the computational domain are defined relatively to the bubble radius R_{bubble} : $l_r = 8R_{bubble}$ and $l_z = 4l_r$. Considering a spherical bubble, reference results on the drag coefficient are provided by boundary-fitted numerical simulations from [19] when $Re_\infty \leq 50$ or by Moore's theory [18] for higher Reynolds numbers. Our computations are performed for low Bond numbers to preserve bubble sphericity. It is worth mentioning that the case of a rising spherical bubble is particularly advised to assess the discretization of the viscous terms because, in this situation, the drag force mainly depends on viscous dissipation and not on pressure effects, unlike the case of a strongly deformed bubble.

The physical properties of liquid and gas are kept identical to water and air for all simulations presented in this section: $\rho_{liq} = 1000 \text{ kg m}^{-3}$, $\mu_{liq} = 0.00113 \text{ kg m}^{-1} \text{ s}^{-1}$, $\rho_{gas} = 1.226 \text{ kg m}^{-3}$, $\mu_{gas} = 1.78 \times 10^{-5} \text{ kg m}^{-1} \text{ s}^{-1}$. Simulation conditions are summarized in Table 2 for five configurations corresponding to $Re_\infty = 20, 40, 60$ for a Bond number $Bo = 0.025$, and to $Re_\infty = 80, 100$ for $Bo = 0.0125$. Bubbles are initially static and evolve under gravity until the terminal velocity is reached. The boundaries of the computational domain are defined as walls (except on symmetry axis) and set sufficiently far from the bubble to have a negligible influence on the results.

Simulations are performed with three different mesh grids (32×128 , 64×256 , 128×512 , 256×1024), which describe respectively one bubble diameter with 8, 16, 32 or 64 mesh points. Fig. 2 shows a snapshot of the interface location and the fully-developed vorticity in the frame moving with the bubble for the case $Re_\infty = 60$ with the thinnest grid. Results of all the simulations are summarized in Tables 3, 4, 5, 6 and 7, which correspond respectively to computations carried out using the *Ghost Fluid Conservative viscous Method* (GFCM), explicit and implicit *Ghost-Fluid Semi-Conservative viscous Method* (GFSCM), *Ghost-Fluid Primitive viscous Method* (GFPM) and *Delta-Function Method* (DFM). Many conclusions can be drawn from these tables. First, all methods fail to provide accurate results on the coarsest grid regardless the value of Re_∞ : this mesh is clearly insufficient to capture accurately viscous effects. When the intermediate grid is used, the GFCM and the explicit and implicit GFSCM provides very good results: whatever Re_∞ , deviations between reference results and numerical simulations are less or equal to 3% (except for $Re_\infty = 80$ when the explicit GFSCM is used). The GFPM also provides acceptable results on this

Table 2

Bubble radius, gravity acceleration, surface tension and final time of simulation for a spherical bubble rising at various Reynolds numbers.

	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
R_{bubble} (μm)	19.55	44.65	70.45	98.95	127.9
g_z (m s^{-2})	-9144.4	-1755.2	-705.6	-357.5	-213.9
σ (N m^{-1})	0.56	0.56	0.56	1.12	1.12
t_f (s)	0.0004	0.00135	0.002375	0.0048	0.007

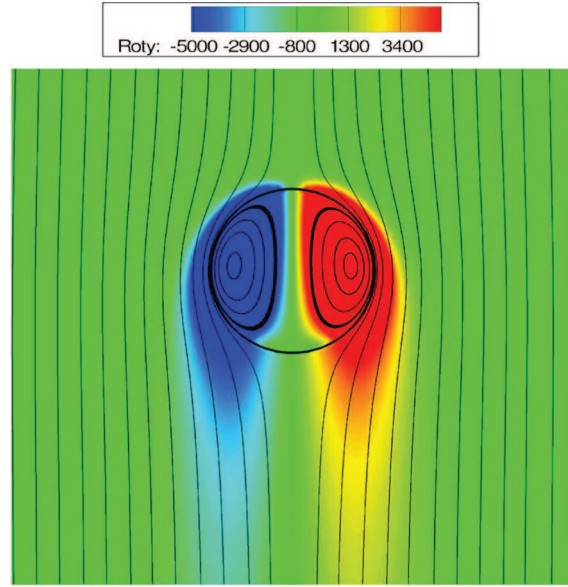


Fig. 2. Streamlines, interface location and vorticity field (s^{-1}) in the moving frame of a rising spherical bubble at $Re_\infty = 60$ (grid 128×512) using the GFCM.

Table 3

Error on the terminal Reynolds number of rising bubbles for different computational grids with the explicit GFCM.

Grids	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	-22.0%	-34.0%	-38.8%	-63.0%	-67.4%
64×256	+2.5%	+1.8%	-1.3%	-1.3%	-3.0%
128×51	+0.5%	+0.8%	-1.0%	+1.9%	+1.9%
256×1024	-0.9%	-0.4%	-1.95%	-1.02%	-0.65%

Table 4

Error on the terminal Reynolds number of rising bubbles for different computational grids with the explicit GFSCM.

Grids	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	-23.5%	-32.5%	-39.0%	-62.5%	-67.0%
64×256	-0.5%	+2.5%	+1.3%	+4.3%	+3.0%
128×512	-3.5%	+0.6%	0.7%	+5.4%	+6%
256×1024	-2.15%	+3.25%	+4.2%	+6.75%	+8.2%

Table 5

Error on the terminal Reynolds number of rising bubbles for different computational grids with the implicit GFSCM.

Grids	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	-18.6%	-25.0%	-30.2%	-52.9%	-55.1%
64×256	-1.2%	1.5%	-0.1%	2.0%	0.9%
128×512	-3.7%	0.1%	-0.2%	3.2%	3.9%
256×1024	-4.2%	3%	4%	6.25%	7.6%

Table 6

Error on the terminal Reynolds number of rising bubbles for different computational grids with the explicit GFPM.

Grids	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	-16.7%	-24.6%	-31.5%	-52.3%	-56.8%
64×256	0.5%	-1.2%	-4.5%	-4.8%	-6.6%
128×512	0.1%	-0.3%	-2.6%	-1.0%	-1.6%
256×1024	-0.7%	-0.67	-2.6%	-1.3%	-1.4%

Table 7

Error on the terminal Reynolds number of rising bubbles for different computational grids with the explicit DFM.

Grids	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	-39.5%	-46.1%	-53.8%	-75.8%	-81.6%
64×256	-16.5%	-18.3%	-21.7%	-29.3%	-31.8%
128×512	-7.5%	-7.9%	-9.9%	-10.5%	-10.9%
256×1024	-4.3%	-4.65%	-6.6%	-6.04%	-6.05%

grid since the discrepancy between numerical simulations and reference solutions is less than 7%; the error increases with the Reynolds number, which is probably due to the difficulty to calculate accurately boundary layers that becomes thinner at high Re_∞ . By contrast, the DFM provides a lower accuracy than the other methods with errors on Re_∞ which vary from 15% to 30% and increase with Re_∞ . Finally, considering results obtained with the most refined grids, it clearly appears that both the GFCM and the GFPM have converged to the correct solution. The explicit and implicit GFSCM provide results with larger errors than the two previous methods with errors varying between 2% and 8%. The DFM seems to converge slowly towards the theoretical solution and still shows deviations varying from 4% to 6% (according to Re_∞) with 64 mesh points on a bubble diameter.

3.2. Shape oscillations of a bubble

Considering small-amplitude-shape-oscillations of a bubble in the absence of gravity, theoretical solutions of the linearized Navier–Stokes equations for an incompressible flow exist and give expressions for the frequency and the damping rate of the oscillations [17,22]. We have took advantage of this to design an accurate and stringent test-case for direct numerical simulations. While the oscillation frequency depends essentially on capillary effects, we have focused on the damping rate in order to test the accuracy of different numerical approaches about viscosity. Like in the previous section, simulations are performed with an axisymmetric coordinate frame. The dimensions of the computational field are: $l_r = 4R_{bubble}$, $l_z = 2l_r$. Wall boundary conditions are used on all domain borders (except on symmetry axis) and confinement effects have been checked to be negligible. Densities and viscosities, corresponding to an air/water system, are the same as in Section 3.2, and the surface tension is $\sigma = 0.07 \text{ N m}^{-1}$. We propose a parametric study by varying the Reynolds number of oscillation, by $Re_{osc} = \frac{\sqrt{\sigma R_{bubble} \rho_{liq}}}{\mu_{liq}}$, from 20 to 300. (Note that the Reynolds number of oscillation is the inverse of the Ohnesorge number.)

In our simulations, the initial condition corresponds to a deformed interface described by a Legendre polynomial of order 2 with the following Level Set function

$$\phi(r, \theta, t = 0) = r - R_{bubble} [1 + \epsilon P_2(\cos \theta)], \quad (100)$$

where (r, θ) are the spherical coordinates, $P_2(\cos \theta) = \frac{1}{2}(3 \cos^2 \theta - 1)$ and ϵ is the dimensionless initial amplitude of the deformation, chosen low enough ($\epsilon = 0.075$) to maintain the oscillations in the linear regime. The computed shape of the interface is extracted at each time step and decomposed into spherical harmonics in order to obtain the time evolution of amplitude $a_2(t)$ of harmonic $l = 2$:

$$r(\theta, t) = R_{bubble} + a_2(t) P_2(\cos \theta). \quad (101)$$

Then, the frequency ω_2 and the damping rate β_2 of the oscillations are obtained from $a_2(t)$ since

$$a_2(t) = \epsilon R_{bubble} \cos(\omega_2 t) e^{-\beta_2 t} \quad (102)$$

The values issued from the simulations are compared with the theoretical values ω_2^{th} and β_2^{th} that are computed by solving numerically the characteristic equation given by Prosperetti (eq. (33) of Ref. [22]). The damping rate is a global parameter that requires an accurate calculation of the viscous terms all over the computation domain.

Snapshots of the interface location and vorticity are shown in Fig. 3 and Fig. 4 for $Re_{osc} = 20$ and $Re_{osc} = 200$, respectively. It is clear from these figures that the vorticity is localized inside the bubble, whereas the flow remains potential outside. For a shape-oscillating bubble, both the frequency and the damping rate are mainly controlled by the potential flow in the liquid phase. The asymptotic development of the damping rate given by Miller and Scriven [17] shows that, for a bubble, the viscous dissipation in the boundary layers is responsible for only 1 to 3% of the total dissipation in a present

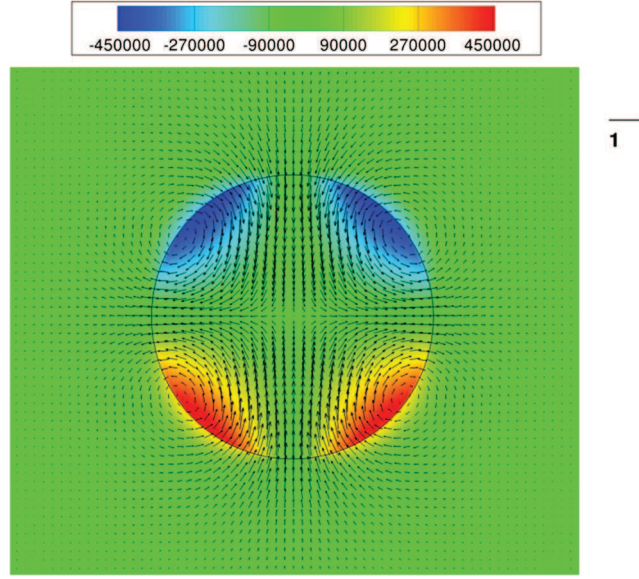


Fig. 3. Velocity field, interface location and vorticity field (s^{-1}) of an oscillating bubble at $Re_{osc} = 20$ (grid 256×512) using the GFCM.

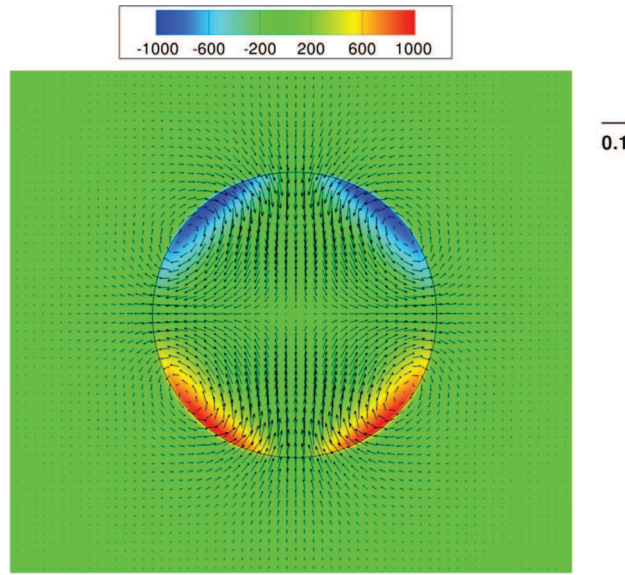


Fig. 4. Velocity field ($m s^{-1}$), interface location and vorticity field (s^{-1}) of an oscillating bubble at $Re_{osc} = 200$ (grid 256×512) using the GFCM.

Table 8

Error on the damping rate β_2 for an oscillating bubble with different computational grids with the explicit GFCM.

Grids	$Re_{osc} = 20$	$Re_{osc} = 50$	$Re_{osc} = 100$	$Re_{osc} = 200$	$Re_{osc} = 300$
64×128	1.49%	1.47%	9.75%	23.16%	41.72%
128×256	-0.20%	0.32%	1.99%	7.30%	13.71%
256×512	-0.31%	0.08%	0.70%	2.55%	5.36%

range of Re_{osc} , i.e. between 20 and 300. The rest of the viscous dissipation comes from the potential flow in the liquid phase.

For each Reynolds number of oscillation, a test has been computed with four different numerical methods (GFCM, explicit GFSCM, GFPM and DFM) and for three different mesh grids (64×128), (128×256) and (256×512), which describe a bubble diameter with 32, 64 and 128 mesh points, respectively. The results of the simulations are summarized in [Tables 8, 9, 10 and 11](#), where the deviation of the numerical simulations relative to the theoretical predictions have been reported. The error on the frequency of oscillation is very small ($<1\%$) whatever the numerical method and the mesh grids; it has therefore

Table 9Error on the damping rate β_2 for an oscillating bubble with different computational grids with the explicit GFSCM.

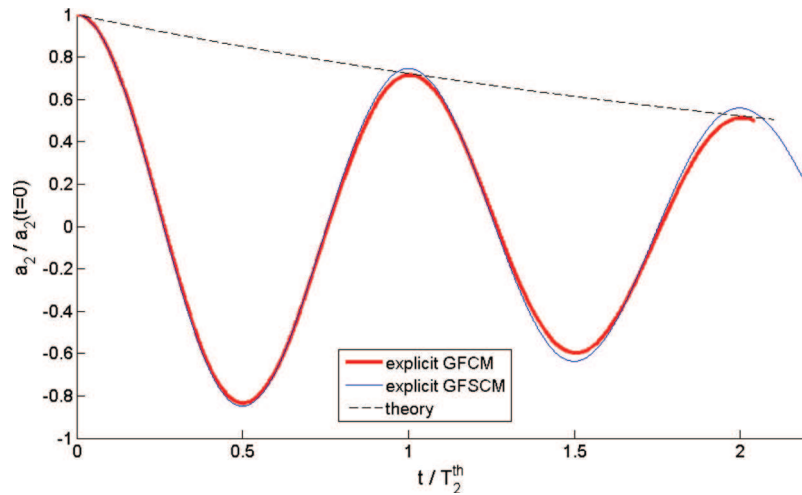
Grids	$Re_{osc} = 20$	$Re_{osc} = 50$	$Re_{osc} = 100$	$Re_{osc} = 200$	$Re_{osc} = 300$
64×128	-4.57%	-7.16%	-0.30%	13.06%	31.04%
128×256	-3.90%	-7.58%	-6.47%	0.38%	8.78%
256×512	-2.69%	-7.52%	-12.13%	-4.21%	-0.77%

Table 10Error on the damping rate β_2 for an oscillating bubble with different computational grids with the explicit GFPM.

Grids	$Re_{osc} = 20$	$Re_{osc} = 50$	$Re_{osc} = 100$	$Re_{osc} = 200$	$Re_{osc} = 300$
64×128	2.88%	3.11%	10.39%	22.34%	39.62%
128×256	0.23%	0.68%	1.66%	5.90%	11.70%
256×512	-0.10%	-0.00%	-1.04%	1.31%	3.52%

Table 11Error on the damping rate β_2 for an oscillating bubble with different computational grids with the explicit DFM.

Grids	$Re_{osc} = 20$	$Re_{osc} = 50$	$Re_{osc} = 100$	$Re_{osc} = 200$	$Re_{osc} = 300$
64×128	5.63%	5.56%	7.07%	-1.66%	-9.57%
128×256	2.63%	2.75%	1.56%	-1.07%	-4.54%
256×512	0.82%	1.02%	-0.10%	-2.10%	-4.35%

**Fig. 5.** Temporal evolution of amplitude of mode 2 at $Re_{osc} = 100$. Comparisons between theory (from Ref. [20]) and numerical simulations with the GFCM and the GFSCM. Time is normalized by the theoretical period of oscillation of mode 2, T_2^{th} .

not been reported. A detailed analysis of the errors on the damping rate leads to the following conclusions. First, the GFCM and the GFPM accurately match with the exact solution when refined grids are used. Both methods present comparable abilities on this test-case, with errors less than 2% in the range $20 \leq Re_{osc} \leq 200$ (the GFPM is just slightly better than the GFCM). For the highest Reynolds numbers of oscillation considered, the error seems to decrease monotonously and the order of convergence can be roughly estimated to 1.5. That is not surprising considering that, even if the discretization of viscous terms remains of the second order far from the interface, the subcell viscosity interpolation in the cells that are crossed by an interface is only of the first order. The DFM also provides good results in this case and converges well, but exhibits a different behavior than the GFCM or the GFPM. For example, when coarse grids are used, the error is relatively low (between 5 and 10%) whatever the Re_{osc} , contrary to the two previous methods where the error was clearly increasing with Re_{osc} to become larger than 20% at the highest Re_{osc} . The rate of convergence of the DFM is not as clear as for the GFCM or the GFPM. Finally, the explicit GFSCM provides poor results with errors up to 12% in some cases with the thinnest grid. Although this discrepancy is not negligible, a more detailed analysis of the temporal signal $a_2(t)$, presented in Fig. 5 for the most disadvantageous case for the GFSCM ($Re_{osc} = 100$, grid 256×512), allows us to realize that this underestimation of 12% of the damping rate does not induce a so significant difference compared to the other methods.

In addition, it is worth mentioning that the GFCM has been successfully used in Ref. [13] to simulate the shape oscillations of rising drops and bubbles in order to determine the effect of the rising motion on the frequency and the damping rate of the oscillations.

Table 12

Errors of the different numerical methods with the grid (128×512) on the terminal Reynolds number of a deformed rising bubble ($Re_\infty = 19.6$).

Numerical method	Measured Re_∞	Error on Re_∞
Explicit GFCM	18.79	-4.1%
Implicit GFSCM	18.25	-6.9%
Explicit GFPM	18.59	-5.2%
Explicit DFM	18.73	-4.4%

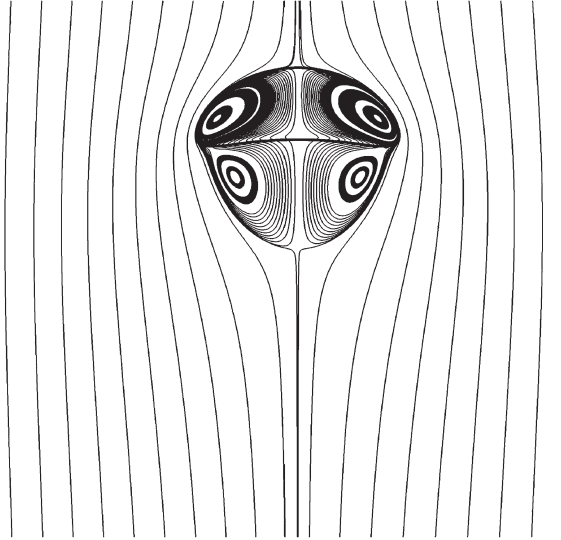


Fig. 6. Streamlines and interface location in the moving frame of a rising deformed bubble at $Re_\infty = 19.6$ (grid 128×512) using the GFCM.

3.3. Deformed rising bubble at $Re_\infty = 19.6$

To complete the assessment of the different numerical methods, we propose a third test-case which consists in computing the rise of a bubble in a very viscous liquid. This test-case is based on experimental data [7] and has already been used for the validation of numerical simulations, see Refs. [2,6,26] for example. Its interest lies in the strong deformation of the bubble, which involves the development of a recirculation zone in the bubble wake. Here, unlike the case of a spherical rising bubble, the drag force results both from pressure and viscous effects. Simulations are performed in axisymmetric coordinates and the computational domain is defined by $l_r = 8R_{bubble}$ and $l_z = 4l_r$. We used the following physical properties for the fluids: $\rho_{liq} = 875.5 \text{ kg m}^{-3}$, $\mu_{liq} = 0.118 \text{ kg m}^{-1} \text{ s}^{-1}$, $\rho_{gas} = 1 \text{ kg m}^{-3}$, $\mu_{gas} = 0.001 \text{ kg m}^{-1} \text{ s}^{-1}$, $\sigma = 0.0332 \text{ N m}^{-1}$, $g_z = -9.81 \text{ m s}^{-2}$, $R_{bubble} = 0.0061 \text{ m}$.

All results obtained for this test-case are summarized in Table 12, the simulations being carried out with a grid (128×512), which counts 32 points in a bubble diameter. We compare the GFCM, GFPM, DFM and implicit GFSCM. A snapshot of interface location and streamlines around and inside the bubble are plotted in Fig. 6; it clearly shows the existence of a recirculation zone in the bubble wake. Temporal evolutions of the Reynolds number have been plotted in Fig. 7 to describe the results obtained during the transient stage. The reference terminal rising velocity issued from experiments corresponds to $Re_\infty = 19.6$. We observe that all methods provide satisfactory results with errors on the Reynolds number varying from 4.1% for the GFCM to 6.9% for the GFSCM. We can conclude that all tested methods have a comparable accuracy for this test-case. Fig. 8 highlights the interest of the implicit GFSCM on a simulation where the time step restriction due to viscosity (96) is dominant.

This time step restriction can be removed if the implicit GFSCM is used. Therefore the numerical stability is maintained by imposing the time step restriction due to capillary forces (97) and the modified time step restriction due to the viscosity jump (98) which have both comparable values in this case.

The cost of a temporal iteration is approximately the same for an explicit discretization or the implicit GFSCM. Indeed, the linear systems resulting from the implicit discretization of the viscous terms are diagonally dominant due to the other time step restrictions. Therefore, their solutions are computed much faster than for the pressure Poisson equation. Thus, for the simulations which are considered in Fig. 8 the implicit GFSCM will run 10 to 20 faster than an explicit version of the GFSCM, GFCM, GFPM or DFM.

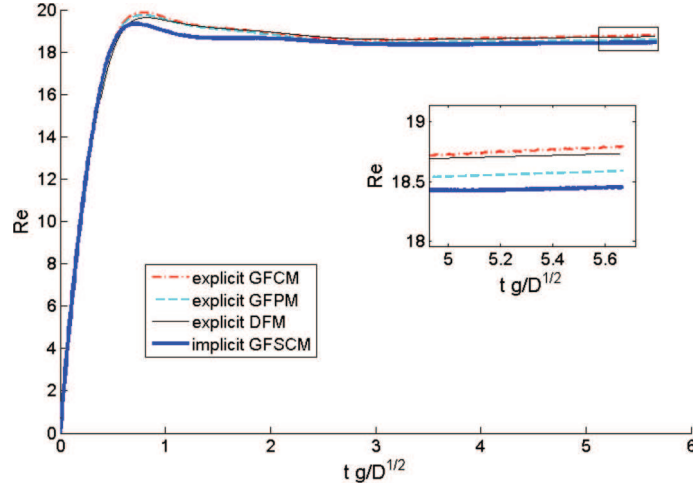


Fig. 7. Temporal evolution of the Reynolds number for the different numerical methods for a deformed rising bubble at $Re_\infty = 19.6$ (grid 128×512).

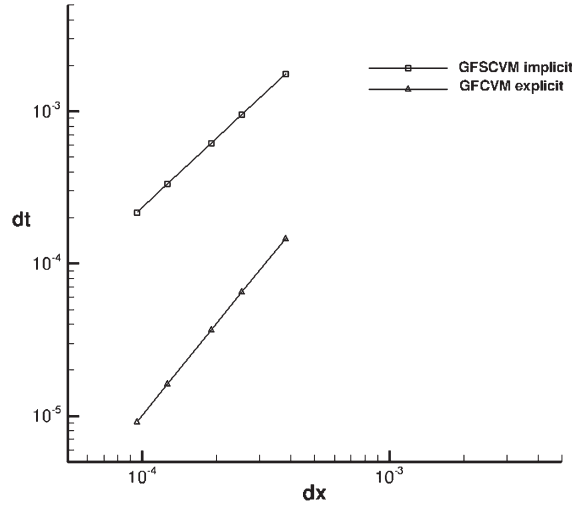


Fig. 8. Time step evolution with the mesh size for the deformed rising bubble. Comparison between the implicit GFSCVM and the explicit GFCVM.

4. Conclusions

In this paper, we have proposed an extensive study to assess different numerical methods for the computation of the viscous terms in the framework of direct numerical simulation of two-phase flows with *Level-Set methods*. An important feature of this work is to provide the theoretical evidence that the *Delta-Function Method* described by Sussman et al. in [24], the original *Ghost-Fluid Method* for incompressible two-phase flows proposed by Kang et al. in [10], and another version of the *Ghost-Fluid Method* proposed later by Sussman et al. in [26] are formally equivalent. For sake of clarification, these three methods are respectively renamed in this paper the *Delta-Function Method* (DFM), the *Ghost-Fluid Primitive viscous Method* (GFPM), and the *Ghost-Fluid Conservative viscous Method* (GFCM). It is shown that all these different methods result from different choices of the predictive velocity \bar{V}^* , but they all lead to solve the same set of formal equations. Thanks to this proof, we have designed a novel method to compute viscous effects in two-phase flows, which is named the *Ghost-Fluid Semi-Conservative viscous Method* (GFSCM). The main interest of this new method lies in its ease of implementation for the implicit temporal integration of the viscous terms.

Next, the respective accuracy of these different methods has been compared from many simulations based on three relevant test-cases: a spherical bubble rising at different Reynolds numbers, damped shape-oscillations of a bubble, and the rise of a non-spherical bubble at low Reynolds number. Important conclusions have been drawn from these benchmarks. The GFCM and the GFPM succeed all tests and clearly appear as the most accurate methods, with comparable results. The DFM presents a poor efficiency in the case of a spherical rising bubble because of its slow spatial convergence, but works well for the other tests. Finally, the GFSCM provides results in simulations with rising bubbles but less satisfactory results for bubble shape oscillations, with a precision on the damping rate of about 10%. Nevertheless, thanks to its simplicity to

compute an implicit discretization of the viscous terms and its ability to treat the case of discontinuous velocities at the interface (due for example to mass transfers or phase changes), this approach is promising to deal with challenging complex problems for which slight inaccuracies can be tolerated.

Acknowledgements

The authors gratefully acknowledge funding by ANR (French Research National Agency) in the frame of the IDHEAS (Interaction of Droplets with HEATED Surface) program.

References

- [1] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* 100 (1992) 335–353.
- [2] T. Bonometti, J. Magnaudet, Transition from spherical cap to toroidal bubbles, *Phys. Fluids* 18 (2006) 052102.
- [3] Y.G. Chang, T.Y. Hou, B. Merriman, S. Osher, A level set formulation of eulerian interface capturing methods for incompressible fluid flows, *J. Comput. Phys.* 124 (1996) 449–464.
- [4] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (The Ghost Fluid Method), *J. Comput. Phys.* 152 (1999) 457–492.
- [5] M. François, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, M.W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, *J. Comput. Phys.* 213 (2006) 141–173.
- [6] D. Gueyffier, J. Lie, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, *J. Comput. Phys.* 152 (1999) 423–456.
- [7] J.G. Hnat, J.D. Buckmaster, Spherical cap bubbles and skirt formation, *Phys. Fluids* 19 (1976) 182.
- [8] J.-M. Hong, T. Shinar, M. Kang, R. Fedkiw, On boundary condition capturing for multiphase interfaces, *J. Sci. Comput.* 31 (2007) 99–125.
- [9] G.S. Jiang, C.W. Shu, Efficient implementation of weighted essentially non-oscillatory schemes, *J. Comput. Phys.* 126 (1996) 202–228.
- [10] M. Kang, R. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, *J. Sci. Comput.* 15 (2000) 323–360.
- [11] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, *J. Comput. Phys.* 59 (1985) 308–323.
- [12] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, *J. Comput. Phys.* 113 (1994) 134–147.
- [13] B. Lalanne, S. Tanguy, F. Risso, Effect of rising motion on the damped shape oscillations of drops and bubbles, *Phys. Fluids* 25 (2013) 112107.
- [14] R.J. Leveque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, *SIAM J. Sci. Comput.* 18 (3) (1997) 709–735.
- [15] X.-D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, *J. Comput. Phys.* 160 (2000) 151–178.
- [16] T. Menard, S. Tanguy, A. Berlemont, Coupling Level Set/VOF/Ghost Fluid methods: validation and application to 3D simulation of the primary break-up of a liquid jet, *Int. J. Multiph. Flow* 33 (5) (2007) 510–524.
- [17] C.A. Miller, L.E. Scriven, The oscillations of a fluid droplet immersed in another fluid, *J. Fluid Mech.* 32 (1968) 417–435.
- [18] D.W. Moore, The boundary layer on a spherical gas bubble, *J. Fluid Mech.* 16 (1962) 161–176.
- [19] D.L.R. Oliver, J.N. Chung, Flow about a fluid sphere at low and moderate Reynolds numbers, *J. Fluid Mech.* 177 (1987) 1–18.
- [20] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* 79 (1988) 12–49.
- [21] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, *Int. J. Numer. Methods Fluids* 30 (6) (1999) 775–793.
- [22] A. Prosperetti, Normal mode analysis for the oscillations of a viscous liquid drop in an immiscible liquid, *J. Méc.* 19 (1980) 149–182.
- [23] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* 31 (1999) 567–603.
- [24] M. Sussman, P. Smereka, S. Osher, A Level Set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* 114 (1994) 146–159.
- [25] M. Sussman, E.G. Puckett, A coupled level and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, *J. Comput. Phys.* 162 (2000) 301–337.
- [26] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, *J. Comput. Phys.* 221 (2007) 469–505.
- [27] S. Tanguy, A. Berlemont, Application of a Level Set method for simulation of droplet collisions, *Int. J. Multiph. Flow* 31 (9) (2005) 1015–1035.
- [28] P. Trontin, S. Vincent, J.L. Estivalezes, J.P. Caltagirone, Direct numerical simulation of a freely decaying turbulent interfacial flow, *Int. J. Multiph. Flow* 36 (11–12) (2010) 891–907.
- [29] B.P. Van Poppel, O. Desjardins, J.W. Daily, A Ghost Fluid Level Set methodology for simulating multiphase electrohydrodynamic flows with application to liquid fuel injection, *J. Comput. Phys.* 229 (2010) 7977–7996.
- [30] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible multi-fluid flows, *J. Comput. Phys.* 100 (1992) 25–37.