

## Article

# Validation of HiG-Flow Software for Simulating Two-Phase Flows with a 3D Geometric Volume of Fluid Algorithm

Aquisson T. G. da Silva <sup>1</sup>, Célio Fernandes <sup>2,3,\*</sup>, Juniormar Organista <sup>1</sup>, Leandro Souza <sup>1</sup>  
and Antonio Castelo <sup>1</sup>

- <sup>1</sup> Institute of Mathematical and Computer Sciences (ICMC), University of São Paulo (USP), São Carlos 13566-590, SP, Brazil; aquisson@usp.br (A.T.G.d.S.); juniormarorganista@usp.br (J.O.); lefraso@icmc.usp.br (L.S.); castelo@icmc.usp.br (A.C.)  
<sup>2</sup> Transport Phenomena Research Center (CEFT), Faculty of Engineering at University of Porto (FEUP), 4200-465 Porto, Portugal  
<sup>3</sup> Center of Mathematics (CMAT), School of Sciences at University of Minho, Campus of Gualtar, 4710-057 Braga, Portugal  
\* Correspondence: cbpf@fe.up.pt

**Abstract:** This study reports the development of a numerical method to simulate two-phase flows of Newtonian fluids that are incompressible, immiscible, and isothermal. The interface in the simulation is located and reconstructed using the geometric volume of fluid (VOF) method. The implementation of the piecewise-linear interface calculation (PLIC) scheme of the VOF method is performed to solve the three-dimensional (3D) interface transport during the dynamics of two-phase flows. In this method, the interface is approximated by a line segment in each interfacial cell. The balance of forces at the interface is accounted for using the continuum interfacial force (CSF) model. To solve the Navier–Stokes equations, meshless finite difference schemes from the HiG-Flow computational fluid dynamics software are employed. The 3D PLIC-VOF HiG-Flow algorithm is used to simulate several benchmark two-phase flows for the purpose of validating the numerical implementation. First, the performance of the PLIC implementation is evaluated by conducting two standard advection numerical tests: the 3D shearing flow test and the 3D deforming field test. Good agreement is obtained for the 3D interface shape using both the 3D PLIC-VOF HiG-Flow algorithm and those found in the scientific literature, specifically, the piecewise-constant flux surface calculation, the volume of fluid method implemented in OpenFOAM, and the high-order finite-element software FEEL. In addition, the absolute error of the volume tracking advection calculation obtained by our 3D PLIC-VOF HiG-Flow algorithm is found to be smaller than the one found in the scientific literature for both the 3D shearing and 3D deforming flow tests. The volume fraction conservation absolute errors obtained using our algorithm are  $4.48 \times 10^{-5}$  and  $9.41 \times 10^{-6}$  for both shearing and deforming flow tests, respectively, being two orders lower than the results presented in the scientific literature at the same level of mesh refinement. Lastly, the 3D bubble rising problem is simulated for different fluid densities ( $\rho_1/\rho_2 = 10$  and  $\rho_1/\rho_2 = 1000$ ) and viscosity ratios ( $\mu_1/\mu_2 = 10$  and  $\mu_1/\mu_2 = 100$ ). Again, good agreement is obtained for the 3D interface shape using both the newly implemented algorithm and OpenFOAM, DROPS, and NaSt3D software. The 3D PLIC-VOF HiG-Flow algorithm predicted a stable ellipsoidal droplet shape for  $\rho_1/\rho_2 = 10$  and  $\mu_1/\mu_2 = 10$ , and a stable cap shape for  $\rho_1/\rho_2 = 1000$  and  $\mu_1/\mu_2 = 100$ . The bubble's rise velocity and evolution of the bubble's center of mass are also computed with the 3D PLIC-VOF HiG-Flow algorithm and found to be in agreement with those software. The rise velocity of the droplet for both the ellipsoidal and cap flow regime's is found, in the initial stages of the simulation, to gradually increase from its initial value of zero to a maximum magnitude; then, the steady-state velocity of the droplet decreases, being more accentuated for the cap regime.



**Citation:** Silva, A.T.G.d.; Fernandes, C.; Organista, J.; Souza, L.; Castelo, A. Validation of HiG-Flow Software for Simulating Two-Phase Flows with a 3D Geometric Volume of Fluid Algorithm. *Mathematics* **2023**, *11*, 3900. <https://doi.org/10.3390/math11183900>

Academic Editor: Vasily Novozhilov

Received: 2 August 2023

Revised: 6 September 2023

Accepted: 12 September 2023

Published: 13 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** two-phase flows; geometric volume of fluid method; piecewise-linear interface calculation; HiG-Flow; numerical validation

**MSC:** 76-10

## 1. Introduction

Interfacial multiphase flows play a crucial role in a wide range of industrial fields, offering opportunities for enhanced process efficiency, product quality, and environmental sustainability. Understanding and manipulating the dynamics of multiple immiscible fluids in contact at their interfaces are key challenges that require accurate numerical computations. These systems exhibit complex behaviors due to the interplay between bulk fluid coupling and surface effects, necessitating advanced modeling and simulation techniques [1]. The ability to accurately predict and control interfacial phenomena is of paramount importance in industrial processes such as oil and gas extraction [2], chemical reactions [3], pharmaceutical manufacturing [4], food processing [5], and microfluidics [6].

Interfacial multiphase flows exhibit complex and dynamic behavior, requiring sophisticated numerical techniques for their accurate calculation and analysis. Over the years, various numerical methods have been developed to tackle the challenges associated with capturing and tracking fluid interfaces [7].

One widely employed approach is the front-tracking method [8,9], which explicitly tracks the fluid interface using a moving mesh. This method accurately captures the detailed dynamics of the interface and enables the precise representation of surface effects. An example of a front-tracking method is the arbitrary Lagrangian–Eulerian (ALE) method [10]. Front-tracking methods are very good at providing an accurate description of the contact surface but often come with high computational costs. In the realm of the front-tracking approach, it is imperative to recognize a critical aspect: the efficacy of these methods can wane in scenarios where intricate surface deformations transpire. While the front-tracking technique demonstrates significant utility in capturing and tracking interfaces in fluid dynamics simulations, its performance can be compromised when dealing with complex and dynamic surface behaviors. This limitation can stem from challenges in accurately representing and maintaining the evolving interface shape, particularly in cases involving substantial distortions, breakups, or coalescence.

Another approach is the fixed-mesh method, which represents the fluid interface implicitly using additional fields. The volume of fluid (VOF) method [11], phase-field method [12], and level-set method [13] are notable examples. These methods use a fixed mesh and provide easier coupling formulations and smaller algebraic problems. However, they may sacrifice some accuracy in the description of the interface compared to front-tracking methods. In the VOF method, a mass conservation equation is solved to track the movement of the interface by computing the volume fraction of each fluid. This method is effective at capturing large-scale interface dynamics and is widely used in various industrial applications. The phase-field method is a diffuse-interface model that introduces an additional scalar field to represent the interface, and one of the notable benefits is arguably the complete avoidance of interface tracking, particularly in cases where there is no explicit necessity for evaluating interface curvature and normal vectors. This approach offers a smooth transition across the interface and is particularly useful for simulating complex interfacial phenomena such as phase separation and morphological changes. Conversely, the primary drawback associated with phase-field methods lies in their tendency to generate relatively wide interfaces. The level-set method evolves a scalar function called the level set function, which tracks the interface as it evolves in time. This method provides accurate interface capturing and is suitable for simulating topological changes, such as the merging and breakup of droplets.

Continued advancements in numerical methods hold the potential for further enhancing our understanding and control of interfacial multiphase flows. Specifically, the modeling and simulation of three-dimensional (3D) interfacial multiphase flows is a critical area of research that has garnered significant attention in recent years [1,14,15]. The geometric complexity and dynamic nature of the fluid interfaces demand advanced modeling techniques that can accurately resolve and track the evolving interfaces in three-dimensional space. Researchers are continuously exploring new approaches to improve the conservation of mass, capture fluid interfaces, and compute interfacial surface tension. Liovic et al. [14]

developed a volume tracking method for accurate calculations of interfaces in 3D geometries. The interface geometries, which are approximated as piecewise planar and fully multidimensional fluxes, are used in a single unsplit step to advect cell volumes. The definition of these fluxes is based on backward-trajectory remapping, ensuring the reliable and precise tracking of the interfaces. To assess the performance of the 3D volume tracking scheme, Liovic et al. [14] conducted 3D shearing flow and deformation field tests. Adelsberger et al. [15] conducted 3D simulations of incompressible two-phase flow to investigate the behavior of rising droplets. Three different flow solvers, namely, DROPS [16], NaSt3D [17], and OpenFOAM [18], were utilized, each employing distinct numerical techniques. DROPS employed a finite element spatial discretization, an implicit theta scheme for time discretization, and a level set method for interface capturing. NaSt3D utilized a finite difference spatial discretization, a second-order Adams-Bashforth scheme for time discretization, and a level set method for interface capturing. Lastly, OpenFOAM employed a finite volume spatial discretization, an implicit Euler scheme for time discretization, and a VOF method for interface capturing. Metivet et al. [1] presented a numerical framework for simulating multiphase flows using the level-set method. The discretization of the fluid equations is carried out using a finite element method. For the spatial discretization, the method can use finite elements of high order. This choice ensures accuracy and stability in representing the spatial domain. As for the time discretization, the implicit backward differentiation formulae (BDF) of arbitrary order is used. This approach allows for precise time stepping and facilitates the accurate simulation of dynamic phenomena in interfacial multiphase flows. This ongoing research aims to enhance the fidelity and predictive capabilities of computational models, enabling more accurate simulations of 3D interfacial multiphase flows, which generally suffer from volume conservation, interface sharpness, boundedness, and shape preservation.

In this study, we present a geometric formulation of the VOF method, known as PLIC (piecewise linear interface construction), specifically designed for simulating 3D interfacial multiphase flows. The PLIC-VOF method is implemented within the HiG-Flow (hierarchical grid) framework, comprising recently developed computational fluid dynamics (CFD) software. HiG-Flow utilizes finite differences on hierarchical grids and enables the simulation of a wide range of flow types, including Newtonian, generalized Newtonian, and viscoelastic flows in any dimension. The modular design of the software allows for the easy implementation of new techniques and methods, making it flexible and adaptable to evolving research needs. Furthermore, HiG-Flow incorporates numerical stabilization techniques to ensure accurate and stable simulations, even in the presence of complex flow phenomena, such as 3D multiphase flows. The user-friendly interface allows researchers to choose specific simulation features, such as the dimension, modules to be used, and numerical techniques, providing a customizable and efficient simulation environment.

The paper is organized as follows: in Section 2, we present the governing equations that describe interfacial multiphase flows. Subsequently, Section 3 describes the numerical method used to discretize the governing equations using the PLIC-VOF method. In Section 4, we focus on the validation of the numerical algorithm through two 3D advection tests: the shearing flow and deformation field problems. Furthermore, the validation of the PLIC-VOF method is also conducted using a 3D multiphase benchmark problem involving the rising of a bubble. The paper ends with the conclusion in Section 5.

## 2. Governing Equations

This study examines the characteristics of an unsteady, laminar, isothermal, and incompressible two-phase flow. The flow consists of two immiscible fluid phases, with no mass transfer occurring across their interface. The governing equations for this flow are the mass conservation equation (Equation (1)),

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

and the linear momentum balance equation (Equation (2)),

$$\rho \left[ \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}(\nabla \cdot \mathbf{u}) \right] = -\nabla p + \nabla \cdot \left\{ \mu \left[ \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right] \right\} + \rho \mathbf{g} + \sigma \kappa \mathbf{n} \delta, \quad (2)$$

where  $\mathbf{u} = (u, v, w)$  is the velocity field of the flow;  $\rho$  is the density,  $p$  is the pressure;  $\mu$  is the absolute viscosity;  $\mathbf{g}$  is the gravitational field;  $\sigma$  is the interfacial tension coefficient;  $\kappa$  and  $\mathbf{n}$  are the curvature and the normal vector to the interface, respectively; and  $\delta$  is a function that takes a value of one at the interface and zero elsewhere in the domain. Equation (2) can be written in dimensionless form as:

$$\rho \left[ \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}(\nabla \cdot \mathbf{u}) \right] = -\nabla p + \frac{1}{Re} \nabla \cdot \left\{ \mu \left[ \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right] \right\} + \rho \frac{\mathbf{g}}{g} + \frac{1}{Bo} \kappa \mathbf{n} \delta, \quad (3)$$

where the non-dimensional parameters  $Re = (\rho_1 g^{1/2} L^{3/2}) / \mu_1$  and  $Bo = (\rho_1 g L^2) / \sigma$  represent, respectively, the Reynolds number, which describes the ratio between inertial and viscous forces, and the Bond number (also known as the Eötvös number), which describes the ratio between gravitational force and surface tension. The parameters  $g$ ,  $\rho_1$ ,  $L$ , and  $\mu_1$  represent the magnitude of the gravitational field, the characteristic density, the characteristic length scale, and the characteristic viscosity, respectively. Note that, for the sake of conciseness, both dimensional and dimensionless variables are written with the same symbols. In addition, also note that the characteristic velocity  $V$  is intrinsically linked to gravity. In this context,  $V = \sqrt{g \cdot 2r} = \sqrt{g \cdot L} = g^{1/2} L^{1/2}$ . This velocity description can also be found in Hysing et al. [19].

In the VOF method, assuming the flow is incompressible, the advection of the volume fraction is modeled by the following transport equation:

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{u}f) = 0. \quad (4)$$

The method used to model the evolution of the interface plays a fundamental role in the accuracy of the VOF method. In HiG-Flow software, a geometric formulation was taken to solve the transport equation (Equation (4)), where the fluid–fluid interface is reconstructed at each time step using the PLIC-VOF method. In this sense, an advection algorithm with a fractional step (split) was implemented. This technique simplifies the method and prevents the transportation of the same fluid portion twice, as happens in single-step (unsplit) methods [20].

### 3. Numerical Procedure

#### 3.1. HiG-Tree and HiG-Flow Software

The HiG-Tree software is responsible for the data structure, domains, approximations, and interpolations, solvers of linear and non-linear systems, used in numerical simulations in HiG-Flow software.

The HiG-Tree data structure is a hierarchical Cartesian mesh representation that can be constituted of elements of varying sizes. This structure is based on the recursive spatial subdivisions (refinements) of its elements, enabling their representation in an m-tree (see Figure 1). In the HiG-Tree data structure, arbitrary mesh refinement processes can be applied at any level of the tree, generating an unstructured mesh. Classic examples of this type of mesh are the data structures Quadtree (two-dimensional) and Octree (three-dimensional). Although the illustration in Figure 1 is in two dimensions, the framework is designed to be generic and can be extended to any number of dimensions.

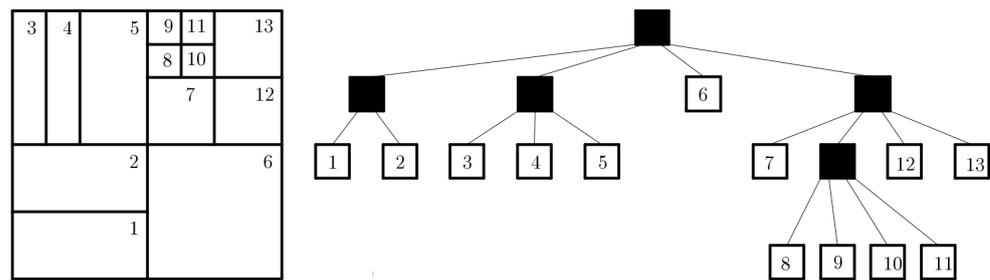


Figure 1. HiG-Tree data structure.

In the module responsible for domain and subdomain partitioning, the generated domains are composed of blocks that can be used to simulate problems in channels with intricate geometries, but this is complicated. Each block is discretized using the HiG-Tree structure, which enables appropriate spatial refinement for the simulated problems. These domains are partitioned using the Zoltan-Trilinos library [21,22], which ensures a good load distribution among processes during execution. This module also allows for the storage and enumeration of properties represented in the cells and facets of the mesh. The module for approximations and interpolations handles the estimation of properties within cells and facets. It allows for the selection of the polynomial approximation order utilized by the moving least squares (MLS) method, as described by Sousa et al. [23]. In terms of meshes composed of elements with different levels of refinement, numerical methods based on approximations of finite differences in Cartesian uniform meshes become limited since they will require spatial interpolations in unknown points of the stencil, as illustrated in Figure 2. The accuracy of this type of method heavily depends on the geometric characteristics of the mesh. The MLS method avoids this geometric dependence by using a cloud of neighboring points to the unknown to apply the interpolation.

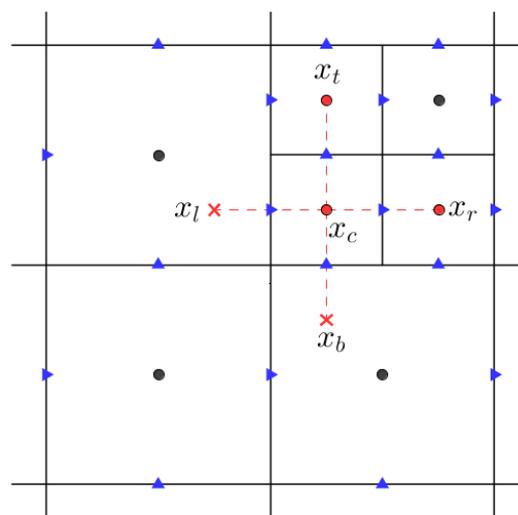


Figure 2. Stencil of the second-order finite difference method, where the unknowns are arranged at the center of the cells.  $x_c$ ,  $x_l$ ,  $x_r$ ,  $x_b$ , and  $x_t$  are approximations of values positioned at the center (c), left (l), right (r), bottom (b), and top (t) in the 5-point stencil.

To obtain an approximation of the second derivative of variable  $u$  in the  $x$  direction at point  $c$ , a standard second-order finite difference scheme can be used. The approximation is given by the following expression:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{1}{\Delta x^2} (u_l - 2u_c + u_r). \tag{5}$$

Note that the point  $u_l$  on the stencil in Figure 2 does not coincide with any point on the mesh and must be approximated by an interpolation technique on the mesh of unknowns in the neighborhood of  $u_c$ . In the HiG-Tree software, the following interpolation is used:

$$u_l = \sum_{k \in \mathcal{I}_l} w_k^l u_k, \tag{6}$$

where  $\mathcal{I}_l = i_k, k = 1, \dots, N_l$  is the set of indices for the unknowns that are in the neighborhood of  $u_c$ , and the weights  $w_k^l$  are calculated using the MLS method. The number of neighbors  $N_l$  depends on the number of points are needed to maintain the order of accuracy of the global approximation. For a detailed description of this interpolation, the reader is referred to Sousa et al. [23].

Lastly, the module designed for solving linear and non-linear systems consists of two powerful libraries that utilize parallel computation techniques for efficient solution. These libraries are specifically tailored for handling both shared and distributed memory architectures, making use of high-performance routines. These libraries are HYPRE (high-performance preconditioners) [24] and PETSc (portable, extensible toolkit for scientific computation) [25]. These libraries offer significant advantages in the development of parallel code, and their performance has been extensively optimized and tested in various computational fluid dynamics codes. HiG-Flow software allows for the numerical simulation of fluid flows in various configurations, specifically, it enables the simulation of single-phase flows of Newtonian, generalized Newtonian, and viscoelastic fluids, as well as two-phase flows of Newtonian fluids using the volume of fluid method (VOF) for interface representation. In HiG-Flow software, the most traditional methods for temporal advancement were implemented. The explicit methods implemented are explicit Euler, second-order TVD (total variation diminishing) Runge–Kutta (or modified Euler method), and third-order TVD Runge–Kutta. The implicit methods implemented in the system are implicit Euler, Crank–Nicholson, and the second-order backward differentiation formula (BDF). To obtain the pressure and velocity fields, the incremental (second order in time) and non-incremental (first order in time) projection methods are implemented.

### 3.2. Discretization Procedure in HiG-Flow

To obtain the pressure fields and flow velocity through the resolution of the Navier–Stokes Equations (1) and (3), the incremental and non-incremental projection methods proposed by Chorin [26], in which the pressure and velocity fields are decoupled, are implemented in HiG-Flow. As stated by Lages [27], in terms of the error arising from the pressure-velocity decoupling, it has been observed that the non-incremental version of the method exhibits first-order accuracy in time, whereas the incremental version achieves second-order accuracy in time. In this work, the incremental projection method was used to solve the equations. Therefore, discretizing Equations (1) and (3) by the explicit Euler method we obtain:

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \tag{7}$$

$$\begin{aligned} \frac{\rho^{n+1} \mathbf{u}^{n+1} - \rho^n \mathbf{u}^n}{\Delta t} &= -\rho^n (\mathbf{u}^n \cdot \nabla \mathbf{u}^n) - \nabla p^{n+1} \\ &+ \frac{1}{Re} \nabla \cdot \left\{ \mu \left[ \nabla \mathbf{u}^n + (\nabla \mathbf{u}^n)^T \right] \right\} + \mathbf{F}^n, \end{aligned} \tag{8}$$

where

$$\mathbf{F}^n = \rho^n \frac{\mathbf{g}}{g} + \frac{1}{Bo} (\kappa \mathbf{n} \delta)^n. \tag{9}$$

The superscripts  $n$  and  $n + 1$  refer to the time level, and  $\Delta t$  represents the time step. The elements with superscript  $n + 1$  are unknowns, and the elements with superscript  $n$  are known.

In the incremental projection method, initially using  $p^n$  instead of  $p^{n+1}$ , an intermediate velocity,  $\mathbf{u}^*$ , is calculated as follows:

$$\frac{\rho^* \mathbf{u}^* - \rho^n \mathbf{u}^n}{\Delta t} = -\rho^n (\mathbf{u}^n \cdot \nabla \mathbf{u}^n) - \nabla p^n + \frac{1}{Re} \nabla \cdot \left\{ \mu \left[ \nabla \mathbf{u}^* + (\nabla \mathbf{u}^*)^T \right] \right\} + \mathbf{F}^n, \quad (10)$$

Subsequently, the difference between the pressure gradients  $p^n$  and  $p^{n+1}$  is calculated to correct the intermediate velocity, and, lastly, the final velocity  $\mathbf{u}^{n+1}$  is obtained:

$$\frac{\rho^{n+1} \mathbf{u}^{n+1} - \rho^n \mathbf{u}^*}{\Delta t} = -\nabla (p^{n+1} - p^n) = -\nabla \Delta p, \quad (11)$$

or rather,

$$\rho^{n+1} \mathbf{u}^{n+1} = \rho^n \mathbf{u}^* - \Delta t \nabla \Delta p. \quad (12)$$

To calculate the pressure, the divergent operator is applied on both sides of Equation (12) and the incompressibility hypothesis, Equation (7), is invoked, to give the Poisson equation for pressure:

$$\nabla \cdot \left( \frac{1}{\rho^n} \nabla \Delta p \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*. \quad (13)$$

Since the phenomena treated in this work involve two-phase flows, that is, they involve two fluids with different physical properties, it follows that  $\rho$  and  $\mu$  are not constant throughout the domain. Thus, the resolution of the Poisson equation for the pressure and momentum equation presents greater difficulties and requires special care. To remediate this issue, in this work, the BiCGS (BiConjugate Gradient Squared) iterative method [28] with Block Jacobi preconditioner [29] was used, considering the convergence tolerance on the residual norm of  $1 \times 10^{-7}$ .

### 3.3. Interface Reconstruction

In the existing literature, various methods have been proposed in conjunction with the VOF technique to handle the geometric reconstruction of the interface. These methods aim to locally approximate the interface within each interfacial cell by utilizing the volume fraction as a fundamental piece of information. Traditional examples of methods for the reconstruction of interfaces are the SLIC (simple line interface calculation) [30], Hirt–Nichols–VOF (original VOF method) [31] and PLIC (piecewise-linear interface calculation) [32]. A comparison of these methods was performed by Rudman [33], where the PLIC method showed more consistent results. Therefore, the PLIC method was implemented in the HiG-Flow solver to represent the interface.

In the PLIC-VOF method, assuming that the volume fraction and the normal vector are known in each interfacial cell, the interface is approached by a plane defined as follows:

$$\mathbf{n} \cdot \mathbf{x} = \alpha, \quad (14)$$

where  $\mathbf{n}$  is the normal vector over the interface,  $\mathbf{x}$  is the position of a point on the plane, and  $\alpha$  is the smallest distance from the plane to the origin of the computational cell. The distance  $\alpha$  must be adjusted so that the volume fraction below the reconstructed plane (say,  $\bar{f}$ ) is equal to  $f$ , i.e.,

$$\bar{f}(\alpha, \mathbf{n}, \Delta) - f = 0. \quad (15)$$

Given a cell with dimension  $\Delta = (\Delta x, \Delta y, \Delta z)$  and the normal vector,  $\mathbf{n} = (n_x, n_y, n_z)$ , the fluid volume below the interface within the interfacial cell is defined as follows:

$$\begin{aligned} \bar{V} = & \frac{1}{6|n_x||n_y||n_z|} [\alpha^3 \\ & - R(\alpha - |n_x|\Delta x) \cdot (\alpha - |n_x|\Delta x)^3 \\ & - R(\alpha - |n_y|\Delta y) \cdot (\alpha - |n_y|\Delta y)^3 \\ & - R(\alpha - |n_z|\Delta z) \cdot (\alpha - |n_z|\Delta z)^3 \\ & + R(\alpha - |n_x|\Delta x - |n_y|\Delta y) \cdot (\alpha - |n_x|\Delta x - |n_y|\Delta y)^3 \\ & + R(\alpha - |n_x|\Delta x - |n_z|\Delta z) \cdot (\alpha - |n_x|\Delta x - |n_z|\Delta z)^3 \\ & + R(\alpha - |n_y|\Delta y - |n_z|\Delta z) \cdot (\alpha - |n_y|\Delta y - |n_z|\Delta z)^3], \end{aligned} \tag{16}$$

where

$$R(a) = \begin{cases} 0, & \text{if } a \leq 0, \\ 1, & \text{if } a > 0. \end{cases} \tag{17}$$

Equation (16) can be simplified based on the intersection of the plane with the cell, allowing one to estimate the distance from the plane to the origin of the cell and the volume of fluid below the plane. To calculate the normal vector, the second-order accurate method described by Mehta et al. [34] was used. In this method, the normal vector is defined as the gradient of the volume fraction (i.e.,  $\mathbf{n} = \nabla f / \|\nabla f\|$ ), where the gradient is approximated using a  $3 \times 3 \times 3$  stencil around each interfacial cell.

### 3.4. Interface Advection

Following the same strategy that Puckett et al. [35] adopted in the two-dimensional case, in order to maintain the conservation of  $f$ , the incompressibility condition was taken into account (i.e.,  $\nabla \cdot \mathbf{u} = 0$ ), resulting in the following modification to the volume fraction transport equation (Equation (4)):

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{u}f) = f \nabla \cdot \mathbf{u}, \tag{18}$$

or rather,

$$\frac{\partial f}{\partial t} + \nabla \cdot (uf) = f \nabla \cdot u, \tag{19}$$

$$\frac{\partial f}{\partial t} + \nabla \cdot (vf) = f \nabla \cdot v, \tag{20}$$

$$\frac{\partial f}{\partial t} + \nabla \cdot (wf) = f \nabla \cdot w. \tag{21}$$

According to Rider and Kothe [36], who also presented this strategy for the two-dimensional case, this “divergence correction” satisfies the local and global volume-filling constraints much more closely since the discrete velocity divergences are not necessarily zero but rather a function of the convergence tolerance used in obtaining the solutions of the linear system.

The calculation of the transport of the volume fraction is divided into three steps: the first in the  $x$  direction, the second is in the  $y$  direction, and the third is in the  $z$  direction. Thus, three consecutive updates are needed to carry  $f$  at each time step, i.e.,  $f^n \rightarrow f^* \rightarrow f^{**} \rightarrow f^{n+1}$ . Thus, discretizing  $f$  implicitly on the right-hand side of

Equation (19), implicitly on the right-hand side of Equation (20), and explicitly on the right-hand side of Equation (21) leads to

$$f_{i,j,k}^* = \frac{f_{i,j,k}^n + \frac{\Delta t}{\Delta x} (F_{i-1/2,j,k}^n - F_{i+1/2,j,k}^n)}{1 - \frac{\Delta t}{\Delta x} (u_{i+1/2,j,k}^n - u_{i-1/2,j,k}^n)}, \tag{22}$$

$$f_{i,j,k}^{**} = \frac{f_{i,j,k}^* + \frac{\Delta t}{\Delta y} (G_{i,j-1/2,k}^* - G_{i,j+1/2,k}^*)}{1 - \frac{\Delta t}{\Delta y} (v_{i,j+1/2,k}^n - v_{i,j-1/2,k}^n)}, \tag{23}$$

$$f_{i,j,k}^{n+1} = f_{i,j,k}^{**} + \frac{\Delta t}{\Delta z} [f_{i,j,k}^{**} (w_{i,j,k+1/2}^n - w_{i,j,k-1/2}^n) - (H_{i,j,k+1/2}^{**} - H_{i,j,k-1/2}^{**})], \tag{24}$$

where  $f_{i,j,k}^n$  is the volume fraction in the  $(i, j, k)$ -th cell; at time  $t^n = n\Delta t$ ,  $u$ ,  $v$ , and  $w$  are the velocity components in each coordinate direction and  $F$ ; and  $G$  and  $H$  denote the fluxes of  $f$  in the facets of cell  $(i, j, k)$ . Suppose, without loss of generality, that the velocity  $u_{i+1/2,j,k}$  is positive; then, the flux volume,  $F_{i+1/2,j,k}^n$ , is given by:

$$F_{i+1/2,j,k}^n = \frac{u_{i+1/2,j,k}^n \cdot \bar{V}_{i+1/2,j,k}}{u_{i+1/2,j,k}^n \cdot \Delta t \cdot \Delta y \cdot \Delta z} = \frac{\bar{V}_{i+1/2,j,k}}{\Delta t \cdot \Delta y \cdot \Delta z}, \tag{25}$$

where  $\bar{V}_{i+1/2,j,k}$  is the volume of fluid that must be transported to the neighboring cell.

### 3.5. Surface Tension Force and Curvature

In this work, the CSF (continuum surface force) method presented by Brackbill et al. [37] was adopted to model the interfacial tension force described in the momentum equation (Equation (3)). In this model, the body force is defined by:

$$F_\sigma = \sigma \kappa \mathbf{n}\delta. \tag{26}$$

In the VOF method, Equation (26) is generally rewritten as:

$$F_\sigma = \sigma \kappa \frac{\nabla f}{\|\nabla f\|} \|\nabla f\| = \sigma \kappa \nabla f. \tag{27}$$

To obtain the curvature  $\kappa$ , the traditional height function method, described by Torrey et al. [38], was used. This method takes nine heights to form a stencil  $7 \times 3 \times 3$ ,  $3 \times 7 \times 3$  or  $3 \times 3 \times 7$  around each interfacial cell, according to the orientation of the interface, where the construction of height functions is defined as follows:

$$h_{i,j} = \sum_{k=3}^{k+3} f_{i,j,k} \Delta z, \quad \text{to } j = j' - 1, j', j' + 1 \quad \text{and } i = i' - 1, i', i' + 1, \tag{28}$$

for the case where the largest component of the normal vector is in the  $z$  direction. Knowing the nine height functions, the interfacial curvature can be calculated as follows:

$$\kappa = \text{sgn}(\mathbf{n}) \cdot \frac{h_{xx} + h_{yy} + h_{xx}h_y^2 + h_{yy}h_x^2 - 2h_{xy}h_xh_y}{(1 + h_x^2 + h_y^2)^{3/2}}, \tag{29}$$

where the derivatives  $h_x$ ,  $h_y$ ,  $h_{xx}$ ,  $h_{yy}$ , and  $h_{xy}$  are calculated using centered finite difference schemes.

### 3.6. Time-Step Constraint

During the geometric advection of the interface, it is necessary to consider certain constraints to ensure the proper functioning of the method. In HiG-Flow, in order to

prevent the same fluid portion from being transported twice, due to the use of a fractional step advection algorithm, and to ensure that the advected fluid volume does not exceed the computational cell volume, the following condition has been imposed:

$$\Delta t < \frac{\Delta}{2|\mathbf{u}|} \equiv \Delta t_{Advection}. \quad (30)$$

The inclusion of the interfacial tension term in the Navier–Stokes equation demands one more restriction for the time step. According to Popinet [7], the explicit discretization of the interfacial tension term leads to a condition of temporal stability of the form:

$$\Delta t < \sqrt{\frac{(\rho_1 + \rho_2)\Delta^3}{4\pi\sigma}} \equiv \Delta t_{\sigma}, \quad (31)$$

where  $\rho_1$  and  $\rho_2$  represent the density of each fluid. According to the rationale provided by Popinet [7], the limitation on the time step size is justified by the need to adequately capture the propagation of the fastest capillary wave within the system.

Furthermore, the CFL (Courant–Friedrichs–Lewy) condition should be taken into account for the parabolic and hyperbolic terms in Equation (8),

$$\Delta t < Re\Delta^2 \equiv \Delta t_{Parabolic}, \quad (32)$$

and

$$\Delta t < \frac{\Delta}{\rho|\mathbf{u}|} \equiv \Delta t_{Hyperbolic}. \quad (33)$$

Therefore, the time step constraint is:

$$\Delta t = \min(\Delta t_{Advection}, \Delta t_{\sigma}, \Delta t_{Parabolic}, \Delta t_{Hyperbolic}). \quad (34)$$

#### 4. Validation

In this section, we present the validation of the numerical results obtained using HiG-Flow software by comparing them with the results reported in the scientific literature. The purpose is to assess the accuracy and reliability of HiG-Flow software by evaluating its agreement with established findings. To assess the performance of the implemented interface reconstruction and advection methods, benchmark problems were simulated. These benchmark problems, namely, the 3D shearing flow test and the 3D deforming field test, were chosen based on their relevance and availability in the literature, particularly in the work by Liovic et al. [14]. In these tests, since the velocity field is given, the sphere is not affected by the numerical dissipation of the methods used to obtain the flow velocity and pressure. In other words, the Navier–Stokes equations do not need to be solved, just the volume fraction transport equation. Therefore, it is expected that the sphere maintains its original shape at the final time of the simulation. By comparing the results obtained from the HiG-Flow simulations with the established solutions, the accuracy and fidelity of the implemented methods can be evaluated. Furthermore, the solver for the Navier–Stokes equations developed in HiG-Flow was validated using the 3D rising drop test proposed by Adelsberger et al. [15]. This test serves as a benchmark to assess the accuracy and performance of the solver in simulating the rising behavior of a three-dimensional droplet. Both quantitative and qualitative analyses of the results obtained from the simulations of these benchmark problems are presented in the following sections.

To quantitatively evaluate the performance of the advection algorithm implemented in HiG-Flow, the volume of fluid at the initial time ( $V_I$ ) and the volume of fluid at the final time ( $V_F$ ) of the simulation were compared for both the 3D shearing flow test and the 3D deforming field test. The absolute error,  $\epsilon_{L_1}$ , between the volumes is defined as follows:

$$\epsilon_{L_1} = |V_I - V_F|. \quad (35)$$

To quantitatively evaluate the 3D rising drop test, the evolution of the bubble’s center of mass and the bubble’s rise velocity were analyzed. The evolution of the bubble’s center of mass is given by:

$$z_c = \int_{\Omega_1} z \, dz, \tag{36}$$

where  $\Omega_1$  is the region of the domain filled by the bubble. By utilizing the volume fraction  $f$ , Equation (36) can be expressed in a different form, as follows:

$$z_c = \sum_{i=1}^N z_i \cdot f \cdot \Delta x \Delta y \Delta z, \tag{37}$$

where  $N$  is the number of cells that contain a non-zero volume fraction.

Lastly, the bubble’s rise velocity is given by:

$$v_c = \frac{d}{dt} z_c. \tag{38}$$

#### 4.1. 3D Shearing Flow Test

The 3D shearing flow test is widely used in the literature to verify the accuracy of interface advection algorithms in two-phase flows [14,39,40]. In this test, a sphere is subjected to a prescribed velocity field defined by the combination of a single vortex in the  $x - y$  plane and laminar pipe flow in the  $z$  direction.

To simulate this test, a sphere with a radius of 0.15 was considered, centered at the point (0.5, 0.75, 0.25), in a domain with dimensions of  $[0, 1] \times [0, 1] \times [0, 2]$ . The velocity field used to move the sphere within the domain is defined by:

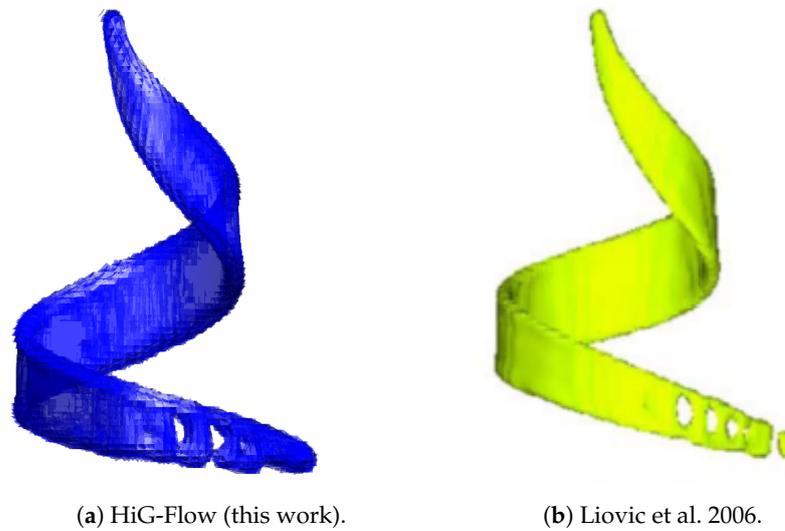
$$\mathbf{u} = \begin{cases} u = -\sin(\pi x)^2 \sin(2\pi y) \cos(\frac{\pi t}{T}), \\ v = \sin(\pi y)^2 \sin(2\pi x) \cos(\frac{\pi t}{T}), \\ w = u_{max} (1 - \frac{r}{R})^2 \cos(\frac{\pi t}{T}), \end{cases} \tag{39}$$

where  $T = 6.0$ ,  $r = \sqrt{(x - 0.5)^2 + (y - 0.5)^2}$ ,  $R = 0.5$  and  $u_{max} = 1.0$ . This velocity field generates a vortical flow, causing the initially spherical interface to take on a spiral shape. After reaching half of the total simulation time (i.e.,  $t = 3.0$ ), the interface reverses its trajectory, returns to its initial position at time  $t = 6.0$ , and reverts to its original shape.

Figure 3a shows the point of maximum interface deformation in the 3D shear flow test simulated in HiG-Flow. A mesh with a resolution of  $64 \times 64 \times 128$  and time-step of  $\Delta t = 1 \times 10^{-3}$  were used in the numerical simulations. The result obtained by HiG-Flow is compared with the findings presented by Liovic et al. [14], as shown in Figure 3b, where a mesh with a resolution of  $64 \times 64 \times 128$  was also used. The interface shape calculated by HiG-Flow demonstrates qualitative agreement with the results obtained by Liovic et al. [14].

To perform a quantitative analysis of the 3D shearing flow test, the volume fraction absolute error was calculated using Equation (35). The obtained result was then compared with the values reported in the scientific literature at  $t = 6$ . This comparison is shown in Table 1, where HiG-Flow software showed superiority, in terms of volume fraction conservation, when compared to the results presented by Liovic et al. [14], Duz et al. [39], and Lopez et al. [40].

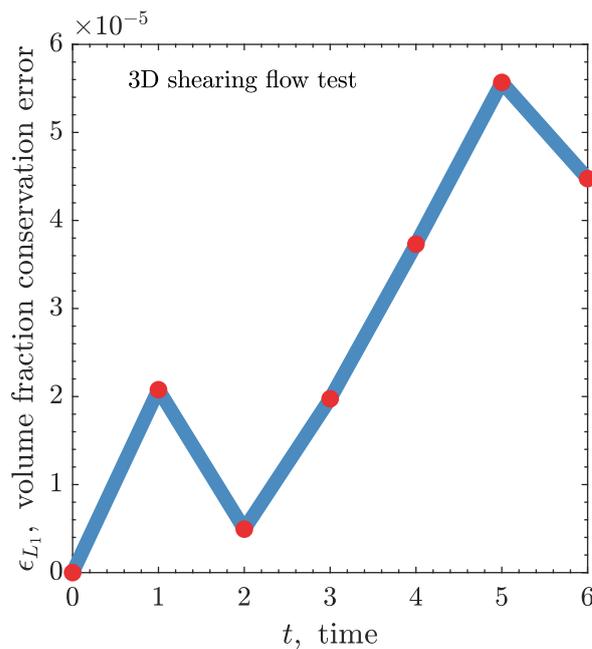
In addition, Figure 4 shows the time history of the volume fraction variation for the 3D shearing flow test when using mesh  $64 \times 64 \times 128$ . It can be seen that the  $L_1$  errors magnitude is in the range of  $10^{-6}$  to  $10^{-5}$  during all the simulation time, thus proving the reliability of the numerical method developed in this work.



**Figure 3.** Comparison of the interface shape at time  $t = 3$  for the 3D shearing flow test. Results obtained using (a) HiG-Flow software and (b) PCFSC (piecewise-constant flux surface calculation) unsplit advection scheme presented by Liovic et al. [14], both with a mesh of  $64 \times 64 \times 128$  elements.

**Table 1.** Comparison of the  $L_1$  errors for the volume fraction conservation in the 3D shearing flow test at  $t = 6$ . The HiG-Flow errors are compared against the ones reported by Liovic et al. [14], Duz et al. [39], and Lopez et al. [40].

Solver	Mesh	Error ( $\epsilon_{L_1}$ )
HiG-Flow	$50 \times 50 \times 100$	$2.86 \times 10^{-4}$
HiG-Flow	$64 \times 64 \times 128$	$4.48 \times 10^{-5}$
CVTNA + unsplit [14]	$64 \times 64 \times 128$	$3.64 \times 10^{-3}$
ELVIRA + COSMIC [39]	$64 \times 64 \times 128$	$3.97 \times 10^{-3}$
CLC-CBIR [40]	$64 \times 64 \times 128$	$3.27 \times 10^{-3}$



**Figure 4.** Transient behaviour of the  $L_1$  errors,  $\epsilon_{L_1}$ , for the volume fraction conservation in the 3D shearing flow test using mesh  $64 \times 64 \times 128$ .

### 4.2. 3D Deforming Field Test

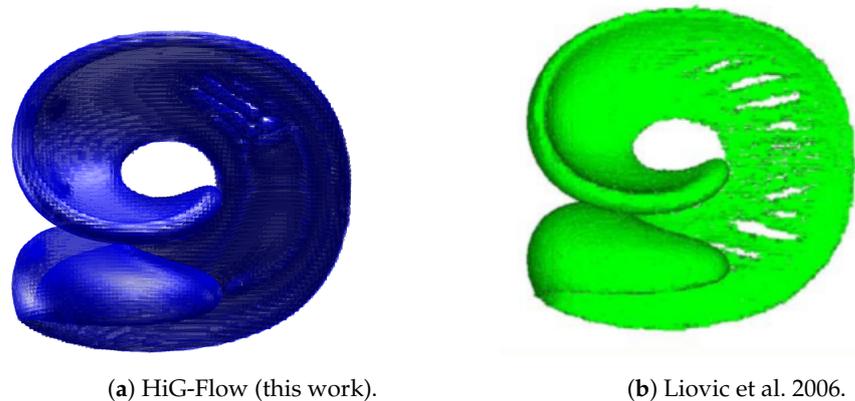
The 3D deforming field test is also widely used in the literature to verify interface advection methods [14,40] as the prescribed velocity field causes the interface to assume a complex geometry during its motion. To perform this test, a sphere with a radius of 0.15 was considered, centered at the point (0.35, 0.35, 0.35), within a cubic cavity with dimensions  $[0, 1] \times [0, 1] \times [0, 1]$ . The velocity field used to move the sphere within the domain is defined by:

$$\mathbf{u} = \begin{cases} u = 2\sin(\pi x)^2\sin(2\pi y)\sin(2\pi z)\cos(\pi t/T), \\ v = -\sin(\pi y)^2\sin(2\pi x)\sin(2\pi z)\cos(\pi t/T), \\ w = -\sin(\pi z)^2\sin(2\pi x)\sin(2\pi y)\cos(\pi t/T), \end{cases} \quad (40)$$

where  $T = 3.0$ .

Similarly to the 3D shearing flow test studied in Section 4.1, the prescribed solenoidal velocity field causes the sphere to deform until reaching half of the total simulation time and then follows a reverse trajectory, tending back to its original shape and position.

Figure 5a shows the point of maximum interface deformation in the 3D deforming field test simulated in HiG-Flow. A mesh with a resolution of  $128^3$  and time-step of  $\Delta t = 1 \times 10^{-3}$  was used in the numerical simulations. The result is compared with the findings presented by Liovic et al. [14], as shown in Figure 5b, where a mesh with a resolution of  $128^3$  was also used. The interface shape obtained by HiG-Flow exhibits qualitative agreement with the results obtained by Liovic et al. [14], with fewer regions of void volume fraction compared to their results, despite using the same level of mesh refinement.



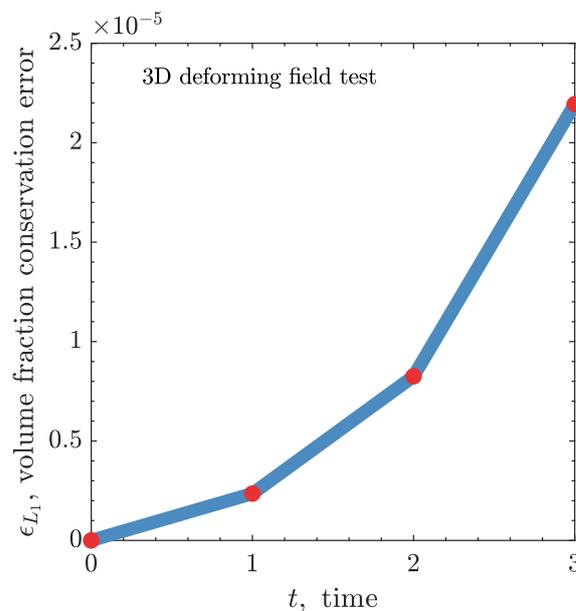
**Figure 5.** Comparison of the interface shape at time  $t = 3$  for the 3D deforming field test. Results obtained using (a) HiG-Flow software and (b) PCFSC (piecewise-constant flux surface calculation) unsplit advection scheme presented by Liovic et al. [14], both using a mesh constituted of  $128^3$  elements.

Analogously to the previous test shown in Section 4.1, we quantitatively analyze the results obtained here for the 3D deforming field test. For that purpose, the volume fraction absolute error was calculated using Equation (35) in order to verify the ability of the volume tracking scheme to conserve volume. The results obtained by our newly developed multiphase algorithm were compared with the results presented by Liovic et al. [14] and by Lopez et al. [40], as shown in Table 2. Across all levels of mesh refinement utilized, our numerical formulation consistently yields lower errors compared to those reported in references [14,40]. These results support the methodology of developing an interface advection scheme with a divergence correction approach inside the volume fraction transport equation (see right-hand side of Equation (18)), which leads to the closer satisfaction of both local and global volume-filling constraints.

**Table 2.** Comparison of the  $L_1$  errors for the volume fraction conservation in the 3D deformation field test. The HiG-Flow errors are compared against the ones reported by Liovic et al. [14] and Lopez et al. [40].

Solver	Mesh	Error ( $\epsilon_{L_1}$ )
HiG-Flow	$50 \times 50 \times 50$	$1.15 \times 10^{-4}$
HiG-Flow	$64 \times 64 \times 64$	$2.19 \times 10^{-5}$
HiG-Flow	$128 \times 128 \times 128$	$9.41 \times 10^{-6}$
CVTNA + PCFSC unsplit [14]	$64 \times 64 \times 64$	$1.99 \times 10^{-3}$
CVTNA + PCFSC unsplit [14]	$128 \times 128 \times 128$	$3.09 \times 10^{-4}$
CLC-CBIR [40]	$64 \times 64 \times 64$	$2.09 \times 10^{-3}$
CLC-CBIR [40]	$128 \times 128 \times 128$	$3.52 \times 10^{-4}$

Lastly, Figure 6 shows the time history of the volume fraction variation for the 3D deforming field test when using mesh  $64 \times 64 \times 64$ . Again, we can conclude that the  $L_1$  errors magnitude is in the range of  $10^{-6}$  to  $10^{-5}$  during all the simulation time, thus proving the reliability of the numerical method developed in this work.



**Figure 6.** Transient behaviour of the  $L_1$  errors,  $\epsilon_{L_1}$ , for the volume fraction conservation in the 3D deforming field test using mesh  $64 \times 64 \times 64$ .

### 4.3. 3D Buoyancy-Driven Rising Drop

To conclude the validation of the newly implemented numerical algorithm for simulating 3D multiphase flows, we now turn our attention to the problem of the buoyancy-driven rise of a bubble immersed in a Newtonian fluid. In this test case, we consider a bubble with a density lower than the fluid filling the column. As a result, the pressure gradient induced by gravity causes the bubble to move in the direction opposite to the gravitational field.

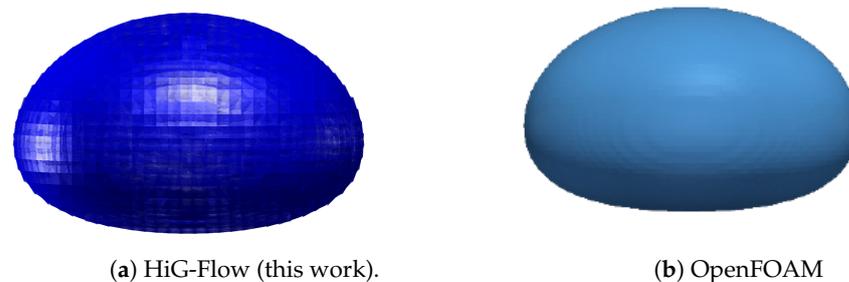
A bubble with a radius of  $R = 0.25$  was positioned at the point  $(0.5, 0.5, 0.5)$  inside a tank with dimensions of  $[0, 1] \times [0, 1] \times [0, 2]$ . The no-slip boundary condition was imposed on the domain walls. The simulations were performed until time reached  $t = 3.0$ . A mesh size of  $50 \times 50 \times 100$  elements and a time-step of  $\Delta t = 2.0 \times 10^{-3}$  were employed in the simulations. The two test cases proposed by Adelsberger et al. [15] were simulated. In the first case, the density and viscosity ratios between the two fluids in the tank are both equal to 10 ( $\rho_1/\rho_2 = \mu_1/\mu_2 = 10$ ). In the second case, the density ratio between the fluids is 1000 ( $\rho_1/\rho_2 = 1000$ ) and the viscosity ratio is 100 ( $\mu_1/\mu_2 = 100$ ). In test case 1, the droplet undergoes a transformation into an ellipsoidal shape. On the other hand, in test

case 2, the droplet has the potential to break up due to the lower surface tension. Test case 1 can be categorized as an ellipsoidal regime, while test case 2 falls somewhere between the skirted and dimpled ellipsoidal-cap regimes. The physical and dimensionless parameters considered for simulating these two cases are presented in Table 3. These parameters were also used by Metivet et al. [1] with high-order finite-element software, so-called FEEL.

**Table 3.** Physical and dimensionless parameters used on the 3D buoyancy-driven rising drop simulation.

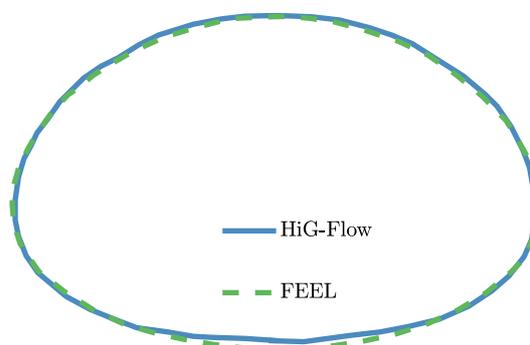
Test Cases	$\rho_1$	$\rho_2$	$\mu_1$	$\mu_2$	$g$	$\sigma$	$Re$	$Bo$
case 1	1000	100	10	1	0.98	24.5	35	10
case 2	1000	1	10	0.1	0.98	1.96	35	125

Figure 7 illustrates a comparison of the bubble shapes at time  $t = 3$  simulated using HiG-Flow and OpenFOAM, for the test case 1 (ellipsoidal regime). The interface shape obtained by HiG-Flow exhibits qualitative agreement with the result obtained by Adelsberger et al. [15] using OpenFOAM software, where the droplet reaches a stable ellipsoidal shape at steady-state regime.



**Figure 7.** Comparison of the bubble shape at time  $t = 3$  for the test case 1 of the 3D buoyancy-driven rising drop problem. The results were compared between (a) HiG-Flow and (b) OpenFOAM (Adelsberger et al. [15]) software.

For a better comparison of the actual droplet shape, Figure 8 illustrates the two-dimensional contour through the droplet’s center generated by the HiG-Flow (this work) and FEEL [1] software at time  $t = 3$ . The contour lines of HiG-Flow (blue) and FEEL (green) exhibit a close resemblance to each other, with a small difference at the bottom of the drop.

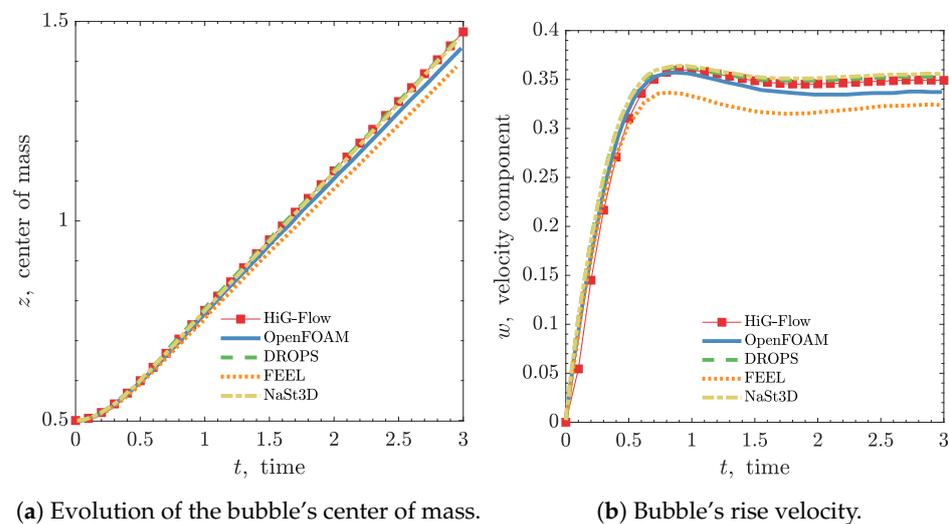


**Figure 8.** Comparison of the bubble contours at time  $t = 3$  for the test case 1 of the 3D buoyancy-driven rising drop problem. The results were compared between HiG-Flow (blue) and FEEL (green) software.

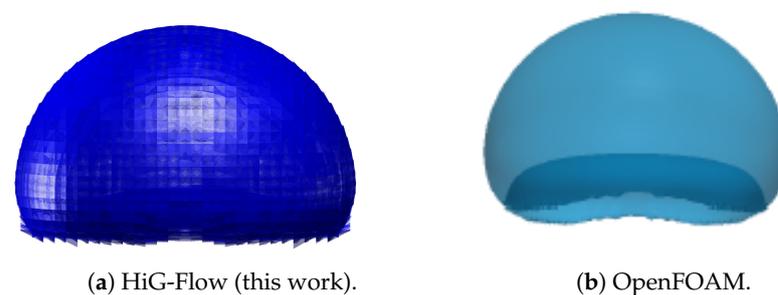
Figure 9 shows the evolution of the bubble’s center of mass and bubble’s rise velocity for test case 1. The results obtained using the newly-developed algorithm in HiG-Flow are compared to the results computed by FEEL [1], as well as OpenFOAM, DROPS, and NaSt3D [15]. In Figure 9a, at the initial stage of the simulation, up until approximately  $t \approx 1$ ,

the drop position is nearly identical in all simulations. Afterwards, the vertical position of the droplet in OpenFOAM and FEEL software is lower than the one obtained in HiG-Flow, DROPS, and NaSt3D. Our results are in excellent agreement with the findings obtained by DROPS and NaSt3D. Figure 9b illustrates the rise velocity of the droplet. In the initial stages of the simulation, the vertical velocity component ( $w$ ) of the fluid velocity gradually increases from its initial value of zero, reaching a maximum magnitude of approximately 0.35–0.36 at around  $t \approx 0.9$ . Then, the steady-state velocity of the droplet decreases, being again in agreement with the findings obtained by DROPS and NaSt3D.

Figure 10 illustrates a comparison of the bubble shapes at time  $t = 3$  simulated using HiG-Flow and OpenFOAM, for the test case 2 (skirted and dimpled ellipsoidal-cap regimes). Again, the interface shape obtained by HiG-Flow exhibits qualitative agreement with the result obtained by Adelsberger et al. [15] using OpenFOAM software, where the droplet reaches a stable cap shape at steady-state regime. In the simulated time, our results do not indicate the presence of satellite droplets, contrary to the OpenFOAM results, which predict 8 satellite droplets, as stated in Adelsberger et al. [15].



**Figure 9.** 3D buoyancy-driven rising drop: Test case 1 (ellipsoidal regime). (a) Bubble’s center of mass  $z$  and (b) bubble’s rise velocity component  $w$ .

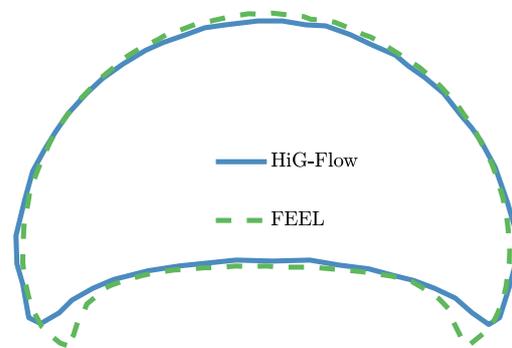


**Figure 10.** Comparison of the bubble shape at time  $t = 3$  for the test case 2 of the 3D buoyancy-driven rising drop problem. The results were compared between (a) HiG-Flow and (b) OpenFOAM (Adelsberger et al. [15]) software.

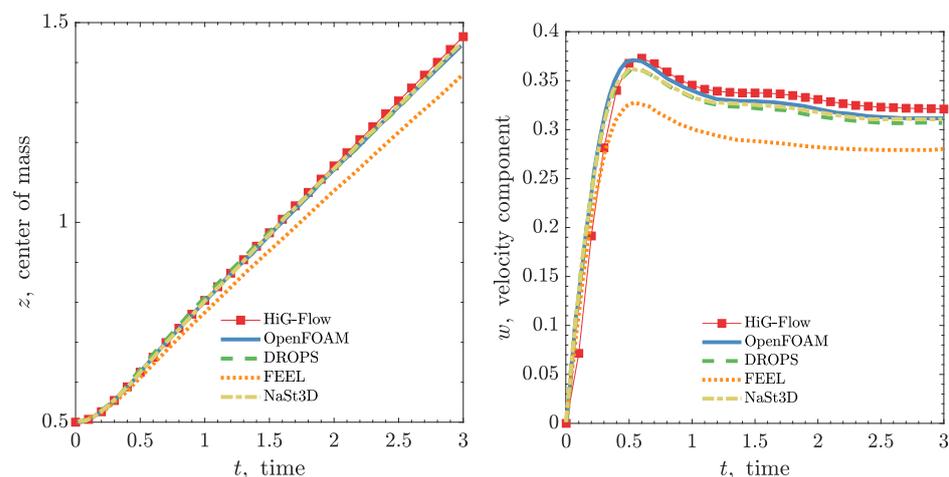
Figure 11 illustrates the comparison between two-dimensional contours through the droplet’s center generated by the HiG-Flow (this work) and FEEL [1] software at time  $t = 3$ . The shape of the contour lines of the bubbles obtained by both software are similar and reach a skirted format since the surface tension is lower and the ratio between the densities and viscosities between the fluids present in the flow is higher. The differences between the two simulations are particularly noticeable along the bottom edge of the droplet, where the curvature is most pronounced. As depicted in the contour plot, the contour line in FEEL

(green) appears more extended than in HiG-Flow. Furthermore, FEEL exhibits a higher contour line at the top edge of the droplet and a smaller contour line at the central part of the bottom side, when compared to that one obtained with HiG-Flow software.

Lastly, Figure 12 shows the evolution of the bubble’s center of mass and bubble’s rise velocity for test case 2. In Figure 12a, we show that the droplet’s position over time is similar for all the simulation times in all the flow solvers, except for FEEL software, which can explain the differences shown in the contour plot in Figure 11. Figure 12b illustrates the rise velocity of the droplet. In the initial stages of the simulation, the vertical velocity component ( $w$ ) of the fluid velocity gradually increases from its initial value of zero, reaching a maximum magnitude of approximately 0.37 at around  $t \approx 0.54$ . Then, the steady-state velocity of the droplet decreases so that a final velocity of 0.32 at  $t = 3$  is obtained using HiG-Flow software. Our result is similar to OpenFOAM, DROPS, and NaSt3D software, which predict a final velocity of 0.31. The final velocity predicted by FEEL is approximately 0.28, showing the most significant difference compared to the velocities obtained from other software.



**Figure 11.** Comparison of the bubble contours at time  $t = 3$  for the test case 2 of the 3D buoyancy-driven rising drop problem. The results were compared between HiG-Flow (blue) and FEEL (green) software.



(a) Evolution of the bubble’s center of mass. (b) Bubble’s rise velocity.

**Figure 12.** 3D buoyancy-driven rising drop: Test case 2 (skirted and dimpled ellipsoidal-cap regimes). (a) Center of mass  $z$  and (b) rise velocity component  $w$ .

### 5. Conclusions

In conclusion, this study presents a newly developed numerical method for simulating two-phase flows of incompressible, immiscible, and isothermal Newtonian fluids. The geometric volume of fluid (VOF) method is employed to locate and reconstruct the interface during the dynamics of the two-phase flow. Specifically, the piecewise-linear interface

calculation (PLIC) technique is implemented for three-dimensional (3D) interface transport. This technique approximates the interface in each interfacial cell using line segments. The methodology of developing an interface advection scheme with a divergence correction approach inside the volume fraction transport equation leads to a closer satisfaction of both local and global volume-filling constraints. In addition, the continuum interfacial force (CSF) model is utilized to account for the balance of forces at the interface. Lastly, the Navier–Stokes equations are solved using meshless (possibly high-order) finite difference schemes from HiG-Flow computational fluid dynamics software.

To validate the numerical implementation, several benchmark two-phase flows are simulated with the 3D PLIC-VOF HiG-Flow algorithm. Two standard advection numerical tests, namely, the 3D shearing flow test and the 3D deforming field test, are conducted, and the following conclusions can be devised:

- Good agreement is observed between the 3D interface shapes obtained by the 3D PLIC-VOF HiG-Flow algorithm and those found in the scientific literature, specifically, the piecewise-constant flux surface calculation, the volume of fluid method implemented in OpenFOAM, and the high-order finite-element software FEEL.
- The absolute errors of the volume tracking advection calculation obtained using our algorithm are  $4.48 \times 10^{-5}$  and  $9.41 \times 10^{-6}$  for both shearing and deforming flow tests, respectively, being two orders lower than the results presented in the scientific literature at the same level of mesh refinement.

Additionally, the 3D bubble rising problem is simulated for different fluid densities ( $\rho_1/\rho_2 = 10$  and  $\rho_1/\rho_2 = 1000$ ) and viscosity's ratios ( $\mu_1/\mu_2 = 10$  and  $\mu_1/\mu_2 = 100$ ), mimicking two different regimes (the ellipsoidal regime and the skirted and dimpled ellipsoidal-cap regime). The results obtained with the newly implemented algorithm for the 3D bubble rising problem can be summarized as follows:

- Good agreement is obtained for the 3D interface shape using both the newly implemented algorithm and OpenFOAM, DROPS, and NaSt3D software. Specifically, the 3D PLIC-VOF HiG-Flow algorithm predicted a stable ellipsoidal droplet shape for  $\rho_1/\rho_2 = 10$  and  $\mu_1/\mu_2 = 10$ , and a stable cap shape for  $\rho_1/\rho_2 = 1000$  and  $\mu_1/\mu_2 = 100$ .
- The bubble's rise velocity and center of mass evolution are computed with the 3D PLIC-VOF HiG-Flow algorithm and found to be consistent with OpenFOAM, DROPS, and NaSt3D software. In the initial phases of the simulation, the rise velocity of the droplet experiences a gradual increment from its initial zero value to reach a maximum magnitude. Following this, in the steady-state phase, the droplet's velocity decreases. This trend is particularly pronounced in the cap regime compared to the ellipsoidal regime.

Overall, the developed method demonstrates promising accuracy and reliability in simulating complex two-phase flows, making it a valuable tool for future fluid dynamics research. Particularly, its applicability extends to the study of complex fluid rheology, including viscoelastic fluids.

**Author Contributions:** Conceptualization, A.T.G.d.S., C.F., and A.C.; methodology, A.T.G.d.S., C.F., and A.C.; software, A.T.G.d.S., J.O., L.S., and A.C.; validation, A.T.G.d.S. and J.O.; formal analysis, A.T.G.d.S., C.F., J.O., L.S., and A.C.; investigation, A.T.G.d.S., C.F., J.O., L.S., and A.C.; writing—original draft preparation, A.T.G.d.S., C.F., J.O., L.S., and A.C.; writing—review and editing, A.T.G.d.S., C.F., J.O., L.S., and A.C.; project administration, C.F., L.S., and A.C.; and funding acquisition, C.F., L.S., and A.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior–Brasil (CAPES)–Finance Code 001; São Paulo Research Foundation (FAPESP) grants 2013/07375-0, 2019/07316-0, and 2023/00062-8; Fundação para a Ciência e a Tecnologia (FCT)

and Centre of Mathematics (CMAT) of the University of Minho projects UIDB/00013/2020 and UIDP/00013/2020; and FCT funding contract 2022.00753.CEECIND.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** A. Castelo gratefully acknowledges financial support from the São Paulo Research Foundation (FAPESP) grants 2013/07375-0 and 2019/07316-0. All of the authors acknowledge the support of the Institute of Mathematics and Computational Sciences (ICMC) at the University of Sao Paulo (USP). C. Fernandes gratefully acknowledges the financial support of the São Paulo Research Foundation (FAPESP) through grant 2023/00062-8 and Fundação para a Ciência e a Tecnologia (FCT) through the Centre of Mathematics (CMAT) at the University of Minho projects UIDB/00013/2020 and UIDP/00013/2020; he would also like to thank FCT for providing funding under the contract FCT/2022.00753.CEECIND. A. T. G. da Silva and J. Organista gratefully acknowledge financial support from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior–Brasil (CAPES)–Finance Code 001. Research was carried out using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by FAPESP (grant 2013/07375-0).

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

## Abbreviations

The following abbreviations are used in this manuscript:

BDF	Backward differentiation formula
BiCGS	Biconjugate gradient squared
Bo	Bond number
CFL	Courant–Friedrichs–Lewy
CSF	Continuum surface force
HiG	Hierarchical grid
HYPRE	High performance preconditioners
MLS	Moving least squares
PETSc	Portable, extensible toolkit for scientific computation
PLIC	Piecewise-linear interface calculation
Re	Reynolds number
SLIC	Simple line interface calculation
TVD	Total variation diminishing
VOF	Volume of fluid

## References

1. Metivet, T.; Chabannes, V.; Ismail, M.; Prud'homme, C. High-Order Finite-Element Framework for the Efficient Simulation of Multifluid Flows. *Mathematics* **2018**, *6*, 203. [\[CrossRef\]](#)
2. Pineda, H.; Biazussi, J.; López, F.; Oliveira, B.; Carvalho, R.D.M.; Bannwart, A.C.; Ratkovich, N. Phase distribution analysis in an Electrical Submersible Pump (ESP) inlet handling water–air two-phase flow using Computational Fluid Dynamics (CFD). *J. Pet. Sci. Eng.* **2016**, *139*, 49–61. [\[CrossRef\]](#)
3. Haroun, Y.; Legendre, D.; Raynal, L. Volume of fluid method for interfacial reactive mass transfer: Application to stable liquid film. *Chem. Eng. Sci.* **2010**, *65*, 2896–2909. [\[CrossRef\]](#)
4. Štěpánek, F.; Rajniak, P.; Mancinelli, C.; Chern, R.T.; Ramachandran, R. Distribution and accessibility of binder in wet granules. *Powder Technol.* **2009**, *189*, 376–384. [\[CrossRef\]](#)
5. Carciofi, B.A.M.; Prat, M.; Laurindo, J.B. Dynamics of vacuum impregnation of apples: Experimental data and simulation results using a VOF model. *J. Food Eng.* **2012**, *113*, 337–343. [\[CrossRef\]](#)
6. Wörner, M. Numerical modeling of multiphase flows in microfluidics and micro process engineering: A review of methods and applications. *Microfluid. Nanofluid.* **2012**, *12*, 841–886. [\[CrossRef\]](#)
7. Popinet, S. Numerical Models of Surface Tension. *Annu. Rev. Fluid Mech.* **2018**, *50*, 49–75. [\[CrossRef\]](#)
8. Unverdi, S.O.; Tryggvason, G. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.* **1992**, *100*, 25–37. [\[CrossRef\]](#)
9. Tryggvason, G.; Bunner, B.; Esmaeeli, A.; Juric, D.; Al-Rawahi, N.; Tauber, W.; Han, J.; Nas, S.; Jan, Y.-J. A front-tracking method for the computations of multiphase flow. *J. Comput. Phys.* **2001**, *169*, 708–759. [\[CrossRef\]](#)
10. Hirt, C.W.; Amsden, A.A.; Cook, J.L. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J. Comput. Phys.* **1974**, *14*, 227–253. [\[CrossRef\]](#)

11. Scardovelli, R.; Zaleski, S. Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech.* **1999**, *31*, 567–603. [[CrossRef](#)]
12. Anderson, D.M.; McFadden, G.B.; Wheeler, A.A. Diffuse-interface methods in fluid mechanics. *Annu. Rev. Fluid Mech.* **1998**, *30*, 139–165. [[CrossRef](#)]
13. Sethian, J.A.; Smereka, P. Level set methods for fluid interfaces. *Annu. Rev. Fluid Mech.* **2003**, *35*, 341–372. [[CrossRef](#)]
14. Liovic, P.; Rudman, M.; Liow, J.-L.; Lakehal, D.; Kothe, D. A 3D unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction. *Comput. Fluids* **2006**, *35*, 1011–1032. [[CrossRef](#)]
15. Adelsberger, J.; Essery, P.; Griebel, M.; Groß, S.; Klitz, M.; Rüttgers, A. 3D incompressible two-phase flow benchmark computations for rising droplets. In Proceedings of the 11th World Congress on Computational Mechanics (WCCM XI) Proceedings, Barcelona, Spain, 20–25 July 2014.
16. DROPS Package for Simulation of Two-Phase Flows. Available online: <http://www.igpm.rwth-aachen.de/DROPS/> (accessed on 21 June 2023).
17. Croce, R.; Griebel, M.; Schweitzer, M.A. Numerical simulation of bubble and droplet deformation by a level set approach with surface tension in three dimensions. *Int. J. Numer. Methods Fluids* **2010**, *62*, 963–993. [[CrossRef](#)]
18. OpenFOAM, The Open Source CFD Toolbox, User Guide Version 2.2.2. Available online: <http://www.openfoam.org> (accessed on 21 June 2023).
19. Hysing, S.; Turek, S.; Kuzmin, D.; Parolini, N.; Burman, E.; Ganesan, S.; Tobiska, L. Quantitative benchmark computations of two-dimensional bubble dynamics. *Int. J. Numer. Methods Fluids* **2009**, *60*, 1259–1288. [[CrossRef](#)]
20. Figueiredo, R.A. Simulação Numérica de Escoamentos Viscoelásticos Multifásicos Complexos. Ph.D. Thesis, University of São Paulo, São Paulo, Brazil, 9 September 2016. [[CrossRef](#)]
21. Devine, K.D.; Boman, E.G.; Riesen, L.A.; Catalyurek, U.V.; Chevalier, C. Getting started with Zoltan: A short tutorial. In *Dagstuhl Seminar Proceedings*; Schloss Dagstuhl-Leibniz-Zentrum für Informatik: Wadern, Germany, 2009.
22. Heroux, M.A.; Bartlett, R.A.; Howle, V.E.; Hoekstra, R.J.; Hu, J.J.; Kolda, T.G.; Lehoucq, R.B.; Long, K.R.; Pawlowski, R.P.; Phipps, E.T.; et al. An overview of the Trilinos project. *ACM Trans. Math. Softw. (TOMS)* **2005**, *31*, 397–423. [[CrossRef](#)]
23. Sousa, F.S.; Lages, C.F.; Ansoni, J.L.; Castelo, A.; Simao, A. A finite difference method with meshless interpolation for incompressible flows in non-graded tree-based grids. *J. Comput. Phys.* **2019**, *396*, 848–866. [[CrossRef](#)]
24. Falgout, R.D.; Yang, U.M. hypre: A library of high performance preconditioners. In *International Conference on Computational Science*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 632–641. [[CrossRef](#)]
25. Balay, S.; Abhyankar, S.; Adams, M.; Brown, J.; Brune, P.; Buschelman, K.; Dalcin, L.D.; Eijkhout, V.; Gropp, W.; Kaushik, D.; et al. *PETSc Users Manual Revision 3.8. United States*; Technical Report; Argonne National Lab.: Argonne, IL, USA, 2017. [[CrossRef](#)]
26. Chorin, A.J. Numerical solution of the Navier–Stokes equations. *Math. Comput.* **1968**, *22*, 745–762. [[CrossRef](#)]
27. Lages, C.F.A. Métodos Numéricos Para Escoamentos Multifásicos em Malhas Hierárquicas. Ph.D. Thesis, University of São Paulo, Brazil, 3 March 2016. [[CrossRef](#)]
28. Sonneveld, P. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* **1989**, *10*, 36–52. [[CrossRef](#)]
29. Anzt, H.; Dongarra, J.; Flegar, G.; Higham, N.J.; Quintana-Ortí, E.S. Adaptive precision in block-Jacobi preconditioning for iterative sparse linear system solvers. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e4460. [[CrossRef](#)]
30. Noh, W.F.; Woodward, P. SLIC (Simple Line Interface Calculation). In *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics 28 June–2 July 1976 Twente University, Enschede, The Netherlands*; Lecture Notes in Physics; van de Vooren, A.I., Zandbergen, P.J., Eds.; Springer: Berlin/Heidelberg, Germany, 1976; Volume 59. [[CrossRef](#)]
31. Hirt, C.W.; Nichols, B.D. Volume of Fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* **1981**, *39*, 201–225. [[CrossRef](#)]
32. Youngs, D.L. Time-dependent multi-material flow with large fluid distortion. In *Numerical Methods in Fluid Dynamics*; Morton, K.W., Baines, M.J., Eds.; Academic Press: Cambridge, MA, USA, 1982; pp. 273–285.
33. Rudman, M. Volume-tracking methods for interfacial flow calculations. *Int. J. Numer. Methods Fluids* **1997**, *24*, 671–691. [[CrossRef](#)]
34. Mehta, S.; Patel, N.; Zinjala, H.; Banerjee, J. Development of 3-D geometric PLIC-VOF solver for two-fluid flow simulation. In Proceedings of the Fortieth National Conference on Fluid Mechanics and Fluid Power, NIT Hamirpur, Himachal Pradesh, India, 12–14 December 2013.
35. Puckett, E.G.; Almgren, A.S.; Bell, J.B.; Marcus, D.L.; Rider, W.J. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *J. Comput. Phys.* **1997**, *130*, 269–282. [[CrossRef](#)]
36. Rider, W.J.; Kothe, D.B. Reconstructing volume tracking. *J. Comput. Phys.* **1998**, *141*, 112–152. [[CrossRef](#)]
37. Brackbill, J.U.; Kothe, D.B.; Zemach, C. A continuum method for modeling surface tension. *J. Comput. Phys.* **1992**, *100*, 335–354. [[CrossRef](#)]
38. Torrey, M.D.; Cloutman, L.D.; Mjolsness, R.C.; Hirt, C.W. NASA-VOF2D: A computer program for incompressible flows with free surfaces. *NASA STI/Recon Tech. Rep.* **1985**, *86*, 30116.

39. Duz, B.; Borsboom, M.J.A.; Veldman, A.E.P.; Wellens, P.; Huijsmans, R. Efficient and accurate PLIC-VOF techniques for numerical simulations of free surface water waves. In Proceedings of the 9th International Conference on Computational Fluid Dynamics-ICCFD9, Istanbul, Turkey, 11–15 July 2016.
40. López, J.; Zanzi, C.; Gómez, P.; Faura, F.; Hernández, J. A new volume of fluid method in three dimensions—Part II: Piecewise-planar interface reconstruction with cubic-Bézier fit. *Int. J. Numer. Methods Fluids* **2008**, *58*, 923–944. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.