fluids

MDPI

Article

The Coupled Volume of Fluid and Brinkman Penalization Methods for Simulation of Incompressible Multiphase Flows

Evgenii L. Sharaborin, Oleg A. Rogozin and Aslan R. Kasimov *

Skolkovo Institute of Science and Technology, 121205 Moscow, Russia; Evgenii.Sharaborin@skoltech.ru (E.L.S.); O.Rogozin@skoltech.ru (O.A.R.)
* Correspondence: a.kasimov@skoltech.ru

Abstract: In this work, we contribute to the development of numerical algorithms for the direct simulation of three-dimensional incompressible multiphase flows in the presence of multiple fluids and solids. The volume of fluid method is used for interface tracking, and the Brinkman penalization method is used to treat solids; the latter is assumed to be perfectly superhydrophobic or perfectly superhydrophilic, to have an arbitrary shape, and to move with a prescribed velocity. The proposed algorithm is implemented in the open-source software Basilisk and is validated on a number of test cases, such as the Stokes flow between a periodic array of cylinders, vortex decay problem, and multiphase flow around moving solids.

Keywords: direct numerical simulation; incompressible multiphase flows; Brinkman penalization; volume of fluid method; continuum surface force; adaptive mesh refinement; finite volume method

Citation: Sharaborin, E.L.; Rogozin, O.A.; Kasimov, A.R. The Coupled Volume of Fluid and Brinkman Penalization Methods for Simulation of Incompressible Multiphase Flows. *Fluids* **2021**, *6*, 334. https://doi.org/10.3390/fluids6090334

Academic Editors: Federico Piscaglia and Jérôme Hélie

Received: 2 August 2021
Accepted: 14 September 2021
Published: 18 September 2021

1. Introduction

In this paper, we develop a numerical approach to the simulation of three-dimensional incompressible multiphase flows involving multiple fluid phases as well as solid inclusions or boundaries of arbitrary shapes. The proposed algorithm is implemented in the open-source environment Basilisk [1] and is validated on several fluid dynamics problems demonstrating its capabilities for solving challenging multiphase multiscale flow problems.

Multiphase flows are of importance to many areas of applied science and engineering. Sprays and jets from nozzles are a common occurrence in mechanical, chemical, and aerospace engineering [2–5]. Chemically reacting flows in multiphase systems are important for combustion applications [6], microreactors [7], and energy storage technologies [8,9]. The motion of bubbles, droplets, and particles in viscous flows are important in the oil and gas industry [10–12], biology, and medicine [13–17]. The design of microfluidic devices requires the understanding of multiphase flows in small channels of complex shapes [14,18–21]. Numerous manufacturing applications also relate to multiphase flows involving, for example, the formation of bubbles during composite manufacturing [22]. Detailed numerical simulation in problems involving multiphase flows is associated with extra challenges compared to single-phase flows due to the need to accurately resolve various features associated with interfaces, which often requires sophisticated algorithms and, in the presence of multiscale dynamics, substantial computational resources [21,23]. The reader can find general and extensive discussions of the theory, computational techniques, and applications of multiphase flows in recent and classical books, such as [23–28].

In multiphase flows, the fluid–fluid interface is generally a dynamic unknown surface that is to be determined as part of the solution. One can describe such flows by writing governing systems of equations for each fluid region separated by the interface and supplementing them by kinematic and dynamic interface conditions. Such equations coupled via the interface conditions can then be discretized and solved numerically. This direct approach is generally quite challenging to implement as it requires one to track the

evolution of the interface and to carefully handle topological changes associated with the break-up or merger of domains with different fluids.

The present work uses a different formulation that provides a simple and efficient way to handle problems with the interface identification. The formulation is based on the introduction of a scalar field that serves as an indicator function for each fluid phase and allows one to track the interface motion by solving an appropriate evolution equation for the function. For example, one can define function $\varphi(\mathbf{x}, t)$ that has a value pf 1 in the first fluid and a value of 0 in the second fluid, or $\varphi(\mathbf{x}, t)$ could have different signs on different sides of the interface and be zero on the interface. Many methods based on such ideas have been designed to solve for the interface evolution, including the front tracking method [29–31], the boundary integral method [32], the phase field method [33,34], the volume of fluid method (VOF) [35,36], and the level set method [37,38].

Our formulation is based on the VOF method, which was originally proposed by Hirt and Nichols in 1981 [35] and has subsequently been improved in many works, for example [39–41]. There exist two versions of the method: the geometric VOF [41,42] wherein the distribution of φ is found geometrically by, e.g., advecting a plane interface with a predefined velocity $\mathbf{u}$, and the algebraic VOF [43] wherein φ is represented by a function, such as a polynomial or trigonometric function. The geometric VOF is more widely used due to its advantages in terms of mass conservation, robustness to topological changes, such as a merger or break-up, and relative ease of extension to high dimensional Cartesian grids [44]. However, it should be noted that the geometric VOF method is harder to implement in parallel.

The surface tension of a fluid interface is treated by the so-called "continuum surface force" (CSF) model, originally proposed by Brackbill et al. [45]. In this model, the surface tension is represented as a body force highly concentrated in the vicinity of the interface. Other related models can be found in the literature [46,47]. The CSF model is easily coupled with various fixed (Eulerian) mesh formulations for interfacial flows, in particular with the VOF [45,48], level set [49,50], and front-tracking methods [3,30,51]. Recent improvements of CSF are associated with the reduction of spurious currents using the balanced force algorithm [46,52] and with improving the interface curvature evaluation [53,54]. We also mention a related continuum surface stress method [55–57] that represents the surface tension as the divergence of the capillary pressure tensor. The main advantage of this method over CSF is that it naturally takes into account tangential stresses, such as Marangoni stresses (which are not considered in the present work).

The presence of solids in multiphase flows is associated with many computational challenges especially if the solids are of a complex shape and are not stationary. When considering only mesh-based schemes for such multiphase flows, there are two approaches to represent the geometry of the solids: (1) based on body-fitted methods, wherein the mesh boundary is aligned with the surface of the solids or (2) the immersed boundary (IB) methods, wherein the boundary conditions are incorporated in the governing equations, and the solid region becomes part of the computational domain. The first of these options allows for an easy formulation of the boundary conditions. However, when bodies experience large deformations or move fast, a grid has to be reconstructed frequently, which leads to large overhead. Moreover, during the frequent reconstructions, one must avoid the appearance of geometrical singularities and small angles in order to obtain high-quality meshes. Although the automatic generation of body-fitted grids has been developed [58,59], the task remains challenging and labor-intensive [60], especially when bodies move. Such methods are also difficult to generalize to high-order schemes, so most body-fitted methods are low-order.

In contrast, the IB methods [61] allow for high-order schemes, arbitrary geometries, and moving solids. The first IB method was proposed by Peskin in 1977 [62] to simulate the interaction of cardiac mechanics and blood flow. The main feature of the method is a non-body conformal Cartesian grid in which a new approach is formulated to consider the effects of the immersed boundary on the flow. Specifically, the fluid is modeled on

a fixed Cartesian grid, but the boundary is modeled on a curvilinear Lagrangian mesh, which moves freely across the fixed mesh. In this technique, a grid is not body-fitted. The immersed boundary is provided by an additional surface grid that cuts across the Cartesian grid. The non-conformity of the wall and the grid leads to the modification of the Navier–Stokes equations [63] by incorporating an artificial force term that models the effects of the boundary in such a way that guarantees boundary conditions by slightly adjusting the fluid velocity field near the solid surface using various kernel functions [61].

One of the IB methods is the Brinkman penalization method (BPM), which was originally developed by Arquis and Caltagirone [64] for the numerical simulation of isothermal obstacles in incompressible single-phase flows. The general idea of the BPM is based on the work of Brinkman [65] in which solids are considered porous media with very small permeability, $\eta \to 0$, and fluids as media with large permeability, $\eta \to \infty$. In BPM, the permeability is taken as a penalization parameter. One of the advantages of BPM is the availability of a convergence proof and error estimates depending on the penalization parameter, which have been thoroughly studied by Angot et al. [66]. Furthermore, the enforcement of boundary conditions on the solid requires no modifications to a numerical scheme near the solid surfaces. It is therefore relatively easy to design an efficient and robust parallel code [67].

In the present work, interfaces are tracked by the VOF method [39], and solids are considered by BPM. The algorithm is implemented in the open-source software Basilisk [1], which uses efficient adaptive meshes. The BPM and VOF methods can be implemented for fixed or adaptive grids. The adaptive mesh refinement reveals the best capabilities of the BPM and VOF methods when the mesh compression ratio (the actual number of cells relative to the number of cells at maximum refinement) is larger than 30% [68].

In [69,70], the authors also consider coupling of VOF and BPM. In [69], non-zero contact angles on the surface of stationary solids are analyzed using fixed meshes, while in [70], the solids are treated with adaptive Cartesian meshes under perfectly wetting conditions. Our work considers fixed as well as moving solids using adaptive meshes; perfectly wetting and perfectly non-wetting surfaces are assumed.

The remainder of the manuscript is organized as follows. In Section 2, we introduce the governing equations together with the main ideas behind the numerical approach. The detailed description of the numerical algorithm is presented in Section 3. In Section 4, we discuss a number of simulation results, including validation tests of the method by solving several multiphase flow problems. Conclusions are presented in Section 5.

2. Governing Equations

The governing equations are based on the Navier–Stokes equations for incompressible Newtonian fluids adapted to treat multiphase flows in a system with complex topology involving multiple fluids and solid inclusions. The solids are assumed to move with a prescribed velocity. The solid surface is considered to be either perfectly superhydrophilic or perfectly superhydrophobic.

The computational domain Ω is assumed to contain two fluids in subdomains $\Omega_{f_i} = \bigcup_k \Omega_{f_i,k}$, $i = 1, 2$, and stationary or moving solids in subdomains $\Omega_s = \bigcup_k \Omega_{s,k}$. Each subdomain can be a multiply connected region with an arbitrary shape, as illustrated in Figure 1. The fluids are assumed to be immiscible and to have different densities ρ_i and viscosities μ_i, $i = 1, 2$. The no-slip boundary condition is set on the surface of solids such that

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{U}_s(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega_s, \tag{1}$$

where the solid velocity field $\mathbf{U}_s(\mathbf{x}, t)$ is assumed prescribed.

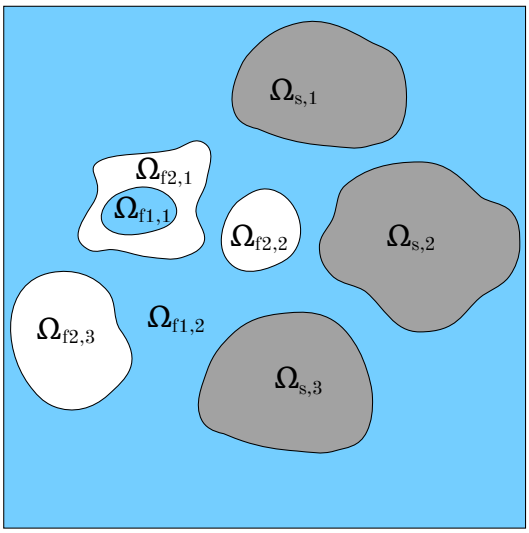

Figure 1. Schematics of the computational domain containing fluids 1 (blue) and 2 (white) in $\Omega_{f_1,k}$ and $\Omega_{f_2,k}$, respectively, and solids in $\Omega_{s,k}$ (gray).

The Navier–Stokes equations for multiphase flow can be reformulated as a single system for the velocity field $\mathbf{u}$ in the whole domain. The effect of surface tension is then included in the momentum equation as a volumetric force $\mathbf{f}_\sigma$ concentrated on the fluid interfaces. The presence of solids is captured by the penalization force $\mathbf{f}_\mathrm{B}$ [66,71] that acts inside solids and also enables imposing the boundary condition (1). Then, the fluid motion is governed by the continuity and modified Navier–Stokes equations:

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0, \tag{2}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \boldsymbol{\tau} + \mathbf{f}_\sigma + \rho \mathbf{g} + \mathbf{f}_\mathrm{B}, \tag{3}$$

where p is pressure, $\boldsymbol{\tau} = \mu\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right)$ is the viscous stress tensor, and $\mathbf{g}$ is the gravitational acceleration. The local average density ρ and dynamic viscosity μ of the fluids are calculated via the local volume fraction $\varphi(\mathbf{x}, t)$ of fluid 1 using a linear interpolation:

$$\begin{aligned} \rho &= \rho_1 \varphi + \rho_2 (1 - \varphi), \\ \mu &= \mu_1 \varphi + \mu_2 (1 - \varphi), \end{aligned} \tag{4}$$

where subscripts 1 and 2 refer to the primary and secondary fluid phases. The parameters ρ_i, μ_i are assumed constant. The volume fraction φ is governed by the advection equation

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \varphi = 0. \tag{5}$$

2.1. Surface Tension Force $\mathbf{f}_\sigma$

The surface tension force is incorporated into the momentum Equation (3) as a source term $\mathbf{f}_\sigma$ that is highly concentrated in the vicinity of the interface. This continuum surface force (CSF) approach was proposed in [45] and is widely used for the simulation of flows with free interfaces. The main advantage of the CSF is the one-shot calculation throughout the computational domain and the relatively straightforward implementation. The surface tension force $\mathbf{f}_\sigma$ in Equation (3) is given by:

$$\mathbf{f}_\sigma(\mathbf{x}, t) = \frac{\rho}{\langle \rho \rangle} \sigma \kappa \boldsymbol{\nabla} \varphi, \tag{6}$$

where $\langle\rho\rangle = (\rho_1 + \rho_2)/2$ is the average density of the two fluid phases, σ is the coefficient of surface tension, and κ is the total curvature of the interface obtained using $\kappa = -\boldsymbol{\nabla}\cdot\mathbf{n}$, where $\mathbf{n} = \boldsymbol{\nabla}\varphi/|\boldsymbol{\nabla}\varphi|$ is the unit normal to the interface.

To model perfectly non-wetting (perfectly superhydrophobic) or perfectly wetting (perfectly superhydrophilic) surfaces, the volume fraction of fluid 1 inside solids is given by:

$$\varphi(\mathbf{x} \in \Omega_s, t) = \begin{cases} 0, \text{ if non-wetting surface,} \\ 1, \text{ if perfectly-wetting surface.} \end{cases} \tag{7}$$

2.2. The Penalization Term $\mathbf{f}_\mathrm{B}$

The presence of the solid phase is reflected in the modified Navier–Stokes equation by the Brinkman penalization term $\mathbf{f}_\mathrm{B}$, and the approach is called the Brinkman penalization method (BPM) [66,72]. The method considers solids as porous media with a vanishing permeability. The source term $\mathbf{f}_\mathrm{B}$ in the governing equations enforces the desired velocity $\mathbf{u} = \mathbf{U}_s$ inside the solids.

For the Dirichlet boundary condition from Equation (1), the term $\mathbf{f}_\mathrm{B}$ in Equation (3) is set as

$$\mathbf{f}_\mathrm{B} = -\rho\frac{\chi}{\eta}(\mathbf{u} - \mathbf{U}_s), \tag{8}$$

where $\eta > 0$ is a penalization coefficient having the dimension of time, and factor χ is a mask field defined as

$$\chi(\mathbf{x}, t) = \begin{cases} 1, \mathbf{x} \in \Omega_s, \\ 0, \mathbf{x} \in \Omega_f. \end{cases} \tag{9}$$

Angot et al. [66] theoretically proved that as $\eta \to 0$, solutions of the penalized Equations (2) and (3) converge to the solutions of the original Navier–Stokes equations with correct boundary conditions. The upper bound on the global errors of the normal and tangential components of velocity were shown to satisfy the estimates

$$\begin{aligned} \left\|u_n - u_{\eta,n}\right\| &\le C_1\eta, \\ \left\|u_\tau - u_{\eta,\tau}\right\| &\le C_2\sqrt{\eta}, \end{aligned} \tag{10}$$

where C_1 and C_2 are constants that depend only on the problem setting. These estimates indicate that the normal component of the velocity converges to the target value faster than the tangential one. Note, however, that even though taking smaller values of η gives better results, there is a practical limitation of the method associated with the presence of discretization errors. As we show in Section 4.2, the error $\left\|\mathbf{u} - \mathbf{u}_\eta\right\|$ reaches a plateau when $\eta \to 0$. The error is seen to decrease as η tends to 0 until the viscous and penalization terms become of the same order of the magnitude inside the solid, i.e., $\nu\Delta\mathbf{u} \approx \mathbf{u}/\eta$. After that, the plateau is reached (see Figure 8). In terms of the characteristic velocity u_0 and thickness δ, the size of the Brinkman layer can be estimated from the balance $\nu u_0/\delta^2 \approx u_0/\eta$ that gives $\delta = \sqrt{\eta\nu}$. This size must be resolved numerically with at least $m = 1, 2$ mesh cells [71,73]. To ensure this, one can take the penalization coefficient as

$$\eta = \frac{(mh)^2}{\nu}. \tag{11}$$

The BPM requires a specification of viscosity and density in the whole domain, even inside solids. The density ρ_3 of the solid can be taken as a real physical value, but the viscosity of the solid, μ_3, has no physical meaning. Nevertheless, it was observed in numerical experiments that taking large values of μ_3 improves the efficiency of computations. Here, μ_3 was chosen (somewhat arbitrarily), assuming that the kinematic viscosities of the primary heavy fluid (fluid 1) and the solid are the same, which makes the kinematic viscosities continuous across the fluid–solid interface: $\mu_3 = \mu_1\rho_3/\rho_1$. Note that taking a μ_3

value that is too small leads to a slow convergence of the solution. One should also avoid taking a μ_3 too large. Either way, the average viscosity and density are found from

$$\mu = (1-\chi)[\mu_1\varphi + \mu_2(1-\varphi)] + \chi\mu_3, \tag{12}$$

$$\rho = (1-\chi)[\rho_1\varphi + \rho_2(1-\varphi)] + \chi\rho_3. \tag{13}$$

These definitions ensure that: $\rho = \rho_3, \mu = \mu_3$ inside solids ($\chi = 1$ and any φ); $\rho = \rho_1, \mu = \mu_1$ inside fluid 1 ($\chi = 0$ and $\varphi = 1$); and $\rho = \rho_2, \mu = \mu_2$ inside fluid 2 ($\chi = 0$ and $\varphi = 0$).

3. Computational Algorithms

The computational domain Ω is spatially discretized using square (cubic in 3D) finite volumes forming a hierarchical structure; the so-called quadtree (octree in 3D) [74] is schematically shown in Figure 2a. Variables are defined either at the cell centers or on their faces, as shown in Figure 2b. The incompressible Navier–Stokes solver in Basilisk uses a standard staggered grid, where the pressure p is specified only at the cell center, but the velocity is specified both at the cell centers as $\mathbf{u}$ and the centers of the cell faces as $\mathbf{u}_f$. Similarly, the solid volume fractions are specified at the cell centers as χ and at the face centers as χ_f. Both χ and χ_f are determined using the distance functions at the vertices. The density ρ and volume fraction φ of fluid 1 are also defined at the cell centers, while viscosity μ, acceleration $\mathbf{a}_f$, and specific volume $\alpha_f = 1/\rho(\varphi_f, \chi_f)$ are defined at the cell faces, where ρ is computed using Equation (13), and the face-centered value φ_f is linearly interpolated using two adjacent cells.

We develop the numerical algorithm for the solution of the main governing system by splitting various physical processes and treating them separately either explicitly or implicitly. The algorithm consists of six main steps, which are explained in detail below: (1) the volume of fluid step, in which the volume fraction φ is updated; (2) the advection step for momentum in Equation (3), in which the velocity field $\mathbf{u}$ is updated using the Bell–Colella–Glaz scheme [75]; (3) the implicit update of the viscous and Brinkman penalization terms as well as of $\mathbf{u}$; (4) the updates of the surface tension and gravity terms together with the face velocity $\mathbf{u}_f$; (5) the Chorin projection step is taken [76], in which the incompressibility condition is imposed and the pressure field p is found; and (6) the mesh adaptation step, where the mesh is refined in the regions with large gradients or coarsened in smooth regions.

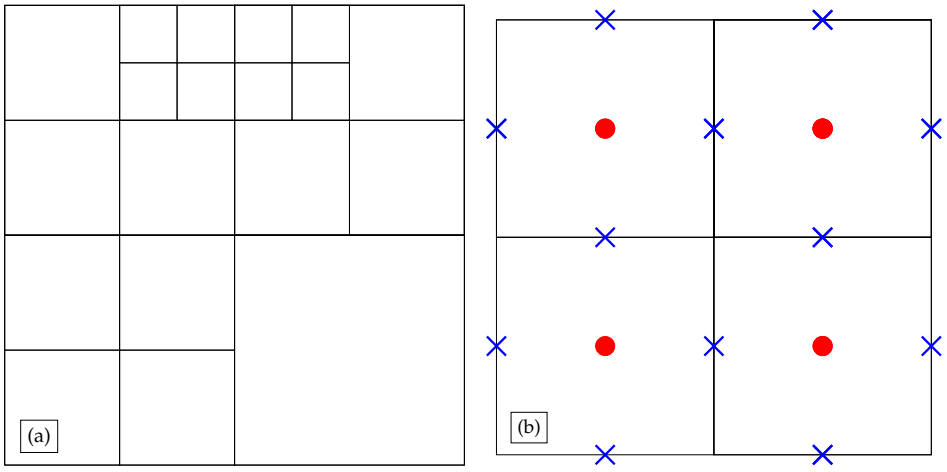

Figure 2. Structure of the adaptive computational mesh and variable definitions. (**a**) The quadtree discretization. (**b**) The staggered grid with some variables defined at the cell centers while others at the cell boundaries. Pressure p, velocity $\mathbf{u}$, volume fraction φ of fluid 1, volume fraction χ of the solid, and density ρ are defined at the cell centers (red circles). Other variables, such as velocity $\mathbf{u}_f$, viscosity μ, acceleration $\mathbf{a}$, and volume fraction of solids χ_f are specified on the faces of cells (crosses) (note that some variables can be defined both at the centers and the faces).

3.1. Step 1: Volume of Fluid (VOF)

The VOF is a sharp interface method that evolves fluid interfaces using the tracer field φ satisfying the advection (Equation (5)). When advecting φ from the time level $n-1/2$ to $n+1/2$ using a conservative, non-diffusive geometric VOF method [39,42], we obtain:

$$\varphi^{n+1/2} = \varphi^{n-1/2} + \Delta t \boldsymbol{\nabla} \cdot \left(\varphi^n \mathbf{u}^n_{f,c} \right), \tag{14}$$

where $\mathbf{u}^n_{f,c}$ is the corrected velocity field defined on the cell faces. The face-velocity correction is required in order to adjust the flow so that the fluid does not penetrate into the solid by more than two cells. We use the following correction of $\mathbf{u}_f$ to obtain $\mathbf{u}^n_{f,c}$:

$$\mathbf{u}^n_{f,c} = (1 - I^n_{\chi_f})\mathbf{u}^n_f + I^n_{\chi_f} \mathbf{U}^n_{f,s}, \tag{15}$$

where $\mathbf{U}^n_{f,s} = \tilde{\mathbf{U}}_s$ is the solid velocity obtained by averaging the cell-centered velocity $\mathbf{U}_s$, and $I^n_{\chi_f}$ is the face-centered indicator function computed from the face-centered solid volume fraction χ^n_f. The indicator function I_{χ_f} is 1 inside the pure fluid domain ($\chi_f = 0$), and 0 even if the cell contains a small volume of the solid body ($\chi_f > 0$). Such a formulation does not strictly respect the mass conservation. Nevertheless, in the verification tests in Sections 4.3 and 4.4, we demonstrate that the mass loss is negligible ($<5 \times 10^{-4}$% for fine mesh and $<10^{-2}$% for coarse mesh). In the first time step, $\mathbf{u}^0_f$ is taken as the average value of the cell-centered velocity $\mathbf{u}^0$. In the multi-dimensional VOF advection scheme, the field φ is advected along each dimension in series using a 1-D scheme.

The VOF method allows one to simulate multiphase flows with complex topological changes using sharp interface representation. However, its implementation is not without challenges. It requires an accurate calculation of the interface normal and the curvature, which involves calculating second derivatives in space (see subsequent steps of the algorithm below). One may face numerical artifacts, such as discretization-induced flow, and hence the absence of convergence with grid refinement [53,56]. Consequently, most approaches do not estimate the curvature directly from the VOF function but instead use a smoothed VOF function via convolution, discrete quadrature [77], and others.

One approach to addressing these problems is based on the construction of a height function H for the calculation of interface normals and curvatures [53]. For each interface cell, fluid "heights" are calculated by summing the fluid volume in the direction of the largest normal component n_x, n_y, or n_z. This summation is performed on a $7 \times 3 \times 3$ stencil around all interface cells with the size h. Note that in 2D, H is a vector, and in 3D, H is a tensor. For example, if for some interface cell the largest normal is directed along z, then the height H for a cell I, J, K can be found from

$$H_{I+i,J+j,K} = \sum_{k=-3}^{3} \varphi_{i,j,k} h, \quad i,j = -1,0,1. \tag{16}$$

The curvature κ and the gradient $\boldsymbol{\nabla}\varphi$ in Equation (6) are then calculated as:

$$\begin{aligned} \kappa &= \nabla_{x,y} \cdot \left(\xi \nabla_{x,y} H \right), \\ \boldsymbol{\nabla}\varphi &= \left(\frac{\varphi_{i,j,k} - \varphi_{i-1,j,k}}{h}, \frac{\varphi_{i,j,k} - \varphi_{i,j-1,k}}{h}, \frac{\varphi_{i,j,k} - \varphi_{i,j,k-1}}{h} \right), \end{aligned} \tag{17}$$

where $\xi = 1/\sqrt{1 + H_x^2 + H_y^2}$, and the nabla operators with x, y indices indicate the divergence or gradient only along the indicated directions.

When the HF stencil crosses a highly curved interface more than once, the standard height-function method stops working. Then the HF method is locally switched to the parabola-fitted curvature method [52]. The idea behind this method is to fit the interface

points with a parabola in 2D and a paraboloid in 3D and differentiating the resulting analytical function to estimate the curvature.

3.2. Step 2: Advection

At this stage, the Bell–Colella–Glaz (BCG) scheme [75] is applied, which is second-order in time and space. The viscous and Brinkman penalization terms from Equation (8) are omitted. For simplicity, the method is described here for the two-dimensional case.

Let $\mathbf{G} = -\alpha_f \boldsymbol{\nabla} p + \sigma\kappa\boldsymbol{\nabla}\varphi/\langle\rho\rangle + \mathbf{g}$ and $\mathbf{u} = (u, v)$. The BCG method estimates the face velocity at the half time step $\mathbf{u}_f^{n+1/2}$ and then advects the cell-centered velocity field $\mathbf{u}$ as a passive tracer. The face velocity $\mathbf{u}_{f,p}^{n+1/2}$ in the intermediate timestep $n+1/2$ is extrapolated along characteristics using the cell velocity $\mathbf{u}^n$ and the term $\mathbf{G}^n$. The velocity $\mathbf{u}_f$ and auxiliary fields $\mathbf{G}_f$ at the faces are estimated by averaging the left and right neighbors for their x components and, correspondingly, bottom and top neighbors for their y components. Such averages are denoted by an overbar: $\bar{\mathbf{f}}_f = (\mathbf{f}^+ + \mathbf{f}^-)/2$.

Thus, the face velocity $\mathbf{u}_{f,p}^{n+1/2}$ at the intermediate temporal layer $n+1/2$ for the x component is given by

$$u_{f,p}^{n+1/2} = u_S^{n+1/2} + \frac{\Delta t}{2}\bar{G}_{f,x}^n + \frac{Sh}{2}(1 - |\bar{r}_x|)\left(\frac{\partial u}{\partial x}\right)_S^n - \frac{\Delta t v_S}{2}\left(\frac{\partial u}{\partial y}\right)_S^n, \tag{18}$$

where h is the cell size, $\bar{r}_x = \bar{u}_f^n \Delta t/h$ is the CFL number, $S = \text{sign}\,(\bar{u}_f^n)$ is responsible for the direction of the flux, and the index S denotes which neighboring variable is used. Since this extrapolation does not guarantee incompressibility, the Chorin projection step is taken (see Section 3.5), which subtracts the non-solenoidal component from the velocity field $\mathbf{u}_{f,p}^{n+1/2}$ and gives the velocity at the half step $\mathbf{u}_f^{n+1/2}$. The x component of the velocity $\mathbf{u}_\mathrm{a}^{n+1}$ is updated as:

$$u_\mathrm{a}^{n+1} = u^n - \sum_{d=x,y} \Delta t \frac{F_d^+ - F_d^-}{h}, \tag{19}$$

where the index a indicates advection, F_d^+, F_d^- are fluxes along direction d, with $()^+$ indicating right or top face values and $()^-$ left or bottom face values. For example, F_x can be obtained by a procedure similar to Equation (18):

$$F_x = u_f^{n+1/2}\left(u_S^n + \frac{\Delta t}{2}\bar{G}_{f,x}^n + \frac{Sh}{2}(1 - |r_x|)\left(\frac{\partial u}{\partial x}\right)_S^n - \frac{\Delta t \bar{v}}{2}\left(\frac{\partial v}{\partial y}\right)_S^n\right), \tag{20}$$

where $r_x = u_f \Delta t/h$ is the CFL number, $S = \text{sign}\left(u_f^{n+1/2}\right)$ is a direction, and $\bar{v}$ is averaged from face values as $\bar{v} = \left(v_{f_{i,j}} + v_{f_{i,j+1}}\right)/2$.

The sum of the pressure gradient, surface tension, and body acceleration, i.e., $\mathbf{G}^n$, is used in the calculation of cell-centered variable $u_{a,*}$ as follows:

$$\mathbf{u}_{\mathrm{a},*}^{n+1} = \mathbf{u}_\mathrm{a}^{n+1} + \mathbf{G}^n \Delta t. \tag{21}$$

Note, that in Equation (20), $\mathbf{G}^n$ is used for more accurate computation of fluxes at cell faces.

The explicit integration gives a restriction on the time step: $\max_\Omega(|\mathbf{u}_i|\Delta t/h_i) < 1$, $i = (x, y)$.

3.3. Step 3: Viscous Force and Brinkman Penalization

At this step, we solve the momentum equation with only the viscous and Brinkman penalization terms. Since the stiff viscous term imposes a strong time-step limitation

($\Delta t \propto h^2$), and the Brinkman penalization term is also stiff, these two terms are integrated with an implicit scheme as follows:

$$\frac{\mathbf{u}_{\mathrm{v}}^{n+1} - \mathbf{u}_{\mathrm{a},*}^{n+1}}{\Delta t} = \frac{1}{\rho^{n+1/2}} \boldsymbol{\nabla} \cdot \mu^{n+1/2} \left(\boldsymbol{\nabla} \mathbf{u}_{\mathrm{v}}^{n+1} + \boldsymbol{\nabla} \left(\mathbf{u}_{\mathrm{v}}^{n+1} \right)^T \right) - \frac{\chi^{n+1}}{\eta} \left(\mathbf{u}_{\mathrm{v}}^{n+1} - \mathbf{U}_s^{n+1} \right), \quad (22)$$

where $\mathbf{u}_{\mathrm{v}}^{n+1}$ is the cell-centered velocity, the penalization coefficient η is computed by Equation (11), and $\chi^{n+1/2}$ is the cell-centered volume fraction of solids. The density $\rho^{n+1/2}$ and viscosities $\mu^{n+1/2}$ are computed using Equations (12) and (13), in which instead of the volume fraction φ, the smoothed volume fraction $\tilde{\varphi}$ is used. The smoothing is done by linearly interpolating values from the nearest cell-centered neighbors. The viscous term has the standard second order central-difference discretization in space. A similar equation without contribution of $\mathbf{f}_{\mathrm{B}}$ is solved by the multigrid method with Jacobi iterations in [78,79].

3.4. Step 4: Surface Tension and Gravity

The cell-centered velocity $\mathbf{u}_{\mathrm{v}}^{n+1}$ is based on an old value of $\mathbf{G}^n$, which is computed from the sum of body acceleration $\mathbf{g}^n$, surface tension acceleration $\sigma \kappa^{n-1/2} \boldsymbol{\nabla} \varphi^{n-1/2} / \langle \rho \rangle$, and the pressure gradient $\boldsymbol{\nabla} p^n$ on the old time layer n. The following steps update the $\mathbf{G}$ field in order to then subtract the old $\mathbf{G}^n \Delta t$: $\mathbf{u}_{\mathrm{v},*}^{n+1} = \mathbf{u}_{\mathrm{v}}^{n+1} - \mathbf{G}^n \Delta t$ and then add the new $\mathbf{G}^{n+1} \Delta t$. Therefore, we first update all accelerations $\mathbf{a}_f^{n+1/2} = \mathbf{g}^{n+1} + \sigma \kappa^{n+1/2} \boldsymbol{\nabla} \varphi^{n+1/2} / \langle \rho \rangle$ and calculate the face velocity $\mathbf{u}_{f,*}^{n+1}$ interpolating between neighboring cells and adding the new acceleration: $\mathbf{u}_{f,*}^{n+1} = \bar{\mathbf{u}}_{\mathrm{v},*}^{n+1} + \mathbf{a}_f^{n+1/2} \Delta t$. The resultant velocity field is not solenoidal, and therefore, the Chorin projection is applied (the next step).

3.5. Step 5: Chorin Projection

The velocity and pressure fields are strongly coupled. The direct integration of Equation (3) does not guarantee the incompressibility of the fluids; that is, the interpolated $\mathbf{u}_{f,*}^{n+1}$ derived from $\mathbf{u}_{\mathrm{v}}^{n+1}$ does not satisfy the incompressibility condition. To obtain a divergence-free velocity field $\mathbf{u}_f^{n+1}$, the Chorin projection step is used [76]. It is based on the Helmholtz decomposition whereby the velocity vector field is split into a solenoidal ($\boldsymbol{\nabla} \cdot \mathbf{u} = 0$) and a curl-free ($\boldsymbol{\nabla} \times \mathbf{u} = 0$) components. The projection begins with solving the pressure Poisson equation

$$\boldsymbol{\nabla} \cdot \alpha_f^{n+1/2} \boldsymbol{\nabla} p^{n+1} = \frac{\boldsymbol{\nabla} \cdot \mathbf{u}_{f,*}}{\Delta t}. \quad (23)$$

Subsequently, the non-solenoidal part of the velocity is subtracted as

$$\mathbf{u}_f^{n+1} = \mathbf{u}_{f,*} - \Delta t \alpha_f^{n+1/2} \boldsymbol{\nabla} p^{n+1}. \quad (24)$$

This guarantees that $\boldsymbol{\nabla} \cdot \mathbf{u}_f^{n+1} = 0$ up to a given threshold in the Poisson equation. After the computation of a new pressure field p^{n+1}, it becomes possible to compute $\mathbf{G}^{n+1}$, first on the faces, then in the centers of the cells using interpolation. Finally, $\mathbf{u}^{n+1}$ is computed as $\mathbf{u}^{n+1} = \mathbf{u}_{\mathrm{v},*}^{n+1} + \mathbf{G}^{n+1} \Delta t$.

The last step of the integration cycle is the grid adaptation [74,80] based on, for example, the volume fraction φ of fluid 1, volume fraction χ of the solid, or velocity $\mathbf{u}$. The grid is refined or coarsened following the standard wavelet-based adaptation criterion [81].

One final note of this section concerns the choice of the integration time step. There are two limiting factors in the time step Δt: the convection with the maximum flow velocity $|\mathbf{u}_{\max}|$, which comes from the advection in Equation (5), and the capillary waves with

the capillary phase speed $c_\sigma = \sqrt{0.5\sigma k/\langle\rho\rangle}$ [45,82], where $\langle\rho\rangle = (\rho_1+\rho_2)/2$, and k is the wavenumber. The maximum possible wavenumber in a finite difference scheme is $k_{\max} = \pi/h$, which corresponds to the minimum wavelength $\lambda_{\min} = 2h$. Hence, the time step is chosen from the minimum estimates:

$$\Delta t < \Delta t_{\text{CFL}} = \min\left(\frac{h}{|\mathbf{u}_f|}\right), \quad \Delta t < \Delta t_\sigma = \frac{h}{2c_\sigma} = \sqrt{\frac{h^3\langle\rho\rangle}{2\sigma\pi}}. \tag{25}$$

Summarizing the steps above, we have the following final algorithm.

We note here that steps 3 and 12 of Algorithm 1 are new and have previously not been implemented in Basilisk. Step 3 addresses the problem of fluid penetration into the solid, and step 12 accounts for the presence of solids implicitly (in numerical integration) rather than explicitly as in SSM.

Algorithm 1 The algorithm of multiphase modeling in Basilisk

Require: Set initial and boundary conditions for $\mathbf{u}$, φ, χ, and χ_f. Notation: $\mathbf{G} = -\boldsymbol{\nabla} p + \mathbf{a}_f$.

1: **while** $t < t_{\max}$ (time integration loop) **do**
2: Set time step Δt based on the CFL condition and surface tension restriction (Equation (25)).
3: Correct face velocity $\mathbf{u}^n_{f,*} = \left(1 - I^n_{\chi_f}\right)\mathbf{u}^n_f + I^n_{\chi_f}\mathbf{U}^n_{f,s}$.
4: Find volume fraction $\varphi^{n+1/2}$ using the VOF method in Equation (14).
5: Calculate fluid properties $\rho^{n+1/2}$, $\alpha^{n+1/2}_f$ and $\mu^{n+1/2}_f$ from $\varphi^{n+1/2}$ using Equations (13) and (12).
6: **advection step:**
7: Predict $u^{n+1/2}_{f,p}$ using Equation (18).
8: Correct $u^{n+1/2}_{f,p}$ to obtain divergence free $u^{n+1/2}_{f,*}$ (Chorin's projection in Equations (23) and (24)).
9: Calculate $\mathbf{u}^{n+1}_{\text{a}}$ using the Bell–Colella–Glaz scheme (Equation (19)).
10: **viscous step:**
11: Calculate $\mathbf{u}^{n+1}_{\text{a},*} = \mathbf{u}^{n+1}_{\text{a}} + \mathbf{G}^n\Delta t$.
12: BPM: find $\mathbf{u}^{n+1}_{\text{v}}$ based on χ^{n+1} using Equation (22).
13: Calculate $\mathbf{u}^{n+1}_{\text{v},*} = \mathbf{u}^{n+1}_{\text{v}} - \mathbf{G}^n\Delta t$.
14: Calculate acceleration $\mathbf{a}^{n+1/2}_f$ using Equation (6) and velocity $\mathbf{u}^{n+1}_{f,*} = \bar{\mathbf{u}}^{n+1}_{\text{v},*} + \mathbf{a}^{n+1/2}_f\Delta t$.
15: Correct $\mathbf{u}^{n+1}_{f,*}$ to obtain divergence free $\mathbf{u}^{n+1}_{f,**}$ and pressure p^{n+1} (Chorin's projection in Equations (23) and (24)).
16: Calculate $\mathbf{G}^{n+1}_f$ on faces and then interpolate to cells $\mathbf{G}^{n+1}$.
17: Update $\mathbf{u}^{n+1} = \mathbf{u}^{n+1}_{\text{v},*} + \mathbf{G}^{n+1}\Delta t$ using the new acceleration $\mathbf{a}_f$ and pressure p^{n+1}.
18: Adapt mesh.
19: **end while**

4. Numerical Simulations

In this section, we present the results of several simulations that are used to assess the validity and accuracy of the proposed algorithm. First, we model the two-dimensional Stokes flow of a single-phase fluid past a periodic array of identical cylinders. We investigate the convergence of the drag force to the theoretical solution and also compare the BPM with several other numerical methods. Second, we study the decaying vortex problem for which an analytical solution exists. Third is a multiphase case, in which we investigate the multiphase flow of a bulk viscous fluid with embedded droplets of a different fluid past a periodic array of cylinders. This problem has no analytical solution and is used for the purpose of verifying mass conservation. Fourth is the case involving a moving solid. We demonstrate the robustness of the proposed method with a situation in which a cylindrical solid covered with a layer of one viscous fluid starts moving inside another viscous fluid, which, at some point, ruptures the layer covering the solid. We have also verified that the algorithm satisfies the Galilean invariance by simulating the motion of a cylinder or a sphere along the interface between two fluids.

4.1. Viscous Flow Past a Periodic Array of Cylinders

Here, we solve numerically a problem that was studied theoretically in [83]. Consider an infinite square array of cylinders of radius r immersed in an incompressible viscous fluid with density $\rho = 1$ and viscosity $\mu = 1$. The cylinders are assumed stationary. The spacing between them is measured by the distance L between the centers of the nearest neighbor cylinders. Since the system is periodic, we consider a square box with size L, as shown in Figure 3. Then the distance between the surfaces of the adjacent cylinders is $L - 2r$, and the solid volume fraction is equal to $\varphi_s = \pi r^2/L^2$. The fluid motion is assumed to begin from rest under a uniform body force $\rho\mathbf{a}$ directed from left to right. The fluid flow reaches a steady state when the drag force becomes equal to the body force. The mean flow velocity in the x direction is denoted by $\hat{U}$. We assume that the Reynolds number $\mathrm{Re} = \rho\hat{U}L/\mu$ is small so that the Stokes flow approximation holds.

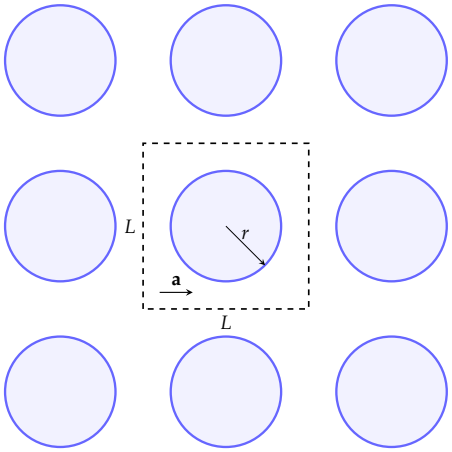

Figure 3. A schematic depiction of the problem studied in [83]. The unit periodic cell is a square with size L containing a cylinder of radius r in the center of the square. The fluid around the cylinder is set in motion by acceleration **a** directed to the right.

The steady-state problem was solved in [83], and we compare our simulation results to that work. In [83], the authors solved the creeping flow equations with the no-slip condition at the surface of the solids and periodic boundary conditions. They obtained an approximate solution and used it to calculate the dimensionless drag force f on a cylinder as

$$f = \frac{F}{\mu\hat{U}} = r\int_0^{2\pi} (\omega \sin\theta - p\cos\theta)\mathrm{d}\theta, \tag{26}$$

where F is the drag force per unit length, p is the pressure, ω is the vorticity, and the integration is taken over the cylinder boundary with polar angle θ. This force f strongly depends on the solid volume fraction φ_s, increasing with the radius of the cylinders, and it was shown to be given by

$$f = -\frac{8\pi(r/L)^2(2a_1 + b_1)}{2\ln(r/L) + 1}, \tag{27}$$

where a_1 and b_1 are certain coefficients from the series expansion of the solution, which are obtained numerically and which depend on φ_s. Asymptotic solutions for two extreme cases can be explicitly derived for: (i) dilute arrays $\varphi_s \ll 1$ and (ii) concentrated arrays $\varphi_s \to \varphi_{\max}$:

$$f = \begin{cases} 4\pi\left(\ln{\varphi_s}^{-1/2} - 0.738 + \varphi_s - 0.877\varphi_s^2 + 2.038\varphi_s^3 + O(\varphi_s^4)\right)^{-1}, & \varphi_s \ll 1, \\ \frac{9\pi}{2\sqrt{2}}\left[1 - \left(\frac{\varphi_s}{\varphi_{\max}}\right)^{1/2}\right]^{-5/2}, & \varphi_{\max} - \varphi_s \ll 1, \end{cases}$$

where $\varphi_{\max}$ is the volume fraction of the particles when the cylinders touch each other.

Values of f for different solid fractions are listed in Table 1. In the same table, we compare the relative error $E = (f_{\text{num}} - f)/f$ in drag force obtained by the Brinkman penalization method with two other methods: the simple splitting method (SSM) and the embedded boundary method (EBM), both available in Basilisk software. Here, f_{num} is the force computed from the numerical simulations. In the simple splitting method (SSM) [84,85], the *cell-centered* velocity update is given by $\mathbf{u} = \chi \mathbf{U}_s + (1 - \chi)\mathbf{u}_*$, where $\mathbf{u}_*$ is the velocity field calculated without accounting for the presence of solids. In the embedded boundary method (EBM) [86,87], the auxiliary forcing term is not explicitly computed as in the IB methods, but instead, the values of the state vector inside the solid body are set so that the fluxes through the solid surface vanish.

The simulation results are compared with the predictions of [83] in Table 1. The grid with $J_{\max} = 10$ is used, and the penalization coefficient is $\eta = 10^{-6}$. Since the dimensionless drag force f has a singularity when φ_s tends to $\varphi_{\max}$, the numerical error E grows with the increasing φ_s. The SSM shows poor convergence, especially for the narrow-gap cases, and the discrepancy between the analytical and numerical solutions diverges rapidly. The EBM is in good agreement over a wide range of porosity. The BPM is seen to have better accuracy at $\varphi_s \leq 0.7$. Overall, the EBM and BPM are comparable in their accuracy in this case. However, the BPM is easier to extend to multiphase flow problems than EBM.

Table 1. The dimensionless drag force f versus the solid volume fraction $\varphi_s = \pi r^2/L^2$ [83]. The relative error in the drag force $E = (f_{\text{num}} - f)/f$ for the simple splitting (SSM), embedded boundary (EBM), and Brinkman penalization methods (BPM). The same maximum level of refinement $J_{\max} = 10$ is used for all three methods. The BPM is used with penalization coefficient $\eta = 10^{-6}$.

φ_s	**0.05**	**0.10**	**0.20**	**0.30**	**0.40**	**0.50**	**0.60**	**0.70**	**0.75**
f	15.56	24.83	51.53	102.90	217.89	532.55	1.763×10^3	1.352×10^4	1.263×10^5
E (SSM)	0.33	0.31	0.34	0.4	0.48	0.59	0.75	0.93	0.99
E (EBM)	0.042	0.023	0.0011	0.013	0.021	0.021	0.06	0.041	0.037
E (BPM)	0.0023	0.0026	0.0029	0.0034	0.0043	0.005	0.0062	0.0046	0.14

To demonstrate the convergence of $E = (f_{\text{num}} - f)/f$ as η decreases, we fixed the maximum level of refinement at $J_{\max} = 9$ and varied η, as shown in Figure 4. The convergence depends on the number of cells $m = \delta/h$ per Brinkman layer $\delta = \sqrt{\eta\nu}$ (see Equation (11)). It can be seen that the increase of m leads to the deterioration of accuracy. As mentioned before, the Brinkman term in the momentum equation models a solid as a porous medium with small permeability, the latter corresponding to small values of η. Thus, η should be taken as small as possible. The decrease of m, and consequently of η, leads to the decrease of the error. However, at $m \ll 1$, when the Brinkman layer is under-resolved, the error oscillates with η, indicating a lack of convergence. The values $m = 1$ or $m = 2$ appear to be optimal in the sense that they are small enough, and yet, the error decreases monotonically. In Figure 5, we display the grid convergence of the error E as $h \to 0$ for different levels of resolution of the Brinkman layer. The rate of convergence is seen to be about 0.9 when $m \geq 1$, while at a smaller m, the monotonicity or even convergence is lost.

To summarize, convergence of the method can be guaranteed provided that: (1) fine features are properly resolved by using sufficiently large levels of refinement $J_{\max}$; (2) the number of cells in the Brinkman layer is $m \geq 1$ but not too large; and (3) the penalization coefficient η satisfies the estimate $\eta \approx (mh)^2/\nu$.

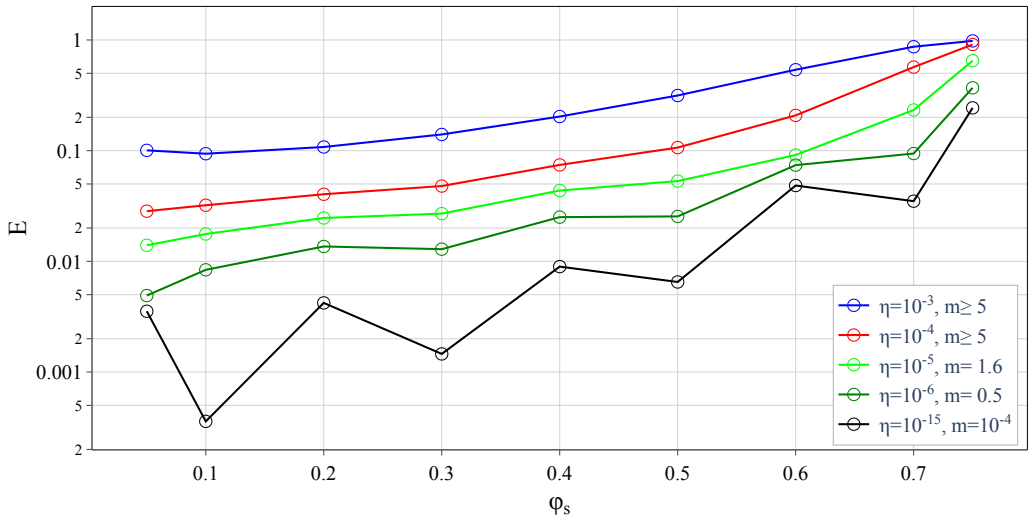

Figure 4. The plot of relative error $E = (f_{\text{num}} - f)/f$ versus the solid volume fraction φ_s for the fixed level of refinement $J_{\max} = 9$ and different penalization coefficients η. The value m denotes the number of cells needed to resolve the Brinkman layer in Equation (11).

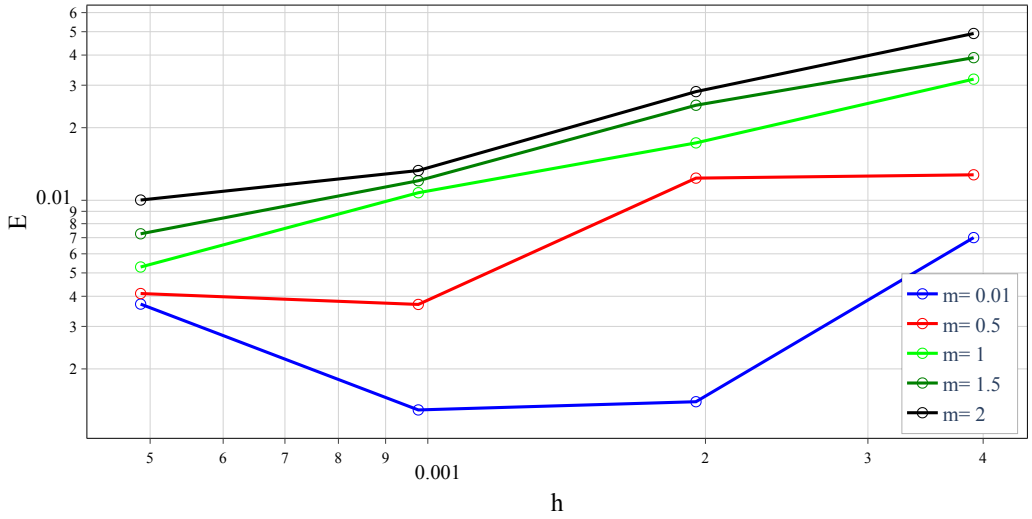

Figure 5. The relative error $E = (f_{\text{num}} - f)/f$ versus the minimum mesh size $h = L/2^{J_{\max}}$ with $J_{\max} = 8, 9, 10$, and 11 for the fixed solid volume fraction $\varphi_s = 0.3$ and different values of $m = \delta/h$, which is the ratio of the Brinkman layer thickness δ to mesh size h.

4.2. Decaying Vortex Problem

The previous test validated the steady-state predictions. To investigate dynamical consistency, we consider the problem of a two-dimensional decaying vortex for which an analytical solution is known. Specifically, the two-dimensional Navier–Stokes equations are solved exactly by the following velocity components: u, v, and pressure p [88] representing a periodic array of vortices decaying exponentially in time:

$$\begin{aligned} u(x,y,t) &= -\cos \pi x \sin \pi y \exp\left(-2\pi^2 t/\,\mathrm{Re}\right), \\ v(x,y,t) &= \sin \pi x \cos \pi y \exp\left(-2\pi^2 t/\,\mathrm{Re}\right), \\ p(x,y,t) &= -\frac{1}{4}(\cos 2\pi x + \cos 2\pi y) \exp\left(-4\pi^2 t/\,\mathrm{Re}\right), \end{aligned} \tag{28}$$

Here, the fluid density is taken as 1, while the characteristic velocity and length are both taken as 1. The Reynolds number Re is then the reciprocal of the kinematic viscosity $\mathrm{Re} = 1/\nu$.

We take as the computational domain a square of size $L_0 = 4$: $\Omega = [-2, 2]^2$, as shown in Figure 6. We split the domain into two parts: an inner square Ω_f of size 2 and a frame Ω_s around the inner square, which is used as a penalization region [89,90]. Therefore, the mask value χ inside Ω_f is 0, while in Ω_s, it is 1. The velocity inside the frame Ω_s is fixed and given by Equation (28), while the velocity in Ω_f is predicted numerically, starting with the analytical solution in Equation (28) at $t = 0$, and its consistency with the exact solution in Equation (28) at time $t = 0.3$ is verified. Some other parameters used in the simulation are: the CFL number $U_{\max}\Delta t/h = 0.3$, and the tolerance for the Poisson solver is 10^{-8}.

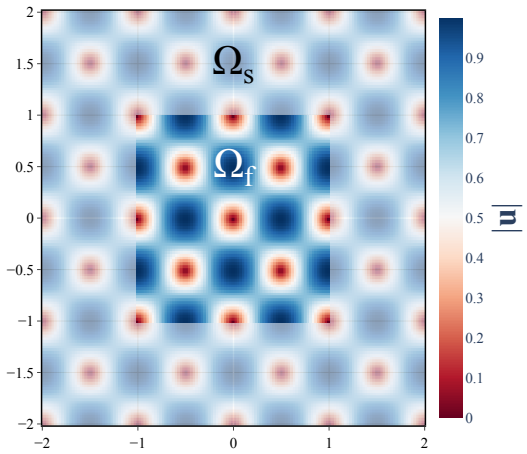

Figure 6. Schematics of the computational domain in the decaying vortex problem. In Ω_s, velocity and pressure fields are prescribed according to Equation (28). The solution in region Ω_f is predicted numerically. The initial field of velocity magnitude is shown by the color.

We point out here an aspect of the present computation that is important in general. One must pay careful attention to the selection of fields according to which the grid is adapted giving a higher priority to the fields that reflect the main physical features of the flow. In the case at hand, adaptation on both the velocity $\mathbf{u}$ and the frame mask χ was used during the first time step of the computations. Subsequently, the adaptation was performed only on the velocity field $\mathbf{u}$. This decision was based on the following observations: (1) the velocity field $\mathbf{u}$ captures the main physical characteristics of the flow; (2) adaptation on sharp masks leads to the maximum refinement for any adaptation threshold ε on interfaces even for smooth physical fields ($\mathbf{u}, \omega, p$) and, therefore, results in excessive use of computational resources without necessarily improving the accuracy of computations.

Ideally, during simulations, the maximum level of refinement $J_{\max}$ should not be reached; otherwise, this would indicate the general lack of resolution and the need for more refinement. The current level of refinement in all cells should be less than $J_{\max}$. The adaptation algorithm should automatically adjust the mesh in the required regions for a specified threshold ε. In practice, however, adaptation is often done on masks or other interfaces (χ, φ, etc.), which typically have sharp boundaries that require maximum adaptation, and thus, the maximum level of refinement is always reached [79]. Then, unless specifically tracked, one may miss flow regions away from the masks in which $J_{\max}$ is also reached and where there is therefore a lack of resolution. One must be careful to avoid such situations.

In what follows, we choose the maximum level of refinement as $J_{\max} = 13$. We observe that this number is never reached in practice, so $J < J_{\max}$. Recall, our adaptation is done based on $\mathbf{u}$, except for the first time step, when it is done additionally on the mask η. The grid size h remains in the range from $L_0/2^{11} = 2^{-9}$ to $L_0/2^6 = 2^{-4}$. To monitor

the resolution, we plot the dependence of the number of active cells on ε in Figure 7. The number of active cells is seen to be inversely proportional to the adaptation threshold, $N_{\text{active}} \propto \varepsilon^{-1}$. According to ([91], Equation 3.9), the number of active cells is related to the order of the numerical scheme as

$$N_{\text{active}} \leq C_1 \varepsilon^{-n/s},$$

where s is the discretization order, n is the dimensionality of the problem ($n = 2$ in the present case), and C_1 is a constant depending on the problem but not on the numerical scheme. From this relation, we derive that the discretization order is $s = 2$, which is consistent with the order of the numerical scheme.

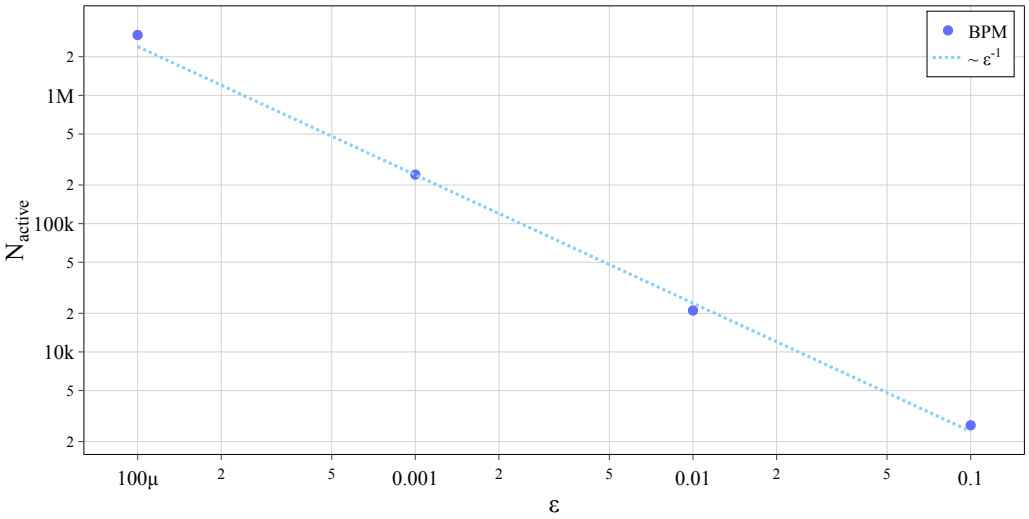

Figure 7. The log-log plot of the number of active cells N_{active} depending on the relative adaptation threshold ε at time $t = 0.3$. The dependence is well approximated by $N_{\text{active}} \propto \varepsilon^{-1}$ consistent with the estimate $N_{\text{active}} \propto \varepsilon^{-n/s}$ [91] if one takes the order of the scheme $s = 2$ (the dimension of the problem is $n = 2$).

The convergence with respect to the penalization coefficient η and adaptation threshold ε is shown in Figure 8. It is observed that the numerical error decreases with η approximately as $\eta^{4/5}$. Note that this order is between $\eta^{1/2}$ and η seen in Equation (10). The convergence rate lies between these functions because both normal and tangential components of the speed exist on the interface of the penalty area Ω_s. The error is seen to reach a plateau as η becomes very small, and the Brinkman layer is no longer resolved. A similar observation of plateau is found in [92], where the BPM is used to simulate compressible flows. The reason behind this phenomenon is that the Brinkman layer is not resolved properly and that $m = 0$ along the plateau. Convergence is also seen with the decrease of ε.

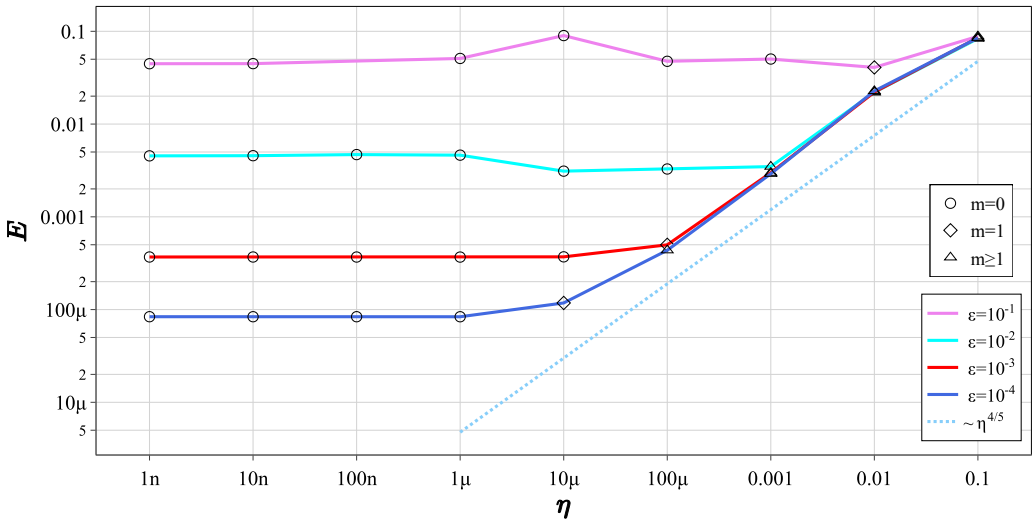

Figure 8. Convergence test for the vortex problem. The error $E = |\mathbf{u} - \mathbf{u}_*|/u_0$ as a function of the penalization coefficient η at time $t = 0.3$ and Reynolds number $\mathrm{Re} = 30$. Here, $\mathbf{u}_*$ is the theoretical velocity field given in Equation (28), and $u_0 = \exp\left(-2\pi^2 t/\,\mathrm{Re}\right)$ is the reference velocity. The plot is given for different adaptation thresholds $\log_{10}\varepsilon = -1, -2, -3, -4$. In all cases, the maximum level of refinement is set to $J_{\max} = 13$, and the current level of refinement satisfies $J < J_{\max}$.

4.3. Mass Conservation

In BPM, the solid phase is modeled as a porous medium with the porosity controlled by the penalization coefficient η. Therefore, the fluid phase can penetrate the porous solid leading to an apparent mass loss of the fluid. Of course, one should minimize this mass loss as much as possible. In order to evaluate the severity of the problem and to verify the conservation properties of the present method, we consider the following test problem. A bubble of radius r_1 is placed initially next to two identical fixed solid cylinders of radius r_2, as shown in Figure 9. The fluids are assumed to be at rest initially. The flow is initiated by gravity $\mathbf{g} = g\hat{\mathbf{i}}$ directed from left to right. Because of the left–right periodic boundary conditions, the bubble passes between the obstacles multiple times (more than 15). The bubble undergoes severe deformations in the process, especially in the case of the large bubbles, as shown in Figure 9. The purpose of this test is to verify conservation of the mass of the bubble after many such cycles.

Important dimensionless parameters that control the dynamics of the bubble in this problem are the capillary number $\mathrm{Ca} = \mu_1 U/\sigma$, the Reynolds number $\mathrm{Re} = UL\rho_1/\mu_1$, and the Froude number $\mathrm{Fr} = U/\sqrt{gL}$, with U denoting the mean velocity of the flow, L the domain size, σ the surface tension coefficient, and ρ_1, μ_1 the density and viscosity of the carrier fluid, respectively. We nondimensionalize the variables as follows:

$$\mathbf{x}_* = \frac{\mathbf{x}}{L},\ \mathbf{u}_* = \frac{\mathbf{u}}{U},\ t_* = \frac{tU}{L},\ \mu_{i,*} = \frac{\mu_i}{\mu_1\,\mathrm{Re}},\ \rho_{i,*} = \frac{\rho_i}{\rho_1},\ \sigma_* = \frac{1}{\mathrm{Re}\,\mathrm{Ca}},\ \mathbf{g}_* = \frac{1}{\mathrm{Fr}^2},$$

where the domain size is $L = 5$ mm, and the velocity is $U = 0.25$ m/s. Then the rescaled domain is a square with a unit side. The mesh adaptation occurs based on the volume fractions φ, χ, and velocity components u_i with the corresponding adaptation thresholds 10^{-3}, 10^{-3}, and 10^{-2}. Other parameters are given in Table 2. The system corresponds to an air bubble in water as a carrier fluid.

Table 2. Physical properties of the fluids and numerical parameters used in the simulation of the bubble passing between two cylinders. Here, $J_{\max}$ is the maximum level of refinement, and ϵ_p is the tolerance of the Poisson solver.

Name	Re	Ca	Fr	σ (mNm^{-1})	g (m^{2}s^{-1})	μ (Pa s)	ρ (kg m^{-3})	$J_{\max}$	ϵ_p
Water	1250	3.42×10^{-3}	1.13	73	9.8	1×10^{-3}	1000	7, 8, 9, 10	1×10^{-8}
Air						1.86×10^{-5}	1		

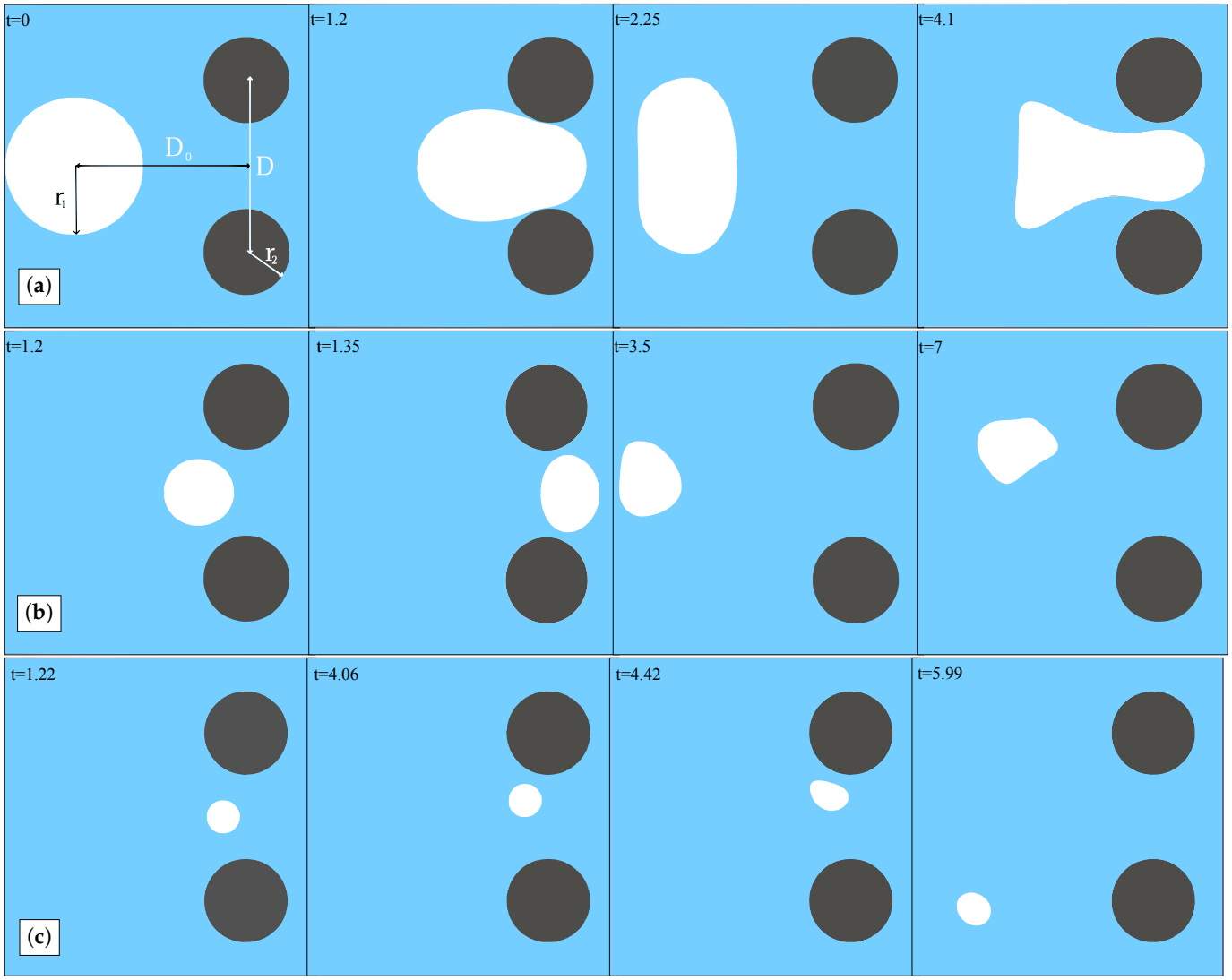

Figure 9. Air bubble passing between two fixed cylinders. Snapshots at different times t are shown. The cylinders of radius $r_2 = 0.125$ are spaced $D = 0.5$ units apart between their centers. The bubble is initially placed at distance $D_0 = 0.5$ from the centerline between the cylinders. The bubble radius is: (**a**) $r_1 = 0.2$; (**b**) $r_1 = 0.1$; (**c**) $r_1 = 0.05$.

Figure 10a displays the mean velocity of the flow, U_{mean}, as a function of time for the bubbles of different size. The steady-state mean velocity is seen to be smaller for the larger bubbles as expected due to their stronger interaction with the cylinders. Figure 10b displays the values of the relative error $E = \left(V_g - V_{g,0}\right)/V_{g,0}$ of the bubble mass as a function of time. As mentioned above, some error in the mass conservation is expected due to the Brinkman penalization term. In addition, the approximation errors of the overall algorithm, such as of the Poisson solver, will also contribute. However, we note that the error in the mass conservation is consistently at a value of the tolerance of the Poisson solver and does not increase appreciably with time during the simulation. If the bubble radius is smaller than the gap, we observe an initial monotonic increase of the error,

which eventually saturates after multiple passages through the gap. In contrast, for the larger bubbles, we notice a sudden jump in the error during the first passage between the cylinders at around time $t = 1.4$, Figure 10b. The large error is evidently due to the stronger interaction between the bubble and the obstacles. Importantly, these larger errors decrease back to the pre-interaction levels once the bubble squeezes between the obstacles. This indicates that even though some fluid mass may be entering the solid during the strong interaction, it is eventually recovered.

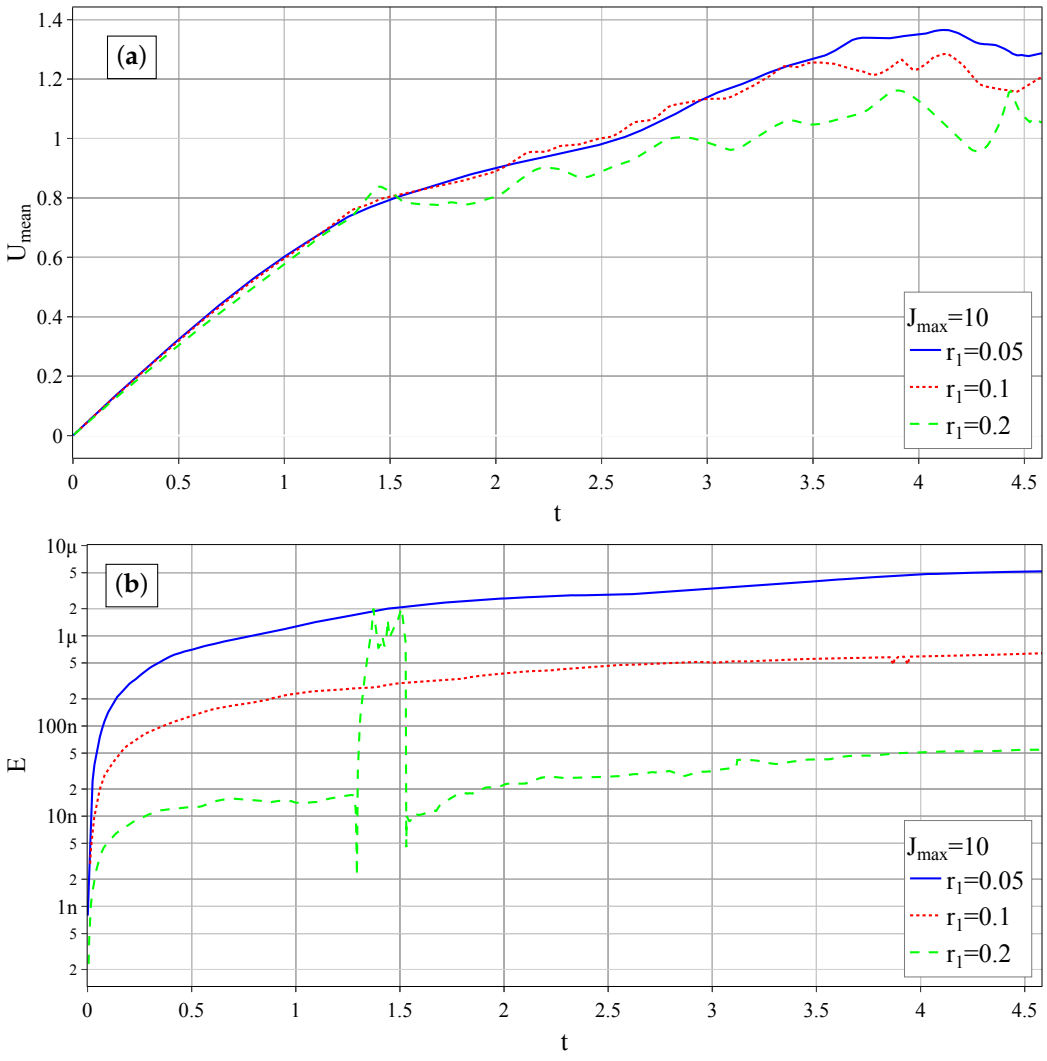

Figure 10. (**a**) The mean flow velocity in the domain, U_{mean}, as a function of time for the bubble passing between two obstacles. (**b**) The relative error $E = \left(V_g - V_{g,0}\right)/V_{g,0}$ in the bubble volume (or mass) as a function of time, where V_g is the current gas volume, and $V_{g,0}$ is the initial gas volume. Both graphs are shown for the bubble radii $r_1 = 0.05, 0.1$, and 0.2. The maximum level of refinement is $J_{\max} = 10$.

In Figure 11, we show the mean velocity and the mass conservation error as functions of time for the hardest case of a large bubble with a radius of $r_1 = 0.2$ when the level of refinement is varied. The important observation is that there is convergence as $J_{\max}$ increases. The level of error is generally small, and $J_{\max} = 10$ is at the level of Poisson solver's tolerance.

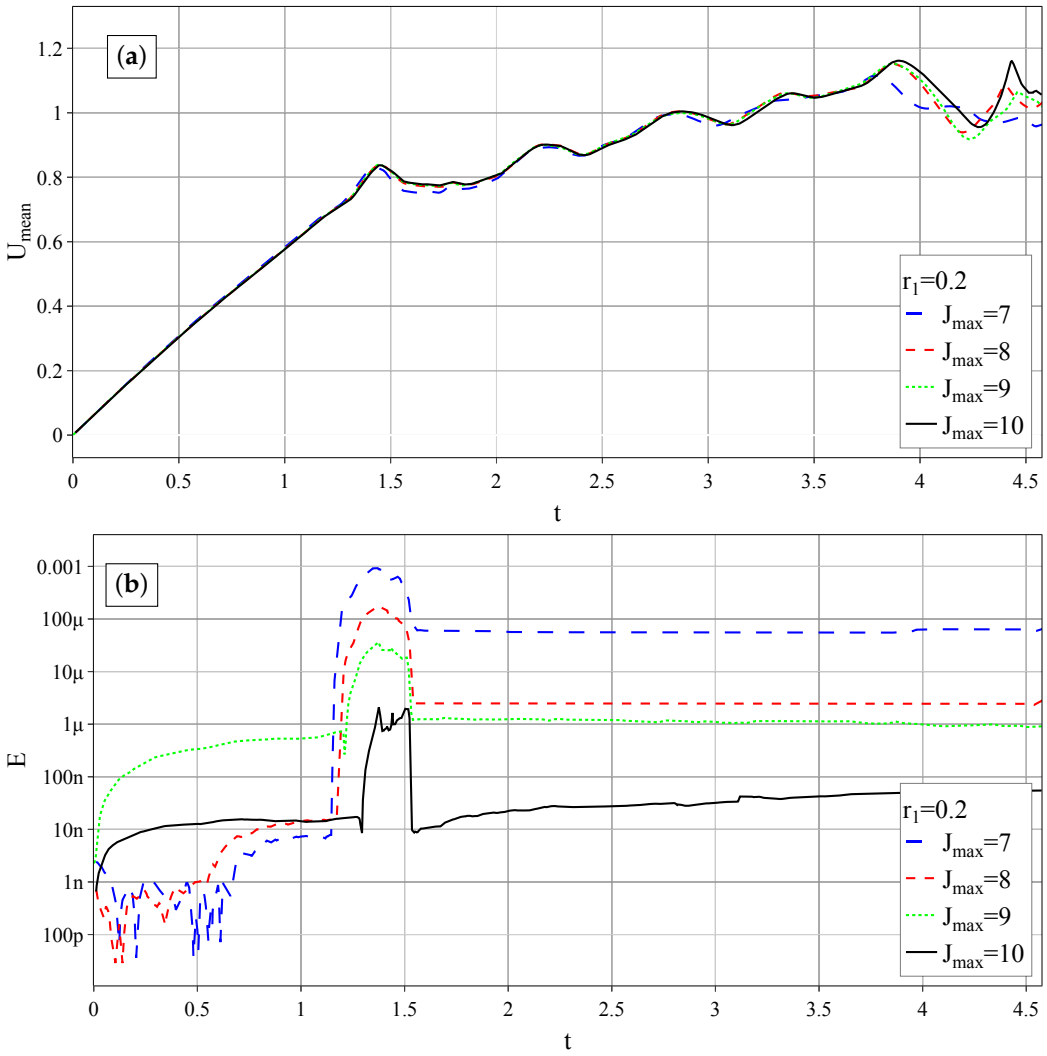

Figure 11. Convergence with the level of refinement for the case with a bubble of radius $r_1 = 0.2$. (**a**) The mean velocity in the domain, U_{mean}, as a function of time for different maximum levels of refinement. (**b**) The relative error $E = \left(V_g - V_{g,0}\right)/V_{g,0}$ in the bubble volume as a function of time.

4.4. Multiphase Flows around Moving Solids

In this section, we consider a two-phase system consisting of a composite droplet moving in a carrier fluid. The computational domain is a square with side $L = 1$ with periodic boundary conditions on the left and right and symmetry boundary conditions on the bottom and top. The composite droplet consists of a solid particle of radius $R_s = 0.0625$ embedded inside a fluid shell with the outer radius $R_b = 2R_s$, as shown in Figure 12a. The hydrophobic surface of the solid is modeled by $\varphi = 0$ inside the solid, see Equation (7). Initially, the fluid and the droplet have zero velocity, but the solid particle has a prescribed constant velocity $\mathbf{U}_s = 1 \cdot \hat{\mathbf{i}}$ in the x direction. The Reynolds and capillary numbers are taken as $\text{Re} = \rho_1 U_s L/\mu_1 = 100$ and $\text{Ca} = \mu_1 U_s/\sigma = 0.01$. The densities and viscosities of both fluids are chosen to be the same and equal to $\rho_1 = \rho_2 = 1$ and $\mu_1 = \mu_2 = \mu = 1/\,\text{Re}$, respectively. The surface tension is equal to $\sigma = 1/(\text{Re}\,\text{Ca})$. As the particle moves, it causes the fluids to move as well, and our goal is to simulate the resultant motion.

The numerical simulations are carried out using various treatments of the interaction between the fluids and the solid. We use $J_{\max} = 9$ as the level of resolution. Figure 12 displays the results. As the solid moves to the right, the fluid shell moves with it but lags behind due to inertia and the drag with the surrounding fluid. The shell turns into a thin film on the front side of the solid, which can subsequently rupture, and the resultant drop surrounding the solid can detach from it. The simulations using different treatments of

the interaction between the solid and the fluid yield different scenarios for the process, and these are explained below.

The method proposed in this work is compared with the simple splitting method (SSM, see Section 4.1). Recall that in SSM, the cell-centered velocity update is given by $\mathbf{u} = \chi \mathbf{U}_s + (1 - \chi)\mathbf{u}_*$, where $\mathbf{u}_*$ is the velocity field calculated without accounting for the presence of solids. Figure 12a shows the results based on the application of SSM without this correction. A close look at the solid surface shows that the shell fluid gradually penetrates into the interior of the solid. A similar artifact is seen on the back side of the solid as well, but this time fluid 2 gets out of the solid region at time $t = 0.4$ (recall that inside the solid, $\varphi = 0$ indicates the presence of the fictitious fluid 2).

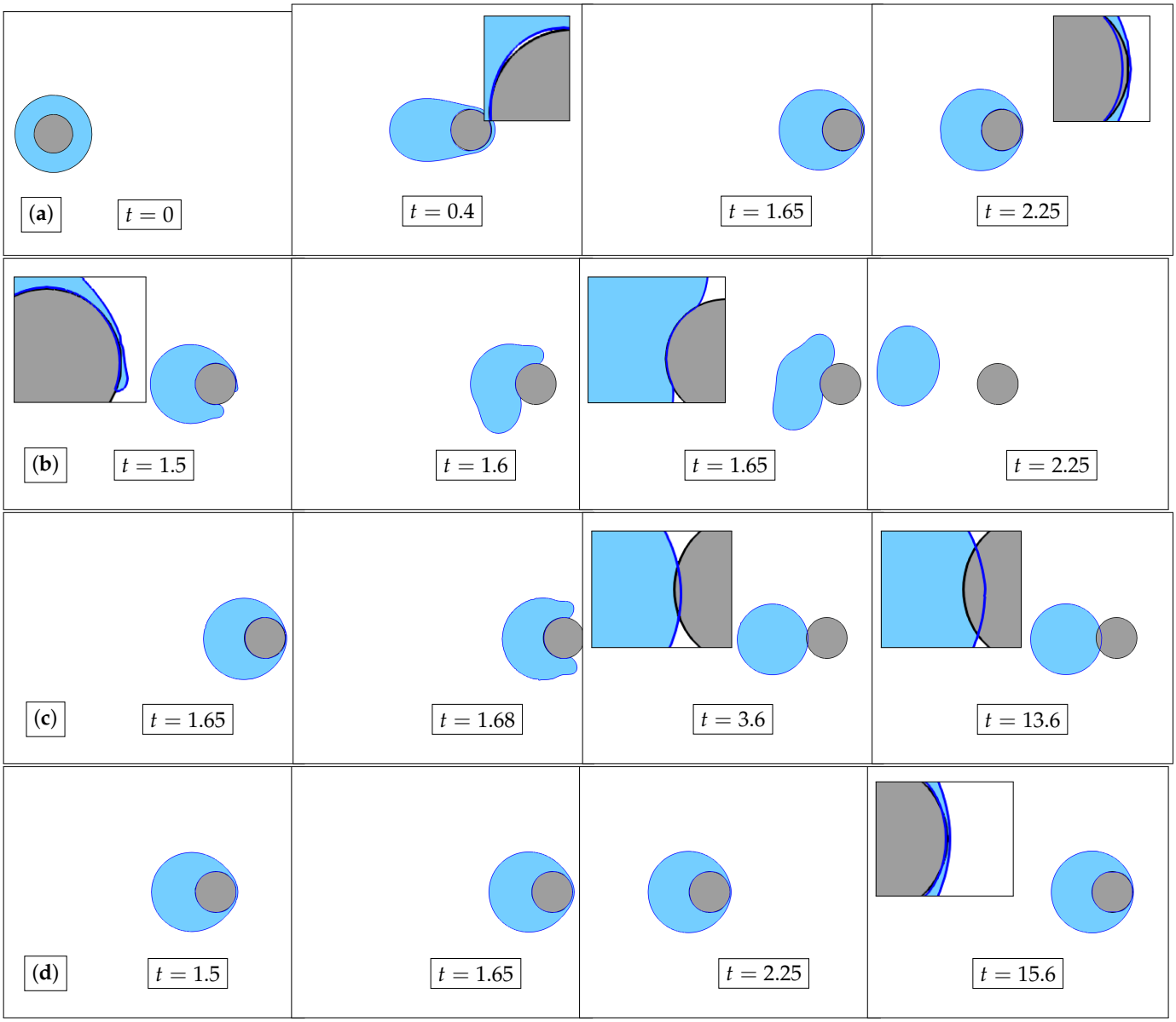

Figure 12. Motion of a solid cylinder (gray) covered by a fluid shell (blue) in a surrounding fluid (white). Left–right periodicity and top–bottom symmetry are assumed as boundary conditions. Simulations with different numerical methods: (**a**) SSM without velocity correction; (**b**) SSM with linear face velocity correction (Equation (29)); (**c**) BPM without velocity correction; and (**d**) BPM with nonlinear correction (Equation (15)). The results are shown at different times t. The insets highlight the relative location of the interfaces, where dark blue indicates the interface of fluid 1 and black indicates the solid boundary.

The penetration problems can be generally handled by various correctors of the *face* velocity $\mathbf{u}_f$. For example, the usage of the linear corrector

$$\mathbf{u}_f = \Big(1 - \chi_f\Big)\mathbf{u}_{f,*} + \chi_f \mathbf{U}_{f,s} \tag{29}$$

in SSM gives topologically different results, as shown in Figure 12b, where at some point the particle ruptures the fluid shell. No significant penetration into the solid is seen for the duration of the interaction. It should be noted that the application of SSM for this problem results in poor convergence of the Poisson solver, and subsequently, at large times, the overall error can be substantial (see the Supplementary Materials Movie S1).

In Figure 12c, we show the results based on BPM without the correction of $\mathbf{u}_f$. It is seen that the shell breaks symmetrically (see $t = 1.68$) in contrast to Figure 12b. Additionally, the film ruptures at a later time compared to the SSM with the corrector from Equation(29). Importantly, now the drop that forms behind the solid does not detach due to the fluid penetration effect. The penetration region continues to increase with time (compare the results at time $t = 3.6$ and $t = 13.6$).

The simulation results based on BPM with the nonlinear (due to the indicator function I_{χ_f}) corrector in Equation (15) are shown in Figure 12d. No appreciable penetration effect is observed until the simulated time $t = 15.6$ at least (see also the Supplementary Materials Movie S2). The mass-conservation error is also generally small. It depends primarily on the accuracy of the Poisson solver and the level of refinement $J_{\max}$. To quantify the mass conservation properties, we introduce the relative volume error $E = |V_d - V_{d,0}|/V_{d,0}$, where $V_{d,0}, V_d$ are fluid volumes at $t = 0$ and $t > 0$, respectively. This value for BPM is $E = 0.01\%$ for $t = 15.6$ compared to 1% for SSM or 0.7% for SSM with the face-velocity correction for the simulation time $t = 2.25$.

There is also a qualitatively different feature in Figure 12d that must be mentioned. The BPM with the nonlinear correction preserves the fluid shell intact. There is no rupture of the thin film on the front side, and eventually, there is a steady-state translation of the solid with the deformed fluid layer on it. We point out in this regard that in order to accurately and physically model the rupture of the thin film correctly, the underlying mathematical formulation has to incorporate the physics of the contact angle as well as the effects of disjoining pressure when the film thickness becomes too small. Since they are not part of our model, it is natural that the simulation algorithm does not lead to the film break-up. The problems of the contact line motion and of thin-film break-up have of course received much attention in the past (e.g., [93,94]). Their incorporation in the present algorithm is the subject of future work.

The final simulation of this problem of a composite droplet considers a situation in which the solid is covered with a very thin layer of fluid 2 that separates the solid from the thicker shell of fluid 1, forming a kind of a lubrication layer with a thickness of $\delta = 0.05R_s$ (see Figure 13). Note that the lubrication layer here and in other simulations below contains 1 or 2 grid points (i.e., $\delta \approx k \cdot h$, with $k = O(1)$) so that they are not required to be fully resolved. This simulation is done with the same method as in Figure 12d, which is BPM with a nonlinear corrector, and is aimed at eliminating the effects associated with the apparent contact angles. We observe that this case differs from the previous one in that the shell fluid now easily detaches from the lubricated solid similar to the case in Figure 12b. Unlike the latter, however, the rupture remains more symmetric, even though the symmetry is lost with time (see also the video of the process in the Supplementary Materials (Movie S3)).

We make one additional remark concerning the BPM with the linear corrector in Equation (29) for which we do not show any simulation results. Such a method gives the result without rupture; however, the fluid penetration into the solid over more than one computational cell is observed at sufficiently large times $t > 2$.

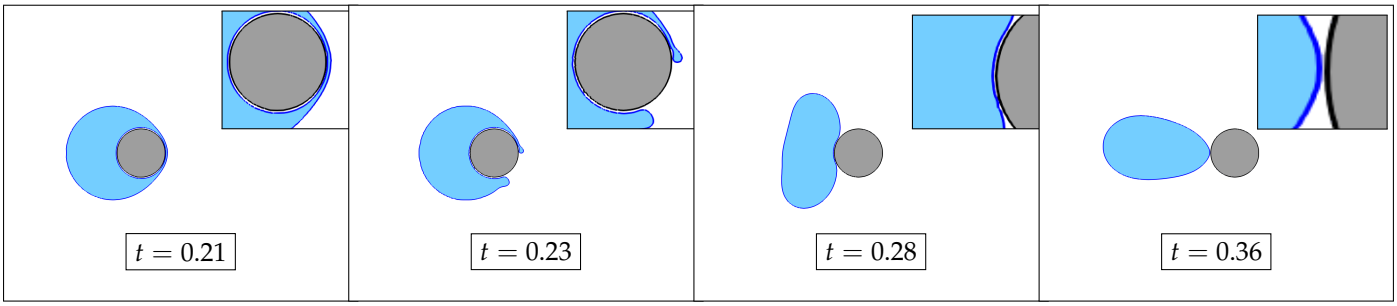

Figure 13. Motion of a solid cylinder (gray) covered by a fluid shell (blue) in a surrounding fluid (white). There is a lubrication layer with the thickness of $\delta = 0.05R_s$ between the fluid 1 and the solid. Left–right periodicity and top–bottom symmetry are assumed as boundary conditions. BPM with the nonlinear correction in Equation (15) is used. The results are shown at different times t. The insets highlight the relative location of the interfaces, where the dark blue indicates the interface of fluid 1 and the black indicates the solid boundary.

In order to further demonstrate the capabilities of the proposed BPM, we simulate a 2D mixer wherein many solids move in a medium consisting of two fluid phases—the carrier fluid with drops of another fluid moving in it. In Figure 14a, the schematics of the initial state are shown. The domain size is $L_0 = 1$, and again, periodic boundary conditions are used. The initial radii of the drops and solid particles are $R_b = R_s = 0.0625$. Initially, the centers of the four solids are placed uniformly along the y-axis at $x_s = 0.1$. One of a series of nine drops is arranged at a distance of $x_b = 0.4$ and $y_b = 0.5$, as shown in the same figure. The drops are placed in a rectangular order at a distance $L_b = 0.25$ in both vertical and horizontal directions. Velocities of the solid particles are directed along the x-axis in alternating orders with speed $|\mathbf{U}_s| = 1$. For simplicity, we assume both fluids have unit density and the same viscosity equal to $\mu = 1/\,\mathrm{Re}$. The surface tension is $\sigma = 1/(\mathrm{Re}\,\mathrm{Ca})$. Three different capillary numbers $\mathrm{Ca} = 0.01, 0.1$, and 1 are used in the simulations to account for the surface tension contribution ranging from dominant to negligible. The solids are assumed to be perfectly wetting.

The results are presented in Figure 14b–d at time $t = 1.39$ after two passes of the top solid across the boundary. In Figure 14b, the simulation is run for $\mathrm{Ca} = 0.01$, and some bubbles are found to merge. Increasing Ca, which is the same as decreasing the surface tension, leads to the formation of elongated bubbles with no coalescence (Figure 14d). A possible qualitative explanation of the latter may be that at large Ca, when the surface tension is small, the bubbles are essentially "enslaved" by the bulk liquid flow and simply follow along that flow. The interaction between bubbles is weak, and hence, coalescence or break-up do not occur easily. In contrast, at large surface tension, the bubbles are more "rigid", and they transfer the forces from different parts of the surrounding liquid, which leads to their stronger interaction with each other, which, in turn, facilitates coalescence or break-up.

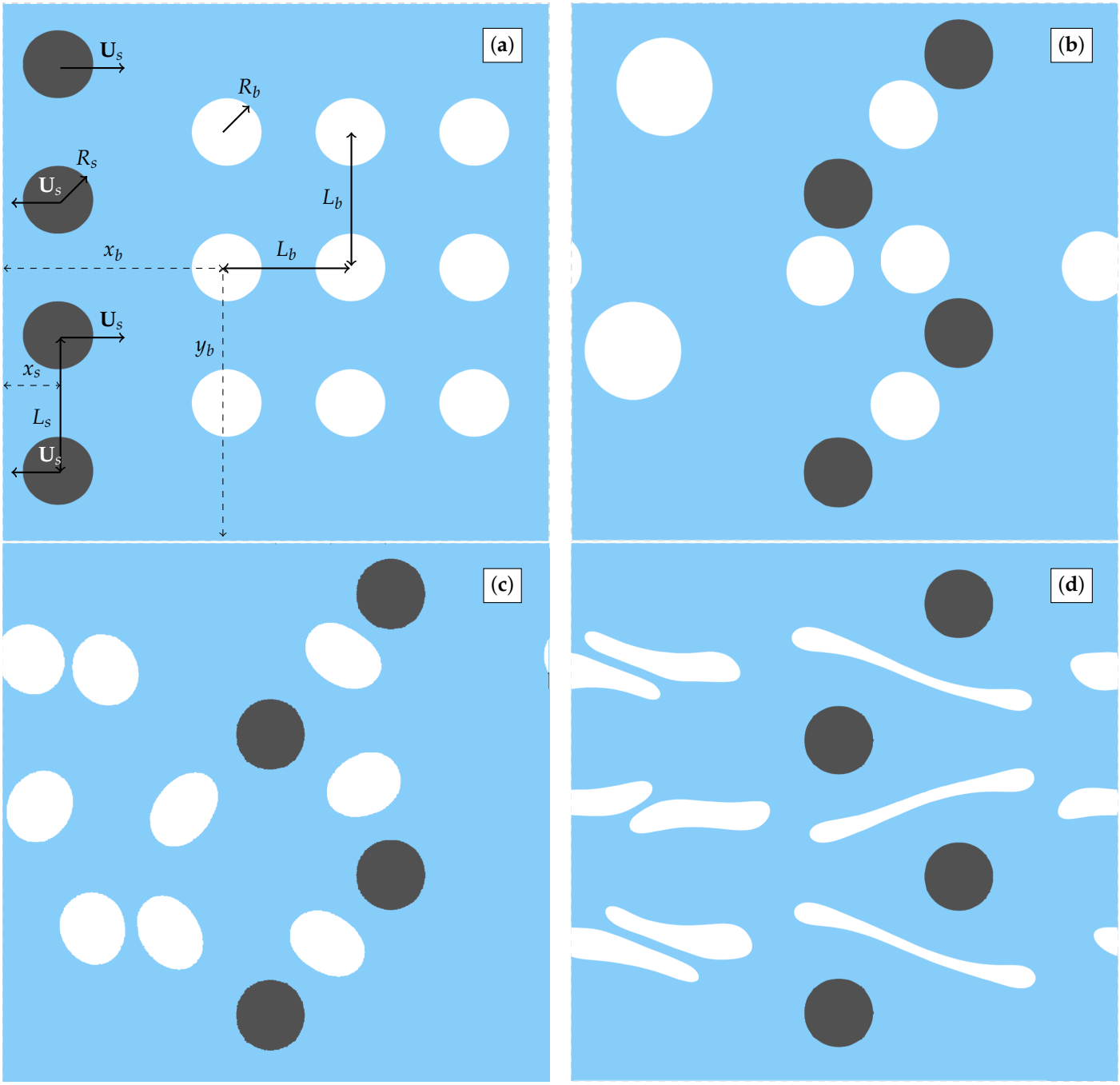

Figure 14. Mixing of a two-fluid medium by moving solids. Dark circles denote solid particles moving along x with a prescribed velocity. White circles indicate droplets of one fluid inside the carrier's second fluid, which is colored blue. The Reynolds number is $\mathrm{Re} = 100$, and different values of the capillary number are considered. (**a**) Initial condition. The other figures correspond to $t = 1.39$ and different Ca: (**b**) $\mathrm{Ca} = 0.01$; (**c**) $\mathrm{Ca} = 0.1$; (**d**) $\mathrm{Ca} = 1$. Time $t = 1.39$ corresponds to two passes of the top solid over the domain.

4.5. On Galilean Invariance

In this section, we verify that the proposed algorithm respects the Galilean invariance. For this purpose, we consider the motion of a cylinder or a sphere along an interface between two adjacent fluid layers. We first consider the two-dimensional case in detail and then show an example in three dimensions.

The computational domain is a square $\Omega = [-0.5, 0.5]^2$, and periodic boundary conditions on the right/left sides and symmetry conditions for top/bottom boundaries are used. We carry out the simulations with two different initial conditions: a fixed cylinder with moving fluids and a moving cylinder with fluids at rest. The radius of the cylinder

is $R_s = 0.0625$. In both cases, the initial difference in speeds is $|U| = 1$, and the velocity is directed along x. In the case of a moving cylinder, it begins its motion to the left from $x_s = 0.3125$. In the stationary case, the cylinder is fixed at $x_s = -0.3125$.

The surface of the solid is assumed to be hydrophobic to the fluid below. Then inside the solid, the fictitious volume fraction field is $\varphi = 0$. To avoid a sharp corner at the initial position of the contact point where the two fluids and the solid meet, we perform a smoothing operation that deforms the shape of the interface locally near the singular region to make it smoothly touch the solid surface. The smoothing is done using the so-called bounded blending operation, which can generate smooth transitions between two or more surfaces [95]. The result of smoothing is shown in Figure 15a.

In Figure 15b, the numerical results for the moving and fixed cases are compared at time $t = 2x_s/U = 0.625$ when the solid positions coincide. This figure shows that the solid surface appears to be partially wet (for example, the apparent contact angle on the back side (i.e., to the right) is about 120°). At the same time, the front side of the moving solid remains non-wetting. Clearly, this behavior misrepresents the real physics of the fluid–solid contact. Such serious discrepancies in the interface position near the solid contact region (and therefore everywhere else) arise due to the various features of the numerical algorithm, such as the flux corrections and the time splitting. As an additional (and likely related) difficulty, we observe no convergence of the contact-point positions with mesh refinement for the levels of refinement $J_{\max} = 9, 10$, or 11 (see Figure 15c).

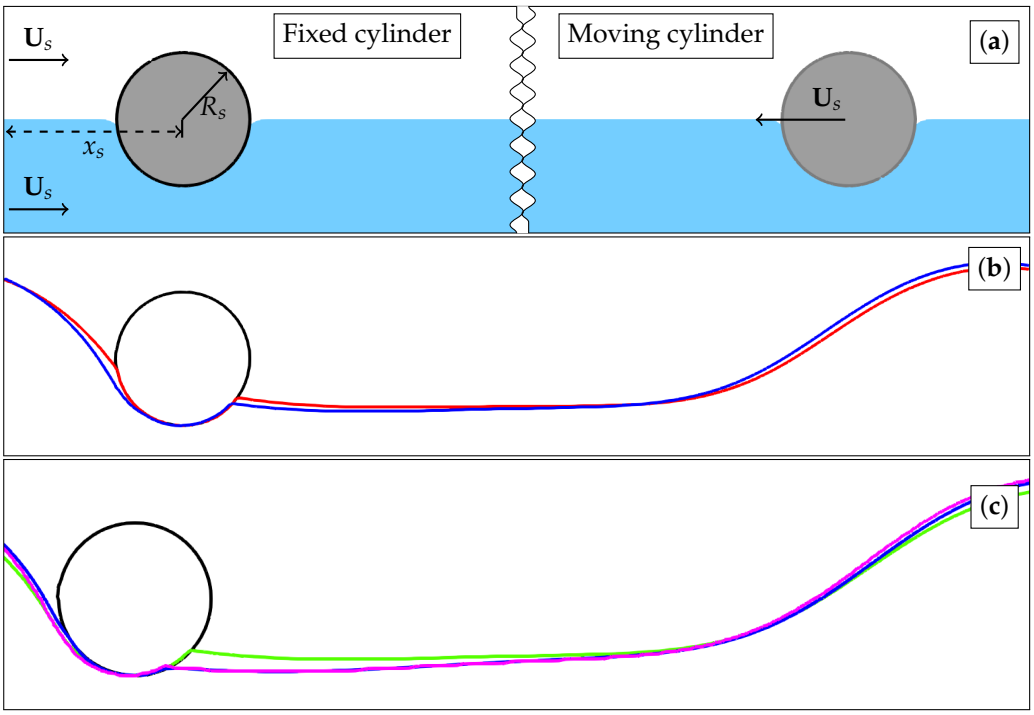

Figure 15. The test for Galilean invariance with a problem of a solid cylinder moving along the interface between two fluids. Figure (**a**) shows the initial conditions for a fixed cylinder (**left**) and moving cylinder (**right**). The arrows indicate the velocity direction in the fixed cylinder case and the cylinder direction for a moving cylinder case. Only a half of the computational domain is shown. Fluid 1 is blue, fluid 2 is white (above the interface and formally inside the solid), and solid is gray. In (**b**), the comparison of the fluid–fluid interface at $t = 0.625$ for a moving (blue) and fixed (red) cylinder are shown with the resolution level $J_{\max} = 10$. In (**c**), we test the convergence of the interface between the fluids in the case of a moving solid: $J_{\max} = 9$ (blue line); $J_{\max} = 10$ (green line); and $J_{\max} = 11$ (pink line).

In order to remedy the problem of the apparent contact angles in Figure 15b, we resort to the same approach as was taken in producing Figure 13. That is, we model the

hydrophobic surfaces with a thin lubrication layer around the solid. The film thickness is taken as $\delta = 0.05R_s$. The results are shown in Figure 16 at three times corresponding to the first three coincidences of the moving and fixed solids (recall, the motion is left–right periodic with the domain length 1, and the cylinder velocity is also 1). In all three time frames, the new algorithm is seen to respect the Galilean invariance with excellent accuracy. See also the Supplementary Movie S4.

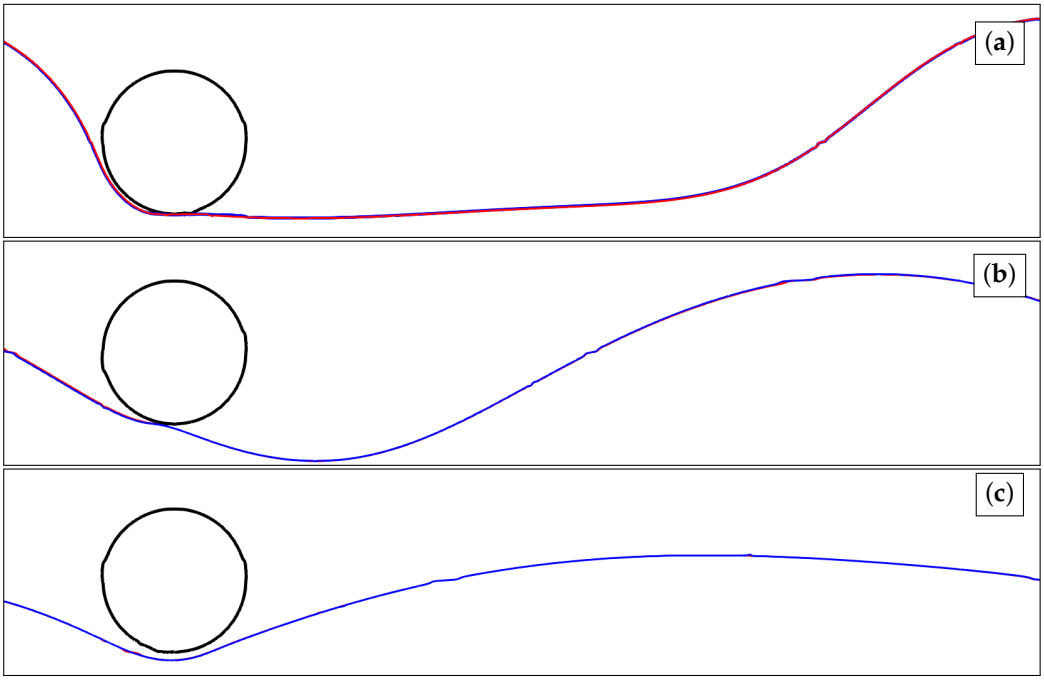

Figure 16. The same as in Figure 15 except for the presence of a thin film of fluid 2, with a thickness of $\delta = 0.05R_s$ on the solid surface. The level of refinement used is $J_{\max} = 8$. The results are shown at times: (**a**) $t = 0.625$; (**b**) $t = 1.625$; (**c**) $t = 2.625$ for a moving (blue) and fixed (red) cylinder. Notice that the blue and red curves are nearly coincident in all cases.

All of the simulations above considered two-dimensional problems for simplicity. The algorithm is, however, developed for general three-dimensional problems. Next, we demonstrate the application of the method to the problem of a sphere moving along the interface between two fluids, which is an extension of the previous result to three dimensions. The simulation was carried out in a domain $\Omega = [-0.5, 0.5]^3$, and the results are shown in Figure 17. The figure displays the solid and fluid interfaces at time $t = 0.2$. Additionally, the color levels and isocontours on the interface indicate the norm of the relative velocity $|\mathbf{u} - \mathbf{U}_s|$. We can see that the form of the isolines is practically the same for both the moving and stationary sphere, which confirms the Galilean invariance in this case as well. The video of the three-dimensional simulation is in the Supplementary Movie S5.

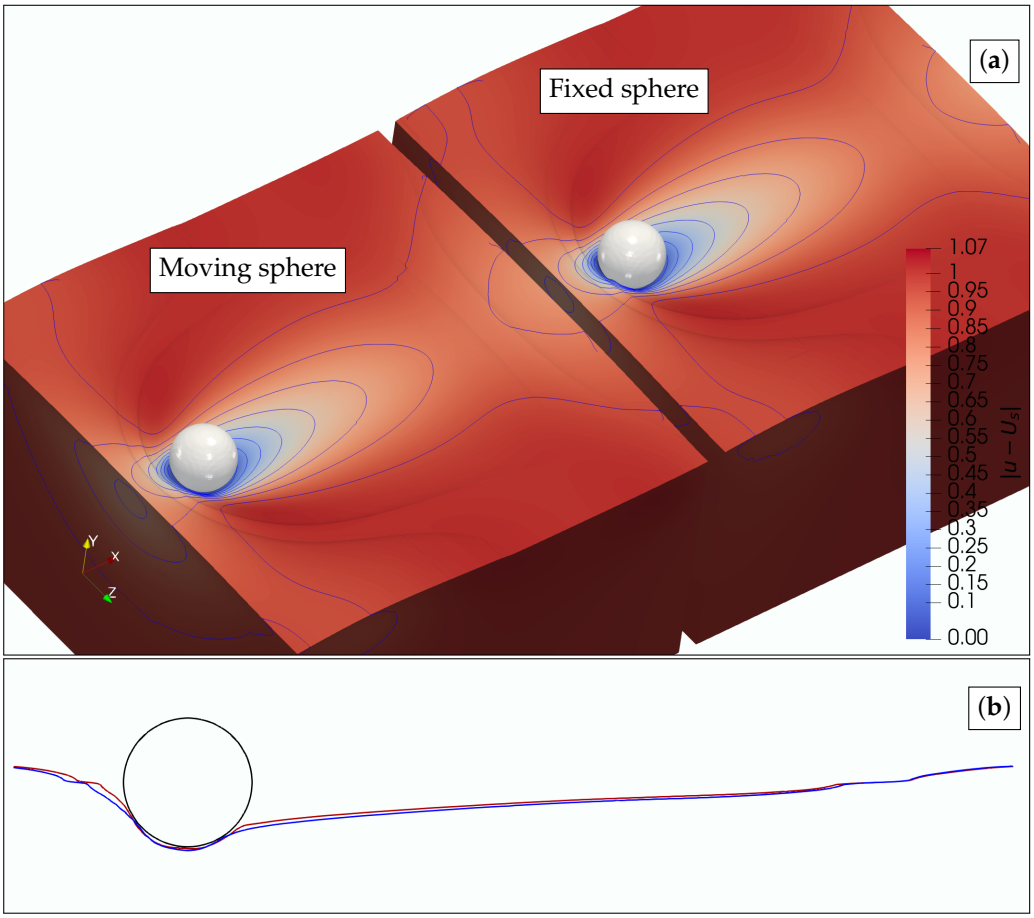

Figure 17. Galilean invariance test in three dimensions. (**a**) The case of a moving sphere and fixed fluids (left) and a fixed sphere and moving fluids (right) at time $t = 0.625$. The interface between the fluids is shown. The color and isolines indicate the norm of the relative velocity $|\mathbf{u} - \mathbf{U}_s|$. (**b**) shows the XY-plane slices through the sphere in (**a**), with the blue curve corresponding to the moving sphere and the red to the fixed sphere. The level of refinement used was $J_{\max} = 7$ so that the mesh is rather coarse compared to the previous two-dimensional case but still shows very good accuracy.

5. Conclusions

In this work, we have contributed to the development of an algorithm for the simulation of multiphase flows consisting of several incompressible fluids and solid objects, the latter either stationary or moving with a prescribed velocity. The solids are assumed to be either perfectly superhydrophilic or perfectly superhydrophobic so that no contact-angle effects are considered. The core of the algorithm consists of the Brinkman penalization method to handle the solids, the volume of fluid method to handle the fluid interfaces, and the continuous surface force to model surface tension phenomena.

The algorithm is implemented in the open source solver Basilisk [1,52] and is validated with a number of test cases. Simulations in Basilisk use adaptive Cartesian meshes that are highly efficient in capturing multiscale features of complex flows, such as the multiphase flows considered in this work. In particular, the algorithm is tested on the problems of Stokes flow past a periodic array of cylinders and of a decaying vortex flow for which there exist analytical solutions. The convergence of the method is demonstrated not only with an increasing grid resolution but also with a decreasing penalization coefficient η.

In addition, we have calculated the flow of a carrier fluid with bubbles and with several stationary or moving obstacles. The mass conservation property of the algorithm and its ability to handle the motion of a solid in a two-phase fluid is verified with a problem of a bubble squeezing between two solid obstacles and a problem of a solid breaking out of a surrounding fluid shell. Another case that we have simulated involves mixing

multicomponent fluids by means of moving solids. This problem demonstrates the ability of the algorithm to handle complex deforming interfaces and interactions between different fluids and solid obstacles. Lastly, we have verified that the algorithm satisfies the Galilean invariance. This test is carried out with a solid cylinder or sphere moving along an interface separating two different fluids.

Thus, we have demonstrated that the algorithm can accurately and efficiently handle a wide range of complex multiphase flow problems. Nevertheless, we point out some of the underlying assumptions that must be relaxed in order to extend the range of problems that can be simulated. One of the main assumptions is that the solid phase motion is assumed prescribed. Obviously, in practice, this is often not the case, and the fluid and solid motions are fully coupled. Another assumption is that of the incompressibility of all the fluid phases involved. Even though this is not such a strong limitation in many applications, the treatment of, say, bubbly liquids may require relaxing the incompressibility assumption about the gas.

Important additional physics that can be incorporated into the modeling and in the algorithm is that of the finite contact angle. As presented, the algorithm assumes that the solids are either perfectly superhydrophobic or perfectly superhydrophylic. The numerical treatment of these cases is also not without difficulties, especially when a solid moves in a two-phase medium. Our simulations with a composite droplet (a solid particle within a fluid shell) moving in an external fluid have shown that many numerical artifacts arising with different approaches to the treatment of the interaction between the solid and the fluids can be eliminated with an approach that allows for a thin (1–2 mesh cells) lubrication layer on the solid surface. Implementing this idea allows for the treatment of both stationary and moving solids with a physically correct and accurate representation of the interaction between the solid and fluids. The solution involving the lubrication layer avoids overlapping different sharp transition layers coming from the fluid–fluid and fluid–solid interfaces contributing to the robustness of the method.

Supplementary Materials: The following movies are available online at https://www.mdpi.com/article/10.3390/fluids6090334/s1, Movie S1: SSM_no_correction_jmax=9.mp4, Movie S2: BPM_ nonlinear_correction_jmax=9.mp4, Movie S3: BPM_nonlinear_correction_with_shell_r=1.05_jmax=9.mp4, Movie S4: 2D_Galilean_invariance_jmax=8.mp4, Movie S5: 3D_Galilean_invariance_jmax=7.mp4.

Author Contributions: Conceptualization, A.R.K.; formal analysis, E.L.S., O.A.R., and A.R.K.; funding acquisition, A.R.K.; investigation, E.L.S., O.A.R., and A.R.K.; project administration, A.R.K.; supervision, A.R.K.; writing—original draft, E.L.S. and A.R.K.; writing—review and editing, E.L.S. and A.R.K. All authors have read and agreed to the published version of the manuscript.

Funding: A.R.K. and O.A.R. were partially supported by the Russian Foundation for Basic Research (Grant No. 20-51-00004).

Data Availability Statement: Raw data were generated at Skoltech. Derived data supporting the findings of this study are available from the corresponding author A.R.K. on request.

Acknowledgments: We acknowledge the use of Skoltech's Arkuda cluster for obtaining the results presented in this paper. We would also like to thank one of the referees for extensive comments and suggestions that helped improve the paper. E.L.S. acknowledges discussions with Oleg V. Vasilyev concerning the BPM approach and the decaying-vortex problem. E.L.S. thanks the members of the Basilisk community for generously providing help on the inner workings of the Basilisk software.

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

References

1. Popinet, S & Collaborators. Basilisk Flow Solver and PDE Library. Available online: http://basilisk.fr (accessed on 1 August 2021).
2. Davanlou, A.; Lee, J.D.; Basu, S.; Kumar, R. Effect of viscosity and surface tension on breakup and coalescence of bicomponent sprays. *Chem. Eng. Sci.* **2015**, *131*, 243–255. [CrossRef]
3. Homma, S.; Koga, J.; Matsumoto, S.; Song, M.; Tryggvason, G. Breakup mode of an axisymmetric liquid jet injected into another immiscible liquid. *Chem. Eng. Sci.* **2006**, *61*, 3986–3996. [CrossRef]

4. Elghobashi, S. Direct numerical simulation of turbulent flows laden with droplets or bubbles. *Annu. Rev. Fluid Mech.* **2019**, *51*, 217–244. [CrossRef]
5. Zenit, R.; Feng, J. Hydrodynamic interactions among bubbles, drops, and particles in non-Newtonian liquids. *Annu. Rev. Fluid Mech.* **2018**, *50*, 505–534. [CrossRef]
6. Kuo, K.K.Y.; Acharya, R. *Fundamentals of Turbulent and Multiphase Combustion*; John Wiley & Sons: Hoboken, NJ, USA, 2012. [CrossRef]
7. Yang, L.; Nieves-Remacha, M.J.; Jensen, K.F. Simulations and analysis of multiphase transport and reaction in segmented flow microreactors. *Chem. Eng. Sci.* **2017**, *169*, 106–116. [CrossRef]
8. Xu, Q.; Zhao, T. Fundamental models for flow batteries. *Prog. Energy Combust. Sci.* **2015**, *49*, 40–58. [CrossRef]
9. Rivera, F.F.; Pérez, T.; Castañeda, L.F.; Nava, J.L. Mathematical modeling and simulation of electrochemical reactors: A critical review. *Chem. Eng. Sci.* **2021**, 116622. [CrossRef]
10. Soulaine, C.; Roman, S.; Kovscek, A.; Tchelepi, H.A. Mineral dissolution and wormholing from a pore-scale perspective. *J. Fluid Mech.* **2017**, *827*, 457–483. [CrossRef]
11. Soulaine, C.; Creux, P.; Tchelepi, H.A. Micro-continuum framework for pore-scale multiphase fluid transport in shale formations. *Transp. Porous Media* **2018**, *127*, 85–112. [CrossRef]
12. Carrillo, F.J.; Bourg, I.C.; Soulaine, C. Multiphase flow modeling in multiscale porous media: An open-source micro-continuum approach. *J. Comput. Phys. X* **2020**, *8*, 100073. [CrossRef]
13. Bourouiba, L. The fluid dynamics of disease transmission. *Annu. Rev. Fluid Mech.* **2021**, *53*, 473–508. [CrossRef]
14. Sebastian, B.; Dittrich, P.S. Microfluidics to mimic blood flow in health and disease. *Annu. Rev. Fluid Mech.* **2018**, *50*, 483–504. [CrossRef]
15. Dollet, B.; Marmottant, P.; Garbin, V. Bubble dynamics in soft and biological matter. *Annu. Rev. Fluid Mech.* **2019**, *51*, 331–355. [CrossRef]
16. Hamby, A.E.; Vig, D.K.; Safonova, S.; Wolgemuth, C.W. Swimming bacteria power microspin cycles. *Sci. Adv.* **2018**, *4*, eaau0125. [CrossRef]
17. Mogilner, A.; Manhart, A. Intracellular fluid mechanics: Coupling cytoplasmic flow with active cytoskeletal gel. *Annu. Rev. Fluid Mech.* **2018**, *50*, 347–370. [CrossRef]
18. Stone, H.; Stroock, A.; Ajdari, A. Engineering flows in small devices: Microfluidics toward a lab-on-a-chip. *Annu. Rev. Fluid Mech.* **2004**, *36*, 381–411. [CrossRef]
19. Tanimu, A.; Jaenicke, S.; Alhooshani, K. Heterogeneous catalysis in continuous flow microreactors: A review of methods and applications. *Chem. Eng. J.* **2017**, *327*, 792–821. [CrossRef]
20. Santra, S.; Mandal, S.; Chakraborty, S. Phase-field modeling of multicomponent and multiphase flows in microfluidic systems: A review. *Int. J. Numer. Methods Heat Fluid Flow* **2020**. [CrossRef]
21. 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]
22. Michaud, V. A review of non-saturated resin flow in liquid composite moulding processes. *Transp. Porous Media* **2016**, *115*, 581–601. [CrossRef]
23. Prosperetti, A.; Tryggvason, G. *Computational Methods for Multiphase Flow*; Cambridge University Press: Cambridge, UK, 2007. [CrossRef]
24. Bear, J. *Dynamics of Fluids in Porous Media*, 2nd ed.; Dover Publications: Mineola, NY, USA, 2013.
25. Brennen, C.E. *Fundamentals of Multiphase Flow*; Cambridge University Press: Cambridge, UK, 2005. [CrossRef]
26. Blunt, M.J. *Multiphase Flow in Permeable Media: A Pore-Scale Perspective*; Cambridge University Press: Cambridge, UK, 2017. [CrossRef]
27. Clift, R.; Grace, J.R.; Weber, M.E. *Bubbles, Drops, and Particles*; Dover Publications: Mineola, NY, USA, 2005.
28. De Gennes, P.G.; Brochard-Wyart, F.; Quéré, D. *Capillarity and Wetting Phenomena: Drops, Bubbles, Pearls, Waves*; Springer: New York, NY, USA, 2013. [CrossRef]
29. Unverdi, S.O.; Tryggvason, G. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.* **1992**, *100*, 25–37. [CrossRef]
30. 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]
31. Du, J.; Fix, B.; Glimm, J.; Jia, X.; Li, X.; Li, Y.; Wu, L. A simple package for front tracking. *J. Comput. Phys.* **2006**, *213*, 613–628. [CrossRef]
32. Boulton-Stone, J.M.; Blake, J.R. Gas bubbles bursting at a free surface. *J. Fluid Mech.* **1993**, *254*, 437–466. [CrossRef]
33. Anderson, D.M.; McFadden, G.B.; Wheeler, A.A. Diffuse-interface methods in fluid mechanics. *Annu. Rev. Fluid Mech.* **1998**, *30*, 139–165. [CrossRef]
34. Sun, Y.; Beckermann, C. Sharp interface tracking using the phase-field equation. *J. Comput. Phys.* **2007**, *220*, 626–653. [CrossRef]
35. 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]
36. Rider, W.J.; Kothe, D.B. Reconstructing volume tracking. *J. Comput. Phys.* **1998**, *141*, 112–152. [CrossRef]

37. Sussman, M.; Smereka, P.; Osher, S. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* **1994**, *114*, 146–159. [CrossRef]
38. Fedkiw, S.O.R.; Osher, S. Level set methods and dynamic implicit surfaces. *Surfaces* **2002**, *44*, 685. [CrossRef]
39. Weymouth, G.; Yue, D.K.P. Conservative volume-of-fluid method for free-surface simulations on Cartesian-grids. *J. Comput. Phys.* **2010**, *229*, 2853–2865. [CrossRef]
40. Pal, S.; Fuster, D.; Zaleski, S. A novel momentum-conserving, mass-momentum consistent method for interfacial flows involving large density contrasts. *arXiv* **2021**, arXiv:2101.04142.
41. Malan, L.; Malan, A.; Zaleski, S.; Rousseau, P. A geometric VOF method for interface resolved phase change and conservative thermal energy advection. *J. Comput. Phys.* **2021**, *426*, 109920. [CrossRef]
42. Youngs, D.L. An interface tracking method for a 3D Eulerian hydrodynamics code. *At. Weapons Res. Establ. (AWRE) Tech. Rep.* **1984**, *44*, 35.
43. Ubbink, O.; Issa, R. A method for capturing sharp fluid interfaces on arbitrary meshes. *J. Comput. Phys.* **1999**, *153*, 26–50. [CrossRef]
44. Tryggvason, G.; Scardovelli, R.; Zaleski, S. *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*; Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press: Cambridge, UK, 2011. [CrossRef]
45. Brackbill, J.U.; Kothe, D.B.; Zemach, C. A continuum method for modeling surface tension. *J. Comput. Phys.* **1992**, *100*, 335–354. [CrossRef]
46. Francois, M.M.; Cummins, S.J.; Dendy, E.D.; Kothe, D.B.; Sicilian, J.M.; Williams, M.W. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *J. Comput. Phys.* **2006**, *213*, 141–173. [CrossRef]
47. Kang, M.; Fedkiw, R.P.; Liu, X.D. A boundary condition capturing method for multiphase incompressible flow. *J. Sci. Comput.* **2000**, *15*, 323–360. [CrossRef]
48. Renardy, Y.; Renardy, M. PROST: A parabolic reconstruction of surface tension for the volume-of-fluid method. *J. Comput. Phys.* **2002**, *183*, 400–421. [CrossRef]
49. Ferrari, A.; Magnini, M.; Thome, J.R. A Flexible Coupled Level Set and Volume of Fluid (flexCLV) method to simulate microscale two-phase flow in non-uniform and unstructured meshes. *Int. J. Multiph. Flow.* **2017**, *91*, 276–295. [CrossRef]
50. Shepel, S.V.; Smith, B.L. On surface tension modelling using the level set method. *Int. J. Numer. Methods Fluids* **2009**, *59*, 147–171. [CrossRef]
51. Hua, J.; Mortensen, D. A front tracking method for simulation of two-phase interfacial flows on adaptive unstructured meshes for complex geometries. *Int. J. Multiph. Flow.* **2019**, *119*, 166–179. [CrossRef]
52. Popinet, S. An accurate adaptive solver for surface-tension-driven interfacial flows. *J. Comput. Phys.* **2009**, *228*, 5838–5866. [CrossRef]
53. Afkhami, S.; Bussmann, M. Height functions for applying contact angles to 2D VOF simulations. *Int. J. Numer. Methods Fluids* **2008**, *57*, 453–472. [CrossRef]
54. Karnakov, P.; Litvinov, S.; Koumoutsakos, P. A hybrid particle volume-of-fluid method for curvature estimation in multiphase flows. *Int. J. Multiph. Flow.* **2020**, *125*, 103209. [CrossRef]
55. Gueyffier, D.; Li, J.; Nadim, A.; Scardovelli, R.; Zaleski, S. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *J. Comput. Phys.* **1999**, *152*, 423–456. [CrossRef]
56. Scardovelli, R.; Zaleski, S. Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech.* **1999**, *31*, 567–603. [CrossRef]
57. Lafaurie, B.; Nardone, C.; Scardovelli, R.; Zaleski, S.; Zanetti, G. Modelling merging and fragmentation in multiphase flows with SURFER. *J. Comput. Phys.* **1994**, *113*, 134–147. [CrossRef]
58. Zhang, Y.; Jia, Y. 2D automatic body-fitted structured mesh generation using advancing extraction method. *J. Comput. Phys.* **2018**, *353*, 316–335. [CrossRef]
59. van Loon, R.; Anderson, P.D.; van de Vosse, F.N. A fluid–structure interaction method with solid-rigid contact for heart valve dynamics. *J. Comput. Phys.* **2006**, *217*, 806–823. [CrossRef]
60. Löhner, R.; Appanaboyina, S.; Cebral, J.R. Comparison of body-fitted, embedded and immersed solutions of low Reynolds-number 3-D incompressible flows. *Int. J. Numer. Methods Fluids* **2008**, *57*, 13–30. [CrossRef]
61. Mittal, R.; Iaccarino, G. Immersed boundary methods. *Annu. Rev. Fluid Mech.* **2005**, *37*, 239–261. [CrossRef]
62. Peskin, C.S. Numerical analysis of blood flow in the heart. *J. Comput. Phys.* **1977**, *25*, 220–252. [CrossRef]
63. Goldstein, D.; Handler, R.; Sirovich, L. Modeling a no-slip flow boundary with an external force field. *J. Comput. Phys.* **1993**, *105*, 354–366. [CrossRef]
64. Arquis, E.; Caltagirone, J. Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieu poreux: Applicationa la convection naturelle. *CR Acad. Sci. Paris II* **1984**, *299*, 1–4.
65. Brinkman, H. A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles. *Appl. Sci. Res.* **1949**, *1*, 27. [CrossRef]
66. Angot, P.; Bruneau, C.H.; Fabrie, P. A penalization method to take into account obstacles in incompressible viscous flows. *Numer. Math.* **1999**, *81*, 497–520. [CrossRef]
67. Schneider, K. Immersed boundary methods for numerical simulation of confined fluid and plasma turbulence in complex geometries: A review. *J. Plasma Phys.* **2015**, *81*, 435810601. [CrossRef]

68. Knipping, J.; du Toit, J.; Vuik, C. *Comparison of Wavelets for Adaptive Mesh Refinement*; Technical Report; Numerical Algorithms Group Ltd.: Oxford, UK, 2020.
69. Horgue, P.; Prat, M.; Quintard, M. A penalization technique applied to the "Volume-Of-Fluid" method: Wettability condition on immersed boundaries. *Comput. Fluids* **2014**, *100*, 255–266. [CrossRef]
70. Frederix, E.; Hopman, J.A.; Karageorgiou, T.; Komen, E.M. Towards direct numerical simulation of turbulent co-current Taylor bubble flow. *arXiv* **2020**, arXiv:2010.03866.
71. Vasilyev, O.; Kevlahan, N.R. Hybrid wavelet collocation–Brinkman penalization method for complex geometry flows. *Int. J. Numer. Methods Fluids* **2002**, *40*, 531–538. [CrossRef]
72. Schneider, K. Numerical simulation of the transient flow behaviour in chemical reactors using a penalisation method. *Comput. Fluids* **2005**, *34*, 1223–1238. [CrossRef]
73. Liu, Q.; Vasilyev, O.V. A Brinkman penalization method for compressible flows in complex geometries. *J. Comput. Phys.* **2007**, *227*, 946–966. [CrossRef]
74. Popinet, S. Gerris: A tree-based adaptive solver for the incompressible Euler equations in complex geometries. *J. Comput. Phys.* **2003**, *190*, 572–600. [CrossRef]
75. Bell, J.B.; Colella, P.; Glaz, H.M. A second-order projection method for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **1989**, *85*, 257–283. [CrossRef]
76. Chorin, A.J. A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.* **1967**, *2*, 12–26. [CrossRef]
77. Cummins, S.J.; Francois, M.M.; Kothe, D.B. Estimating curvature from volume fractions. *Comput. Struct.* **2005**, *83*, 425–434. [CrossRef]
78. Agbaglah, G.; Delaux, S.; Fuster, D.; Hoepffner, J.; Josserand, C.; Popinet, S.; Ray, P.; Scardovelli, R.; Zaleski, S. Parallel simulation of multiphase flows using octree adaptivity and the volume-of-fluid method. *Comptes Rendus Mécanique* **2011**, *339*, 194–207. [CrossRef]
79. Popinet, S. A quadtree-adaptive multigrid solver for the Serre–Green–Naghdi equations. *J. Comput. Phys.* **2015**, *302*, 336–358. [CrossRef]
80. van Hooft, J.A.; Popinet, S.; van Heerwaarden, C.C.; van der Linden, S.J.A.; de Roode, S.R.; van de Wiel, B.J.H. Towards adaptive grids for atmospheric boundary-layer simulations. *Bound. Layer Meteorol.* **2018**, *167*, 421–443. [CrossRef]
81. The Adaptive Wavelet Algorithm. Available online: http://basilisk.fr/sandbox/Antoonvh/the_adaptive_wavelet_algorithm (accessed on 12 September 2021).
82. Lamb, H. *Hydrodynamics*, 6th ed.; Cambridge Mathematical Library, Cambridge University Press: Cambridge, UK, 1975.
83. Sangani, A.; Acrivos, A. Slow flow past periodic arrays of cylinders with application to heat transfer. *Int. J. Multiph. Flow.* **1982**, *8*, 193–206. [CrossRef]
84. Beckermann, C.; Diepers, H.J.; Steinbach, I.; Karma, A.; Tong, X. Modeling melt convection in phase-field simulations of solidification. *J. Comput. Phys.* **1999**, *154*, 468–496. [CrossRef]
85. Al-Rawahi, N.; Tryggvason, G. Numerical simulation of dendritic solidification with convection: Two-dimensional geometry. *J. Comput. Phys.* **2002**, *180*, 471–496. [CrossRef]
86. Schwartz, P.; Barad, M.; Colella, P.; Ligocki, T. A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions. *J. Comput. Phys.* **2006**, *211*, 531–550. [CrossRef]
87. Johansen, H.; Colella, P. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *J. Comput. Phys.* **1998**, *147*, 60–85. [CrossRef]
88. Taylor, G. LXXV. On the decay of vortices in a viscous fluid. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1923**, *46*, 671–674. [CrossRef]
89. Huang, W.X.; Sung, H.J. Improvement of mass source/sink for an immersed boundary method. *Int. J. Numer. Methods Fluids* **2007**, *53*, 1659–1671. [CrossRef]
90. Kim, J.; Kim, D.; Choi, H. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *J. Comput. Phys.* **2001**, *171*, 132–150. [CrossRef]
91. Vasilyev, O.V. Solving multi-dimensional evolution problems with localized structures using second generation wavelets. *Int. J. Comut. Fluid Dyn.* **2003**, *17*, 151–168. [CrossRef]
92. Kasimov, N.; Dymkoski, E.; De Stefano, G.; Vasilyev, O.V. Galilean-invariant characteristic-based volume penalization method for supersonic flows with moving boundaries. *Fluids* **2021**, *6*, 293. [CrossRef]
93. Sui, Y.; Ding, H.; Spelt, P.D. Numerical simulations of flows with moving contact lines. *Annu. Rev. Fluid Mech.* **2014**, *46*, 97–119. [CrossRef]
94. Craster, R.V.; Matar, O.K. Dynamics and stability of thin liquid films. *Rev. Mod. Phys.* **2009**, *81*, 1131. [CrossRef]
95. Pasko, G.I.; Pasko, A.A.; Kunii, T.L. Bounded blending for function-based shape modeling. *IEEE Comput. Graph. Appl.* **2005**, *25*, 36–45. [CrossRef]