Journal of Computational Physics 391 (2019) 91–118

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

# Sharp numerical simulation of incompressible two-phase flows

Maxime Theillard [a,*], Frédéric Gibou [b,c], David Saintillan [d]

[a] School of Natural Sciences, Applied Mathematics Unit, University of California, Merced, CA 95340, United States of America
[b] Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106, United States of America
[c] Department of Computer Science, University of California, Santa Barbara, CA 93106, United States of America
[d] Department of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA 92093, United States of America

ARTICLE INFO

*Article history:*
Received 16 September 2018
Received in revised form 7 March 2019
Accepted 11 April 2019
Available online 17 April 2019

*Keywords:*
Two-phase flows
Incompressible Navier-Stokes
Projection method
Sharp jump conditions
Level-set
MAC grid

ABSTRACT

We present a numerical method for simulating incompressible immiscible fluids, in two and three spatial dimensions. It is constructed as a modified pressure correction projection method on adaptive non-graded Oc/Quadtree Cartesian grids, using the level-set framework to capture the moving interface between the two fluids. The sharp treatment of the interface position, of the fluid parameter discontinuities, and of the interfacial jump conditions ensures convergence in the $L^\infty$-norm. Using a novel construction for the pressure guess, we are able to alleviate the standard time step restriction incurred by capillary forces. The solver is validated numerically and employed to simulate the dynamics of physically relevant problems such as rising bubbles and viscous droplets in electric fields.

## 1. Introduction

Non-miscible incompressible two-phase flows are the cornerstone of a myriad of real-life science and engineering applications. Modeling and understanding them helps us to capture the physics of droplets, sprays and waves among other examples. In the engineering world, it has a tremendous impact on many areas, including the oil and gas, water treatment, biomedical and movie industries. While theoretical models of multiphase flows that couple the fluid dynamics of each phase with the motion of the interface separating them are well established, their mathematical complexity greatly restricts analytical studies and calls for numerical approaches. The complexity arises from many aspects of the problem, among which the sharp discontinuities in the fluid parameters and flow variables across the interface, which have to be addressed carefully.

The first robust numerical treatment enabling the simulation of multiphase flows is the discrete $\delta$-function method introduced by Peskin [28,29]. This immersed-boundary method allowed the coupling of a fluid and a solid on both sides of an irregular surface, even in the case where the solution varies rapidly. It was used to simulate multiphase flows in [45], where the interface between phases was captured using a front-tracking method. The level-set method [27] was used instead in [41] and [5] for its natural handling of topological changes. This approach was later extended to a coupled Volume of Fluid (VOF) Level-Set [39,42] to reduce the mass loss inherent to the level-set method. This work has inspired a

* Corresponding author.
*E-mail address:* mtheillard@ucmerced.edu (M. Theillard).

https://doi.org/10.1016/j.jcp.2019.04.024
0021-9991/© 2019 Published by Elsevier Inc.

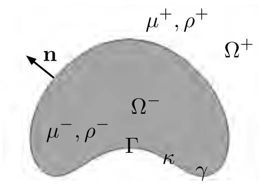


**Fig. 1.** Problem schematic and definitions: two immiscible fluids with densities $\rho^\pm$ and viscosities $\mu^\pm$ occupy subdomains $\Omega^\pm$ separated by interface $\Gamma$ with total curvature $\kappa$ and surface tension $\gamma$.

considerable amount of other methods, including the approach of Popinet [31] that employs adaptive graded Oc/Quadtree grids. Sussman et al. have also proposed a hybrid VOF/Level-Set approach [40].

However, the use of a $\delta$-function formulation artificially smears physical quantities across the interface, an artifact that was partially solved with the advent of the Ghost Fluid Method [9]. Specifically, the idea behind the Ghost Fluid Method was to develop a solver that preserves the sharp jump in physical quantities in the normal direction in the context of the projection method [17]. This strategy was then applied to multiphase incompressible flows in [26]. However, tangential jumps are smeared out within this framework. The Voronoi Interface Method [13] introduced a new approach that preserves the sharp jump in the normal and tangential directions simultaneously, recognizing that a Voronoi tessellation naturally makes the interface orthogonal to the flux.

Another limitation that is shared by most numerical strategies for this problem is the time step restriction, dictating how the time step should be chosen to ensure numerical stability. Brackbill proposed the first restriction in [4]. It is constructed with the intuition that in order for the method to be stable, surface capillary waves should be correctly resolved. While this is an intuitive condition, in practice it is very restrictive and prohibits simulations with very high spatial resolutions or in certain fluid parameter regimes. Schroeder et al. [36] reported that by treating capillary forces semi-implicitly stability can be achieved at time steps that exceed the classical restriction. More recently, Galusinsky et al. [12] proved that the time step can be greatly improved when the two fluids have identical viscosities and densities.

In this paper we present a novel computational approach for the two-phase flow problem that is entirely sharp, i.e., in which the interface and the discontinuities in fluid parameters and flow variables are treated in a sharp manner. It is found to be stable under a much less severe time step restriction inspired by [12]. It is constructed as a modified pressure correction projection method [6] and extends our stable solver for single phase flows [14]. The sharpness of the approach results from both the sharp representation of the interface using the framework presented in [24], and the sharp treatment of the continuity conditions at the interface, which extends our previous approach [25] for the Poisson problem with jump conditions. The improvement on the time step restriction is a consequence of the pressure correction method, and in particular, of the way we construct the pressure guess. It is implemented on non-graded Oc/Quadtree grids, which excel at capturing the inherent multiple length scales present in multiphase flows and provide broad refinement capabilities.

This paper is organized as follows. We first define the two-phase flow problem and introduce the relevant mathematical context and governing equations in Section 2. We introduce our overall method and describe the main steps by which the numerical solution is advanced in Section 3. We then discuss in Section 4 the various numerical techniques involved in the algorithm. Numerical examples and validations in two and three spatial dimensions are presented in Section 5 before we conclude in Section 6.

## 2. Governing equations

We consider a system $\Omega$ composed of two incompressible, immiscible, Newtonian fluids with constant densities $\rho^\pm$ and viscosities $\mu^\pm$. The two fluids occupy subdomains $\Omega^+$ and $\Omega^-$, which are separated by an interface $\Gamma$ (see Fig. 1). The velocities and pressure fields $(\mathbf{u}^\pm, p^\pm)$ satisfy the incompressible Navier-Stokes equations in each subdomain:

$$\rho^\pm \left( \frac{\partial \mathbf{u}^\pm}{\partial t} + \mathbf{u}^\pm \cdot \nabla \mathbf{u}^\pm \right) = \mu^\pm \Delta \mathbf{u}^\pm - \nabla p^\pm + \mathbf{f} \qquad \forall \mathbf{x} \in \Omega^\pm/\Gamma, \tag{1}$$

$$\nabla \cdot \mathbf{u}^\pm = 0 \qquad \forall \mathbf{x} \in \Omega^\pm/\Gamma, \tag{2}$$

where $\mathbf{f}$ is an arbitrary external body force. The moving interface $\Gamma$ is assumed to have constant surface tension $\gamma$, and may be subject to an external interfacial stress to be specified later. Continuity of the velocity and tractions across the interface is expressed as

$$[\![\mathbf{u}]\!] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{3}$$

$$[\![\boldsymbol{\sigma} \cdot \mathbf{n} - p\mathbf{n}]\!] = \gamma\kappa\mathbf{n} + \mathbf{g} \qquad \forall \mathbf{x} \in \Gamma, \tag{4}$$

where $\boldsymbol{\sigma} = \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ is the viscous stress tensor, $\kappa$ is the total curvature of the interface, and $[\![h]\!] = h^+ - h^-$ denotes the jump across the interface of a quantity $h$ defined in both phases. The interface unit normal vector $\mathbf{n}$ is chosen to be pointing from $\Omega^-$ into $\Omega^+$. Representing the fluid-fluid interface by a level-set function $\phi(\mathbf{x})$ such that

$$\phi(\mathbf{x}) = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{5}$$

the kinematic boundary condition at the interface is simply expressed as

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \qquad \forall \mathbf{x} \in \Omega. \tag{6}$$

This system of equations is completed with boundary conditions for the velocity and pressure fields on the outer boundary $\partial\Omega$ of the domain, which will be specified for each application.

## 3. Numerical approach

### 3.1. Overview of the algorithm

Our method is constructed as a traditional pressure correction projection method [6], where at every step a pressure guess needs to be constructed and viscous and projection steps are iterated until the interface and wall boundary conditions are satisfied within a desired tolerance. Specifically, at every time step $t_{n+1}$ the new solution $(\phi^{n+1}, \mathbf{u}^{n+1}, p^{n+1})$ is constructed from the previous solution $(\phi^n, \mathbf{u}^n, p^n)$ according to the algorithm presented in Fig. 2.

**(0) Initialization of the corrective terms**

Initialize the corrective velocity jump $\mathbf{X}_0$ and interfacial stress $\mathbf{\Sigma}_0$.

**(1) Pressure guess**

Construct the pressure guess $\tilde{p}$ as the solution of the Poisson problem

$$\triangle \tilde{p} = 0 \qquad \forall \mathbf{x} \in \Omega/\Gamma, \tag{7}$$

$$[\![\tilde{p}]\!] = -\gamma\kappa - \mathbf{g}\cdot\mathbf{n} \qquad \forall \mathbf{x} \in \Gamma, \tag{8}$$

$$\left[\!\left[\frac{1}{\rho}\nabla\tilde{p}\cdot\mathbf{n}\right]\!\right] = 0 \qquad \forall \mathbf{x} \in \Gamma. \tag{9}$$

**(2) Repeat until convergence**

**2a – Viscosity step**

Compute the intermediate velocity field $\mathbf{u}^*$ as the solution of

$$\rho\frac{D\mathbf{u}^*}{Dt} = \mu\triangle\mathbf{u}^* - \nabla\tilde{p} + \mathbf{f} \qquad \forall \mathbf{x} \in \Omega/\Gamma, \tag{10}$$

$$[\![\mathbf{u}^*]\!] = \mathbf{X}_k \qquad \forall \mathbf{x} \in \Gamma, \tag{11}$$

$$[\![\mu\nabla\mathbf{u}^*\cdot\mathbf{n}]\!] = \mathbf{g} - \mathbf{n}(\mathbf{g}\cdot\mathbf{n}) + \mathbf{\Sigma}_k \qquad \forall \mathbf{x} \in \Gamma. \tag{12}$$

**2b – Projection step**

The Hodge variable $\Phi$ is computed as the solution of

$$\triangle\Phi = \nabla\cdot\mathbf{u}^* \qquad \forall \mathbf{x} \in \Omega/\Gamma, \tag{13}$$

$$[\![\rho\Phi]\!] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{14}$$

$$[\![\nabla\Phi\cdot\mathbf{n}]\!] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{15}$$

and used to project the intermediate velocity on the divergence-free space:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \nabla\Phi. \tag{16}$$

Compute the new corrections $\mathbf{X}_{k+1}$ and $\mathbf{\Sigma}_{k+1}$ from the above velocity.

**(4) Interface evolution**

Construct the new level set $\phi^{n+1}$ from $\phi^n$ and $\mathbf{u}^{n+1}$ by solving Eq. (6).

**(5) Update**

Adapt the mesh to $\phi^{n+1}$ and $\mathbf{u}^{n+1}$ and update all the variables accordingly.

**Fig. 2.** Outline of the algorithm for the construction of the solution $(\phi^{n+1}, \mathbf{u}^{n+1}, p^{n+1})$ at time $t_{n+1}$ from the solution $(\phi^n, \mathbf{u}^n, p^n)$ at the previous time step $t_n$.

## 3.2. Remarks

### 3.2.1. Interfacial conditions

The corrective terms $\mathbf{X}_k$ and $\mathbf{\Sigma}_k$ appearing in the boundary conditions (11) and (12) are used to strictly enforce the interfacial jump conditions (3) and (4). They are designed such that

$$\lim_{k\to\infty} [\![\mathbf{u}_k^{n+1}]\!] = \mathbf{0} \qquad \forall \mathbf{x} \in \Gamma, \tag{17}$$

$$\lim_{k\to\infty} \mathbf{\Sigma}_k = -[\![\mu(\mathbf{u}^{n+1})^T \cdot \mathbf{n}]\!] \qquad \forall \mathbf{x} \in \Gamma. \tag{18}$$

The exact definition of the corrective velocity and stress as well as the proof of the convergence of the functional iterations will be detailed in Section 4.6. Clearly, Eq. (17) implies that continuity of the velocity across the interface is correctly enforced once convergence has been achieved. We note that even if convergence has not been reached, the jump in the normal velocity $[\![\mathbf{u}^{n+1}\cdot\mathbf{n}]\!]$ is zero by the no-flux boundary condition (15) imposed on the Hodge variable. Similarly, no-slip boundary conditions on the wall of the computational domain $\partial\Omega$ are enforced by imposing at every functional iteration that

$$\mathbf{u}^*_{k+1} = \mathbf{X}^{\partial\Omega}_{k+1} \qquad \forall \mathbf{x} \in \partial\Omega. \tag{19}$$

Proving that Eq. (18) enforces the dynamic boundary condition (4) is less straighforward. The proof relies on the following pressure reconstruction formula that will be justified in Section 4.3.1:

$$p^\pm = \tilde{p} + \frac{\alpha\rho^\pm}{\Delta t_n}\Phi - \mu^\pm \nabla\cdot\mathbf{u}^*, \tag{20}$$

where $\alpha$ is a dimensionless coefficient that we define later, and $\Delta t_n = t_{n+1} - t_n$. From the interface conditions (8) and (14) imposed on the pressure guess and the Hodge variable, it results that the imposed pressure jump is

$$[\![p]\!] = -\gamma\kappa - \mathbf{g}\cdot\mathbf{n} - [\![\mu\nabla\cdot\mathbf{u}^*]\!] \qquad \forall \mathbf{x} \in \Gamma. \tag{21}$$

In the limit of $k \to \infty$, dotting Eq. (21) with the unit normal, substracting it from (12) and inserting (18) yields

$$\left[\!\!\left[\mu\left(\nabla\mathbf{u}^* + (\nabla\mathbf{u}^{n+1})^T\right)\cdot\mathbf{n} - \left(p + \nabla\cdot\mathbf{u}^*\right)\mathbf{n}\right]\!\!\right] = \gamma\kappa\mathbf{n} + \mathbf{g} + [\![\mu\nabla\cdot\mathbf{u}^*]\!]\mathbf{n} \qquad \forall \mathbf{x} \in \Gamma. \tag{22}$$

Since by construction $\mathbf{u}^{n+1} = \mathbf{u}^* - \nabla\Phi$,

$$\nabla\mathbf{u}^* = \nabla(\mathbf{u}^{n+1} + \nabla\Phi) = \nabla\mathbf{u}^{n+1} + \nabla\nabla\Phi, \tag{23}$$

and because the Hessian of the Hodge variable $\nabla\nabla\Phi$ and the divergence of the intermediate velocity field $\nabla\cdot\mathbf{u}^*$ are nothing but second-order corrections, we find to first order:

$$\left[\!\!\left[\mu\left(\nabla\mathbf{u}^{n+1} + (\nabla\mathbf{u}^{n+1})^T\right)\cdot\mathbf{n} - p\mathbf{n}\right]\!\!\right] = \gamma\kappa\mathbf{n} + \mathbf{g} \qquad \forall \mathbf{x} \in \Gamma, \tag{24}$$

which is precisely the dynamic boundary condition (4). Similarly, in the monophasic case, Dirichlet boundary conditions for the velocity (no slip) can be enforced exactly while Neumann boundary conditions can only be enforced up to first order. We note that in many two-phase computational approaches, the stress term $[\![\mu(\nabla\mathbf{u}^{n+1})^T\cdot\mathbf{n}]\!]$ is neglected, which greatly simplifies the discretizations but is strictly incorrect. The jump conditions for the pressure guess (8) and the flux of the intermediate velocity field (12), and specifically the distinct treatment of normal stresses (which are enforced in the pressure guess) and tangential stresses (which enter the boundary condition for the viscosity step), is inspired by the work of Leveque and Li [20]. The main difference between their formulation and ours lies in the jump in the pressure flux (9): we set it to zero while in [20] it depends on the tangential derivative of the total interfacial stress, which cannot be computed accurately in the present case.

### 3.2.2. Necessity of the pressure guess definition

Within the context of the modified pressure correction projection method, the definition of the pressure guess is not unique, and in fact, any arbitrary pressure field can theoretically act as a pressure guess. Furthermore, even though we could intuit that some pressure guess definitions are better than others, there is *a priori* no strong argument for choosing one guess over another.

In this work, as Eqs. (7), (8) and (9) indicate, the pressure guess is constructed as the pressure induced by the capillarity and prescribed interfacial normal forces, along with the pressure boundary conditions imposed on the wall of the computational domain. As can be seen from the pressure reconstruction equation (20), one of the consequences of this definition is that, in order to satisfy the stress continuity and the pressure wall boundary conditions, homogeneous interface and boundary conditions have to be imposed for the Hodge variable (see Eqs. (14) and (15)). As we will see in Section 4.4.2,

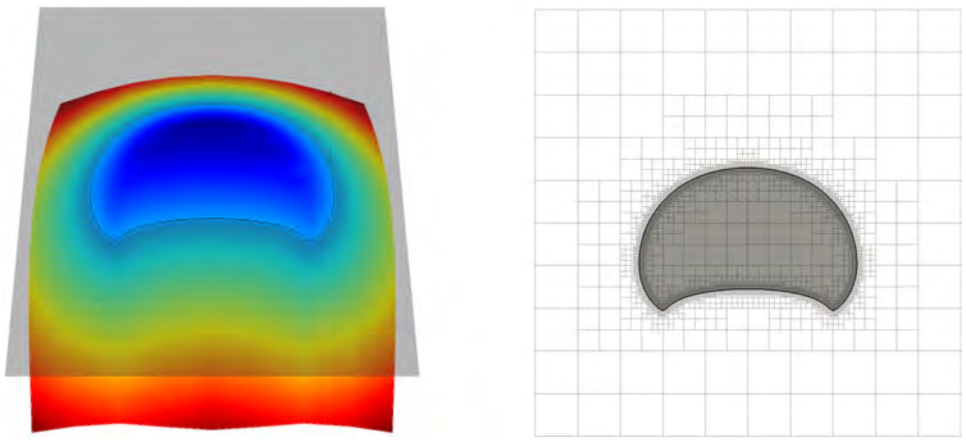


**Fig. 3.** (Left) The interface between the two fluids is represented using the level-set method. (Right) Sampling: the computational domain is discretized using Quadtree (Octree) grids in 2D (3D). In the case shown, the mesh is refined based on distance from the interface.

the homogeneity of the jump conditions for the projection system is a necessary assumption to prove the stability of the projecting step, a property of the highest importance here.

We note that this pressure guess definition is not the only one ensuring that the projection jump conditions are homogeneous. Indeed, any smooth pressure guess obeying the same interfacial and wall boundary conditions would lead to the same property. The pressure guess we consider here is probably among the easiest (since it uses the same solver as for the projection step), smoothest (since, by construction, it is among all these functions the one with the smallest gradient) and arguably cheapest to compute.

The main key finding of the present work is that using this new definition of the pressure guess allows for stable calculations to be performed under a less prohibitive time step restriction, without having to introduce artificial damping. This new restriction is discussed in detail in Section 4.7.

### *3.2.3. Convergence criterion*

The convergence criterion for the functional iterations in step 2 is based on the relative error of the corrective terms, which we impose to be below an arbitrary tolerance in the interfacial norm $\|\cdot\|_\Gamma$:

$$\max\left(\frac{\|\mathbf{X}_{k+1}-\mathbf{X}_k\|_\Gamma}{\|\mathbf{X}_{k+1}\|_\Gamma}, \frac{\|\boldsymbol{\Sigma}_{k+1}-\boldsymbol{\Sigma}_k\|_\Gamma}{\|\boldsymbol{\Sigma}_{k+1}\|_\Gamma}\right) < \epsilon, \tag{25}$$

where we choose $\epsilon = 10^{-5}$ in simulations. In our numerical experiments, the iterative method was found to converge rapidly (typically in 2 to 5 iterations), with the residual defined in (25) decreasing by a factor of 2 at every functional iteration. Final corrective terms at a given time step were found to be very good initial guesses for the next time step's functional iteration.

## 4. Numerical techniques

### *4.1. Interface representation: level-set method*

As illustrated in Fig. 3, the interface $\Gamma$ between the two fluids is represented using the level-set method. With this implicit representation, the fluid interface $\Gamma$ is represented by the zero contour of a so-called level-set function $\phi(\mathbf{x})$:

$$\Gamma = \{\mathbf{x} \quad |\phi(\mathbf{x}) = 0\}. \tag{26}$$

The subdomains $\Omega^-$ and $\Omega^+$ are naturally defined as:

$$\Omega^+ = \{\mathbf{x} \quad |\phi(\mathbf{x}) > 0\} \tag{27}$$

$$\Omega^- = \{\mathbf{x} \quad |\phi(\mathbf{x}) < 0\}. \tag{28}$$

Geometric quantities such as the normal $\mathbf{n}$ or the curvature $\kappa$ of the interface $\Gamma$ are given by:

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \qquad \text{and} \qquad \kappa = \nabla\cdot\mathbf{n} = \nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right). \tag{29}$$

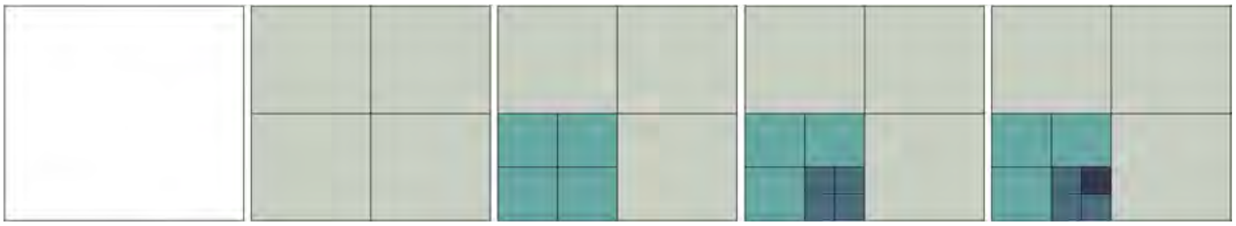


**Fig. 4.** Quadtree mesh generation (from left to right): we start from the root cell of the tree representing the entire domain, which is split into four identical cells. From there, we decide whether to split each of these four children cells. In the case depicted here, only the bottom left child cell is split. This process is then repeated recursively on each newly constructed cell until some terminating condition is reached.

The kinematic boundary condition on the deforming interface translates into an advection equation for the level-set function $\phi$

$$\frac{\partial \phi}{\partial t} + \mathbf{V}_\Gamma \cdot \nabla \phi = 0 \qquad \forall \mathbf{x} \in \Omega, \tag{30}$$

where the calculation of the interfacial velocity $\mathbf{V}_\Gamma$ is explained in Section 4.5. The level-set function is systematically reinitialized at each time step to ensure that it remains a signed distance function, i.e. $|\nabla\phi| = 1$. This is achieved by solving the following equation

$$\frac{\partial \phi}{\partial \tau} = \text{sign}(\phi)\,(|\nabla\phi| - 1) \qquad \forall \mathbf{x} \in \Omega, \tag{31}$$

for a fictitious time $\tau$. Mathematically, the reinitialization condition $|\nabla\phi| = 1$ uniquely defines the level-set function for any given contour. In practice, it has important consequences for the robustness and stability of the method.

Following the work of Min and Gibou [24], the level-set evolution equation (30) is solved using a second-order semi-Lagrangian method. The reinitialization equation (31) is solved using a Total Variation Diminishing second-order Runge-Kutta (TVD-RK2) method [38]. For the calculation of the normal vector and curvature, we employ the second-order accurate finite-difference formula given in [24] to discretize the definition (29). For an external velocity field, this framework was shown to yield second-order accuracy for the interface position and first-order accuracy for the curvature, both in $L^\infty$-norm and close to the interface. As we will see later on, our projection method is only first-order accurate in space, meaning that ultimately the numerical error in the interface position will be mostly due to the error in the velocity field rather than a consequence of the interface advection procedure.

## 4.2. *Sampling and data structure*

### 4.2.1. *Adaptive mesh refinement*

Our approach is implemented on non-graded Quadtree (2D) and Octree (3D) grids, which are constructed as depicted in Fig. 4: we start from a root cell representing the entire computational domain and split it into four (eight) children cells in 2D (3D). Based on some arbitrary splitting criterion, we then decide whether to split any of these children cells. This process is then recursively repeated on the newly created cells, and stopped when a terminating criterion is reached.

We define the level of a cell as the number of splits required for its construction. By definition, the level of the root cell is 0. If the root cell is of size 1, any cell of level $N$ will be of size $2^{-N}$. We also define $\min_{\text{level}}$ and $\max_{\text{level}}$ of a Quad/Octree as the minimum and maximum levels of any of its cells. A tree is called graded (or balanced) if the level difference between any two adjacent cells is always less than or equal to one; this is the case of Gerris [30]. Following [24], the mesh is systematically refined at its maximum level at the location of the interface using the following splitting criterion: split any cell $C$ if

$$\min_{v \in \text{nodes}(C)} |\phi(v)| \leq \text{Lip}(\phi)\,\text{diag}(C) \qquad \text{and} \qquad \text{level}(C) < \max_{\text{level}}, \tag{32}$$

where Lip($\phi$) is the Lipschitz constant of the level-set function $\phi$, diag($C$) is the length of the diagonal of cell $C$, and $\max_{\text{level}}$ is the predefined maximum level of the tree. We note that the above criterion can be easily modified to impose a uniform band of arbitrary width around the interface.

On top of this interface-based criterion, additional ones can be considered. For example, as previously done in [14,30, 23], the mesh can be automatically refined where high velocity gradients occur, which can be achieved using the following condition: split any cell $C$ if

$$\max_{v \in \text{nodes}(C)} \frac{||\nabla \mathbf{u}(v)||_\infty}{||\mathbf{u}||_{L^\infty(\Omega)}} \geq T_{\text{V}} \qquad \text{and} \qquad \text{level}(C) < \max_{\text{V}}, \tag{33}$$

where $||\nabla \mathbf{u}(v)||_\infty$ is the $L^\infty$-norm of the gradient of the velocity field at node $v$, $T_{\text{V}}$ is a predefined threshold and $\max_{\text{V}}$ is the maximum level of refinement allowed for this refinement criterion.

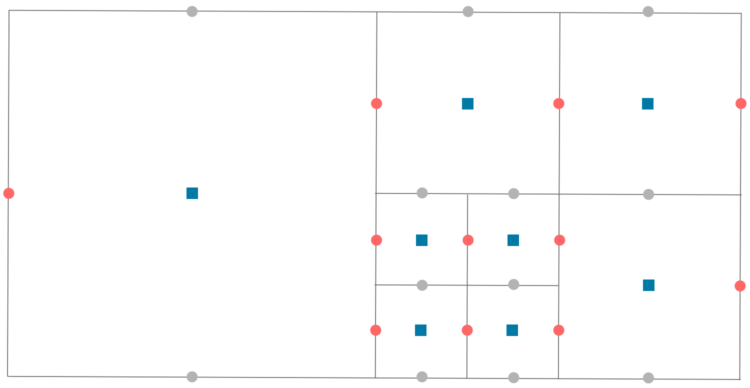


**Fig. 5.** MAC grid layout in 2D: the Hodge variable and pressure coefficients are stored at the cell centers (blue squares), while the velocity components are stored at the center of the faces (red and gray dots). Velocity components in the $x$-direction are stored on faces with a normal vector pointing in the same direction (red dots). Similarly, velocity components in the $y$-direction are stored on faces with a normal vector pointing in the $y$-direction (gray dots). The level-set coefficients are stored at the nodes. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

#### 4.2.2. *MAC grid layout*

As illustrated in Fig. 5, the quantities of interest (velocity and Hodge variables) are stored according to the Marker and Cell (MAC) layout [16]. The Hodge variable and pressure guess are stored at the cell centers, the velocity components are stored at the faces, and the level-set values are stored at the nodes. As pointed out by Guittet et al. [14], while on uniform grids this data structure easily leads to a stable discretization of the viscosity and projection steps, its implementation on adaptive grids is more challenging as we will see in Sections 4.3 and 4.4.

Constructing interpolation and extrapolation procedures for quantities defined at the cell centers and faces also becomes more challenging. Following our previous work, the interpolations are based on the third-order accurate least-squares method while the extensions are constructed geometrically with the same accuracy. We refer the interested reader to [14] for a detailed presentation of these procedures. For computational efficiency, copies of interpolated velocities $\mathbf{u}^{\pm}_{\text{nodes}}$ are stored at the nodes. This nodal velocity field is used to discretize the convective terms during the viscosity step as well as to reconstruct the interfacial velocity in Section 4.5.

### 4.3. *Viscosity step*

#### 4.3.1. *Temporal discretization: SLBDF scheme*

The temporal discretization of the viscosity step is chosen to ensure its stability: we combine a second-order Semi-Lagrangian (SL) method with a second-order Backward Difference Formula (BDF) scheme, leading to the following discretization for the modified momentum equation (10) in each phase:

$$\rho\left(\alpha\frac{\mathbf{u}^*-\mathbf{u}^n_d}{\Delta t_n}+\beta\frac{\mathbf{u}^n_d-\mathbf{u}^{n-1}_d}{\Delta t_{n-1}}\right)=\mu\Delta\mathbf{u}^*+\mathbf{f}-\nabla\tilde{p}, \tag{34}$$

where the weighting coefficients $\alpha$ and $\beta$ are defined in terms of time steps $\Delta t_n$ and $\Delta t_{n-1}$ as

$$\alpha=\frac{2\Delta t_n+\Delta t_{n-1}}{\Delta t_n+\Delta t_{n-1}} \quad \text{and} \quad \beta=-\frac{\Delta t_n}{\Delta t_n+\Delta t_{n-1}}. \tag{35}$$

Following the semi-Lagrangian implementation of [23], the departing velocities $\mathbf{u}^n_d$, $\mathbf{u}^{n-1}_d$ on the face centered at $\mathbf{x}^{n+1}$ are defined as the velocities at times $t_n$, $t_{n-1}$ and departure points $\mathbf{x}^n_d$, $\mathbf{x}^{n-1}_d$ from which the characteristic curves originate. These points are calculated using a backward RK2 scheme:

$$\mathbf{x}^*=\mathbf{x}^{n+1}-\frac{\Delta t_n}{2}\mathbf{u}^n(\mathbf{x}^{n+1}), \tag{36}$$

$$\mathbf{u}^*=\left(1+\frac{\Delta t_n}{2\Delta t_{n-1}}\right)\mathbf{u}^n(\mathbf{x}^*)-\frac{\Delta t_n}{2\Delta t_{n-1}}\mathbf{u}^{n-1}(\mathbf{x}^*), \tag{37}$$

$$\mathbf{x}^n_d=\mathbf{x}^{n+1}-\Delta t_n\mathbf{u}^*, \tag{38}$$

and

$$\mathbf{x}^*=\mathbf{x}^{n+1}-\Delta t_n\mathbf{u}^n(\mathbf{x}^{n+1}), \tag{39}$$

$$\mathbf{u}^*=\mathbf{u}^n(\mathbf{x}^*), \tag{40}$$

$$\mathbf{x}_d^{n-1} = \mathbf{x}^{n+1} - (\Delta t_n + \Delta t_{n-1})\,\mathbf{u}^*. \tag{41}$$

The quantities $\mathbf{u}^n(\mathbf{x}^*)$, $\mathbf{u}^{n-1}(\mathbf{x}^*)$ are computed using third-order accurate interpolations. In practice, for computational efficiency, they are calculated from the nodal representation of the velocity fields using third-order quadratic interpolations. Provided that all the eigenvalues of the discrete Laplacian in (34) are strictly negative and that the interpolation procedures do not introduce local extrema, the above scheme and thus the viscosity step is unconditionally strongly stable.

### *4.3.2. Pressure reconstruction*

The pressure reconstruction Eq. (20) is a direct consequence of this temporal discretization: inserting the Hodge decomposition (16) into the discretized Eq. (34) leads us to

$$\rho\left(\alpha\frac{\mathbf{u}^* - \mathbf{u}_d^n}{\Delta t_n} + \beta\frac{\mathbf{u}_d^n - \mathbf{u}_d^{n-1}}{\Delta t_{n-1}}\right) + \frac{\rho\alpha}{\Delta t_n}\nabla\Phi = \mu\Delta\mathbf{u}^* + \mathbf{f} - \nabla\tilde{p}. \tag{42}$$

Assuming that the momentum Eq. (1) is discretized in time using the scheme of Eq. (34), the velocity $\mathbf{u}^{n+1}$ satisfies

$$\rho\left(\alpha\frac{\mathbf{u}^{n+1} - \mathbf{u}_d^n}{\Delta t_n} + \beta\frac{\mathbf{u}_d^n - \mathbf{u}_d^{n-1}}{\Delta t_{n-1}}\right) = \mu\Delta\mathbf{u}^{n+1} + \mathbf{f} - \nabla p,$$

which, using the Hodge decomposition, can be expressed as

$$\rho\left(\alpha\frac{\mathbf{u}^{n+1} - \mathbf{u}_d^n}{\Delta t_n} + \beta\frac{\mathbf{u}_d^n - \mathbf{u}_d^{n-1}}{\Delta t_{n-1}}\right) = \mu\Delta\mathbf{u}^* - \mu\Delta\nabla\Phi + \mathbf{f} - \nabla p. \tag{43}$$

Substracting Eq. (43) from Eq. (42) and using the Hodge decomposition $\mathbf{u}^{n+1} = \mathbf{u}^* - \nabla\Phi$ yields

$$\nabla p = \nabla\left(\tilde{p} + \frac{\alpha\rho}{\Delta t_n}\Phi - \mu\nabla\cdot\mathbf{u}^*\right), \tag{44}$$

which implies that $p$ and the quantity between parentheses are equal up to a constant $c_0$. If Dirichlet boundary conditions are imposed on at least one point on the wall of the computational domain, then these two quantities must have the same value at that point since $\mu\nabla\cdot\mathbf{u}^*$ is a second-order correction, and therefore $c_0 = 0$. If Neumann boundary conditions are imposed everywhere on $\partial\Omega$, then the pressure, pressure guess and Hodge variable are rigorously only defined up to a constant, and we can use the convention that $c_0 = 0$. Therefore,

$$p = \tilde{p} + \frac{\alpha\rho}{\Delta t_n}\Phi - \mu\nabla\cdot\mathbf{u}^*. \tag{45}$$

### *4.3.3. Spatial discretization: Voronoi finite volume solver*

Without loss of generality, the spatial discretization of Eq. (34) subject to the continuity conditions (11)–(12) can be reformulated as a jump problem for each component $\xi$ of the velocity of the form

$$d(\mathbf{x})\xi - \mu\Delta\xi = h(\mathbf{x}) \qquad \forall\mathbf{x}\in\Omega/\Gamma, \tag{46}$$

$$[\![\xi]\!] = a(\mathbf{x}) \qquad \forall\mathbf{x}\in\Gamma, \tag{47}$$

$$\left[\!\!\left[\mu\frac{\partial\xi}{\partial n}\right]\!\!\right] = b(\mathbf{x}) \qquad \forall\mathbf{x}\in\Gamma, \tag{48}$$

where $d$ is a positive function. In the present case $d$ is uniform in each phase, but as we will see later on our approach is still valid if this last assumption is no longer satisfied. Before discretizing the above system, we first rewrite it so that the jump in the solution disappears from the formulation. To this effect, we introduce an extension $\tilde{a}$ of $a$ to the entire domain $\Omega$ such that

$$\tilde{a} = 0 \quad \forall\mathbf{x}\in\Omega^+, \qquad (d - \mu\Delta)\tilde{a} = 0 \quad \forall\mathbf{x}\in\Omega^-, \qquad \tilde{a} = a \quad \forall\mathbf{x}\in\Gamma. \tag{49}$$

Because $\tilde{a}$ is the solution of the above well-posed Poisson problem, we know that it exists and can be approximated numerically. We define a new unknown function $\psi = \xi - \tilde{a}$, solution of

$$d\psi - \mu\Delta\psi = h \qquad \forall\mathbf{x}\in\Omega/\Gamma \tag{50}$$

$$[\![\psi]\!] = 0 \qquad \forall\mathbf{x}\in\Gamma \tag{51}$$

$$\left[\!\!\left[\mu\frac{\partial\psi}{\partial n}\right]\!\!\right] = \tilde{b} \qquad \forall\mathbf{x}\in\Gamma, \tag{52}$$

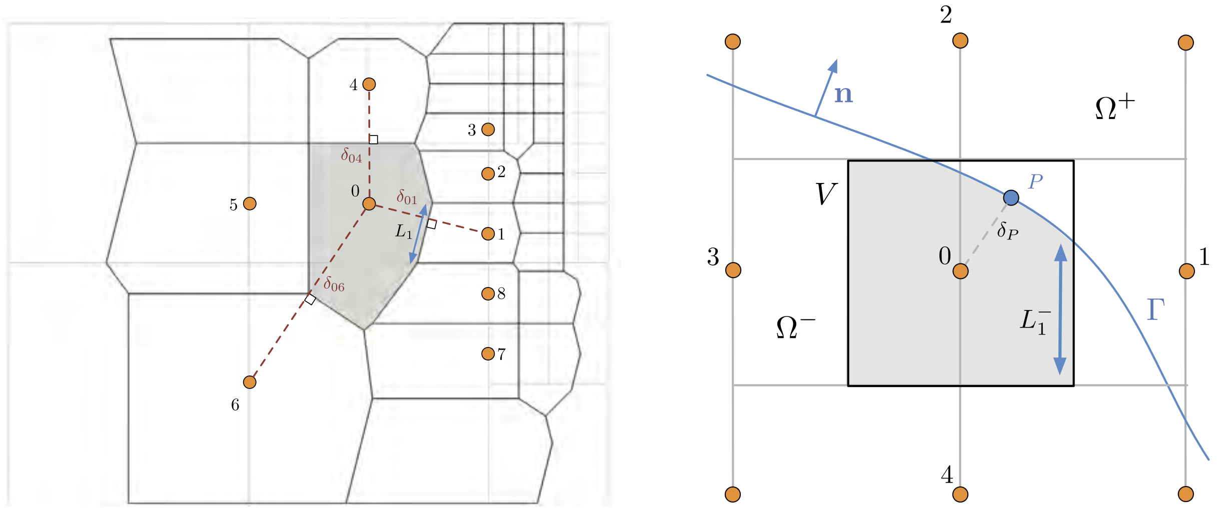


**Fig. 6.** Voronoi finite-volume method for the face-based jump solver in two dimensions. (Left) Far away from the interface, the mesh is arbitrary and non-graded. The control volumes are defined as the Voronoi cells. For example in this picture, the control volume of face 0 is colored in gray. (Right) Close to the interface where the mesh is uniform, the control volumes are squares of size equal to the finest resolution that are truncated by the interface. In this case, the control volume for the face 0 in the $\Omega^-$ subdomain is shaded in gray.

where $\tilde{b} = b - [\![\mu(\partial\tilde{a}/\partial n)]\!]$. In practice, the extension $\tilde{a}$ is constructed numerically by solving (49) at the nodes using the method presented in [44], which provides second-order accuracy for the solution and its gradient. We note that the second-order accuracy of the gradient $\partial\tilde{a}/\partial n$ is crucial here since it appears in the computation of the jump values $\tilde{b}$.

The discretization of the above system (50) combines two different approaches. Far away from the interface, in each phase, it follows exactly the face-based Voronoi finite volume method described in [14]. In this method, the control volumes $V_k$ are defined as the Voronoi cells of the faces of the grid (see Fig. 6), which are constructed locally using our geometric algorithm in two dimensions and the voro++ library [33] in three dimensions. Once the Voronoi partition has been built, we follow the standard finite volume procedure. Assuming that face $k$ is in $\Omega^-$, Eq. (50) is integrated over each control volume $V_k$, and after applying Gauss's theorem we obtain

$$\int_{V_k} d\psi^- - \int_{\partial V_k} \mu^- \frac{\partial \psi^-}{\partial n} = \int_{V_k} h. \tag{53}$$

Discretizing the volume integrals leads to

$$|V_k| d_k \psi^- - \sum_{i\in N(k)} \int_{L_i} \mu \frac{\partial \psi^-}{\partial n} = |V_k| h_0 k + O(\Delta \mathbf{x}^3), \tag{54}$$

where $\psi_k$ is the value of $\psi$ at the face $k$, $|V_k|$ is the volume of $V_k$, $N(k)$ is the set of Voronoi neighbors of $k$, the $L_i$ are the line segments between faces $k$ and $i$ (see Fig. 6) and $\partial V_k = \cup_{i\in N(k)} L_i$. The contour integrals in Eq. (54) are discretized as

$$-\int_{L_i} \mu \frac{\partial \psi^-}{\partial n} = |L_i| \mu \frac{\psi_k^- - \psi_i^-}{\delta_{ik}} + O(\Delta \mathbf{x}^3), \tag{55}$$

where $\delta_{ik}$ is the distance between faces $i$ and $k$. The third-order accuracy in Eq. (55) (and thus the second-order accuracy of the scheme) is a direct consequence of the properties of the line segment $L_i$, which is equidistant from $i$ and $k$ and orthogonal to the segment connecting these two points.

Close to the interface where the jump conditions have to be enforced and the mesh is uniform (Fig. 6), we follow the approach we developed in [25]. In this case, Eq. (50) must be discretized in both $\Omega^+$ and $\Omega^-$. Focusing on the discretization in $\Omega^-$, and adopting the notations of Fig. 6, we consider face 0 at the center of the fictitious cell $V$ and its corresponding control volume $V^- = V \cap \Omega^-$. Integrating (50) over the control volume $V^-$ and applying Gauss's theorem provides

$$\int_{V^-} d\psi^- - \int_{\partial V^-} \mu^- \frac{\partial \psi^-}{\partial n} = \int_{V^-} h, \tag{56}$$

which, after discretizing the volume integrals, becomes

**Table 1**
Convergence of the face-based jump Poisson solver. The test solution is $u^-(x,y)=\cos(x)\sin(y)$, $u^+(x,y)=\sin(x)\cos(y)$, the computational domain is $\Omega=[-2,2]^2$, and the interface is represented by the level-set function $\Phi(x,y)=0.58-\sqrt{x^2+y^2}$. We observe second-order convergence.

| Quadtree level (min/max) | $\mu^-/\mu^+=10^{10}$ | | $\mu^-/\mu^+=10^{-7}$ | | $\mu^- - \mu^+=10^{-8}$ | |
|---|---|---|---|---|---|---|
| | $L^\infty$ error | order | $L^\infty$ error | order | $L^\infty$ error | order |
| 1/5 | $2.002\times10^{-1}$ | - | $5.575\times10^{-2}$ | - | $1.139\times10^{-1}$ | - |
| 2/6 | $7.046\times10^{-2}$ | 1.507 | $1.915\times10^{-2}$ | 1.588 | $4.178\times10^{-2}$ | 1.448 |
| 3/7 | $2.202\times10^{-2}$ | 1.677 | $5.848\times10^{-3}$ | 1.712 | $1.225\times10^{-2}$ | 1.769 |
| 4/8 | $6.263\times10^{-3}$ | 1.814 | $1.690\times10^{-3}$ | 1.791 | $3.364\times10^{-3}$ | 1.865 |
| 5/9 | $1.750\times10^{-3}$ | 1.839 | $4.826\times10^{-4}$ | 1.808 | $8.945\times10^{-4}$ | 1.911 |
| 6/10 | $4.436\times10^{-4}$ | 1.981 | $1.420\times10^{-4}$ | 1.765 | $2.364\times10^{-4}$ | 1.920 |
| 7/11 | $1.144\times10^{-5}$ | 1.954 | $3.801\times10^{-5}$ | 1.901 | $5.858\times10^{-5}$ | 1.947 |

$$|V^-|d_0\psi^- - \int_{\partial V^-}\mu^-\frac{\partial\psi^-}{\partial n} = |V^-|h_0 + O(\Delta\mathbf{x}^3). \tag{57}$$

The surface integrals in Eq. (57) can be decomposed into two parts:

$$\int_{\partial V^-}\mu^-\frac{\partial\psi_0^-}{\partial n} = \int_{\partial V\cap\Omega^-}\mu^-\frac{\partial\psi_0^-}{\partial n} + \int_{\Gamma\cap V}\mu^-\frac{\partial\psi_0^-}{\partial n}. \tag{58}$$

The first integral in Eq. (58) can be decomposed as a sum of integrals over every face $F_i$ of $V$, which we discretize independently. For example, the integral on the right face $F_1$, between faces 0 and 1, is approximated by

$$-\int_{F_1}\mu^-\frac{\partial\psi_0^-}{\partial n} = L_1^-\frac{\psi_0^- - \psi_1^-}{\Delta\mathbf{x}} + O(\Delta\mathbf{x}^3), \tag{59}$$

where $L_1^-$ is the length of face $F_1$ in $\Omega^-$ (see Fig. 6). Following [25], the second integral in Eq. (58) can be approximated by

$$-\int_{\Gamma_V}\mu^-\frac{\partial\psi_0^-}{\partial n} = J(\psi_0^- - \psi_0^+) - \frac{|\Gamma_V|\mu^-}{(\mu^- - \mu^+)}\tilde{b}_P + O(\Delta\mathbf{x}^2), \tag{60}$$

where $\Gamma_V=\Gamma\cap V$, $J=-|\Gamma_V|\mu^+\mu^-/\delta(\mu^\mp-\mu^\pm)$, $\delta$ is the signed distance between the face center and the interface, $P$ is the orthogonal projection of that face center on the interface and $\tilde{b}_P$ is the value of $\tilde{b}$ at that point. To prevent the magnitude of $J$ from being arbitrarily large while preserving the accuracy of the scheme, the distance $\delta$ is truncated to $\epsilon=O(\Delta\mathbf{x}^2)$. Similarly when discretizing in the $\Omega^+$ domains, the flux across the interface becomes

$$-\int_{\Gamma_V}\mu^+\frac{\partial\psi_0^+}{\partial n} = J(\psi_0^+ - \psi_0^-) + \frac{|\Gamma_V|\mu^+}{(\mu^- - \mu^+)}\tilde{b}_P + O(\Delta\mathbf{x}^2). \tag{61}$$

As illustrated in Table 1, the resulting numerical solution is second-order accurate in $L^\infty$-norm, even in extreme cases where the viscosities are seven or eight orders of magnitude apart from each other. We also remark that the reported orders of accuracy in this paper are significantly better than those reported in our previous study [25]. We believe that this improvement is a consequence of the homogenization procedure that is employed here and was not in our previous work.

Even though the scheme is compact and the resulting linear system is symmetric, it is no longer guaranteed to be positive definite, which can have dramatic consequences for the stability of the algorithm. In fact, the additional diagonal term $J$ introduced by formula (60) can be either positive or negative. A simple way (implemented here) of ensuring that the system remains positive definite and therefore of ensuring unconditional stability for the viscosity step is to arbitrarily impose that $J$ be positive on all the faces close to the interface, by replacing $J$ by its absolute value in formulae (61)–(60). By doing so, the order of accuracy of the viscosity step drops to first-order. Since the projection step is already first-order accurate, it does not reduce the overall order of the method.

Once the intermediate velocity $\mathbf{u}^{*\pm}$ has been calculated, it is extended to the faces in the $\Omega^\mp$ subdomain domain that have a control volume $V^\pm$ that does not intersect with $\Omega^\pm$ and that are located within a distance $2\Delta\mathbf{x}$ of the interface. These extended velocity fields are used to compute the term $\nabla\cdot\mathbf{u}^*$ during the projection step.

### *4.4. Pressure guess and projection steps*

For the pressure guess and the projection steps, the two Poisson problems (7) and (13) with jump boundary conditions are discretized using a cell-based finite volume solver. Close to the interface where the mesh is uniform, the discretization procedure is identical to the one described in the above section. Far away from the interface, our approach follows the one we used in [14], which was first developed by Losasso et al. [21] and is presented here in the context of the projection step.

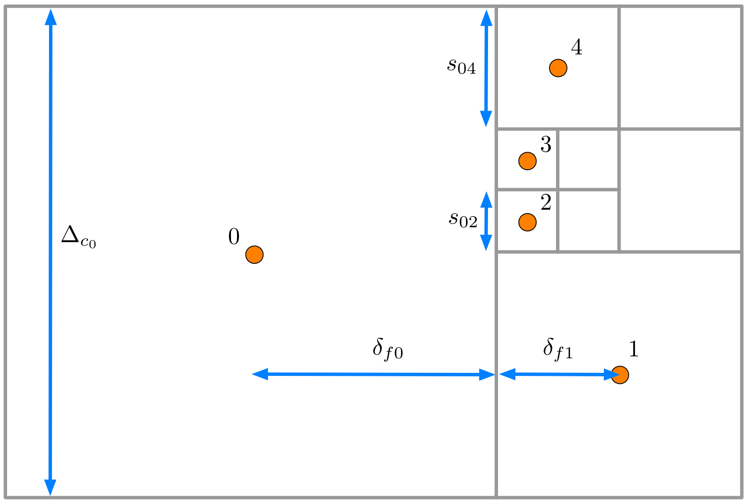


**Fig. 7.** Cell-based finite volume solver: general configuration far away from the interface.

**Table 2**
Convergence of the cell-based jump Poisson solver. The test solution is $u^-(x,y) = \cos(x)\sin(y)$, $u^+(x,y) = \sin(x)\cos(y)$, the computational domain is $\Omega = [-2,2]^2$ and the interface is represented by the level-set function $\Phi(x,y) = 0.58 - \sqrt{x^2+y^2}$. We observe second-order convergence in $L^\infty$ norm, even when the two viscosities differ dramatically.

| Quadtree level (min/max) | $\mu^-/\mu^+ = 10^8$ | | $\mu^-/\mu^+ = 10^{-7}$ | | $\mu^- - \mu^+ = 10^{-6}$ | |
|---|---|---|---|---|---|---|
| | $L^\infty$ error | order | $L^\infty$ error | order | $L^\infty$ error | order |
| 1/5 | $1.057 \times 10^{-1}$ | - | $2.916 \times 10^{-2}$ | - | $6.943 \times 10^{-2}$ | - |
| 2/6 | $4.319 \times 10^{-2}$ | 1.315 | $7.481 \times 10^{-3}$ | 1.963 | $2.727 \times 10^{-2}$ | 1.347 |
| 3/7 | $1.394 \times 10^{-2}$ | 1.315 | $1.940 \times 10^{-3}$ | 1.947 | $8.910 \times 10^{-3}$ | 1.614 |
| 4/8 | $4.589 \times 10^{-3}$ | 1.604 | $5.296 \times 10^{-4}$ | 1.873 | $2.817 \times 10^{-4}$ | 1661 |
| 5/9 | $1.256 \times 10^{-3}$ | 1.869 | $1.987 \times 10^{-4}$ | 1.413 | $8.070 \times 10^{-4}$ | 1.803 |
| 6/10 | $3.189 \times 10^{-4}$ | 1.977 | $3.991 \times 10^{-5}$ | 2.316 | $1.964 \times 10^{-4}$ | 2.038 |

### *4.4.1. Cell-based finite volume solver*

For each component of the Hodge variable $\Phi_0$, we define its associated control volume as the Quadtree cell $c_0$ it is at the center of. As we saw in the previous section, the finite volume method requires approximating the fluxes of the unknown quantity on the contour of the control volumes, which in this case means the normal gradient of $\Phi$ on each of the four faces of $c_0$. Each of these gradients is approximated using the second-order approximation constructed in [21], and, for example, takes the following form on the right face $f$ of $c_0$:

$$\nabla\Phi|_f = \sum_{i \in N_R(c_0)} \frac{s_{0i}}{\Delta_{c_0}} \left( \frac{\Phi_0 - \Phi_i}{\Delta_f} \right) + O(\Delta \mathbf{x}^2), \tag{62}$$

where $N_R(c_0)$ is the set of indices of the cells connected to cell $c_0$ through its right face. In the case depicted in Fig. 7 where the face $f$ belongs to a big cell connected to smaller cells, $N_R(c_0)$ is simply the set of adjacent cells to $c_0$ across face $f$, which in the current case is $N_R(c_0) = 1,2,3,4$. In the inverse case, where a small cell $c_S$ is adjacent on its right to a bigger cell $c_B$, $N_R(c_S)$ is defined as $(N_L(c_B) \cup B) - S$. For example, in Fig. 7 the small cell $c_3$ is adjacent to its left to cell $c_0$, therefore $N_L(c_3) = (N_R(c_0) \cup 0) - 3 = 0,1,2,4$. The coefficient $s_{0i}$ is the length of the segment connecting cell $c_0$ and $c_i$, $\Delta_{c_0}$ is the size of cell $c_0$, and the average distance $\Delta_f$ is defined as:

$$\Delta_f = \sum_{i \in N_R(c_0)} \frac{s_{0i}}{\Delta_{c_0}} (\delta_{f0} - \delta_{fi}), \tag{63}$$

where $\delta_{fi}$ is the signed distance from cell $c_i$ to face $f$. From these definitions, we can see that this discretization leads to a symmetric linear system. Also, as previously mentioned, the above formula is second-order accurate, and we refer the interested reader to [14,21] for a detailed proof.

Convergence results are provided in Table 2 and confirm the second-order accuracy of our approach in $L^\infty$-norm, even for viscosity coefficients separated by several orders of magnitude. As pointed out in our previous work, one order of accuracy may be lost at T-junctions when projecting the velocity field $\mathbf{u}^*$ by taking the gradient of $\Phi$, even though $\Phi$ is known with second-order accuracy. Therefore, the resulting velocity field $\mathbf{u}^{n+1}$ is expected to be only first-order accurate.

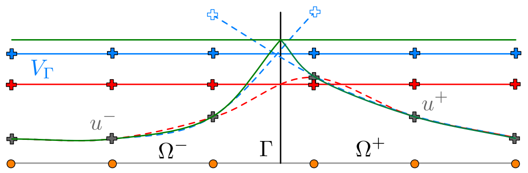


**Fig. 8.** Interfacial velocity reconstruction procedure: first we extend the nodal velocities $\mathbf{u}^+$ to $\Omega^-$ and $\mathbf{u}^-$ to $\Omega^+$ using a third-order extension (dash blue lines). From these extended quantities, we construct two constant extrapolations from the interface in the normal direction, which we average to obtain the interfacial velocity $\mathbf{V}_\Gamma$ (solid blue line). The exact fluid and interfacial velocities are represented in green. Defining the interfacial velocity as the combination of $\mathbf{u}^+$ and $\mathbf{u}^-$ (dash red line) or even as the constant extrapolation of this quantity (red line), smears out the discontinuity in the gradient of the velocity and therefore leads to a very inaccurate representation of the interfacial velocity.

### *4.4.2. Stability of the projection step*

A core property of the projection method presented here is its orthogonality:

$$||\mathbf{u}^{n+1}||^2_{L_F} = ||\mathbf{u}^* - G\Phi||^2_{L_F} = ||\mathbf{u}^*||^2_{L_F} - ||G\Phi||^2_{L_F}, \tag{64}$$

where $G$ is the discrete gradient appearing in the discretization of the projection equation (13), and $L_F$ is a diagonal operator defining a norm $||\cdot||_{L_F}$ on the space of the faces of the mesh. This property is of the greatest importance, since it implies that

$$||\mathbf{u}^{n+1}||^2_{L_F} \le ||\mathbf{u}^*||^2_{L_F}, \tag{65}$$

which ensures that the projection step is stable in the $||\cdot||_{L_F}$ norm, i.e., that it is stable in any norm since we are working in finite dimension where all norms are equivalent.

The orthogonality property (64) results from the homogeneity of the jump conditions (14)–(15), and also of the definition of the discrete divergence $D$, which we construct to be proportional to the transconjugate of the gradient operator:

$$L_C D = -(GL_F)^T, \tag{66}$$

where similarly to $L_F$, $L_C$ is a diagonal operator on the space of the cell centers of the mesh. Specifically, the discrete divergence of $\mathbf{u}^*$ at cell $c$ is computed as

$$D\mathbf{u}^*|_c = -\frac{1}{\Delta_c}\sum_{f_i\in N_f(c)} \frac{s_i u_i^*}{A_{f_i}}\frac{\delta_{f_i c}}{|\delta_{f_i c}|} \qquad \text{where} \qquad A_f = \frac{1}{2}\sum_{c_i\in N_c(f)} s_{c_i}. \tag{67}$$

$N_f(c)$ is the set of faces in contact with cell $c$. We also define $N_c(f)$ as the set of the cells in contact with the largest direct cell neighbor of face $f$. Property (66) is easily shown by reformulating (62) using the new notation

$$G\Phi|f = \frac{1}{A_f}\sum_{c_i\in N_c(f)} \frac{s_{c_i}\Phi_i}{\Delta f}\frac{\delta_{f_i c}}{|\delta_{f_i c}|}, \tag{68}$$

and then identifying the two diagonal operators $L_F$, $L_C$ as:

$$(L_C)_{cc} = s_c\Delta_c \qquad \text{and} \qquad (L_F)_{ff} = s_f\Delta_f. \tag{69}$$

The stability of the projection, under the assumption that the jump conditions are homogeneous and that condition (66) is satisfied, can easily be proven by expanding the squared norm $||\mathbf{u}^* - G\Phi||^2_{L_F}$. We refer the interested reader to [14] for the detailed proof in the single-phase flow case.

## *4.5. Interfacial velocity*

At each new time step, the evolution of the interface starts with the reconstruction of the interfacial velocity $\mathbf{V}_\Gamma$ from the newly calculated fluid velocity $\mathbf{u}^{n+1}$. This process, illustrated in Fig. 8, consists of the following steps. First, both velocity fields $\mathbf{u}^\pm$ are interpolated from the faces to the nodes of the mesh using a third-order least-squares method. These nodal representations $\mathbf{u}^\pm_{\text{nodes}}$ are then extended to the nodes across the interface (i.e. those in $\Omega^\mp$) using a third-order extension procedure described in [24,1]. From these two extensions, two constant extrapolations from the interface to the entire domain in the normal direction are constructed. Any of these two extrapolations can be used to define $\mathbf{V}_\Gamma$. For better accuracy, and since there is no *a priori* reason for selecting one or the other, we define the interfacial velocity $\mathbf{V}_\Gamma$ as their average.

Once the interfacial velocity is computed, the interface is evolved by solving the level-set advection equation (30) using a second-order Semi-Lagrangian (SL) method and reinitialized using a TVD-RK2 method.

### 4.6. Definition and calculation of the corrective terms

As discussed earlier, the role of the corrective jump velocity $\mathbf{X}_k$ and stress $\mathbf{\Sigma}_k$ is to ensure that the continuity of the velocity (11) and the interfacial stress balance (12) are strictly enforced. Naively, the corrective jump velocity $\mathbf{X}_{k+1}$ could be defined as the jump of the Hodge variable at the previous functional iteration:

$$\mathbf{X}_{k+1} = [\![\nabla \Phi_k]\!], \tag{70}$$

so that, if the functional iterations converge, we have $[\![\mathbf{u}^*]\!] = [\![\nabla \Phi]\!]$ upon convergence, which directly implies $[\![\mathbf{u}^{n+1}]\!] = \mathbf{0}$. From our experience, this can often fail unpredictably with disastrous consequences for the overall simulation. Motivated by these observations, we instead construct the corrective terms such that the functional iterations are guaranteed to converge. Specifically, we use the following definition for the corrective velocity and stress:

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \omega\, [\![\mathbf{u}_k^{n+1}]\!], \tag{71}$$

$$\mathbf{\Sigma}_{k+1} = \mathbf{\Sigma}_k - \lambda \left( [\![\mu (\nabla \mathbf{u}_k^{n+1})^T \cdot \mathbf{n}]\!] + \mathbf{\Sigma}_k \right), \tag{72}$$

where $\omega, \lambda$ are strictly positive coefficients.

**Theorem 1.** *For any initial corrective terms $\mathbf{X}_0$, $\mathbf{\Sigma}_0$ and for any $(\omega, \lambda) \in ]0, 1[^2$, in the limit $\Delta x, \Delta t \to 0$, we have*

$$\mathbf{X}_k \to \mathbf{X}, \qquad \mathbf{\Sigma}_k \to \mathbf{\Sigma}, \qquad [\![\mathbf{u}_k^{n+1}]\!] \to \mathbf{0}, \qquad [\![\boldsymbol{\sigma}^{n+1} \cdot \mathbf{n}]\!] \to \mathbf{g} - \mathbf{n}(\mathbf{g} \cdot \mathbf{n}) \qquad \text{as } k \to \infty. \tag{73}$$

**Proof of Theorem 1.** First, given two functions $\mathbf{Y}^+, \mathbf{Y}^-$ defined on the faces $\Omega_F$ of the mesh, we define $\mathbf{Y} = (\mathbf{Y}^+, \mathbf{Y}^-) \in \Omega_F^2$ and their jump as $[\![\mathbf{Y}]\!] = \mathbf{Y}^+ - \mathbf{Y}^-$. We also define the function $\mathcal{F}(\mathbf{X}) : \Omega_F \to \Omega_F^2$, which for any arbitrary jump velocity $\mathbf{X} \in \Omega_F$ returns the velocity field $\mathbf{u}^{n+1} \in \Omega_F^2$ computed through steps 2a (viscosity) and 2b (projection) with the jump condition $[\![\mathbf{u}^*]\!]\big|_\Gamma = \mathbf{X}$. In order to define $\mathcal{F}(\mathbf{X})$ explicitly we need to interpret the viscosity and projection steps as operators. First, we recognize that the viscosity step is an affine transformation of the prescribed jump velocity $\mathbf{X}$, and can therefore be rewritten in terms of a linear operator $\mathcal{V} : \Omega_F \to \Omega_F^2$ and a right hand side vector $\mathbf{R} \in \Omega_F^2$ as

$$\mathbf{u}^* = \mathcal{V}\mathbf{X} + \mathbf{R}. \tag{74}$$

As we discussed in the previous section, the jump conditions for the projection step are homogeneous and therefore $D\mathbf{u}^* = DG\Phi = -\mathcal{A}\Phi$, where $\mathcal{A}$ is symmetric positive definite. Thus, we can write

$$\Phi = -\mathcal{A}^{-1} D\mathbf{u}^* = -\mathcal{A}^{-1} D\big(\mathcal{V}\mathbf{X} + \mathbf{R}\big). \tag{75}$$

With these new notations, the Hodge decomposition $\mathbf{u}^{n+1} = \mathbf{u}^* - \nabla\Phi$ becomes

$$\mathcal{F}(\mathbf{X}) = \big(I + G\mathcal{A}^{-1}D\big)\big(\mathcal{V}\mathbf{X} + \mathbf{R}\big). \tag{76}$$

The jump of the solution $\mathbf{u}^{n+1}$ is expressed as

$$[\![\mathcal{F}(\mathbf{X})]\!] = [\![\mathcal{V}\mathbf{X}]\!] + [\![G\mathcal{A}^{-1}D\big(\mathcal{V}\mathbf{X} + \mathbf{R}\big)]\!]. \tag{77}$$

By definition of $\mathcal{V}$, the first term in the right hand side is equal to $\mathbf{X}$ up to the numerical errors ($O(\Delta x)$). The second one, the jump of the non-divergence part of the auxiliary velocity, is $O(\Delta t)$. We obtain

$$[\![\mathcal{F}(\mathbf{X})]\!] = \mathbf{X} + O(\Delta x + \Delta t). \tag{78}$$

We define two functions $\mathcal{G}_N(\mathbf{\Sigma}), \mathcal{G}_T(\mathbf{\Sigma}) : \Omega_C \to \Omega_C^2$, which for an arbitrary corrective stress $\mathbf{\Sigma} \in \Omega_C$ return the normal flux and normal transposed flux of the solution $\mathbf{u}^{n+1}$:

$$\mathcal{G}_N(\mathbf{\Sigma}) = [\![\mu \nabla \mathbf{u}^{n+1} \cdot \mathbf{n}]\!], \tag{79}$$

$$\mathcal{G}_T(\mathbf{\Sigma}) = [\![\mu \big(\nabla \mathbf{u}^{n+1}\big)^T \cdot \mathbf{n}]\!], \tag{80}$$

where the jump operator is now defined on the space of all the cells $\Omega_C$ the same way it was on $\Omega_F$ previously. ~~Under the assumption that the second order corrections $\nabla\nabla\Phi$ (terms involving the gradient $G\Phi$ above) can be neglected, and~~ Recognizing that the viscosity step is an affine transformation and noting that $\nabla\nabla\Phi = O(\Delta t)$, the above definitions can be rewritten as

$$\mathcal{G}_N(\mathbf{\Sigma}) = [\![\mathcal{N}\big(\mathcal{W}\mathbf{\Sigma} + \mathbf{Q}\big)]\!] + O(\Delta t), \tag{81}$$

$$\mathcal{G}_T(\mathbf{\Sigma}) = [\![\mathcal{T}\big(\mathcal{W}\mathbf{\Sigma} + \mathbf{Q}\big)]\!] + O(\Delta t), \tag{82}$$

where $\mathcal{N}, \mathcal{T} : \Omega_C^2 \to \Omega_C$ are the normal flux and normal transposed flux linear operators, $\mathcal{W} : \Omega_C \to \Omega_C^2$ is bijective, and $\mathbf{Q} \in \Omega^2$.

Finally, assuming that $\omega$ and $\lambda$ are constant and using Eq. (78), the correction sequences (71) and (72) can be rewritten with these notations as

$$\mathbf{X}_{k+1} = \quad \big(1-\omega\big)\mathbf{X}_k + O\left(\Delta x + \Delta t\right) \quad = \mathcal{X}\mathbf{X}_k + O\left(\Delta x + \Delta t\right), \tag{83}$$

$$\mathbf{\Sigma}_{k+1} = \big((1-\lambda)\mathbf{\Sigma}_k - \lambda \mathcal{G}_T(\mathbf{\Sigma}_k)\big) + O\left(\Delta t\right) = \mathcal{S}\mathbf{\Sigma}_k + O\left(\Delta t\right), \tag{84}$$

where $\mathcal{X}$ and $\mathcal{S}$ are two linear operators. In the limit $\Delta x, \Delta t \to 0$, convergence of the functional iterations is guaranteed by the fact that we are dealing with a fixed point problem, which must converge to a unique solution independent of the initial conditions so long as the absolute values of the real parts $\Re\left(\mathcal{X}\right)$ and $\Re\left(\mathcal{S}\right)$ of the eigenvalues of operators $\mathcal{X}$ and $\mathcal{S}$ are less than 1. Since $\mathcal{X} = (1-\omega)\mathcal{I}$, the parameter $\omega$ has to be in $]0, 1[$ for the method to converge. To study the spectrum of $\mathcal{S}$, we first recognize that the derivative operators $\mathcal{N}$ and $\mathcal{T}$ are identical up to a permutation: we can express them as

$$\mathcal{N} = ng, \tag{85}$$

$$\mathcal{T} = npg, \tag{86}$$

where $g : \Omega_F \to \Omega_C^4$ is the cell-based gradient operator, $p : \Omega_C^4 \to \Omega_C^4$ is a permutation matrix permuting the partial derivatives at each cell, and the operator $n : \Omega_C^4 \to \Omega_C$ returns the flux in the normal direction. Because of the structures of $n$ and $p$, we have that $\|\mathcal{N}\| = \|\mathcal{T}\|$ and therefore $\big\|[\![\mathcal{N}\mathcal{W}]\!]\big\| = \big\|[\![\mathcal{T}\mathcal{W}]\!]\big\|$ since $\mathcal{W}$ is bijective. Neglecting numerical errors, $[\![\mathcal{N}\mathcal{W}]\!]$ is nothing else than the identity operator (since, for any $\mathbf{\Sigma}$, $[\![\mathcal{N}\mathcal{W}\mathbf{\Sigma}]\!]$ is the jump of the normal flux of the solution with prescribed normal flux jump $\mathbf{\Sigma}$), and therefore

$$\big\|[\![\mathcal{N}\mathcal{W}]\!]\big\| = \big\|[\![\mathcal{T}\mathcal{W}]\!]\big\| = 1 \qquad \Rightarrow \qquad \Re\left(\mathcal{G}_T\right) \le 1. \tag{87}$$

Additionally, we know that all the eigenvalues of $\mathcal{T}\mathcal{W}$ must be non-zero due to the relationship between $\mathcal{N}$ and $\mathcal{T}$. Therefore $0 < \Re(\mathcal{G}_T) \le 1$ and

$$1 - 2\lambda \le \Re(\mathcal{S}) < 1 - \lambda, \tag{88}$$

which means that $\lambda$ has to be in $]0, 1[$ for the algorithm to converge.

To conclude we note that, by construction, if $\mathbf{X}_k$ and $\mathbf{\Sigma}_k$ converge as $k \to \infty$, then automatically from the definitions (71)–(72) we also have

$$[\![\mathbf{u}_k^{n+1}]\!]\big|_\Gamma \to \mathbf{0}, \qquad [\![\mu\left(\nabla\mathbf{u}^{n+1}\right)^T \cdot \mathbf{n}]\!]\big|_\Gamma \to -\mathbf{\Sigma}. \tag{89}$$

The last limit directly implies that

$$\lim_{k\to\infty} [\![\boldsymbol{\sigma} \cdot \mathbf{n}]\!]\big|_\Gamma = \mathbf{g} - \mathbf{n}\left(\mathbf{g} \cdot \mathbf{n}\right). \quad \square \tag{90}$$

One interesting remark is that, under the naive definition of the corrective velocity (70), the convergence of the functional iterations is not guaranteed and the method can reasonably be expected to diverge. Specifically, using the same notations we used to prove Theorem 1, definition (70) can be reformulated as:

$$\mathbf{X}_{k+1} = -[\![G\mathcal{A}^{-1}D(\mathcal{V}\mathbf{X}_k + \mathbf{R})]\!]. \tag{91}$$

Following the same reasoning, the above construction is convergent if $\Re(G\mathcal{A}^{-1}D\mathcal{V}) \in ]0, 1[$. The nullspace of $G\mathcal{A}^{-1}D\mathcal{V}$ is potentially non-zero because it contains all the vectors $\mathbf{X}$ such that $\mathcal{V}\mathbf{X}$ is divergence-free. In this case, the limit $\mathbf{X}$ is no longer uniquely defined. $\Re\big(G\mathcal{A}^{-1}D\mathcal{V}\big)$ may also contain the critical value 1, which would prevent the method from converging. In fact any vector $\mathbf{X}$ such that $\mathcal{V}\mathbf{X}$ is in the space orthogonal to the divergence-free space is an eigenvector of $G\mathcal{A}^{-1}D\mathcal{V}$ with eigenvalue 1.

We also note that Theorem 1 still holds if the coefficients $\omega_k$, $\lambda_k$ are no longer constant but bounded below by $\omega_{\min} > 0$ and $\lambda_{\min} > 0$ and less than 1. The proof is identical and therefore not reproduced.

In practice we take the arbitrary values $\omega = \lambda = 0.95$. At each new time step, the initial corrections $\mathbf{X}_0$ and $\mathbf{\Sigma}_0$ are defined as the final corrections at the previous time step. Our numerical experiments suggest that advecting these corrections with the interfacial velocity (in the same manner as the level-set function) produces slightly better initial guesses.

The velocity correction is constructed from the constantly extrapolated nodal velocities used in the interface velocity reconstruction, which are conveniently already extrapolated. The stress correction is updated by differentiating the nodal velocity fields. Both corrections are also systematically constantly extrapolated from the interface in the normal direction.

### 4.7. Improved time step restriction

In [4], Brackbill first proposed the following stability condition for the time step $\Delta t$:

$$\Delta t < \sqrt{\frac{\left(\rho^+ + \rho^-\right)\Delta x^3}{4\pi\gamma}} = \Delta t_B, \tag{92}$$

where $\Delta x$ is the minimum spatial resolution. This restriction comes from the intuition that in order for the method to be stable, the propagation of capillary waves must be correctly resolved and that therefore the numerical propagation speed must be less than the theoretical one. In the present context, it implies that the following inequality must be satisfied for any wavenumber:

$$\frac{c_k \Delta t}{\Delta x} < \frac{1}{2}, \tag{93}$$

where $c_k$ is the phase velocity associated to the wavenumber $k$. Since $c_k = \sqrt{\gamma k/(\rho^+ + \rho^-)}$ is maximum for the highest observable wavenumber $\pi/\Delta x$, condition (92) is obtained. This condition has been widely found to ensure stability. Yet it can be extremely prohibitive for simulations with high spatial resolutions since $\Delta t \propto \Delta x^{\frac{3}{2}}$, or when inertial effects become comparable to the effects of capillarity and viscosity, for example as $\rho^\pm \to 0$.

More recently, Galusinski and Vigneaux [12] presented a numerical method for two-phase flows in the case where viscosities and densities are identical in each domain, which they proved to be stable under the following time step restriction:

$$\Delta t < \min\left( c_0 \frac{\Delta x}{||\mathbf{u}^\pm||_{L^\infty}} \quad , \quad \frac{c_1 \mu \Delta x}{\gamma} + \sqrt{\left(\frac{c_1 \mu \Delta x}{\gamma}\right)^2 + c_2 \frac{\rho \Delta x^3}{2\pi\gamma}} \right) = \Delta t_{GV}, \tag{94}$$

where $c_0$, $c_1$ and $c_2$ are constant coefficients independent of fluid properties. The proof they present is only valid for small enough Reynolds numbers that the flow can be considered laminar, assumes that the surface tension term is treated explicitly (which is generally the case) and that the interface is advected using a stable explicit scheme. They do not provide an analytical expression for the three constant coefficients but numerically estimate them to be close to one. The first term between parentheses in Eq. (94) enforces that no fluid element or any point on the interface is displaced by more than $c_0 \Delta x$ over the course of a time step, which is in practice a very reasonable assumption that is usually made. The second term is always greater than $\Delta t_B$ (assuming that indeed $c_2$ is equal to one), particularly in the problematic parameter regimes evoked previously. Most importantly, it scales linearly with $\Delta x$, which has a tremendous impact for highly spatially refined simulations.

Numerically, we found our approach to be stable under the following restriction:

$$\Delta t < \min\left( c_0 \frac{\Delta x}{||\mathbf{u}^\pm||_{L^\infty}} \quad , \quad \frac{c_1 \mu_{\min} \Delta x}{\gamma} + \sqrt{\left(\frac{c_1 \mu_{\min} \Delta x}{\gamma}\right)^2 + c_2 \frac{\left(\rho^+ + \rho^-\right)\Delta x^3}{4\pi\gamma}} \right) = \Delta t_{\max} \tag{95}$$

where $\mu_{\min} = \min(\mu^+, \mu^-)$, for $c_0 \le 1$ and $c_1, c_2 < 1$. It is a very natural extension of the restriction proposed by Galusinski and Vigneaux to systems where the viscosities and densities are not identical in each phase. In agreement with what has been commonly reported, we also observed that our simulation engine is only stable under the original time step restriction (92) if the pressure guess step is omitted by setting $\tilde{p} = 0$ and imposing the pressure jump condition (8) directly on the Hodge variable.

In practice, the time step restriction is recomputed at each time step as the right hand side of (95) taking $c_0 = 1$ and $c_1 = c_2 = 0.95$. We note that since the convective terms are treated with a semi-Lagrangian method that is unconditionally stable, we could in principle use a larger value for $c_0$. It would however significantly reduce the accuracy of the advection procedure for both the fluid velocity and the level-set function.

## 5. Numerical examples and validations

In this section, we present numerical evidence that our method reproduces the dynamics of two-phase flows accurately and efficiently. We first consider analytical solutions and canonical two-phase flow test cases to verify the accuracy and stability of our approach. In a second time, we consider physically relevant applications in two and three spatial dimensions to further validate it and illustrate its full capabilities.

All the presented simulations were run on desktop computers. Most of the code is parallelized with OpenMP. The linear systems are inverted using a Bi-Conjugate Gradient Stabilized solver with an algebraic multigrid preconditioner (HYPRE) from the PETSC library [2].

**Table 3**
Convergence of the overall method for the analytical solution described in Section 5.1. The error in the interface location is quantified by the maximum error in the level-set function within a band of thickness $2\Delta x$ around the interface.

| Quadtree level (min/max) | Velocity | | Interface location | |
|---|---|---|---|---|
| | $L^\infty$ error | order | $L^\infty$ error | order |
| 0/4 | $6.880 \times 10^{-1}$ | - | $7.830 \times 10^{-1}$ | - |
| 1/5 | $3.457 \times 10^{-1}$ | 0.992 | $8.539 \times 10^{-1}$ | -0.125 |
| 2/6 | $1.293 \times 10^{-1}$ | 1.418 | $8.528 \times 10^{-1}$ | 0.002 |
| 3/7 | $4.407 \times 10^{-2}$ | 1.553 | $5.698 \times 10^{-2}$ | 3.903 |
| 4/8 | $1.998 \times 10^{-2}$ | 1.141 | $3.231 \times 10^{-2}$ | 0.818 |
| 5/9 | $4.180 \times 10^{-3}$ | 2.256 | $1.229 \times 10^{-2}$ | 1.394 |
| 6/10 | $1.752 \times 10^{-3}$ | 1.254 | $6.948 \times 10^{-3}$ | 0.823 |

## 5.1. Analytical solution

We first validate our method by considering the following exact analytical solution

$$u^\pm = u_e(x,y) = \sin(x)\cos(y)\sin(t), \tag{96}$$

$$v^\pm = v_e(x,y) = -\cos(x)\sin(y)\sin(t), \tag{97}$$

$$p^\pm = p_e(x,y) = 0, \tag{98}$$

$$\phi_e(x,y) = 0.1 - \sin(x)\sin(y), \tag{99}$$

for the two-phase incompressible Navier-Stokes equations with an external body force $\mathbf{f}^\pm(x,y,t)$ and interfacial stress $\mathbf{g}$ given by:

$$\mathbf{f}^\pm(x,y,t) = \rho^\pm\left(\frac{\partial \mathbf{u}_e}{\partial t} + \mathbf{u}_e\cdot\nabla\mathbf{u}_e\right) - \mu^\pm\Delta\mathbf{u}_e \tag{100}$$

$$\mathbf{g}(x,y,t) = -\gamma\nabla\cdot\mathbf{N}(x,y) + [\![\mu\nabla\mathbf{u}_e\cdot\mathbf{N}(x,y)]\!], \tag{101}$$

where $\mathbf{N}(x,y) = \nabla\Phi_e/|\nabla\Phi_e|$ is the analytical normal of the exact level-set $\Phi_e$. This solution is constructed from the analytical solution for the monophasic case presented in [14,23]. The results presented here are obtained for the following parameters

$$\mu^+ = \rho^+ = 1, \qquad \mu^- = \rho^- = 0.1, \qquad \gamma = 0.1, \qquad \Omega = \left[-\frac{\pi}{3}, \frac{4\pi}{3}\right]^2, \qquad t_{final} = \pi. \tag{102}$$

No-slip boundary conditions are imposed on the walls of the computational domain. The mesh is only refined near the interface location (no vorticity-based refinement).

Calculations are initially performed on a coarse mesh, and then recursively reproduced on a mesh where the minimum and maximum tree levels are increased by one. We note that with this sequence of meshes the local resolution is not necessarily divided by two between two consecutive grids. Therefore, rigorously, the computed orders of accuracy are only lower bounds of the actual ones.

The errors at the final time and orders of accuracy for both the velocity field and the interface location in $L^\infty$-norm are reported in Table 3. As we see, because of the sharp treatment of the jump interfacial conditions, our method converges in $L^\infty$-norm for both quantities. The velocity field is only first-order accurate, which is expected as the gradient of the Hodge variable is only first-order accurate at the T−junctions of the mesh. As also observed in the single phase case [14], our results seem to indicate that the actual order of the method might be between 1 and 2 but indeed not 2. Since the interface accuracy is bounded by the accuracy of the velocity, it is also first-order accurate in $L^\infty$-norm.

## 5.2. Parasitic currents

In this second example, we reproduce the canonical example of parasitic currents presented in previous studies such as [32,11,42]. We consider an initially circular drop (2D) of radius $R$ and measure the velocity and interface displacement induced by the parasitic currents. The fluid parameters are chosen to be:

$$\mu^- = \rho^- = 1, \quad \mu^+ = \rho^+ = 1.001, \quad \gamma = \frac{1}{12\,000}, \quad \Omega = [-1.25, 1.25]^2, \quad t_{final} = 250, \quad R = 0.5. \tag{103}$$

Calculations are performed on grids refined close to the interface (no vorticity-based refinement), with a minimum level of 4 and increasing maximum levels.

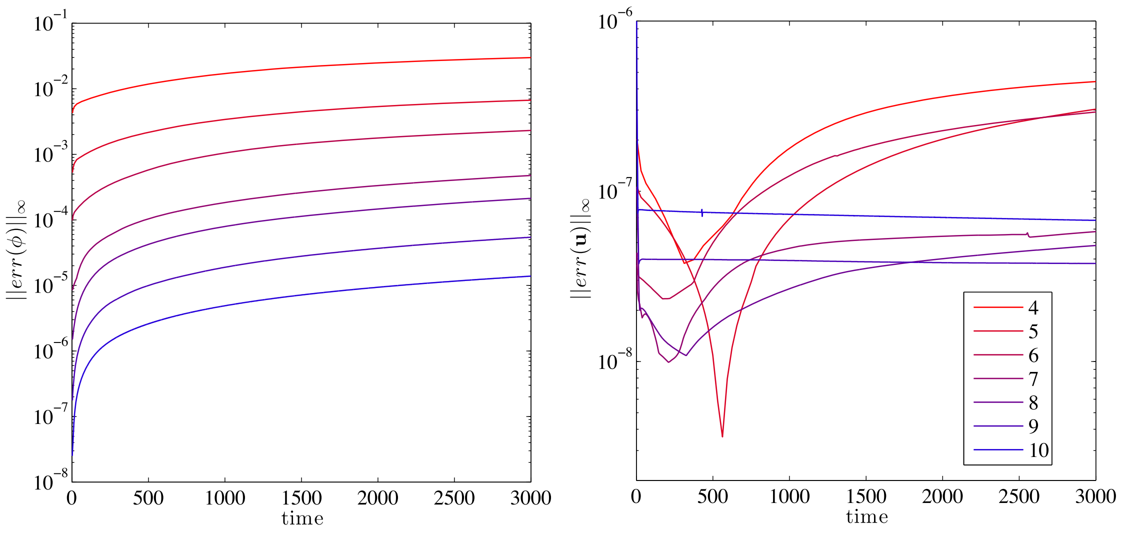


**Fig. 9.** Parasitic currents: interface location and velocity fields errors. As for the analytical case, the interface maximum error is computed within a band of $2\Delta x$ close to the interface. The red curves are obtained on the coarsest grid (max level 4), while the blue ones are obtained on the finest one (max level 10).

**Table 4**
Parasitic currents: evolution of the parasitic velocity and interface displacement as the grid is refined close to the interface. The reported values are computed at the final time $t_f = 3\,000$.

| Quadtree level $\max_{\text{level}}$ | Velocity | | Interface location | |
|---|---|---|---|---|
| | $L^\infty$ error | order | $L^\infty$ error | order |
| 4 | $4.441 \times 10^{-7}$ | - | $4.876 \times 10^{-2}$ | - |
| 5 | $3.064 \times 10^{-7}$ | 0.525 | $1.382 \times 10^{-2}$ | 1.818 |
| 6 | $2.931 \times 10^{-7}$ | 0.064 | $2.112 \times 10^{-3}$ | 2.703 |
| 7 | $5.803 \times 10^{-8}$ | 2.336 | $7.941 \times 10^{-4}$ | 1.418 |
| 8 | $4.807 \times 10^{-8}$ | 0.271 | $2.708 \times 10^{-4}$ | 1.552 |
| 9 | $3.764 \times 10^{-8}$ | 0.352 | $6.057 \times 10^{-5}$ | 2.161 |
| 10 | $6.749 \times 10^{-8}$ | -0.842 | $1.717 \times 10^{-5}$ | 1.818 |

The resulting $L^\infty$-errors for both the interface location and the velocity as functions of time are depicted in Fig. 9 and Table 4. We note that the reported orders of accuracy are in fact only a poor approximation of the actual ones since the mesh remains coarse far away from the interface with the refinement strategy employed here. Still, the convergence of the level-set indicates a second-order accuracy, while the error in the velocity, despite not converging for this refinement strategy, remains way below any tolerance level and is comparable to – if not smaller than – that reported in other studies [32,11,42]. Because the velocity is so small, we expect the error in the interface location to be mostly due to advection and reinitialization of the level-set, which have been shown to be second-order accurate [24]. The second-order convergence of the interface location is therefore not surprising.

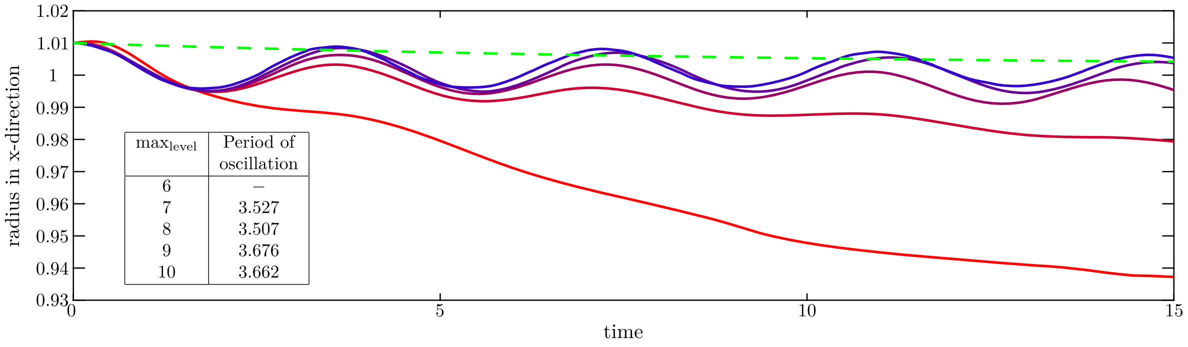
For this example, our improved time step (95) is much larger than the traditional one (92), which explains why we can run simulations for much longer times and on much finer grids than in other studies. For example, on the finest tree ($\max_{\text{level}} = 10$) our time step is about 6 000 times larger than the traditional one.

## 5.3. Oscillating bubble

In this third example, we consider the dynamics of an oscillating bubble. The bubble is initially perfectly circular and undergoes a slight shape deformation. The resulting non-uniform curvature induces a non-uniform capillary force that, coupled to the effect of inertia, causes the bubble to oscillate. These oscillations are eventually damped by viscous effects.

In the limit of small deformations and in three dimensions, Lamb [18] calculated the radius $R(\theta, \phi, t)$ as an eigenmode expansion in spherical coordinates $(r, \theta, \phi)$:

$$R(\theta, \phi, t) = R_a + \sum_{n=0}^{\infty} \epsilon_n \cos(\omega_n t) S_n(\theta, \phi) e^{-\frac{t}{\tau_n}}, \tag{104}$$



| $\max_{\text{level}}$ | Period of oscillation |
|---|---|
| 6 | – |
| 7 | 3.527 |
| 8 | 3.507 |
| 9 | 3.676 |
| 10 | 3.662 |

**Fig. 10.** Oscillating bubble: Radius in the $x$-direction as a function of time for various maximum levels of refinement. The blue curve corresponds to the results obtained for $\max_{\text{level}} = 10$, while the red one is for $\max_{\text{level}} = 6$. The green curve shows the three-dimensional exponential decay predicted by Lamb [18]. Table inset shows the average period of oscillation. The theoretical prediction of [37] is 3.629.

where $R_a$ is the radius of the undeformed bubble, $\epsilon_n$ is the amplitude of mode $n$ and $S_n$ is a surface harmonic of degree $n$. The pulsations of oscillation $\omega_n$ and characteristic times $\tau_n$ are given by:

$$\omega_n = \sqrt{\gamma \frac{n(n-1)(n+1)(n+2)}{R_a^3\left(\rho^-(n+1)+\rho^+ n\right)}}, \tag{105}$$

$$\tau_n = \frac{R_a^2 \rho^-}{\mu^-(2n+1)(n-1)}. \tag{106}$$

In two dimensions, the pulsations are predicted to be [37]

$$\omega_n^{2D} = \sqrt{\frac{(n^3-n)\gamma}{(\rho^+ + \rho^-)R_\infty^3}}, \tag{107}$$

and we were unable to find corresponding expressions for the damping times $\tau_n^{2D}$. Here, we consider a two-dimensional bubble that is initially deformed according to the first mode, for which the corresponding level-set function in polar coordinates is

$$\phi(r,\theta) = r - (R + \epsilon P_2(\theta)) = r - \left(R + \frac{\epsilon}{2}(3\cos^2\theta - 1)\right), \tag{108}$$

with $R = 1$ and $\epsilon = 0.01$. The asymptotic radius toward which the bubble is expected to converge is given by the equivalent radius:

$$R_a = \sqrt{\frac{A}{\pi}} = \sqrt{R^2 + \frac{R\epsilon}{2} + \frac{11}{32}\epsilon^2}, \tag{109}$$

where $A$ is the bubble area. The fluid properties and computational parameters are

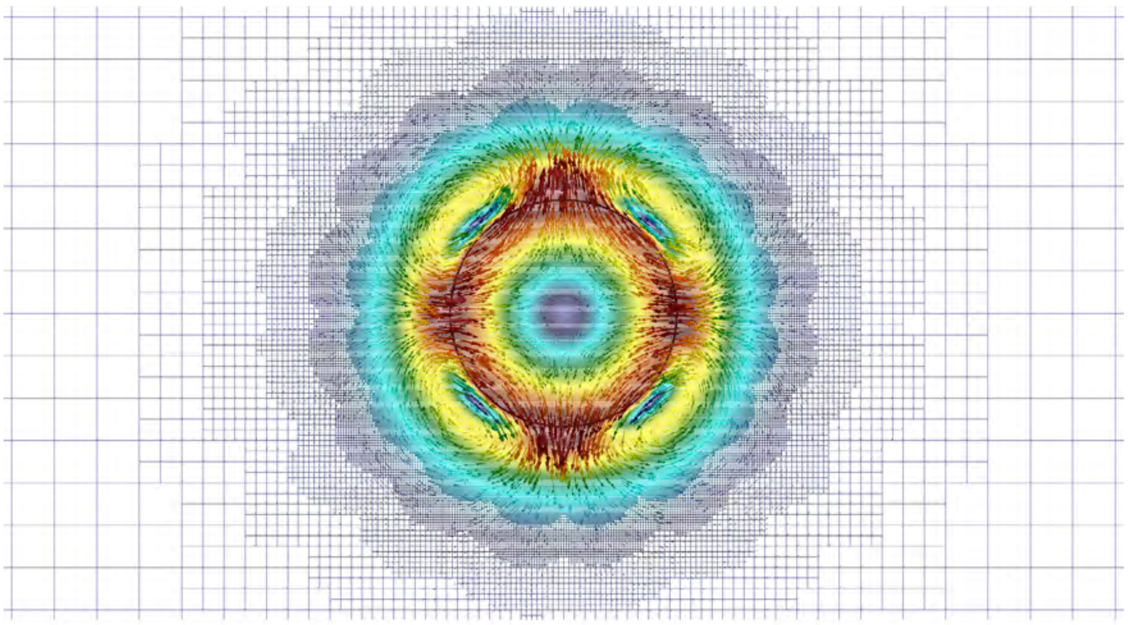
$$\mu^- = 0.02 \quad \rho^- = 1, \quad \frac{\mu^-}{\mu^+} = \frac{\rho^-}{\rho^+} = 10^3, \quad \gamma = 0.5, \quad \Omega = [-6,6]^2, \quad t_{final} = 16, \quad \min_{\text{level}} = 4. \tag{110}$$

No-slip boundary conditions are imposed on the walls of the computational domain. Homogeneous no-flux conditions are imposed on the pressure.

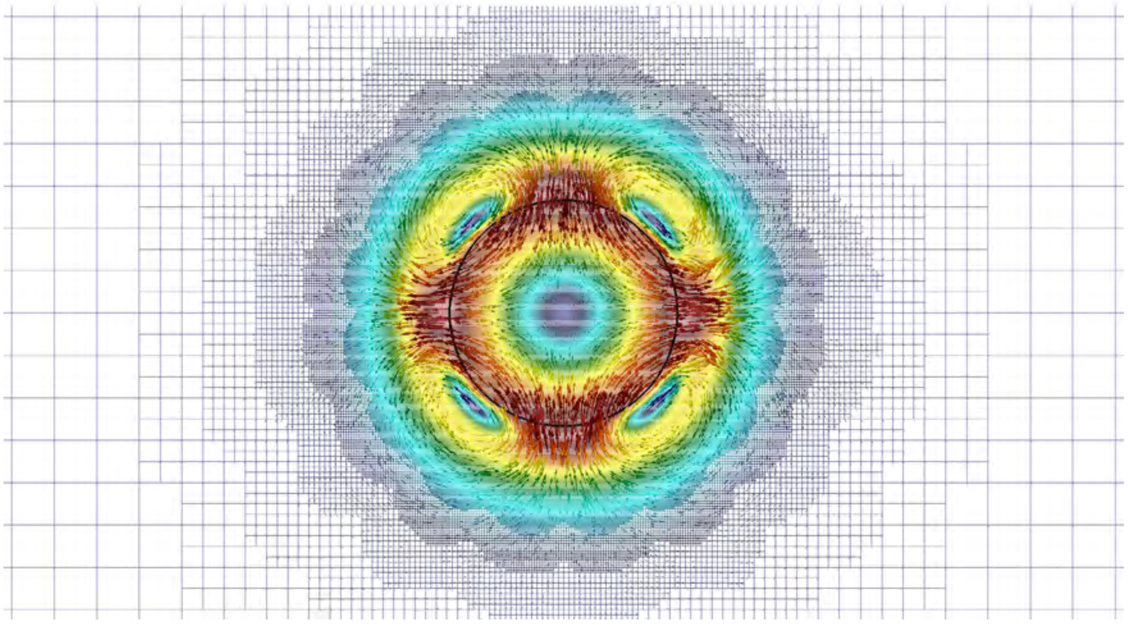
Fig. 10 depicts the evolution of the bubble radius in the $x$-direction as a function of time for various levels of refinement. The measured oscillation period $T$ reported in Fig. 10 converges to the theoretical two-dimensional prediction of $T_2 = 2\pi/w_2^{2D} = 3.629$ as the mesh is refined. On the coarsest mesh, the radius variations are dominated by the mass loss, and it is impossible to measure the oscillation period. The observed exponential decay in the oscillations is similar to that predicted by Lamb [18] in 3D. Figs. 11a and 11b show the velocity profiles at $T/4$ and $3T/4$, when the drop is spherical (i.e. $R_x = R_a$) and is expanding and compressing rapidly in the $y$- and $x$-directions, respectively.

## 5.4. Electrohydrodynamics of a viscous droplet

We now turn our attention to the electrohydrodynamics of viscous liquid dielectric droplets placed in a constant and uniform electric field (see Fig. 12). This classic problem was first addressed by Taylor [43], who first recognized that the oblate drop shapes observed in experiments [34] can only be explained by the emergence of an interfacial charge distribution at the liquid-liquid interface, which results from the weak conductivity of the media. This led him to the formulation



(a) $t = 0.907 = \frac{T_2}{4}$



(b) $t = 2.722 = \frac{3T_2}{4}$

**Fig. 11.** Oscillating bubble: Velocity profile at two different times obtained on the finest mesh ($\text{max}_{\text{level}} = 8$).

of the classic Taylor-Melcher leaky dielectric model [22,35], which we also use here and is presented in detail below. While most simulations of this problem have used boundary-integral methods and have thus been limited to the Stokes regime [19,8,7], our approach allows us to extend these results to inertial regimes. Given the extensive numerical and theoretical literature on droplet electrohydrodynamics, this problem provides an excellent example for validating and illustrating the capabilities of our method, in particular in the case where external interfacial (or volumic) forces have to be taken into account. For the range of parameters considered here, we will see that our new time step is much larger than the standard one, thus drastically reducing the computational cost and in fact allowing for fully three-dimensional simulations.

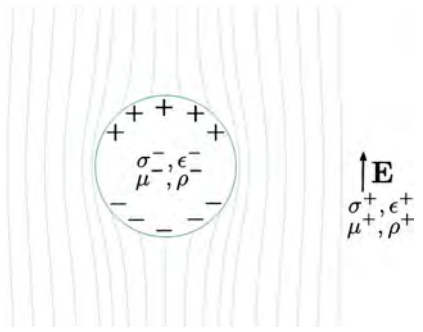


**Fig. 12.** Electrohydrodynamics of a viscous droplet: problem definition. A viscous droplet of a leaky dielectric fluid is placed in a uniform applied electric field, which induces a charge distribution at the interface. This distribution deforms the electric field and thereby induces electric stresses on the interface that generate drop deformations and drive a fluid flow. The plot shows electric field lines around a spherical drop.

### *5.4.1. Problem definition and theoretical predictions*

We consider an initially perfectly spherical leaky dielectric droplet $(-)$ with no surface charge that is suspended in another leaky dielectric fluid $(+)$ and is subject to an applied voltage. Both fluids are incompressible and have constant viscosities and densities, as well as electric permittivities $\epsilon^{\pm}$ and conductivities $\sigma^{\pm}$.

The voltage difference generates an electric field $\mathbf{E} = -\nabla\Psi$, where the electric potential $\Psi$ satisfies the following Poisson problem:

$$\triangle\Psi = 0 \qquad \forall \mathbf{x} \in \Omega, \tag{111}$$

$$[\![\Psi]\!] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{112}$$

$$[\![\epsilon\nabla\Psi \cdot \mathbf{n}]\!] = q \qquad \forall \mathbf{x} \in \Gamma, \tag{113}$$

completed by appropriate boundary conditions (given by the voltage difference) on the walls of the computational domain. The mismatch in material properties results in an interfacial charge density $q$, which satisfies a conservation equation [22,35]:

$$\frac{\partial q}{\partial t} + \nabla_S \cdot (q\mathbf{u}) = -[\![\sigma\nabla\Psi \cdot \mathbf{n}]\!] \qquad \forall \mathbf{x} \in \Gamma, \tag{114}$$

where $\nabla_S$ is the surface divergence. The velocity $\mathbf{u}$ in the above equation is the fluid velocity, which is obtained by solution of the two-phase flow problem with an imposed interfacial stress $\mathbf{g} = [\![\mathbf{n}\cdot\mathbf{M}]\!]$ capturing electrostatic forces, where $\mathbf{M}$ denotes the Maxwell stress tensor defined as

$$\mathbf{M} = \epsilon\mathbf{E}\mathbf{E} - \frac{\epsilon}{2}E^2\mathbf{I}. \tag{115}$$

This interfacial forcing causes the drop to deform and induces a fluid flow both inside and outside of the drop as first discussed by Taylor [43].

The results we present here are limited to the case of field strengths such that $E_0 \ll E_c$, where the critical electric field $E_c$ is defined as:

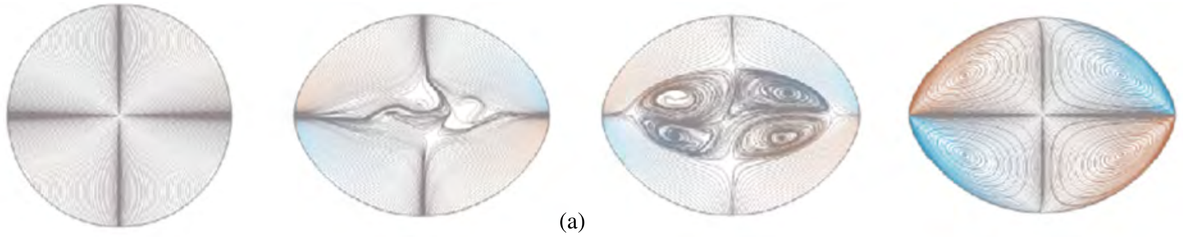
$$E_c = \sqrt{\frac{2\sigma^+\mu^+(R+2)^2}{3\epsilon^+\epsilon^-(1-RS)}} \qquad \text{where} \quad R = \frac{\sigma^-}{\sigma^+} \qquad \text{and} \qquad S = \frac{\epsilon^+}{\epsilon^-}.$$

This ensures that drop deformations are axisymmetric and allows for easy comparison to existing measurements and theoretical predictions from the literature. The dynamics in electric fields that exceed $E_c$ is known to become non-axisymmetric and lead to electrorotation [8,7]; our simulations were also shown to be able to capture that regime. In the axisymmetric case, we define Taylor's deformation parameter $D$ as $D = (d_{\parallel} - d_{\perp})/(d_{\parallel} + d_{\perp})$, where $d_{\parallel}$ and $d_{\perp}$ are the drop dimensions in the directions parallel and perpendicular to the electric field, respectively. In the case where $D > 0$, the drop elongates in the direction parallel to the field and adopts a prolate shape. In the opposite case, the shape is described as oblate. In the limit of weak deformations characterized by small values of the capillary number $Ca = E_0^2 r\epsilon^+/\gamma \ll 1$, it is possible to estimate $D$ theoretically as [43,22,10,8]

$$D_{2D} = \frac{Ca}{3S(1+R)^2}\left(SR(R+1) + S - 3\right) + O(Ca^2), \tag{116}$$

$$D_{3D} = \frac{9Ca}{16S(2+R)^2}\left(S(R^2+1) - 2 + 3(RS-1)\frac{2\lambda+3}{5\lambda+5}\right) + O(Ca^2), \tag{117}$$

in two and three dimensions, respectively, where $\lambda = \mu^-/\mu^+$ is the viscosity ratio.



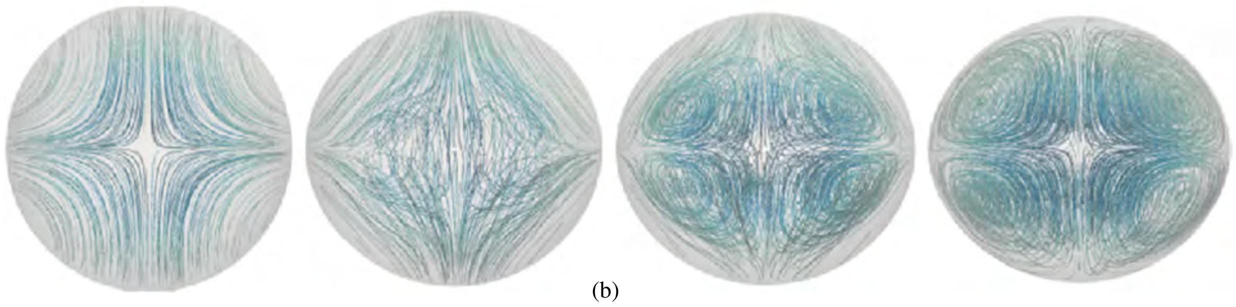


**Fig. 13.** Electrohydrodynamics of a viscous droplet: time evolution (from left to right) of an initially spherical drop placed in a uniform electric field, in both two (a) and three (b) spatial dimensions. The fluid parameters are chosen so that the drop is expected to deform into an oblate shape and the imposed electric field is $E_0 = 0.6E_c$. For the 2D results, the streamlines of the velocity are colored according to the vorticity: blue indicating a clockwise rotation, red indicating a counter-clockwise rotation. For the 3D results, the streamlines are colored according to the level-set function: dark blue corresponds to points far away from the interface while light blue corresponds to points close to the interface.

### 5.4.2. Numerical method and results

Our numerical approach for this problem is as follows. At every time step, we first update the interfacial charge distribution using a semi-Lagrangian method to discretize the advective term in Eq. (114) while treating the terms arising from conduction on the right-hand side explicitly. From there, the new electric potential is computed by solving the Poisson problem (112) using a node-based jump solver similar to that described in Section 4.3.3. The choice of a node-based rather than cell- or face-based solver is motivated by the need for an accurate calculation of the solution gradient, which provides the electric field. The Maxwell stress tensor is then determined and input into the two-phase flow solver to update the velocity field.

For the examples presented here, the parameters (in SI units) are chosen to match the experimental values reported by Salipante et al. [34]:

$$\rho^+ = 0.961 \times 10^3 \qquad \rho^- = 0.970 \times 10^3 \qquad \mu^+ = 0.69 \qquad \frac{\mu^-}{\mu^+} = 1.4 \qquad \gamma = 4.5 \times 10^{-3} \tag{118}$$

$$\sigma^+ = 4.5 \times 10^{-11} \qquad \sigma^- = 1.23 \times 10^{-12} \qquad \epsilon^+ = 5.3\epsilon_0 \qquad \epsilon^- = 3\epsilon_0 \qquad r = 10^{-3}, \tag{119}$$

where $\epsilon_0$ is the permittivity of vacuum. The computational domain is chosen to be 10 times larger than the initial drop radius so that the effects of domain boundaries are minimized. Calculations are performed on a tree of maximum level 8 in 2D and 7 in 3D. For these resolutions and chosen parameters, our method allows us to use a time step 200 times larger than previously, thus dramatically reducing the computational cost. Without this new time step criterion, carrying out 3D simulations would not have been possible. For this choice of parameters, formulae (116) and (117) predict that the drop adopts an oblate shape (i.e. elongates in the directions orthogonal to the electric field), which our simulations indeed capture as depicted in Fig. 13.

For a more quantitative validation in two dimensions, we gradually vary the strength of the applied field and compare measured drop deformations to the theoretical prediction of Eq. (116). The results showed in Fig. 14 display an excellent agreement. For comparison, experimental measurements by Salipante et al. [34] along with the 3D theoretical prediction of Eq. (117) are also depicted.

## 5.5. Dynamics and deformations of rising bubbles

### 5.5.1. Single bubble

As a final example, we analyze the dynamics of bubbles rising under gravity in both two and three dimensions, and we validate our simulation results against the experimental observations of Bhaga and Weber [3]. We consider an initially spherical bubble suspended in another denser and more viscous fluid. The density difference between the two fluids induces

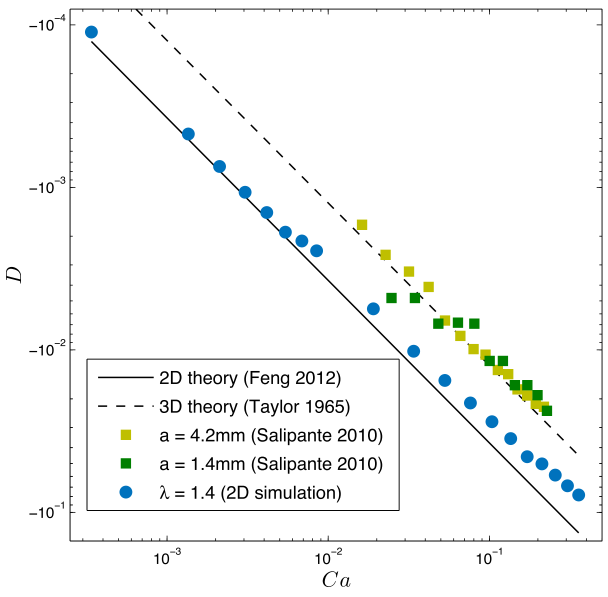


**Fig. 14.** Electrohydrodynamic deformation $D$ as a function of capillary number $Ca$ in the case of oblate drops. Our 2D numerical results are compared to the 2D theoretical prediction of Feng [10] given in Eq. (116). The 3D theoretical prediction of Eq. (117) [43] and experimental measurements [34] are also reported for comparison.

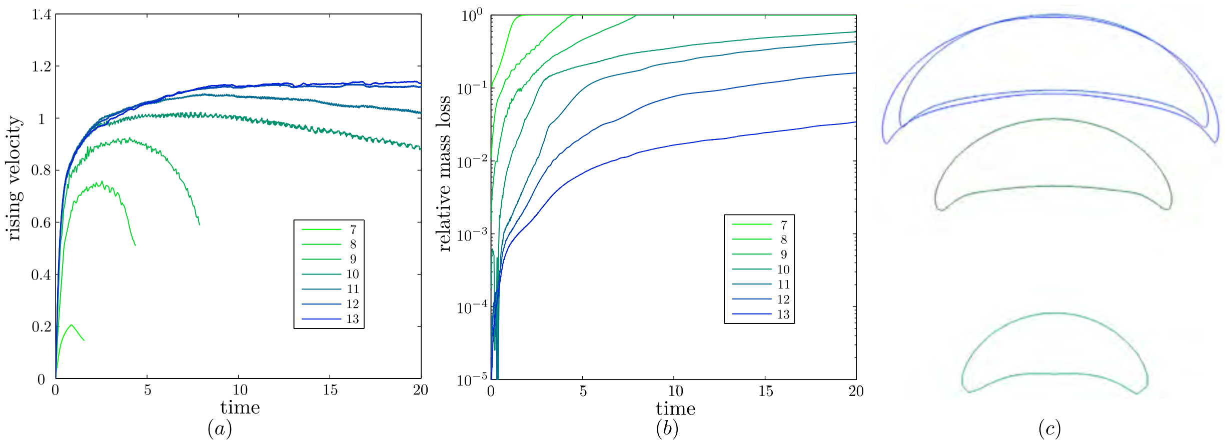


**Fig. 15.** Rising bubble: convergence analysis in two dimensions: (a) rising velocity measured at the tip of the bubble, (b) relative mass loss and (c) final interface shape and position are reported for increasing maximum resolution. The results are labeled by the maximum level of the tree they are obtained on.

a buoyancy force that causes the bubble to rise and deform. As in [3], the dynamics is governed by three dimensionless groups: the Morton, Eotvos and Reynolds numbers, respectively defined as

$$Mo = \frac{g\mu^{+4}}{\rho^+\sigma^3} \qquad Eo = \frac{gd^2\rho^+}{\sigma} \qquad Re = \frac{\rho^+ dU}{\mu^+}, \tag{120}$$

where $U$ is the asymptotic rising velocity measured at the tip of the drop, $g$ is the acceleration of gravity, and $d$ is the undeformed drop diameter. In the simulations presented here, the parameters are constructed from these three numbers, setting the rising velocity $U$ and undeformed diameter $d$ to 1:

$$\rho^+ = 1 \qquad \frac{\rho^+}{\rho^-} = 10^3 \qquad \mu^+ = \frac{\rho^+}{Re} \qquad \frac{\mu^+}{\mu^-} = 10^2 \qquad \sigma = \frac{\mu^{+2}}{\rho^+}\sqrt{\frac{Eo}{Mo}} \qquad g = \frac{Mo\rho^+\sigma^3}{\mu^{+4}}. \tag{121}$$

The density and viscosity ratios are chosen to be close to those for air and water. By setting $U = 1$, we expect the asymptotic rising velocity to be 1.

Initially, a spherical drop is placed in a container of length $L = 64$ at a height of $H = 8$. We purposely choose a large computational domain and initially place the bubble far away from the bottom wall to ensure that wall effects are negligible.

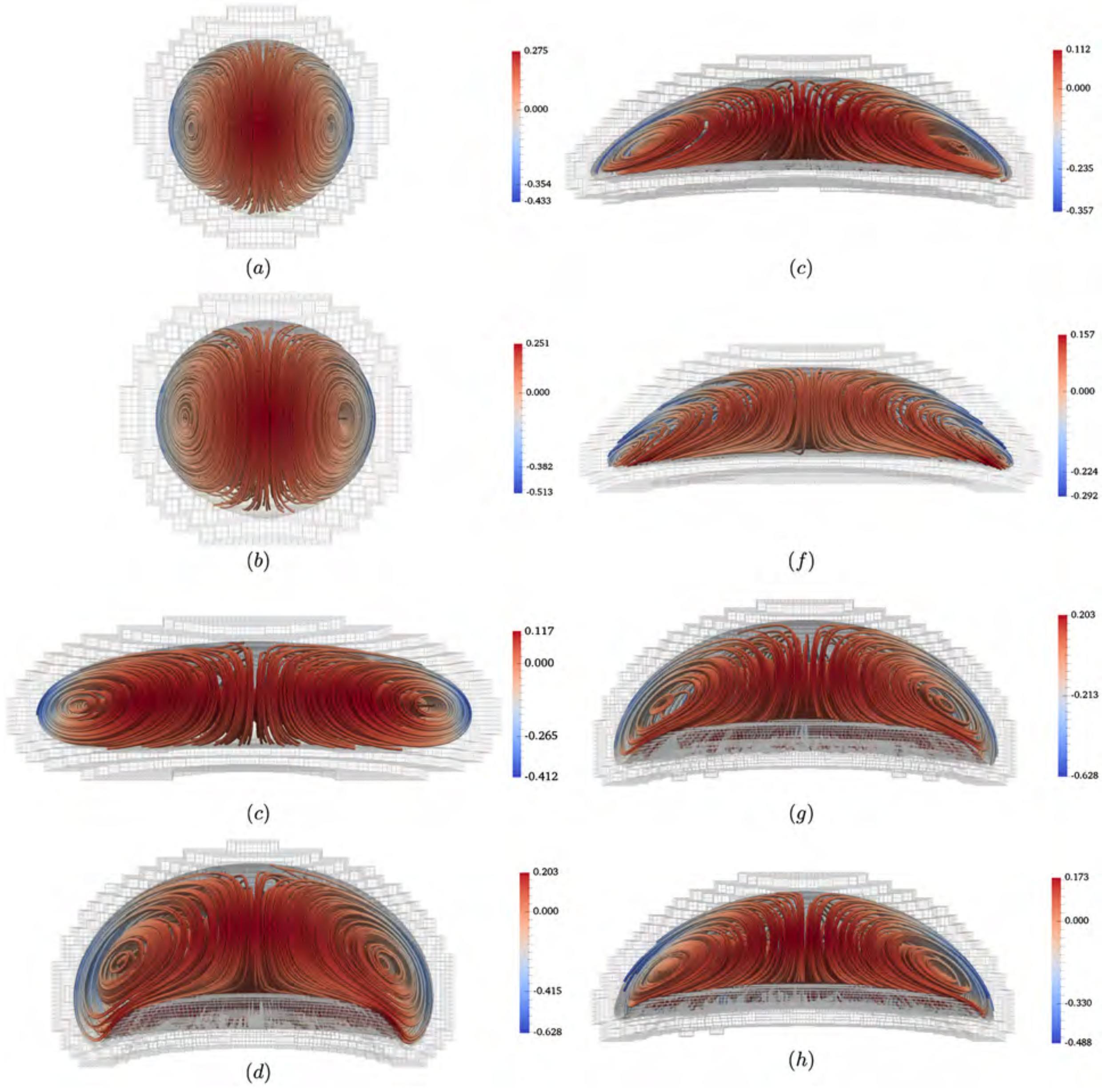


**Fig. 16.** Final bubble shapes and interior flow fields corresponding to examples 2(a–h) from Bhaga and Weber [3]. The streamlines are colored according to the apparent velocity, or velocity of the fluid in the reference frame of the rising bubble. For clarity, only half of the bubble along with the smallest mesh cells are shown.

We prescribe a no-slip condition for the velocity on all walls, except on the top boundary where we impose a no-flux condition. The pressure conditions are defined inversely. In all our calculations, the minimum level is set to 1, producing an extremely coarse mesh far away from the bubble. In addition to the interface-based refinement criterion, the grid is also refined where high velocity gradients arise using the splitting criterion (33), with $T_\mathrm{V} = 0.01$ and $\max_\mathrm{V} = \max_\mathrm{level} - 1$.

We begin by analyzing the convergence of our method for a specific case from [3] (Fig. 2d of that paper), for which $Mo = 243$, $Eo = 266$, and $Re = 7.77$. The simulations are conducted in two spatial dimensions for an arbitrary final time of $t_f = 20$ and using increasing maximum levels of refinement. The results depicted in Fig. 15 illustrate the convergence of our method. In particular, as the mesh is refined, the asymptotic rising velocity converges to a value of around 1.1, which is close to the expected rising velocity of 1 in 3D. On the coarsest levels $(7, 8, 9)$, mass losses are so important that the bubble disappears before the final time is reached. We see that by increasing the maximum resolution, these mass losses can be efficiently reduced below any arbitrary error tolerance. Graphically, both the convergence of the rising velocity and the mass loss are consistent with the first-order accuracy predicted theoretically and observed for the analytical solution

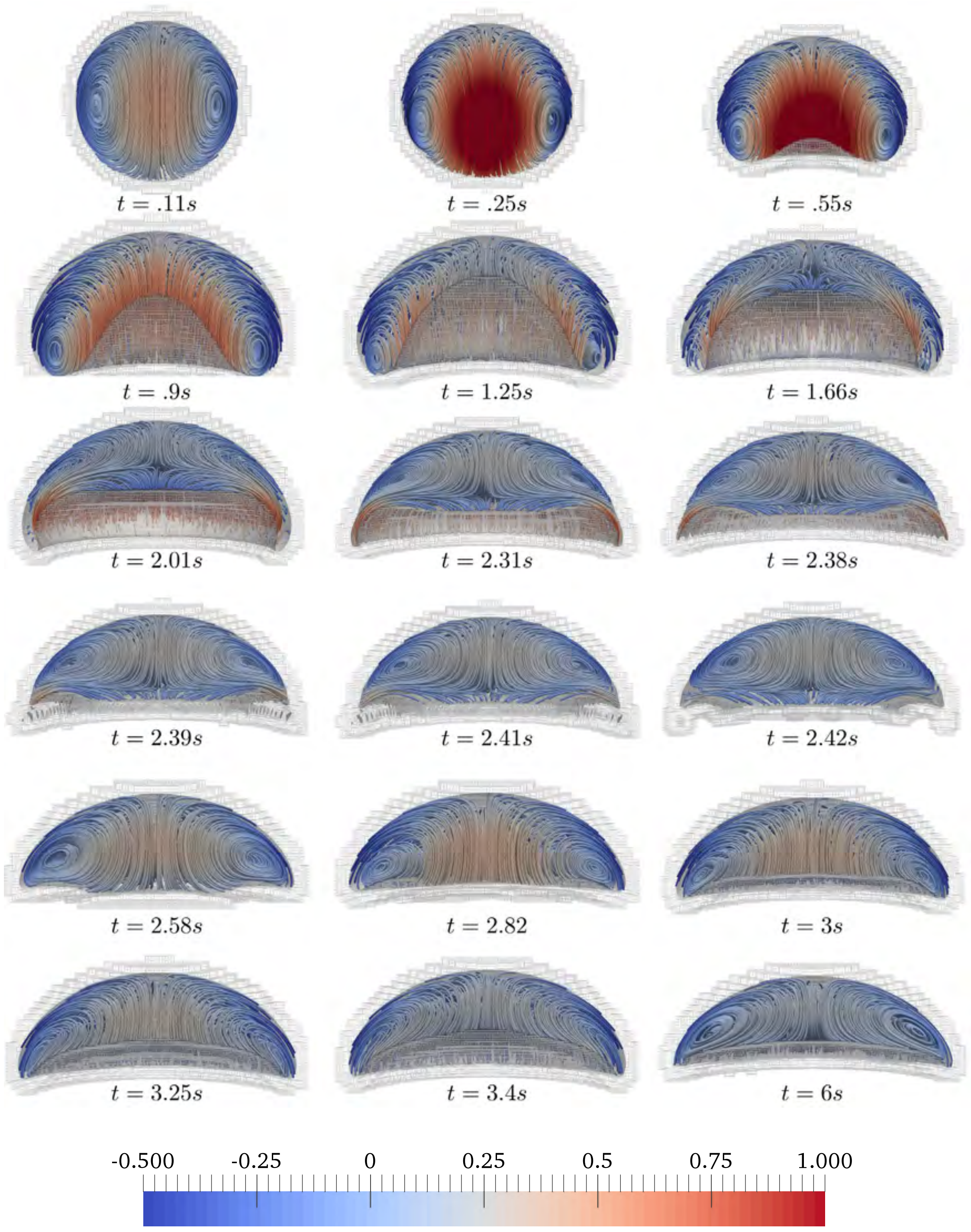


**Fig. 17.** Rising bubble: time evolution of the instantaneous shape and apparent velocity for example 2(h) from Bhaga and Weber [3]. The streamlines are colored by the apparent velocity in the ascending direction.

in Section 5.1. The final drop shapes and positions for different levels of refinement demonstrate the convergence of the method for both the interface location and velocity field. Rigorously, the comparison between these 2D contours and the 3D experimental observations of Bhaga and Weber [3] is limited, yet we note that the final bubble shape on the finest resolution resembles the experimentally observed shape.

For a more meaningful comparison, we pursue our investigation in full 3D and systematically reproduce in Fig. 16 the observations of Bhaga and Weber (see figure 2 of [3]). Qualitatively, our simulation engine does an excellent job at capturing

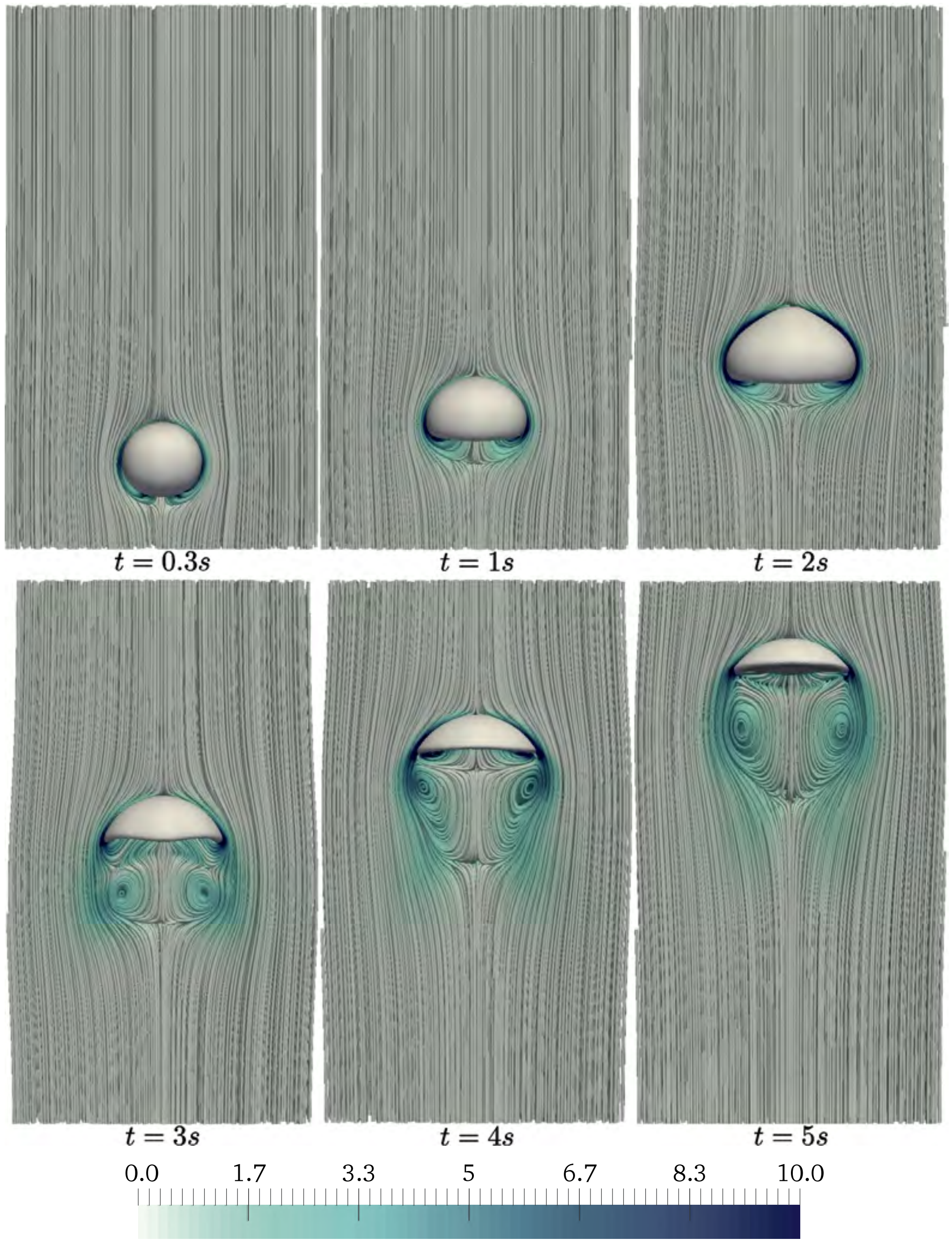


**Fig. 18.** Rising bubble: time evolution of the instantaneous shape and apparent exterior flow for example 2(f) from Bhaga and Weber [3]. The flow streamlines in the suspending fluid are colored according to the vorticity.

the various configurations observed experimentally over a wide range of parameters. The only significant difference is the presence of trailing skirts in experiments for cases (g) and (h), which we were not able to capture faithfully. Following [15], the thickness $\delta_S$ of these skirts can be estimated to be

$$\delta_S \approx 2\sqrt{\frac{3\mu^- U}{\rho^+ g}}. \tag{122}$$

In the present cases, this corresponds to 6.8% and 5.4% of the equivalent drop radius, which is clearly too small to be correctly resolved with the grid resolutions accessible here. There is therefore very little hope to observe such persistent structures. Yet, as depicted in Fig. 17, the bubble shape in case 2(h) does exhibit a skirt in the transient regime, which

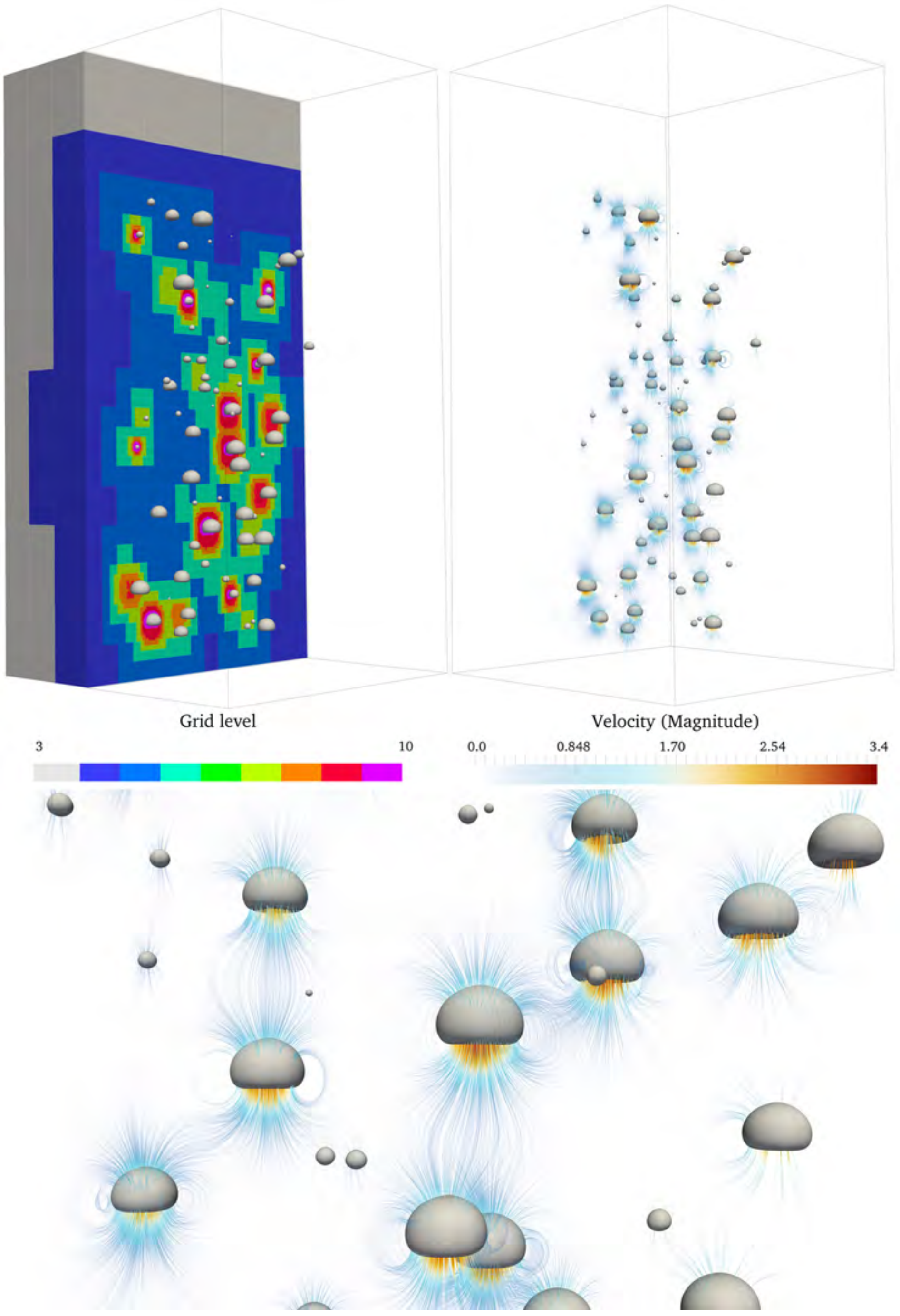


**Fig. 19.** Final configuration in a 3D simulation of a collection of 100 rising bubbles with random initial positions and sizes. The level 10 Octree (top left), composed of 4 619 885 nodes (and 4 125 493 cells) is systematically refined near the interface location and where strong velocity gradients occur. The flow is visualized through the velocity streamlines (top right), which are colored by the magnitude of the velocity. The zoomed view (bottom) further highlights the complex flow structure coupling the bubble dynamics.

progressively shrinks to the point where it cannot be correctly resolved and eventually disappears. This example is a good illustration of the ability of our method to handle complex and even singular topological changes. Looking more closely at the streamlines of the apparent velocity field inside the bubbles in Fig. 16, we note that close to the interface streamlines are tangential to the surface, as expected in the asymptotic regime as the bubbles are no longer deforming but only translating. Of course, such is not the case during the transient regime as Fig. 17 illustrates. The typical apparent flow in the suspending fluid, and in particular the establishment of a wake below the bubble, is also illustrated in Fig. 18.

**Table 5**
Rising bubbles in 3D: parameters and measurements associated with the cases shown in Fig. 16. The rising velocity is measured at the final time at the front tip of the bubble. The adaptivity ratio $r$ is defined as the ratio of the final number of grid nodes to the number of nodes in a uniform mesh with the same spatial resolution. $\Delta t_{GV}/\Delta t_B$ is the ratio of our improved time step $\Delta t_{GV}$ to the standard one $\Delta t_B$. The domain length in each direction is $L = 32$ and the drop is initially placed at a distance $H = 4$ from the bottom wall.

| | a | b | c | d | e | f | g | h |
|---|---|---|---|---|---|---|---|---|
| Final time | 1 | 2 | 5.6 | 5 | 10 | 6.6 | 4 | 6 |
| Rising velocity | 1.03 | 0.94 | 0.97 | 0.95 | 0.93 | 0.95 | 0.98 | 0.98 |
| Mass lost | 3.5 % | 2.8 % | 12.1% | 7.2% | 30.1% | 28.7% | 12.6% | 30.6 % |
| $\max_{\text{level}}$ | 10 | 10 | 11 | 11 | 11 | 11 | 11 | 11 |
| Adaptivity ratio $r$ | 0.093 % | 0.095 % | 0.0065 % | 0.0153 % | 0.0067 % | 0.0066 % | 0.0063% | 0.0064 % |
| $\Delta t_{GV}/\Delta t_B$ | 2.84 | 2.48 | 1.04 | 1.76 | 1.04 | 1.02 | 1.41 | 1.34 |

More quantitative results are presented in Table 5, where the asymptotic numerical rising velocity measured at the top of the bubble at the final time step is in great agreement with the expected value of 1. The total mass loss during the rise of the bubble in Table 5 is reasonable and illustrates once again the accuracy and convergence of the method. As expected, those cases with longer final times (cases (c) and (e)) or for which the bubble undergoes large deformations (cases (f) and (h) also shown in Figs. 18 and 17) are accompanied by higher mass losses. The efficiency of the adaptive mesh refinement strategy at maintaining a low computational cost while producing high-fidelity simulations is quantified by the adaptivity ratio $r$, of ratio of the final number of grid nodes to the number of nodes in a uniform mesh with the same maximum resolution: as shown in Table 5, $r$ is on the order of 0.01% in all eight simulations shown. For these eight examples, the difference between our improved time step and the standard one, as measured by the ratio $\Delta t_{GV}/\Delta t_B$ in Table 5, is not as impressive as it is for the previous examples. The improvement, nonetheless, is on average of a factor of two, which already significantly reduces computational time.

### 5.5.2. *Multiples bubbles*

To illustrate the full capabilities of our method, we simulate a collection of a hundred bubbles rising together. The bubbles are arbitrarily generated in a container of size $\Omega = 30 \times 30 \times 60$ with random positions and radii. The radii range from 0.3 to 3 and the fluid parameters are taken to be identical to case (d) discussed above. The boundary conditions and refinement strategy remain the same as in the previous analysis. The simulation is carried on an Octree of level 10 over a time interval of two seconds. The average number of grid cells is about $4 \times 10^6$. The simulation is performed on a desktop computer (40 cores and 64 GB RAM), on which each time step takes approximately 15 minutes.

The final configuration is depicted in Fig. 19. Qualitatively, the average asymptotic bubble shape is consistent with our previous observations on isolated bubbles (see Fig. 16(d)). Zooming in, we observe that the drops are no longer perfectly axisymmetric. The loss of axisymmetry is even more noticeable when considering the fluid flow, which illustrates the multiscale hydrodynamic interactions between the bubbles that our adaptive framework excels at capturing efficiently.

## 6. Conclusion

We have presented a novel numerical approach for the simulation of two-phase flows in two and three spatial dimensions. Our approach is fully sharp, in the sense that the representation of the interface, the jump in material parameters and flow variables as well as relevant boundary conditions at the interface are treated in a sharp manner to ensure that the physics is correctly captured – or, in more mathematical terms, that the overall method converges in $L^\infty$-norm. Within the context of a stable modified pressure correction method, we have proposed a new definition for the pressure guess, which *in fine* allows for calculations to be made under a much less prohibitive time step restriction. The computational cost was further reduced by using adaptive Octree/Quadtree grids. The numerical examples we have presented fully illustrate the validity and capabilities of our method. In particular, we believe that its accuracy, stability, and efficiency properties make it a unique tool for simulating and investigating the dynamics of complex multiphysics two-phase flow problems.

## Acknowledgements

Maxime Theillard gratefully acknowledges support through startup funds from the University of California, Merced. The research of Frederic Gibou was supported by ONR under MURI award N00014-17-1-2676 and by ARO W911NF-16-1-0136. David Saintillan acknowledges support from NSF Grant CBET-1705377, and from the Donors of the American Chemical Society Petroleum Research Fund through Grant 53240-ND9.

## References

[1] T. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (2004) 349–355.
[2] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, D.A. May, L. Curfman McInnes, K. Rupp, B.F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc Web page, http://www.mcs.anl.gov/petsc, 2017.
[3] D. Bhaga, M.E. Weber, Bubbles in viscous liquids: shapes, wakes and velocities, J. Fluid Mech. 105 (1981) 61–85.

[4] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[5] Y.-C. Chang, T. Hou, B. Merriman, S. Osher, Eulerian capturing methods based on a level set formulation for incompressible fluid interfaces, J. Comput. Phys. 124 (1996) 449–464.
[6] A. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.
[7] D. Das, D. Saintillan, Electrohydrodynamics of viscous drops in strong electric fields: numerical simulations, J. Fluid Mech. 8829 (2017) 127–152.
[8] D. Das, D. Saintillan, A nonlinear small-deformation theory for transient droplet electrohydrodynamics, J. Fluid Mech. 810 (2017) 225–253.
[9] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[10] J.Q. Feng, A 2D electrohydrodynamic model for electrorotation of fluid drops, J. Colloid Interface Sci. 246 (2002) 112–121.
[11] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, M.W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, J. Comput. Phys. 213 (2006) 141–173.
[12] C. Galusinski, P. Vigneaux, On stability condition for bifluid flows with surface tension: application to microfluidics, J. Comput. Phys. 227 (2008) 6140–6164.
[13] A. Guittet, M. Lepilliez, S. Tanguy, F. Gibou, Solving elliptic problems with discontinuities on irregular domains - the Voronoi interface method, J. Comput. Phys. 298 (2015) 747–765.
[14] A. Guittet, M. Theillard, F. Gibou, A stable projection method for the incompressible Navier–Stokes equations on arbitrary geometries and adaptive Quad/Octrees, J. Comput. Phys. 292 (2015) 215–238.
[15] R.I.L. Guthrie, A.V. Bradshaw, The stability of gas envelopes trailed behind large spherical cap bubbles rising through viscous liquids, Chem. Eng. Sci. 24 (1969) 913–917.
[16] F. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.
[17] M. Kang, R. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.
[18] H. Lamb, Hydrodynamics, Cambridge University Press, London, 1932.
[19] J.A. Lanauze, L.M. Walker, A.S. Khair, The influence of inertia and charge relaxation on electrohydrodynamic drop deformation, Phys. Fluids 25 (2013) 112101.
[20] R.J. LeVeque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (1997) 709–735.
[21] Frank Losasso, Ron Fedkiw, Stanley Osher, Spatially adaptive techniques for level set methods and incompressible flow, Comput. Fluids 35 (2006) 995–1010.
[22] J.R. Melcher, G.I. Taylor, Electrohydrodynamics: a review of the role of interfacial shear stresses, Annu. Rev. Fluid Mech. 1 (1969) 111–146.
[23] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier-Stokes equation on non-graded adaptive grids, J. Comput. Phys. 219 (2006) 912–929.
[24] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive Cartesian grids, J. Comput. Phys. 225 (2007) 300–321.
[25] M. Mirzadeh, M. Theillard, A. Helgadöttir, D. Boy, F. Gibou, An adaptive, finite difference solver for the nonlinear Poisson-Boltzmann equation with applications to biomolecular computations, Commun. Comput. Phys. 13 (2013) 150–173.
[26] D.Q. Nguyen, R.P. Fedkiw, M. Kang, A boundary condition capturing method for incompressible flame discontinuities, J. Comput. Phys. 172 (2001) 71–98.
[27] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[28] C. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.
[29] C. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[30] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries, J. Comput. Phys. 190 (2003) 572–600.
[31] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, J. Comput. Phys. 228 (2009) 5838–5866.
[32] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, Int. J. Numer. Methods Fluids 30 (1999) 775–793.
[33] C.H. Rycroft, Voro++: a three-dimensional Voronoi cell library in C++, Chaos 19 (2009) 041111.
[34] P.F. Salipante, P.M. Vlahovska, Electrohydrodynamics of drops in strong uniform dc electric fields, Phys. Fluids 22 (2010) 112110.
[35] D.A. Saville, Electrohydrodynamics: the Taylor-Melcher leaky dielectric model, Annu. Rev. Fluid Mech. 29 (1997) 27–64.
[36] C. Schroeder, W. Zheng, R. Fedkiw, Semi-implicit surface tension formulation with a Lagrangian surface mesh on an Eulerian simulation grid, J. Comput. Phys. 231 (2012) 2092–2115.
[37] S. Shin, D. Juric, Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity, J. Comput. Phys. 180 (2002) 427–470.
[38] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[39] M. Sussman, A. Almgren, J. Bell, P. Colella, L. Howell, M. Welcome, An adaptive level set approach for incompressible two-phase flows, J. Comput. Phys. 148 (1999) 81–124.
[40] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2000) 301–337.
[41] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[42] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, J. Comput. Phys. 221 (2007) 469–505.
[43] G.I. Taylor, Studies in electrohydrodynamics. I. The circulation produced in a drop by electrical field, Proc. R. Soc. Lond. A 291 (1966) 159–166.
[44] M. Theillard, C. Rycroft, F. Gibou, A multigrid method on non-graded adaptive Octree and Quadtree Cartesian grids, J. Sci. Comput. 55 (2013) 1–15.
[45] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multifluid flows, J. Comput. Phys. 100 (1992) 25–37.