



A volume-preserving reference map method for the level set representation

Maxime Theillard

School of Natural Sciences, Applied Mathematics Department, University of California, Merced, CA 95340, USA



ARTICLE INFO

Article history:

Available online xxxx

Keywords:

Interfacial problems
Level set
Reference map
Mass conservation
Volume-preserving
Two-phase flows

ABSTRACT

This paper presents an implicit interface representation, where the geometry is captured by a level set function and its deformations are reconstructed from diffeomorphism between the warped and original geometries (the reference map). A key advantage of this representation is that it provides a local estimation of numerical local mass losses. Using this metric, we design a novel projection for the reference map on the space of volume-preserving diffeomorphisms which results in enhanced, but inexact, mass conservation. In the limit of small deviations from this space, the projection is shown to be uniquely defined, and the correction can be computed as the solution of a Poisson problem. The method is analyzed and validated in two and three spatial dimensions. Both the theoretical and computational results show it excels at correcting the mass loss due to inaccuracy in the advection process or the velocity field. While this error reduction does not manifest for analytical problems, it is evident for practical applications such as the simulation of multiphase flows over long time intervals and offers improved computational exploration capabilities.

© 2021 Published by Elsevier Inc.

1. Introduction

Interfacial problems, in which the overall system dynamic is dictated by the motion of a complex interface, are ubiquitous to physical and engineering sciences. They are used to model a myriad of real-life applications among which solidification processes [58], dynamics of multiphase flows [59], or the biological morphogenesis [40,7] are the most paradigmatic examples. Simulating accurately and efficiently such problems remains a challenging task. It requires to model multiple physics interacting over several lengths and time scales, and, at the most fundamental level, challenge our ability to accurately represent and model deforming interfaces.

For problems involving small or negligible deformations, explicit interface representations, such as the front tracking method [48,17,16,61,68], where the interface is modeled by a conforming mesh, have been recognized as the most accurate [15] and often most practical strategies. They are, for example, at the core of most Finite or Boundary element methods. Yet, as the deformations increase, conforming the mesh to the geometry becomes challenging, in particular, if topological changes occur.

Implicit methods represent the geometry through an auxiliary function and model its deformations through an evolution equation. Within the level set framework, introduced by Osher and Sethian [43], the interface is represented by the

E-mail address: mtheillard@ucmerced.edu.

zero-contour of a so-called level set function. The mesh conformation is replaced by the numerical resolution of an advection equation, automatically handling large deformations and potential topological changes. The discretization introduces numerical errors, which in this context can manifest as mass losses. While in theory these losses can be kept arbitrarily small, using for example adaptive data structures [25,63,30,19,8,37,34], massively parallel techniques [32,22,14], advanced discretization techniques such as the Discontinuous Galerkin method [52,46,47,22,44] or high-order reconstruction procedures [36,2], in practice, for long or under resolved simulations, they can become undesirably large, deteriorating the overall accuracy and *in fine* restricting computational exploration capabilities.

Designing variations of the level set method that reduce or alleviate the mass loss is challenging because, within the standard framework, their definition is purely global, while any accurate modification would require a local treatment. The mathematical model has to be augmented for the mass loss to become measurable locally. Sussman and Puckett [55,54] proposed the Coupled Level Set and Volume of Fluid method, where both the level set and the volume fraction are used to reconstruct the interface evolution while preserving the total mass. Olsson et al. [41,42] proposed to replace the level set function with a smeared Heaviside function so that the equation of motion can be written and discretized in conservative forms, providing good mass conservation. It has been further improved and employed to simulate multiphase flows [64,13,44,39].

Recently, Bellotti and Theillard introduced the Coupled Level Set and Reference Map method (CLSRM [5]). The central idea of this new interface representation, inspired by the work of Kamrin et al. [21], is to keep track of the entire space deformation through the evolution of a diffeomorphism (the reference map) and to reconstruct the moving interface (the level set function) as the deformation of the initial interface. As it was demonstrated, this new approach grants appreciably better accuracy at marginal additional costs. The reference map is a key concept in solid mechanics, and therefore been used to simulate fluid-structure interactions [10,26], coupling with elastic [21,62] and soft [20] materials, two-phase flows [5], and most recently it has been employed for general advection problems [28,67]. Its association with the level set method was first proposed by Pons et al. in [45], and inspired by [1,65], to monitor interfacial quantities. Beyond the accuracy improvement, the CLSRM method allows us to estimate the mass loss locally by measuring at every point how much the reference-map is not volume-preserving.

This paper presents how this new metric can be used to construct a correction for the CLSRM method, where the reference map is systematically projected on the space of volume-preserving diffeomorphisms to improve the total mass conservation. It is organized as follows. We start by introducing the Coupled Level Set Reference Map method in section 2. The novel volume-preserving projection is presented in section 3. Its implementation on adaptive Oc/QuadTree is detailed in section 4. The resulting numerical errors are analyzed in section 5. Two and three-dimensional numerical examples are presented in section 6 for validation and further analysis. Concluding remarks are presented in section 7.

2. Coupled level set reference map method

In this section, we present the coupled level set and reference map method [5] on which the new approach is based and discuss the implications of the volume-preserving condition for the level set and the reference map. The volume-preserving method is constructed in the following section.

2.1. Definitions

We consider a deforming domain $\mathcal{B}(t) \subseteq \mathbb{R}^d$, possibly unbounded and defined $\forall t \geq 0$ in dimension $d = 2, 3$ as the entire domain of interest. An interface $\Gamma(t)$ represented by the zero contour of a level set function $\phi(t, \mathbf{x})$

$$\Gamma(t) = \{\mathbf{x} \in \mathcal{B}(t) \mid \phi(t, \mathbf{x}) = 0\}. \quad (1)$$

We denote by \mathcal{B}_0 , Γ_0 , and ϕ_0 the initial domain, interface, and level set function respectively. We assume that both geometries are deformed under the action of a known velocity field \mathbf{u} . The motion map $\chi(t, \cdot)$ is the morphism that transform the initial domain \mathcal{B}_0 into the deformed one $\mathcal{B}(t)$. In particular, it maps every material point $\mathbf{x}_0 \in \mathcal{B}_0$ to its corresponding image $\mathbf{x}(t)$ on the deformed geometry $\mathcal{B}(t)$ (see Fig. 1)

$$\mathbf{x}(t) = \chi(t, \mathbf{x}_0) \quad \forall t \geq 0, \quad \forall \mathbf{x}_0 \in \mathcal{B}_0. \quad (2)$$

Assuming that $\chi(t, \cdot)$ is an isomorphism (which is for example the case if $\det(\nabla \chi) > 0$)¹ we call its inverse the reference map $\xi(t, \mathbf{x})$:

$$\mathbf{x}_0 = \xi(t, \mathbf{x}) \quad \forall t \geq 0, \quad \forall \mathbf{x} \in \mathcal{B}(t). \quad (3)$$

¹ Even though the bijectivity of the motion map is guaranteed for an external velocity field because of the structure of Eq. (4) (i.e. linear advection), this fact is not general. For example, if the geometry is self-advection (e.g. motion in the normal direction or under curvature) the bijectivity of χ is lost to shocks. In such cases, the rarefactions make it impossible to formally define the motion map.

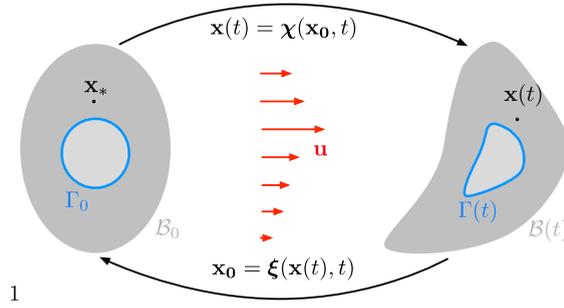


Fig. 1. Coupled level set and reference map method: definitions and notations. We consider a moving interface $\Gamma(t)$ defined on a moving domain $B(t)$. To any point $\mathbf{x}_0 \in B_0$, the motion map $\chi(t, \cdot)$ associates its image $\mathbf{x}(t)$ on $B(t)$. The inverse map $\xi(t, \cdot)$ is called the reference map. The interface $\Gamma(t)$ is represented by the level set function $\phi(t, \mathbf{x}) = \phi_0(\xi(t, \mathbf{x}))$.

The deformation of the domain $B(t)$ is captured by the transport equation for the reference map

$$\frac{\partial \xi}{\partial t} + \mathbf{u} \cdot \nabla \xi = 0 \quad \forall t \geq 0, \quad \forall \mathbf{x} \in B(t), \tag{4}$$

$$\xi(t = 0, \mathbf{x}) = \mathbf{x} \quad \forall \mathbf{x} \in B_0. \tag{5}$$

Because the domain and the interface are advected by the same velocity field \mathbf{u} , the image of Γ_0 by $\chi(t, \cdot)$ is $\Gamma(t)$, and reciprocally the image of $\Gamma(t)$ by $\xi(t, \cdot)$ is Γ_0

$$\{\mathbf{x} \in B(t) \mid \xi(t, \mathbf{x}) \in \Gamma_0\} = \Gamma(t). \tag{6}$$

Expressing the definition of the initial interface

$$\phi_0(\mathbf{x}_0) = 0 \quad \forall \mathbf{x}_0 \in \Gamma_0, \tag{7}$$

in terms of the reference map

$$\phi_0(\xi(t, \mathbf{x})) = 0 \quad \forall \mathbf{x} \in B(t) \mid \xi(t, \mathbf{x}) \in \Gamma_0 \tag{8}$$

and using the above set equality (6), we obtain

$$\phi_0(\xi(t, \mathbf{x})) = 0 \quad \forall \mathbf{x} \in \Gamma(t), \tag{9}$$

which tells us that $\phi_0(\xi(t, \cdot))$ is a level set function for the contour $\Gamma(t)$. Therefore we define

$$\phi(t, \mathbf{x}) = \phi_0(\xi(t, \mathbf{x})) \quad \forall \mathbf{x} \in B(t). \tag{10}$$

2.2. Coupled method

The central idea of the coupled method is to advect the reference map using the transport Eq. (4) and then reconstruct the level set from the definition (10). Compared to the standard level set method where the interface is directly advected, this new approach is more accurate for two main reasons. First, the advected object is more regular, hence spatial discretization and interpolations are more accurate. Typically, for an infinitely smooth advecting velocity field, the level set function carries discontinuities in its derivatives while the reference map is uniformly continuously differentiable. Second, the reinitialization errors are minimized since, as we will discuss next, reinitializations are performed only when the mapping needs to be restarted.

2.3. Restarting procedure

The entire method relies on the bijectivity of $\chi(t, \mathbf{x})$, and thus of $\xi(t, \mathbf{x})$. While in theory, this can be guaranteed by the structure of the velocity field \mathbf{u} , in practice, it can be lost due to numerical errors [21,5]. To ensure that the bijectivity is preserved, we use the following restarting procedure and refer the interested reader to [5] for additional details.

From an algebraic perspective, the bijectivity of $\xi = (\xi_1, \dots, \xi_d)$ means that the normalized columns of $\nabla \xi$ are linearly independent. Since the loss of bijectivity is only problematic close to the interface, we decide to restart the reference map when at least two columns are critically close to colinear in the vicinity of the interface:

$$\text{restart } \xi \text{ if : } \left\| \det \left(\frac{\nabla \xi_1}{\|\nabla \xi_1\|}, \dots, \frac{\nabla \xi_d}{\|\nabla \xi_d\|} \right) \right\|_{L^\infty(\gamma)} < \sin(\theta_{crit}), \tag{11}$$

where the critical angle is chosen as $\theta_{crit} = \pi/9$, and the vicinity of the interface \mathcal{V} is defined as:

$$\mathcal{V} = \{\mathbf{x} \in \mathcal{B}(t) : |\phi(t, \mathbf{x})| < \epsilon_r\} \quad t \geq 0, \quad \epsilon = 0.1 \left(\max_{\mathbf{x} \in \mathcal{B}_0} \phi_0(\mathbf{x}) - \min_{\mathbf{x} \in \mathcal{B}_0} \phi_0(\mathbf{x}) \right) \quad (12)$$

If the above criterion identifies a critical time t_c at which a restarting is needed, we set

$$\begin{aligned} \phi_0(\mathbf{x}) &= \mathcal{R}(\phi(t_c, \mathbf{x})) \quad \forall \mathbf{x} \in \mathcal{B}(t_c), \\ \xi(t_c, \mathbf{x}) &= \mathbf{x} \quad \forall \mathbf{x} \in \mathcal{B}(t_c), \end{aligned}$$

where $\mathcal{R}(\phi(t_c, \mathbf{x}))$ is the reinitialization of the level set $\phi(t_c, \mathbf{x}) = \phi_0(\xi(t_c, \mathbf{x}))$. It is the signed-distance function for the contour $\Gamma(t)$

$$\mathcal{R}(\phi(t_c, \mathbf{x})) = 0 \Leftrightarrow \phi(t_c, \mathbf{x}) = 0, \quad |\nabla \mathcal{R}(\phi(t_c, \mathbf{x}))| = 1 \quad \forall \mathbf{x} \in \mathcal{B}(t_c) \quad (13)$$

and is computed as the steady state solution in fictitious time τ of the reinitialization equation:

$$\begin{cases} \frac{\partial \phi}{\partial \tau} + \text{sign}(\phi)(|\nabla \phi| - 1) = 0 & \tau \geq 0, \quad \forall \mathbf{x} \in \mathcal{B}(t_c), \\ \phi(\tau = 0, \mathbf{x}) = \phi(t_c, \mathbf{x}) & \forall \mathbf{x} \in \mathcal{B}(t_c). \end{cases} \quad (14)$$

2.4. Motion under a volume-preserving velocity field

When the advecting velocity \mathbf{u} is divergence-free, as it is, for example, the case for incompressible fluid flows, the volume $\Omega(t)$ contained inside the interface $\Gamma(t)$ is conserved

$$\int_{\Omega(t)} d\omega = |\Omega(t)| = |\Omega_0|. \quad (15)$$

In practice, the above condition allows us to measure spurious mass loss and estimate the accuracy of the numerical method. Unfortunately, this is its only relevant implication as it only tells us whether mass was globally lost, but not specifically where it happened. If we use it to reduce the artificial mass loss we could only design a global strategy, which would destroy the local accuracy. For the reference map, the divergence-free condition implies that the map is volume-preserving: for any closed domain $\mathcal{D}(t) \subseteq \mathcal{B}(t)$ deformed by an incompressible velocity field \mathbf{u}

$$\int_{\mathcal{D}(t)} d\omega - |\mathcal{D}_0| = |\mathcal{D}(t)| - |\mathcal{D}_0| = 0, \quad (16)$$

which after a change of variable can be rewritten as

$$\int_{\mathcal{D}(t)} d\omega - |\mathcal{D}_0| = \int_{\xi(\mathcal{D}(t))} |\det(\nabla \xi^{-1})| d\omega - |\mathcal{D}_0| = \int_{\mathcal{D}_0} (|\det(\nabla \chi)| - 1) d\omega. \quad (17)$$

Since Eq. (17) must be true $\forall \mathcal{D}_0 \subseteq \mathcal{B}_0$, we obtain that

$$\det(\nabla \chi(t, \mathbf{x}_0)) = \pm 1 \quad \forall \mathbf{x}_0 \in \mathcal{B}_0. \quad (18)$$

Assuming that \mathbf{u} is smooth enough, $\det(\nabla \chi(t, \mathbf{x}_0))$ is a continuous function of time, and since $\det \nabla \chi(0, \mathbf{x}_0) = \det \mathcal{I} = 1$, we conclude that

$$\det(\nabla \chi(t, \mathbf{x}_0)) = 1 \quad \forall \mathbf{x}_0 \in \mathcal{B}_0. \quad (19)$$

Similarly, we can prove that

$$\det(\nabla \xi(t, \mathbf{x})) = 1 \quad \forall \mathbf{x} \in \mathcal{B}(t). \quad (20)$$

Unlike Eq. (16), these last two conditions are local. In practice, at any point (\mathbf{x}, t) they can be used to evaluate how much the maps deviate from being volume-preserving, and therefore estimate the local error and mass loss. The following section explains how to use this unique metric to create a volume-preserving correction.

It should be pointed out that conditions (15) and (20) are not equivalent. The second one is stronger and implies that the velocity field is incompressible. The first one only means that the velocity field preserved the volume enclosed in $\Gamma(t)$. Enforcing condition (20) would preserve any enclosed volume $\mathcal{D}(t)$, independently of the velocity field preserving the volume enclosed in $\Gamma(t)$ only or being fully incompressible.

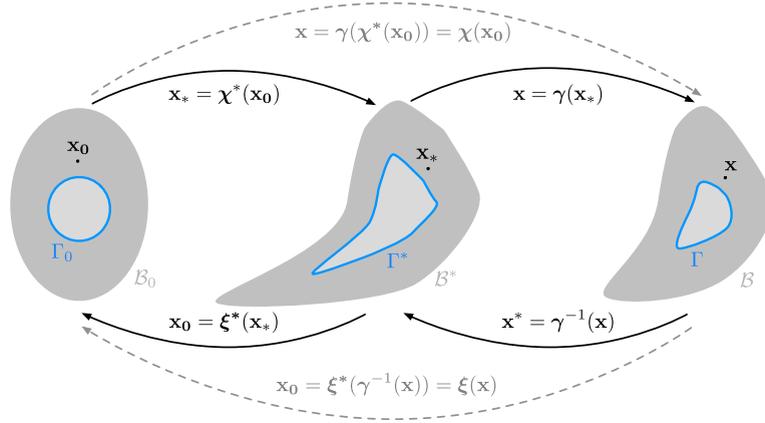


Fig. 2. Volume-preserving projection. We consider a volume preserving transformation, and assume that at some given time the constructed maps (χ^*, ξ^*) are not volume-preserving due to numerical errors. We correct these maps into a volume-preserving pair (χ, ξ) by composing them with a diffeomorphism γ .

This is an important remark as for most interfacial problems, the velocity is only known on the interface. It can be preserving the enclosed volume but cannot be incompressible as its divergence is not formally defined. For example, with Stefan’s problem [58], the interfacial velocity is defined as the heat flux jump across the interface. It may be preserving the mass of the phases it separates in some specific cases, but it cannot, by definition, be incompressible. Another canonical example is the two-phase flow problem: the velocity field is continuous across the interface, but not its gradient due to the stress balance condition. If the fluids are incompressible, the interfacial velocity must preserve the enclosed volume. The incompressibility condition is satisfied everywhere away from the interface, but on the interface itself, the divergence operator is ill-defined. In practice, for both cases, the interfacial velocity may be evaluated at the interface and then extended to the entire domain (see [58,59]). These extensions may not be incompressible.

3. Volume-preserving projection

The central idea of the volume-preserving projection is to measure how much the reference map deviates from a volume-preserving diffeomorphism and to use this measure to locally correct the reference map to ensure that the condition (20) is satisfied. This correction is a projection onto the space of volume-preserving diffeomorphisms, as we will see. It is defined from the following observations and, using standard variational calculations, can be proven to be uniquely defined in the small deviations limit.

3.1. General observations

As Fig. 2 illustrates, we consider a pair of motion and reference maps (χ^*, ξ^*) (typically the numerical solution of the coupled method described in section 2 at time step t_n) and the associated domain B^* and interface Γ^* . For the sake of simplicity we will omit the time dependence in the notations. We assume that the transformation is theoretically preserving the volume enclosed by $\Gamma(t)$, but that the maps are not volume-preserving due to errors in their construction, and would like to correct them into a volume-preserving pair (χ, ξ) . To do so, we compose the mapping with a correction diffeomorphism $\gamma : B^* \rightarrow B$ such that

$$\chi(\mathbf{x}_0) = \gamma(\chi^*(\mathbf{x}_0)), \quad \det \nabla \chi(\mathbf{x}_0) = 1, \quad \forall \mathbf{x}_0 \in B_0, \tag{21}$$

$$\xi(\mathbf{x}) = \xi^*(\gamma^{-1}(\mathbf{x})), \quad \det \nabla \xi(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in B. \tag{22}$$

For any diffeomorphism γ , this correction is stable since from Eq. (22)

$$\|\xi\|_{L^\infty(B)} \leq \|\xi^*\|_{L^\infty(B^*)}. \tag{23}$$

Independently of the errors sources (e.g. the inaccuracy on the velocity or the advecting scheme), the deviation of the reference map from the volume-preserving operator space should be small, typically comparable to the overall accuracy of the numerical method ($\mathcal{O}(\Delta t^a + \Delta x^b)$). Therefore the correction γ should be close to the identity operator

$$\gamma(\mathbf{x}_*) = \mathbf{x}_* + \epsilon(\mathbf{x}_*) = \mathbf{x}_* + \mathcal{O}(\Delta t^a + \Delta x^b), \quad \forall \mathbf{x}_* \in B^*, \tag{24}$$

and the projected map should be close to the original one

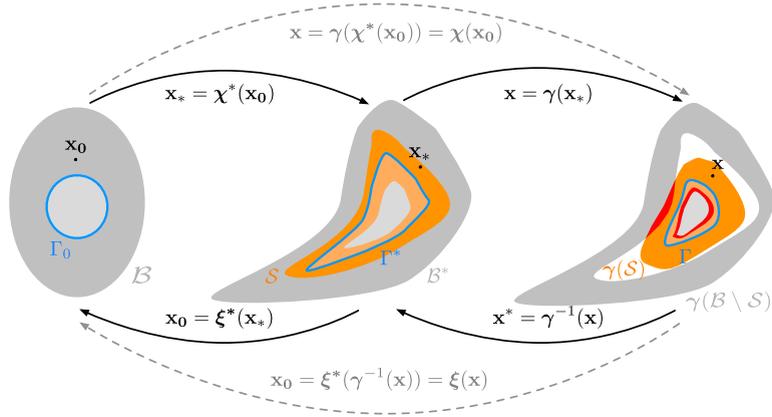


Fig. 3. Potential parasitic topological changes induced by the thin shell correction: loss of bijectivity (overlapping region in red) or artificial holes in the domain \mathcal{B} (white region between $\gamma(\mathcal{S})$ and $\gamma(\mathcal{B}^* \setminus \mathcal{S})$). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\chi(\mathbf{x}_0) = \chi^*(\mathbf{x}_0) + \epsilon(\chi^*(\mathbf{x}_0)) + \mathcal{O}(|\epsilon|^2) = \chi^*(\mathbf{x}_0) + \mathcal{O}(\Delta t^a + \Delta x^b), \quad \forall \mathbf{x}_0 \in \mathcal{B}_0, \tag{25}$$

which - provided that a and b are large enough, and that the time step is adequately defined - guarantees that the method is consistent if the pair (χ^*, ξ^*) is as well. In addition, if the correction is close to the identity, it is bijective, and therefore the composed mappings will also be. Looking at Eq. (22) we realize that the inverse of the correction is needed to compute the corrected reference map ξ . While in theory this can be directly computed from the correction using a fixed-point method, in practice we can use the approximations

$$\gamma^{-1}(\mathbf{x}) = \mathbf{x} - \epsilon(\mathbf{x}) + \mathcal{O}(|\epsilon|^2), \quad \forall \mathbf{x} \in \mathcal{B}, \tag{26}$$

which will preserve the overall accuracy of the method. Since, in the present study, ultimately the interface location is the only information we are interested in, we could perform that correction only in a shell $\mathcal{S} \subseteq \mathcal{B}^*$ containing the interface, defining the correction as

$$\gamma(\mathbf{x}^*) = \begin{cases} \mathbf{x}^* & \text{if } \mathbf{x}^* \in \mathcal{B}^* \setminus \mathcal{S} \\ \mathbf{x}^* + \epsilon(\mathbf{x}) & \text{if } \mathbf{x}^* \in \mathcal{S} \end{cases} \tag{27}$$

and the above observations about the stability and consistency of the method would remain valid. The main difference induced by this discontinuous definition is that the correction is now only bijective between \mathcal{S} and $\gamma(\mathcal{S})$ and between $\mathcal{B}^* \setminus \mathcal{S}$ and $\gamma(\mathcal{B}^* \setminus \mathcal{S})$ but not between \mathcal{B}^* and $\gamma(\mathcal{S})$. Furthermore, while for $\mathcal{S} = \mathcal{B}^*$, \mathcal{B} is guaranteed to be connected if \mathcal{B}^* is, using $\mathcal{S} \subsetneq \mathcal{B}^*$ the sets $\gamma(\mathcal{S})$ and $\gamma(\mathcal{B}^* \setminus \mathcal{S})$ are connected but not necessarily their union. As Fig. 3 illustrates, these properties can have disastrous consequences: the loss of bijectivity can create overlapping regions, while the loss of connectedness can drill artificial holes in \mathcal{B} .

Performing the shell correction only guarantees that the volume of any domain enclosed in the shell is preserved through the corrected transformation. Therefore, if the shell contains $\Omega(t)$, the volume $|\Omega(t)|$ will be conserved. If $\Omega(t) \not\subseteq \mathcal{S}$ the enclosed volume $|\Omega(t)|$ may not be preserved.

Finally, to minimize the potential numerical errors, if such a correction exist, we would like to pick the smallest possible correction in $L^2(\mathcal{S})$ norm, so that (χ, ξ) are the projection of (χ^*, ξ^*) on the space of volume-preserving diffeomorphisms for that norm. Because of the correction definition (21), (22), the smallest correction means the closest to the identity operator, i.e. the smallest ϵ .

3.2. Definition of the correction

From the above observations, for any prescribed shell \mathcal{S} , we define $\gamma(\mathbf{x}) = \mathbf{x} + \epsilon(\mathbf{x})$ as the solution of the constrained optimization problem:

$$\min_{\epsilon} \int_{\mathcal{S}} \frac{\epsilon \cdot \epsilon}{2} ds \quad \text{such that} \quad \det \nabla \xi(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in \mathcal{S}. \tag{28}$$

For the correction to appear in the optimization problem, we need to reformulate the constraint. Starting from the definition of the correction (22), we rewrite it as

$$\det(\nabla(\xi^* \gamma^{-1})(\mathbf{x})) = 1 = \det(\mathcal{I}) = \det(\nabla \mathbf{x}) = \det(\nabla(\gamma \gamma^{-1})(\mathbf{x})) \quad \forall \mathbf{x} \in \gamma(\mathcal{S}). \tag{29}$$

Using the chain rule and determinant properties, this last expression can be expressed as

$$\det(\nabla \xi^*(\boldsymbol{\gamma}^{-1}(\mathbf{x}))) \det(\nabla \boldsymbol{\gamma}^{-1}(\mathbf{x})) = \det(\nabla \boldsymbol{\gamma}(\boldsymbol{\gamma}^{-1}(\mathbf{x}))) \det(\nabla \boldsymbol{\gamma}^{-1}(\mathbf{x})) \quad \forall \mathbf{x} \in \boldsymbol{\gamma}(\mathcal{S}), \quad (30)$$

simplifying as

$$\det(\nabla \xi^*(\boldsymbol{\gamma}^{-1}(\mathbf{x}))) = \det(\nabla \boldsymbol{\gamma}(\boldsymbol{\gamma}^{-1}(\mathbf{x}))) \quad \forall \mathbf{x} \in \boldsymbol{\gamma}(\mathcal{S}), \quad (31)$$

and since $\boldsymbol{\gamma}$ is an isomorphism,

$$\det(\nabla \xi^*(\mathbf{x})) = \det(\nabla \boldsymbol{\gamma}(\mathbf{x})) \quad \forall \mathbf{x} \in \mathcal{S}. \quad (32)$$

Using the small correction expansions (24), the above equation becomes

$$\det(\nabla \xi^*(\mathbf{x})) = \det(\mathcal{I} + \nabla \boldsymbol{\epsilon})(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{S}, \quad (33)$$

which to the leading order in $\boldsymbol{\epsilon}$ gives us

$$\det(\nabla \xi^*(\mathbf{x})) = 1 + \text{Tr}(\nabla \boldsymbol{\epsilon})(\mathbf{x}) = 1 + \nabla \cdot \boldsymbol{\epsilon}(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{S}, \quad (34)$$

implying that

$$\nabla \cdot \boldsymbol{\epsilon}(\mathbf{x}) + 1 - \det(\nabla \xi^*(\mathbf{x})) = 0 \quad \forall \mathbf{x} \in \mathcal{S}. \quad (35)$$

Using this new expression for the constraint, we reformulate the optimization problem in terms of $\boldsymbol{\epsilon}$ into the following definition.

Definition 1. Under the assumption of small deviations from the volume-preserving condition, $\forall \mathcal{S} \subseteq \mathcal{B}^*$, we define the correction $\boldsymbol{\gamma}$ and its inverse $\boldsymbol{\gamma}^{-1}$ as

$$\boldsymbol{\gamma}(\mathbf{x}) = \mathbf{x} + \boldsymbol{\epsilon}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{B}, \quad (36)$$

$$\boldsymbol{\gamma}^{-1}(\mathbf{x}) = \mathbf{x} - \boldsymbol{\epsilon}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{B}, \quad (37)$$

where the diffeomorphism $\boldsymbol{\epsilon}$ is the first component of a saddle point $(\boldsymbol{\epsilon}, \lambda)$ of the Lagrangian

$$\mathcal{L}(\boldsymbol{\mu}, \rho) = \int_{\mathcal{S}} \frac{\boldsymbol{\mu}(\mathbf{x}) \cdot \boldsymbol{\mu}(\mathbf{x})}{2} ds + \int_{\mathcal{S}} (\nabla \cdot \boldsymbol{\mu}(\mathbf{x}) + 1 - \det(\nabla \xi^*(\mathbf{x}))) \rho(\mathbf{x}) ds \quad (38)$$

defined for $\boldsymbol{\mu}$ and ρ in the Sobolev spaces $H^1(\mathcal{S})^2$ and $H^1(\mathcal{S})$ respectively.

We should remark that it was only assumed that the deviations from the volume-preserving space are small, which allowed us to linearize the inverse correction and the non-linear volume-preserving condition (32). In particular, we did not assume anything on the structure of $\boldsymbol{\epsilon}(\mathbf{x})$.

3.3. Existence and uniqueness

Using standard functional analysis [50] and optimization theory [38], the saddle point of the Lagrangian $\mathcal{L}(\boldsymbol{\mu}, \rho)$ can be shown to be uniquely defined. Moreover we can derive an explicit construction method for the adjoint λ from which the diffeomorphism $\boldsymbol{\epsilon}$ can be directly obtained.

Theorem 1. $\forall \mathcal{S} \subseteq \mathcal{B}^*$, the Lagrangian $\mathcal{L}(\boldsymbol{\mu}, \rho)$ has a unique saddle point $(\boldsymbol{\epsilon}, \lambda)$. The adjoint λ is the unique solution of the following Poisson problem

$$\begin{cases} -\Delta \lambda(\mathbf{x}) = 1 - \det(\nabla \xi^*(\mathbf{x})) & \forall \mathbf{x} \in \mathcal{S}, \\ \lambda(\mathbf{x}) = 0 & \forall \mathbf{x} \in \partial \mathcal{S}, \end{cases} \quad (39)$$

and the correction $\boldsymbol{\epsilon}$ is defined as

$$\boldsymbol{\epsilon}(\mathbf{x}) = \nabla \lambda(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{S}. \quad (40)$$

Proof. At any saddle point (ϵ, λ) , the derivatives of the Lagrangian are null in all direction, i.e.

$$\left\langle \frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}}(\epsilon, \lambda) \middle| \boldsymbol{\Theta} \right\rangle = 0 \quad \forall \boldsymbol{\Theta} \in H^1(\mathcal{S})^2, \quad \left\langle \frac{\partial \mathcal{L}}{\partial \rho}(\epsilon, \lambda) \middle| \theta \right\rangle = 0 \quad \forall \theta \in H^1(\mathcal{S}). \quad (41)$$

The second condition in (41) tells us that

$$\int_{\mathcal{S}} (\nabla \cdot \boldsymbol{\epsilon}(\mathbf{x}) + 1 - \det(\nabla \boldsymbol{\xi}^*(\mathbf{x}))) \theta(\mathbf{x}) ds \quad \forall \theta \in H^1(\mathcal{S}), \quad (42)$$

and since it must hold $\forall \theta \in H^1(\mathcal{S})$, we immediately recover the constraint

$$\nabla \cdot \boldsymbol{\epsilon}(\mathbf{x}) + 1 - \det(\nabla \boldsymbol{\xi}^*(\mathbf{x})) = 0 \quad \forall \mathbf{x} \in \mathcal{S}. \quad (43)$$

The first condition in (41) yields

$$\int_{\mathcal{S}} \boldsymbol{\Theta}(\mathbf{x}) \cdot \boldsymbol{\epsilon}(\mathbf{x}) ds + \int_{\mathcal{S}} \nabla \cdot \boldsymbol{\Theta}(\mathbf{x}) \lambda(\mathbf{x}) ds = 0 \quad \forall \boldsymbol{\Theta} \in H^1(\mathcal{S})^2, \quad (44)$$

which after integrating by part can be rewritten as

$$\int_{\mathcal{S}} \boldsymbol{\Theta}(\mathbf{x}) \cdot (\boldsymbol{\epsilon}(\mathbf{x}) - \nabla \lambda(\mathbf{x})) ds + \int_{\partial \mathcal{S}} \boldsymbol{\Theta}(\mathbf{x}) \cdot \mathbf{n} \lambda(\mathbf{x}) ds = 0 \quad \forall \boldsymbol{\Theta} \in H^1(\mathcal{S})^2, \quad (45)$$

where \mathbf{n} denote the normal to contour $\partial \mathcal{S}$. Considering first $\boldsymbol{\Theta}$ with compact support in \mathcal{S} , we see that the surface integral is null, and therefore obtain Eq. (40)

$$\boldsymbol{\epsilon}(\mathbf{x}) = \nabla \lambda(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{S}. \quad (46)$$

Varying the trace of $\boldsymbol{\Theta}$, we recover the Dirichlet boundary condition

$$\lambda(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \partial \mathcal{S}. \quad (47)$$

Plugging Eq. (46) into the constraint (43) and re-arranging, we obtain the Poisson equation

$$-\Delta \lambda(\mathbf{x}) = 1 - \det(\nabla \boldsymbol{\xi}^*(\mathbf{x})) \quad \forall \mathbf{x} \in \mathcal{S}. \quad (48)$$

The above Poisson equation completed by the Dirichlet boundary condition (47) has a unique solution, and since Eq. (46) uniquely defines $\boldsymbol{\epsilon}$ from λ , we conclude that the saddle point $(\boldsymbol{\epsilon}, \lambda)$ is unique. \square

We should note that the theorem does not state anything about the bijectivity of the final projected reference map. It is important to keep in mind that the artificial loss of bijectivity or connectivity could happen if $\mathcal{S} \subsetneq B^*$, in particular, if the deviations from the volume-preserving space are large.

To ensure that $\chi(\mathcal{S})$ and $\chi(\mathcal{B} - \mathcal{S})$ remain connected, we can redefine the Lagrangian $\mathcal{L}(\boldsymbol{\mu}, \rho)$ for $\boldsymbol{\mu} \in H_0^1(\mathcal{S})^2$, so that $\boldsymbol{\gamma}$ is the identity operator on $\partial \mathcal{S}$. In doing so, the homogeneous Dirichlet boundary condition in (39) is replaced by a homogeneous Neumann condition. Alternatively, a double correction can be done: inside and outside of the shell and enforcing continuous jump conditions across the shell. The outside correction could potentially regularize the maps away from the interface, where it is only really important that it remains smooth and bijective. Both of these strategies were considered and tested. They were found to be either less effective or to grant no clear benefits. For the remainder of the paper, we will work with Definition 1 only. In practice, even though the correction is piecewise continuous, with the proposed implementation, it is treated as a continuous function with rapid spatial variations, which effectively stitch $\chi(\mathcal{S})$ and $\chi(\mathcal{B} - \mathcal{S})$ together.

In a broader context, the exactly volume-preserving correction defined by the non-linear condition (32) can be shown to exist using the Dacorogna-Moser Theorem [11,66], a generalization of the results of Moser [35] and Banyaga [3].

4. Implementation on adaptive non-graded Oc/Quadtree grids

This section focuses on the implementation of the volume-preserving method on adaptive grids. The overall description of the method is followed by an in-depth description of each of its constituting steps and the numerical techniques they are built on.

4.1. Overall method

The general algorithm for the volume-preserving method is described in Fig. 4. Steps (1), (4), (5), (6) are the main steps of the Coupled Level Set Reference Map method, as described in section 2. Steps (2) and (3) are specific to the volume-preserving method.

(1) Advection of the reference map

Compute the intermediate reference map ξ^* by solving the advection equation

$$\begin{cases} \frac{\partial \xi}{\partial t} + \mathbf{u} \cdot \nabla \xi = 0 & \forall t \geq 0, \quad \forall \mathbf{x} \in \mathcal{B}(t), \\ \xi(t = t_n, \mathbf{x}) = \xi^n & \forall \mathbf{x} \in \mathcal{B}_0. \end{cases} \quad (49)$$

(2) Projection on the volume preserving space

Compute the adjoint λ in the shell \mathcal{S} as the solution of

$$\begin{cases} -\Delta \lambda = 1 - \det(\nabla \xi^*) & \forall \mathbf{x} \in \mathcal{S}, \\ \lambda = 0 & \forall \mathbf{x} \in \partial \mathcal{S}, \end{cases} \quad (50)$$

and compute the correction $\gamma^{-1}(\mathbf{x}) = \mathbf{x} - \nabla \lambda$.

(3) Correction

The new reference map is computed from the correction and intermediate reference map

$$\xi^{n+1}(\mathbf{x}) = \xi^*(\gamma^{-1}(\mathbf{x})), \quad (51)$$

(4) Update

We construct the updated the level set function

$$\phi^{n+1}(\mathbf{x}) = \phi_0(\xi^{n+1}(\mathbf{x})). \quad (52)$$

(5) Restarting

Evaluate the restarting criterion and perform restarting if needed.

(6) Remeshing and update

Adapt the mesh to ϕ^{n+1} and interpolate all quantities on the new mesh.

Fig. 4. Outline of the algorithm for the construction of the solution (ξ^{n+1}, ϕ^{n+1}) at time t_{n+1} from the solution (ξ^n, ϕ^n) at the previous time step t_n .

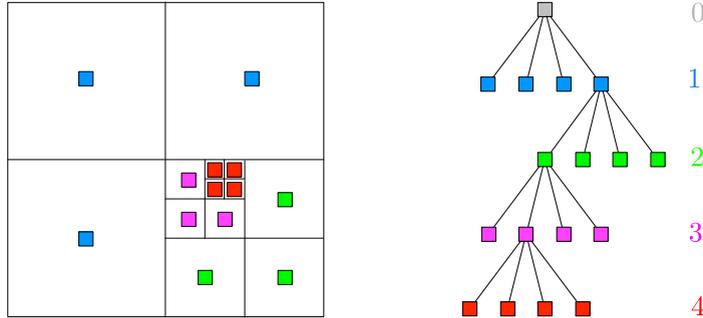


Fig. 5. Non-graded Quadtree grids construction and representation. The root cell (level 0, representing the entire domain) is splitted into four identical children cells. The splitting process is then recursively applied based on a predefined splitting criterion.

4.2. Data structure and discretizations

The computational domain is represented by an adaptive non-graded Quadtree (Octree) in 2D (resp. 3D). As Fig. 5 illustrate, the root cell (level 0) represents the entire domain. It is systematically split into four identical children cells (level 1). Then we recursively decide to split or not the newly created cells based on some predefined meshing criterion. Each time a cell of level N is split, its children are of level $N + 1$. For any given octree grid, we define by \min_{level} and \max_{level} the prescribed minimum and maximum levels of its cells. If the whole domain is a square of size $L \times L$, the coarsest and finest spatial resolution are $\Delta x_{\max} = L/2^{\min_{\text{level}}}$ and $\Delta x_{\min} = L/2^{\max_{\text{level}}}$ respectively. Because here, the level difference between neighboring cells is not bounded by 1, the trees are described as non-graded.

As it was done in previous studies [5,59], we will systematically ensure that the interface lies within a uniform band of half width $B\Delta x_{\min}$, where B is an integer larger than one. To achieve this, starting from any arbitrary tree, we recursively apply the following splitting criterion at each cell \mathcal{C}

$$\begin{aligned} \text{split } \mathcal{C} & \quad \text{if} \quad : \quad \min_{n \in \text{nodes}(\mathcal{C})} |\phi(n)| \leq B \cdot \text{Lip}(\phi) \cdot \text{diag}(\mathcal{C}) \quad \text{and} \quad \text{level}(\mathcal{C}) \leq \max_{\text{level}}, \\ \text{merge } \mathcal{C} & \quad \text{otherwise.} \end{aligned} \quad (53)$$

A cell is merged by removing all its descendants. $\text{Lip}(\phi)$ is an upper estimate of the minimal Lipschitz constant of the level set function ϕ . Since the level set used to create the mesh will be reinitialized (*i.e.* $|\nabla \phi| = 1$), we take $\text{Lip}(\phi) = 1.2$, and $\text{diag}(\mathcal{C})$ is the length of the diagonal of cell \mathcal{C} .

The reference map and level set functions are stored at the nodes of the grid. The discretizations of all differential operators will be done using the second-order finite-Differences framework proposed by Min et al. [30], which relies on the definition of third-order accurate ghost values at T-junctions. The interested reader is referred to [31,30,33,57,58] for the detailed construction and specific formulae.

4.3. Advection

The advection of the reference map is performed using a second-order accurate semi-Lagrangian scheme. At each iteration, the new reference ξ^{n+1} at any grid point \mathbf{x}^{n+1} is obtained by following the characteristic curve passing through $(t_{n+1}, \mathbf{x}^{n+1})$ to find the departure point \mathbf{x}_d^n at time t_n . Once the departure point is known, $\xi^{n+1}(\mathbf{x}^{n+1})$ is obtained by evaluating ξ^n at the departure point \mathbf{x}_d^n . Because the characteristic are unknown, the departure point must be approximated. It is done using a fourth-order Backward Runge-Kutta method (RK4):

$$\mathbf{x}_1 = \mathbf{x}^{n+1}, \quad \mathbf{x}_2 = \mathbf{x}^{n+1} - \frac{\Delta t_n}{2} \mathbf{u}^{n+\frac{1}{2}}(\mathbf{x}_1), \quad \mathbf{x}_3 = \mathbf{x}^{n+1} - \frac{\Delta t_n}{2} \mathbf{u}^{n+\frac{1}{2}}(\mathbf{x}_2), \quad \mathbf{x}_4 = \mathbf{x}^{n+1} - \Delta t_n \mathbf{u}^n(\mathbf{x}_3),$$

leading to $\mathbf{x}_d^n = \mathbf{x}^{n+1} - \frac{\Delta t_n}{6} (\mathbf{u}^{n+1}(\mathbf{x}_1) + 2\mathbf{u}^{n+\frac{1}{2}}(\mathbf{x}_2) + 2\mathbf{u}^{n+\frac{1}{2}}(\mathbf{x}_3) + \mathbf{u}^n(\mathbf{x}_4))$, where Δt_n is the time step from t_n to t_{n+1} . Since the reference map ξ^n is only known at the nodes, the quantity $\xi^n(\mathbf{x}_d^n)$ must be interpolated. Following [30], for each component ψ of ξ^n , we used the following second-order interpolation: if $\mathbf{x}_d^n = (x, y)$ is in cell \mathcal{C} , which we assume to be $[0, 1]^2$ for the sake of presentation

$$\psi(x, y) = \psi_{00}(1-x)(1-y) + \psi_{01}(1-x)y + \psi_{10}x(1-y) + \psi_{11}xy - \psi_{xx} \frac{x(1-x)}{2} - \psi_{yy} \frac{y(1-y)}{2}, \tag{54}$$

where the second-order derivatives are computed in a Weighted Essentially Non-Oscillatory (WENO [24]) fashion, e.g.

$$\psi_{xx} = \frac{1}{W_x} \sum_{n \in \text{nodes}(\mathcal{C})} \psi_{xx}^n w_x^n, \quad w_x^n = \frac{1}{|\phi_{xx}^n|^2}, \quad W_x = \sum_{n \in \text{nodes}(\mathcal{C})} w_x^n$$

where ψ_{xx}^n are the discrete second-order derivatives at node n , approximated using the finite-differences method presented in [30].

4.4. Volume-preserving projection on arbitrary shells

The Poisson system (50) in the projection step is solved using the multigrid method presented in [60], itself based on the finite-difference framework introduced by Chen et al. in [9]. As they showed, this scheme is supra-convergent, meaning that both the solution and its gradient are second-order accurate. It guarantees here that the correction $\gamma^{-1}(\mathbf{x}) = \mathbf{x} - \nabla \lambda$ is computed with second-order accuracy. The shell is defined as the band of width $|S|$ centered on the interface. Because it is by definition the set of points less than $\frac{|S|}{2}$ away from the interface, at any time step t_n , we define its corresponding level set function $\phi_S^n(\mathbf{x})$, from $\mathcal{R}(\phi^n)(\mathbf{x})$ (the reinitialized level set function at t_n) as

$$\phi_S^n(\mathbf{x}) = |\mathcal{R}(\phi^n)(\mathbf{x})| - \frac{|S|}{2}. \tag{55}$$

Because this finite difference approach requires the mesh to be uniform close to the shell interface, the projection system (50) is solved on an auxiliary mesh obtained from the current mesh by refining where the shell boundary lies. The adjoint λ is then interpolated back to the original mesh using the non-oscillatory interpolation (54). From there, the gradient $\nabla \lambda$ is calculated and used to construct the inverse correction γ . The corrected map $\xi^*(\gamma^{-1}(\mathbf{x}))$ is calculated by interpolating ξ^* at $\gamma^{-1}(\mathbf{x})$. We use again the above WENO interpolation to avoid introducing local extrema and therefore preserve the overall stability.

This step is arguably the most expensive one as it requires solving a Poisson system. In comparison, other correcting strategies such as the Coupled Level Set and Volume of Fluid method [55,54] or the Conservative Level Set method [41,42, 64,13,39,44] only require to solve conservation laws, which can be done faster and with better complexity, but incur more restrictive time step restrictions.

4.5. Reinitialization equation for the restarting procedure

Within the restarting step (4), the level set function is reinitialized by solving the reinitialization Eq. (14) with a Total Variation Diminishing second-order Runge-Kutta scheme (TVD-RK2) first presented in [53]. The Godunov Hamiltonian is defined from the first one-sided derivatives $\phi_{x^\pm}, \phi_{y^\pm}$ as

$$\mathcal{H}_{\text{GD}}(a, b, c, d) = \text{sign}(\phi^0) \cdot \begin{cases} \sqrt{\max(a^+, b^-)^2 + \max(c^+, d^-)^2} - 1 & \text{if } \text{sign}(\phi^0) \geq 0 \\ \sqrt{\max(a^-, b^+)^2 + \max(c^-, d^+)^2} - 1 & \text{if } \text{sign}(\phi^0) < 0, \end{cases}$$

Table 1

Error analysis: notations and main convergence results for the CLSRM and VP methods, with the current implementation. We allow the velocity to carry a parasitic compressible part to account for potential inaccuracies in the velocity field. The errors defined in the first column are global. The expressions in the third column are obtained under the assumption that $\Delta t \propto \Delta x$. The values in the last column are obtained under the extra assumption that the compressible part of the velocity is at least second-order (*i.e.* $V = \mathcal{O}(\Delta x^2)$).

Quantity	Notation	Expression	Value
Magnitude of the compressible part	V	$\mathcal{O}(\Delta x^\alpha)$	$\mathcal{O}(\Delta x^2)$
Discretization errors	D	$\mathcal{O}(\Delta x^2)$	$\mathcal{O}(\Delta x^2)$
Linearization errors	L	$\mathcal{O}(\Delta x^3 + \Delta x^{2+\alpha} + \Delta x^{1+2\alpha})$	$\mathcal{O}(\Delta x^3)$
Advection errors	A	$\mathcal{O}(\Delta x^2)$	
Reinitialization errors	R	$\mathcal{O}(\Delta x^2)$	
Error on interface position	E_{CLSRM}	$\mathcal{O}(A + V)$	$\mathcal{O}(\Delta x^2)$
	E_{VP}	$E_{CLSRM} + \mathcal{O}\left(\frac{A}{\Delta x} + V\right)$	$\mathcal{O}(\Delta x)$
Mass loss	M_{CLSRM}	$\mathcal{O}(A + V + R)$	$\mathcal{O}(\Delta x^2)$
	M_{VP}	$\mathcal{O}(D + L + R)$	$\mathcal{O}(\Delta x^2)$
Deviation from volume-preserving space	Δ_{CLSRM}	$\mathcal{O}\left(\frac{A}{\Delta x} + V\right)$	$\mathcal{O}(\Delta x)$
	Δ_{VP}	$\mathcal{O}(D + L)$	$\mathcal{O}(\Delta x^2)$

where $a^+ = \max(a, 0)$ and $a^- = \min(a, 0)$. The equation is then advanced in fictitious time using a RK2 scheme: we first compute ϕ^1 and ϕ^2

$$\phi^1 = \phi^n - \Delta \tau \mathcal{H}_{GD}(\phi_{x^+}^n, \phi_{x^-}^n, \phi_{y^+}^n, \phi_{y^-}^n), \quad \phi^2 = \phi^1 - \Delta \tau \mathcal{H}_{GD}(\phi_{x^+}^1, \phi_{x^-}^1, \phi_{y^+}^1, \phi_{y^-}^1)$$

and then constructing $\phi^{n+1} = \frac{\phi^2 + \phi^n}{2}$. The one-sided differences ϕ_x^\pm, ϕ_y^\pm are computed using second-order ENO discretizations. Following [30], at nodes close to the interface, the interface position is directly incorporated to the discretization using the sub-cell resolution method [51].

4.6. Remeshing and update

During step (5), the mesh is adapted to the new geometry using the criterion (53) and the new solution (ξ^{n+1}, ϕ^{n+1}) is interpolated on this new mesh using third-order weighted essentially non-oscillatory interpolations (see Eq. (54)).

5. Error analysis

In this section, we analyze the accuracy of the volume-preserving method, focusing on three errors metric: the interface position, the mass loss, and the deviations from the volume-preserving space for both the coupled level set reference map and volume-preserving methods, which will be abbreviated CLSRM and VP. Even though the results are specific to the numerical techniques employed here, the analysis is applicable to any other implementation. For the entire analysis, we will assume that the shell is wide and refined enough so that the volume-preserving correction can be correctly performed. The impact of the shell width will be quantified in the next section. We will also allow the velocity field to carry a parasitic compressible part of magnitude $\mathcal{O}(\Delta x^\alpha)$, to account for potential numerical errors in the velocity field. For the sake of simplicity, we will assume that the step size is proportional to the grid resolution (*i.e.* $\Delta t \propto \Delta x$).

All notations and results are summarized in Table 1. The main observation is that both methods converge for all three metrics. Unsurprisingly the CLSRM method is more accurate for the interface position, but the VP method has smaller deviations from the volume-preserving space. Provided that the compressible part of the velocity is small enough, the mass conservation is second-order accurate in both cases, and both methods are equally affected by the reinitialization procedure (within the restarting). For larger compressible components, only the VP preserves the total mass with second-order accuracy.

5.1. Interface position

We should start this analysis noting that the volume-preserving correction is likely to deteriorate the interface position accuracy as the additional projection step might introduce additional numerical errors. In fact, as we will now see, we can even expect to drop one order of accuracy. The local errors for both methods can be expressed in terms of the restarting operator \mathcal{R} and the exact solution ϕ_{exact}

$$e_{CLSRM}^{n+1}(\mathbf{x}) = \mathcal{R}(\phi_0(\xi^*(\mathbf{x}))) - \phi_{exact}(\mathbf{x}) \quad (56)$$

$$e_{VP}^{n+1}(\mathbf{x}) = \mathcal{R}(\phi_0(\xi^*(\mathcal{Y}^{-1}(\mathbf{x})))) - \phi_{exact}(\mathbf{x}). \quad (57)$$

Defining the restarting local error for any level set function ψ as

$$e_{\mathcal{R}}(\psi)(\mathbf{x}) = \mathcal{R}(\psi)(\mathbf{x}) - \psi(\mathbf{x}), \quad (58)$$

we can rewrite the above definitions (56), (57) as

$$e_{CLSRM}^{n+1}(\mathbf{x}) = \phi_0(\xi^*(\mathbf{x})) + e_{\mathcal{R}}(\phi_0(\xi^*(\mathbf{x}))) - \phi_{exact}(\mathbf{x}) \quad (59)$$

$$e_{VP}^{n+1}(\mathbf{x}) = \phi_0(\xi^*(\boldsymbol{\gamma}^{-1}(\mathbf{x}))) + e_{\mathcal{R}}(\phi_0(\xi^*(\boldsymbol{\gamma}^{-1}(\mathbf{x})))) - \phi_{exact}(\mathbf{x}). \quad (60)$$

Because ultimately the accuracy only matters close to the interface, we will focus our analysis to the points less than a grid cell away from the interface (i.e. $\{\mathbf{x} \in \mathcal{S} \mid |\phi(\mathbf{x})| < \Delta x\}$). We will assume that the contour Γ is smooth and well refined, so that the level set function is also smooth in this band. Using the expression for the correction $\boldsymbol{\gamma}^{-1}(\mathbf{x}) = \mathbf{x} - \nabla\lambda$ and a Taylor expansion we can express the local error for the volume preserving method as

$$E_{VP}^{n+1}(\mathbf{x}) = \phi_0(\xi^*(\mathbf{x})) + \mathcal{O}(|\nabla\lambda|) + E_{\mathcal{R}}(\phi_0(\xi^*(\boldsymbol{\gamma}^{-1}(\mathbf{x})))) - \phi_{exact}(\mathbf{x}), \quad (61)$$

which can be expressed in terms of the CLSRM error as

$$E_{VP}^{n+1}(\mathbf{x}) = E_{CLSRM}^{n+1}(\mathbf{x}) + \mathcal{O}(|\nabla\lambda|) + E_{\mathcal{R}}(\phi_0(\xi^*(\boldsymbol{\gamma}^{-1}(\mathbf{x})))) - E_{\mathcal{R}}(\phi_0(\xi^*(\mathbf{x}))). \quad (62)$$

Because the restarting step is only performed selectively (according to criterion (11)), the restarting error $e_{\mathcal{R}}(\psi)$ is not a continuous function of ψ , and therefore the last two terms in Eq. (62) cannot be expanded in Taylor series. They can only be shown to be of the same order as the reinitialization procedure, which is third-order accurate [30] close to the interface. Therefore

$$E_{VP}^{n+1}(\mathbf{x}) = E_{CLSRM}^{n+1}(\mathbf{x}) + \mathcal{O}(|\nabla\lambda|) + \mathcal{O}(\Delta x^3), \quad \forall \mathbf{x} \in \mathcal{S} \mid |\phi(\mathbf{x})| < \Delta x. \quad (63)$$

From the projection equation (50), and the fact that the Poisson solver is second-order accurate, we know that

$$\lambda = \mathcal{O}(1 - \det(\nabla\xi^*)) + \mathcal{O}(\Delta x^2). \quad (64)$$

In the limit of small deviations from the volume-preserving space, the local deviation from the VP space (i.e. $1 - \det(\nabla\xi^*)$) is the sum of the numerical error due to advection and the deviations due to the velocity field. Since the local truncation error of the advection scheme is $A^n = \mathcal{O}(\Delta x^3)$, and that one spatial order might be lost when approximating the gradient of ξ^* , the first contribution is $\mathcal{O}\left(\frac{A^n}{\Delta x}\right) = \mathcal{O}(\Delta x^2)$. The local deviations due to the velocity field are proportional to the magnitude of the parasitic velocity field ($\mathcal{O}(\Delta x^\alpha)$) times the step size ($\Delta t \propto \Delta x$). Adding the two terms we obtain

$$1 - \det(\nabla\xi^*) = \mathcal{O}\left(\Delta x^2 + \Delta x^{1+\alpha}\right). \quad (65)$$

Because the Poisson solver is supra convergent (i.e. both the solution and its gradient are second-order accurate), we also have that

$$\nabla\lambda = \mathcal{O}(\lambda) + \mathcal{O}(\Delta x^2). \quad (66)$$

Combining Eqs. (64), (65) and (66) we obtain that

$$\nabla\lambda = \mathcal{O}\left(\Delta x^2 + \Delta x^{1+\alpha}\right), \quad (67)$$

and conclude that

$$E_{VP}^{n+1}(\mathbf{x}) = E_{CLSRM}^{n+1}(\mathbf{x}) + \mathcal{O}\left(\Delta x^2 + \Delta x^{1+\alpha}\right), \quad \forall \mathbf{x} \in \mathcal{S} \mid |\phi(\mathbf{x})| < \Delta x. \quad (68)$$

The global error for the volume-preserving method is therefore bounded by

$$E_{VP}(\mathbf{x}) \leq E_{CLSRM}(\mathbf{x}) + \mathcal{O}\left(\Delta x + \Delta x^\alpha\right), \quad \forall \mathbf{x} \in \mathcal{S} \mid |\phi(\mathbf{x})| < \Delta x. \quad (69)$$

It shows that the global interface position error is expected to be larger with the VP method, which is expected to be at best first-order accurate. Meanwhile, for accurate velocity fields, the CLSRM method is $\mathcal{O}(\Delta x^2)$ as shown in [5].

5.2. Volume preservation

To quantify the volume preservation, we should look at the distance from the volume-preserving space

$$\Delta^n(\mathbf{x}) = 1 - \det \nabla \xi^n(\mathbf{x}). \quad (70)$$

With the CLSRM method, the local contribution for this error is given by Eq. (65), and thus the global deviation from the volume-preserving space is

$$\Delta_{CLSRM}(\mathbf{x}) = \mathcal{O}(\Delta x + \Delta x^\alpha). \quad (71)$$

On the other hand, with the volume-preserving method, this error is by construction zero up to the local discretization and linearization errors, D^n and L^n . The discretizations errors are introduced because the chain rule used in (30) is not preserved numerically and because in practice the reconstruction formula for the corrected map involves an interpolation. They are $\mathcal{O}(\Delta x^3)$. The linearization errors are $\mathcal{O}(|\nabla \lambda|^2)$. Using expression (65), (64), (67), and recalling that the parasitic velocity is of magnitude $\mathcal{O}(\Delta x^\alpha)$ we obtain

$$\Delta_{VP}^n(\mathbf{x}) = D^n + L^n = \mathcal{O}(\Delta x^3) + \mathcal{O}(\Delta x^4 + \Delta x^{3+\alpha} + \Delta x^{2+2\alpha}), \quad (72)$$

and so the global deviation is

$$\Delta_{VP}(\mathbf{x}) = \mathcal{O}(\Delta x^2 + \Delta x^{2+\alpha} + \Delta x^{1+2\alpha}). \quad (73)$$

As long as the parasitic part of the velocity is converging (i.e. $\alpha > 0$), the VP method will outperform the CLSRM method for this metric. For $\alpha > 0.5$, the error is dominated by the first term, the global discretization errors ($\mathcal{O}(\Delta x^2)$), and therefore, to the leading order, the deviations for the VP method are independent of the advection or velocity errors.

We should note that restarting procedure does not introduce any error for this metric since the restarted reference map satisfies exactly the volume-preserving condition. As we will see in the following subsection, it does impact the total mass conservation.

5.3. Mass conservation

The total mass variations are defined as

$$M^n = \int_{\Omega^n} d\omega_n - \int_{\Omega^0} d\omega_0, \quad (74)$$

where Ω^n is the domain contained inside Γ^n (i.e. $\phi^n < 0$). Identifying the steps at which the map was restarted as $0 = r_0^n < \dots < r_m^n \leq n$, and performing a change of variable between the geometry at step n and at the most recent restarting step r_m^n , we can rewrite the above definition as

$$M^n = \int_{\Omega_{r_m^n}^n} |\det(\nabla \chi^n)| d\omega_{r_m^n} - \int_{\Omega^0} d\omega_0. \quad (75)$$

To incorporate the impact of the restarting, we need to distinguish the solution at any r_m^n before $\xi_{r_m^-}^n$ and after $\xi_{r_m^+}^n$ and define R^m as the mass loss due to the reinitialization involved in the restarting at step r_m . The mass conservation becomes

$$M^n = \int_{\Omega_{r_m^+}^n} |\det(\nabla \chi^n)| d\omega_{r_m^+} - \int_{\Omega^0} d\omega_0 \quad (76)$$

$$= \int_{\Omega_{r_m^-}^n} |\det(\nabla \chi^n)| d\omega_{r_m^-} + R^m - \int_{\Omega^0} d\omega_0. \quad (77)$$

For the CLSRM method

$$|\det(\nabla \chi^n)| = 1 + A_{r_m}^n + V_{r_m}^n \quad (78)$$

where $A_{r_m}^n$ and $V_{r_m}^n$ are the numerical errors accumulated between $t_{r_m^n}$ and t_n due to the advection and compressibility of the velocity field. We plug (78) into (77)

$$M_{CLSRM}^n = \int_{\Omega_{r_m^-}^n} (A_{r_m}^n + V_{r_m}^n) d\omega_{r_m^-} + \int_{\Omega_{r_m^-}^n} d\omega_{r_m^-} + R^m - \int_{\Omega^0} d\omega_0, \quad (79)$$

and by recurrence obtain

$$M_{CLSRM}^n = \int_{\Omega_{r_m^n}^n} (A_{r_m}^n + V_{r_m}^n) d\omega_{r_m^n} + \sum_{r_i^n=r_0^n}^{r_{m-1}^n} \int_{\Omega_{r_i^n}^n} (A_{r_i}^{r_{i+1}} + V_{r_i}^{r_{i+1}}) d\omega_{r_i^n} + \sum_{r_i^n=r_1^n}^{r_m} R^{r_i}. \quad (80)$$

Assuming that there is no topological change, mass can only be gained or lost close to the interface where $|\phi(\mathbf{x})| < \Delta x$. We can therefore replace the domains $\Omega_{r_i^n}$ in the above integrals by the shell $S_{r_i^n}$, centered on the interface and of width $2\Delta x$:

$$M_{CLSRM}^n = \int_{S_{r_m^n}^n} (A_{r_m}^n + V_{r_m}^n) d\omega_{r_m^n} + \sum_{r_i^n=r_0^n}^{r_{m-1}^n} \int_{S_{r_i^n}^n} (A_{r_i}^{r_{i+1}} + V_{r_i}^{r_{i+1}}) d\omega_{r_i^n} + \sum_{r_i^n=r_1^n}^{r_m} R^{r_i}. \quad (81)$$

In terms of the global Advection (A), Velocity (V) and Reinitialization ($R = \mathcal{O}(\Delta x^2)$) errors, the global mass loss is

$$M_{CLSRM} = \mathcal{O}(A + V + R) = \mathcal{O}(\Delta x^2 + \Delta x^\alpha + R). \quad (82)$$

For the volume-preserving method, using the results of the previous section and to the leading

$$|\det(\nabla \chi^n)| = 1 + D^n + L^n \quad (83)$$

where D^n and L^n are the discretizations and linearization errors at time step t_n . Through a similar reasoning we obtain

$$M_{VP}^n = \int_{S_{r_m^n}^n} (D^n + L^n) d\omega_{r_m^n} + \sum_{r_i^n=r_0^n}^{r_{m-1}^n} \int_{S_{r_i^n}^n} (D^{r_{i+1}} + L^{r_{i+1}}) d\omega_{r_i^n} + \sum_{r_i^n=r_1^n}^{r_m} R^{r_i}, \quad (84)$$

and in global form

$$M_{VP} = \mathcal{O}(D + L + R) = \mathcal{O}(\Delta x^2 + \Delta x^{2+\alpha} + \Delta x^{1+2\alpha} + R) \quad (85)$$

Comparing expression (82) and (85), we notice that the mass conservation of the VP method is unaffected by the errors due to the advection process or the velocity field. These errors are replaced by discretization and linearization errors. The reinitialization procedure (within the restarting) impacts the mass conservation for both methods equally. In other words, the VP method does not correct or compensate for the reinitialization errors. For reasonably accurate velocity fields ($\alpha > 0.5$), the VP method will conserve the mass with second-order accuracy. Thus, for practical applications, where the velocity is inexact but slowly converging (*i.e.* $0.5 < \alpha < 2$) or where the geometry undergoes non-trivial transformations, we expect the VP to excel at correcting the mass loss due to the velocity field or the advection procedure. On the other hand, for highly resolved problems (high-order advection scheme and/or low polynomial order solution), or highly-accurate flows ($\alpha > 2$), the VP method may not grant better mass conservation and only deteriorate the interface accuracy.

6. Computational examples

In this section, computational examples are presented to illustrate the performances of the volume-preserving method in both two and three spatial dimensions, using either exact or approximated velocity fields. The first example, the analytic vortex, is the canonical test for advection schemes [4,49,30,19,18,34,12,5]. For the second one, the dynamic of a rising droplet, the volume-preserving method is incorporated as the interface representation within the two-phase flow solver presented in [59]. The results are validated against the experimental measurements of Mandel et al. [27] and Bhaga et al. [6]. The reader is referred to these papers for detailed descriptions of the numerical solver and the experimental techniques.

6.1. Analytic examples

6.1.1. Vortex

For this first example, we consider a disk of radius $R = 0.15$, centered at $(0.5, 0.75)$, contained in a square domain $[0, 1]^2$. Between time $t = 0$ and $t = \pi$, the disk is distorted under the incompressible velocity field

$$\mathbf{u}(x, y) = \left(-\sin^2(\pi x) \sin(2\pi y), \sin^2(\pi y) \sin(2\pi x) \right). \quad (86)$$

At $t = \pi$, the signs of the velocity components are flipped and the geometry is advected until $t = 2\pi$, at which point the geometry recovers its initial circular shape. Numerically, we set the time step as

$$\Delta t = 5 \frac{\Delta x_{\min}}{\|\mathbf{u}\|_{L^\infty(\Omega)}}, \quad (87)$$

where Δx_{\min} represent the minimum grid resolution. For all simulations we continuously enforce that the interface is surrounded by a uniform band of $10\Delta x_{\min}$, to ensure it remains highly resolved at all time. As it was done in [5], we monitor the accuracy on the interface position through the L^∞ and L^1 norms defined from the distance to the interface function $\delta(\mathbf{x})$

$$e_{L^\infty}(\phi) = \max_{\mathbf{x} \in N(\phi)} |\delta(\mathbf{x}) - \delta_{\text{exact}}(\mathbf{x})|, \quad e_{L^1}(\phi) = \int_{\Gamma_{\text{exact}}} |\delta(\mathbf{x})| dl. \quad (88)$$

The distance function is constructed numerically at the set of nodes $N(\phi)$ near the interfaces (i.e. at least one of their direct neighbors is across the interface), using standard third-order approximations. The contour integral in the above definitions and the volume integral intervening in the mass loss calculations are approximated using the second-order quadrature rules [29]. We also monitor the average distance d_{VP} to the volume-preserving space

$$d_{VP}(\xi) = \frac{1}{2\pi} \int_0^{2\pi} \int_{\Gamma_{\text{exact}}} |\det \nabla \xi - 1| dl. \quad (89)$$

Calculations are performed on Quadtree grids of increasing minimum and maximum levels. Fig. 6 illustrates the advection process and depicts the evolution of the reference map, level set function, distance from the volume-preserving space, and magnitude of the correction. The distance from the volume-preserving space is locally minimal close to the interface, where it matters the most, and where the entire algorithm is designed to be the most accurate. The correction is small as expected, but unlike the previous quantity, its spatial variations are much smoother. The correction is globally defined on the shell, while the distance from the volume-preserving space regroups local and global terms. It is, therefore, unsurprising that this second quantity displays sharper spatial variations. The interface location for both methods (see Fig. 7) are indistinguishable, but the deviations from the volume-preserving space are a few orders of magnitude smaller in the vicinity of the interface. For both methods, these deviations are fairly uniform, suggesting that mass is also lost uniformly along the interface.

The errors and estimated orders of accuracy for both the CLSRM and VP methods, and various shell widths are compiled in Table 2. In the case $|\mathcal{S}|/2 = \infty$, the shell is taken to be the entire domain. The CLSRM method is second-order accurate in all norms, as expected from our previous results [5]. The VP method converges in all norms and for all shell widths. The error on the interface position appears to be first-order accurate for the largest shells ($|\mathcal{S}|/2 = R, 2R, \infty$) and second-order accurate for the smallest shells ($|\mathcal{S}|/2 = 2\Delta x_{\min}, 5\Delta x_{\min}$). This apparent accuracy drop for large shells is consistent with the predicted first-order accuracy for the interface location (see Table 2). The mass loss and distance to the volume-preserving space are, on average, close to second-order accurate, which is also consistent with the conclusion of sections 5.2 and 5.3.

Since the velocity field is exactly incompressible, and the advection process is essentially a “back-and-forth” motion, reverting the solution and thus compensating the errors, this example falls into the category of highly resolved problems identified in section 5 for which the CLSRM excels, and the VP method does not confer any advantage. Indeed the mass loss and errors on the interface position are similar or better with the CLSRM method. The distance to the volume-preserving space is systematically smaller with the VP method.

Closer inspection indicates that all four errors are almost identical for the CLSRM method and the two thinnest shells ($|\mathcal{S}|/2 = 2\Delta x_{\min}, 5\Delta x_{\min}$). These shells should be interpreted as too narrow to resolve the volume-preserving projection. On the other end, for the largest shell ($|\mathcal{S}|/2 = \infty$), the accuracy on the interface position is the most deteriorated. It is unsurprising since these are the cases where the correction is expected to be the largest and also constructed with the most under resolved information (far away from the interface). These larger corrections are also associated with smaller distances to the volume-preserving space.

Since the grid is only refined close to the interface, it should be noted that the local resolution will not necessarily be divided by two between any two refinement iteration; hence the orders for all four metrics are systematically underestimated. These estimations get worse as the shell width increases, as more under resolved information gets incorporated in the construction of the projection. As the last column of Table 2 illustrates, this effect is noticeable on the convergence of the distance to the volume-preserving space ($d_{VP}(\xi)$). The restarting frequency could also affect the apparent convergence of this metric. Every time the reference map is restarted, it becomes exactly volume-preserving. Therefore a higher restarting frequency will produce a smaller average distance to the volume-preserving space. Since the restarting criterion is triggered by numerical errors, this frequency will be higher on coarser grids. It will decrease as the resolution increases, slowing down the apparent convergence.

6.1.2. Artificial expansion

This example focuses on the ability of the VP method to correct the mass loss due to inaccuracy in the velocity field. To do so, we retain the same setup as in the previous example and change the velocity to be a uniformly and constant radial expansion

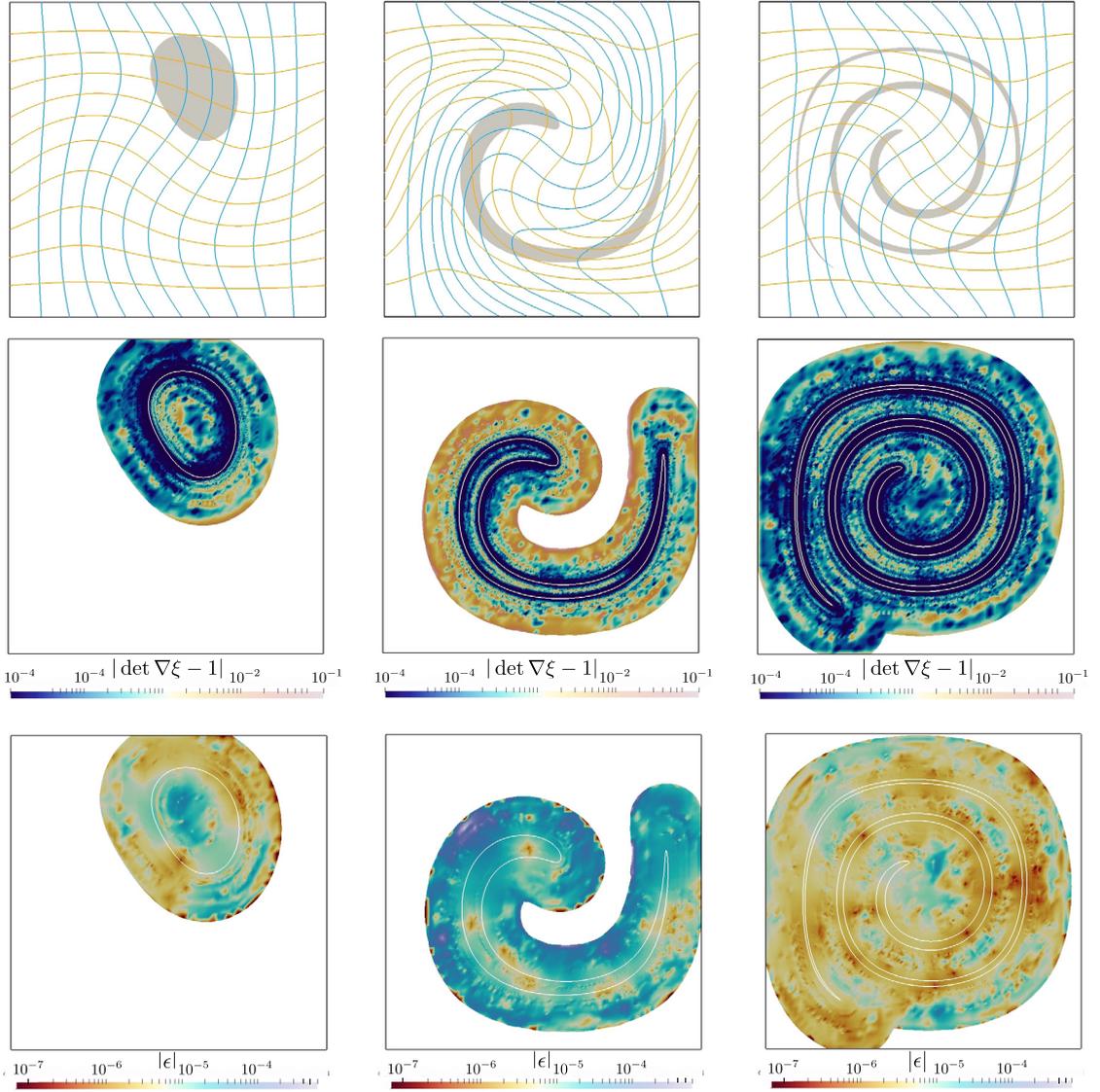


Fig. 6. Volume-preserving method for the analytic vortex example. (top) Snapshots of the geometry (in grey) and iso-contours of the x (blue) and y (orange) components of the reference map. (middle) Deviations from the volume-preserving space in the shell. The interface is represented by a solid white line. (bottom) Magnitude of the correction ϵ . Calculations were performed on a Quadtree with $\max_{\text{level}} = 9$, $\min_{\text{level}} = 5$, and using a shell of constant width $|S|/2 = R$. Snapshots are taken at time $t = 0.097, 1.11$ and π .

$$\mathbf{u}(x, y) = 0.1 \cdot \Delta x^\alpha \frac{\mathbf{r}}{|\mathbf{r}|}, \quad \mathbf{r} = (x - 0.5, y - 0.5), \quad \alpha \in [0.01, 1], \quad \forall t \in [0, 2\pi], \quad (90)$$

which will cause the domain to expand. As the resolution increases, this expansion will diminish ($\alpha > 0$), and we expect the final domain to converge to be the initial one. Table 3 displays the resulting mass variation using both the VP and CLSRM methods and considering five different values of α . With the CLSRM method, the mass loss converges at the rate α as it should. With the VP method, the mass loss is systematically orders of magnitude smaller. As predicted from expression (85), it is second-order for $\alpha > 0.5$, and progressively drops to first-order as α goes to zero. For the smallest α values, the domain artificially expands by over 200% as the CLSRM results show, yet the VP method keeps the mass variations around 1% on the four finest grids.

6.1.3. Topological change: split in two

For this last analytic example, we center the disk at $(0.5, 0.5)$ and advect it under the asymmetric splitting velocity

$$\mathbf{u}(x, y) = \begin{cases} (0.1, 0) & \text{if } y > 0.4 \\ (-0.1, 0) & \text{elsewhere} \end{cases}, \quad (91)$$

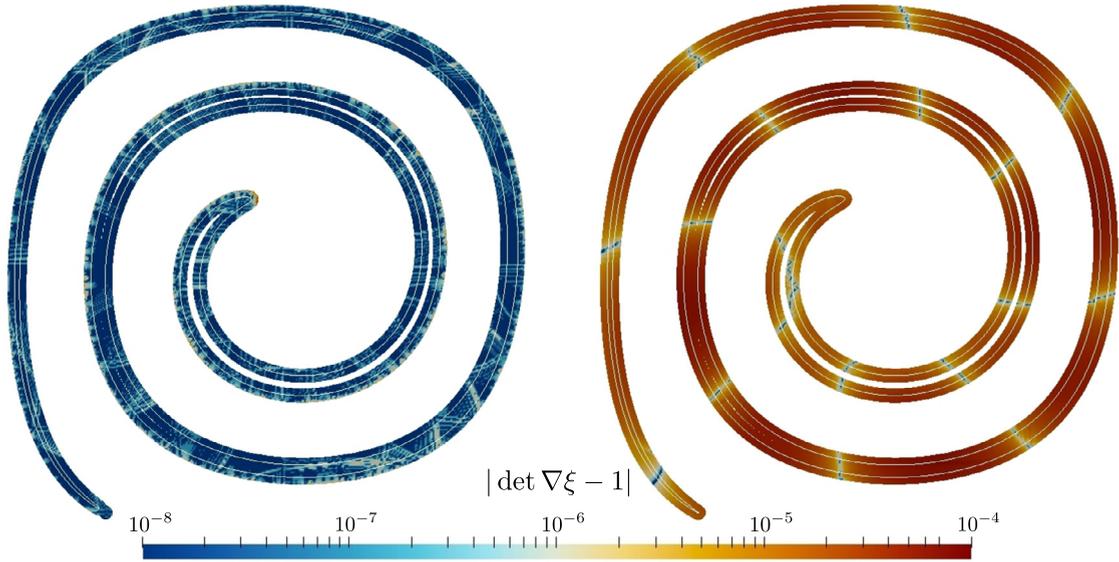


Fig. 7. Interface and deviations from the volume-preserving space close to the interface and a time $t = \pi$, using either the VP (left) or the CLSRM (right) method. Calculations were performed using the same parameters as in Fig. 6.

and over the time interval $[0, \pi]$. At the final time, we compare the numerical solution to the exact one given by the discontinuous equivalent of the initial circular level set function

$$\phi_{exact}(x, y) = \begin{cases} \phi_0(x - 0.1\pi, y) & \text{if } y > 0.4 \\ \phi_0(x + 0.1\pi, y) & \text{elsewhere} \end{cases} \quad (92)$$

The convergence results for the interface position and mass loss are given in Table 4. Because the velocity field is discontinuous, the reference map is always discontinuous, and the restarting criterion is triggered at every single iteration. This cause the distance to the volume-preserving space to be identically zero since the restarted reference map is the identity transformation, which is exactly volume-preserving. For this reason it is not reported.

Additionally, the discontinuity of the solution triggers the limiters in the interpolation and reinitialization procedures, and therefore the interface location and the mass loss for the CLSRM method drop to first-order accurate. Surprisingly, in L^1 -norm, the interface location remains second-order accurate. With the VP method, in both norms the interface location is only first-order accurate, as the analysis predicted in the continuous case. The mass loss with either method is almost identical (first-order). The VP method is about two to three times slower because of the Poisson solver.

6.2. Rising droplet

For this second example we simulate an oil droplet (Ω_d) rising in water (Ω_w). The fluids are assumed to have uniform densities ρ_d, ρ_w , dynamic viscosities μ_d, μ_w and surface tension γ . The velocities ($\mathbf{u}_d, \mathbf{u}_w$) and pressure (p_d, p_w) fields are solutions of the incompressible Navier-Stokes equations

$$\begin{cases} \rho_i \left(\frac{\partial \mathbf{u}_i}{\partial t} + \mathbf{u}_i \cdot \nabla \mathbf{u}_i \right) = -\nabla p_i + \mu_i \Delta \mathbf{u}_i + \rho_i \mathbf{g} & \forall \mathbf{x} \in \Omega_i \setminus \Gamma \\ \nabla \cdot \mathbf{u}_i = 0 & \forall \mathbf{x} \in \Omega_i \setminus \Gamma, \end{cases} \quad (93)$$

subjected to the interface jump conditions

$$\begin{cases} \llbracket \mathbf{u} \rrbracket = 0 & \forall \mathbf{x} \in \Gamma \\ \llbracket \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \mathbf{n} - p \mathbf{n} \rrbracket = \sigma \kappa \mathbf{n} & \forall \mathbf{x} \in \Gamma, \end{cases} \quad (94)$$

where the jump operator is defined for any quantity ψ as $\llbracket \psi \rrbracket = \psi^d - \psi^w$, \mathbf{n} is the normal vector to the fluids interface, and κ is its curvature. Homogeneous Dirichlet and Neumann boundary conditions are enforced on the wall of the computational domain for the velocity and pressure field respectively.

To compare the performances of the new volume-preserving method with the traditional level set and coupled methods, we compute the numerical solution of the above system (93), (94) using all three interface representations within the incompressible two-phase flow solver [59]. In each case, we monitor the tip velocity (measured as the ascending velocity at the tip of the droplet), the relative mass loss, the relative jump error e_j^n

Table 2

Vortex example: error table for the analytic using either the Coupled Level set Reference Map method of the new volume-preserving method, and various shell widths. Results are indexed by their maximum and minimum levels ($\max_{\text{level}} : \min_{\text{level}}$).

CLSRM								
Levels	$e_{L^\infty}(\phi)$	Order	$e_{L^1}(\phi)$	Order	Mass loss (%)	Order	$d_{VP}(\xi)$	Order
6:2	$2.43 \cdot 10^{-2}$	-	$7.77 \cdot 10^{-4}$	-	$3.87 \cdot 10^1$	-	$1.28 \cdot 10^{-2}$	-
7:3	$1.30 \cdot 10^{-2}$	0.90	$1.39 \cdot 10^{-3}$	-0.83	$6.63 \cdot 10^0$	2.55	$2.92 \cdot 10^{-3}$	2.13
8:4	$6.90 \cdot 10^{-3}$	0.91	$6.19 \cdot 10^{-4}$	1.17	$7.92 \cdot 10^{-1}$	3.06	$6.43 \cdot 10^{-4}$	2.18
9:5	$1.91 \cdot 10^{-3}$	1.85	$1.34 \cdot 10^{-4}$	2.21	$1.20 \cdot 10^{-1}$	2.72	$1.51 \cdot 10^{-4}$	2.09
10:6	$5.15 \cdot 10^{-4}$	1.89	$2.28 \cdot 10^{-5}$	2.55	$1.80 \cdot 10^{-2}$	2.73	$3.83 \cdot 10^{-5}$	1.98
11:7	$1.22 \cdot 10^{-4}$	2.08	$3.57 \cdot 10^{-6}$	2.67	$2.71 \cdot 10^{-3}$	2.73	$1.76 \cdot 10^{-5}$	1.12
12:8	$3.69 \cdot 10^{-5}$	1.72	$7.12 \cdot 10^{-7}$	2.32	$4.83 \cdot 10^{-4}$	2.49	$2.29 \cdot 10^{-5}$	-0.37
average		1.56		1.68		2.71		1.52
VP using $ S /2 = 2\Delta x_{\min}$								
Levels	$e_{L^\infty}(\phi)$	Order	$e_{L^1}(\phi)$	Order	Mass loss (%)	Order	$d_{VP}(\xi)$	Order
6:2	$1.62 \cdot 10^{-2}$	-	$4.57 \cdot 10^{-4}$	-	$4.83 \cdot 10^1$	-	$5.00 \cdot 10^{-3}$	-
7:3	$1.29 \cdot 10^{-2}$	0.32	$1.43 \cdot 10^{-3}$	-1.64	$7.27 \cdot 10^0$	2.73	$1.01 \cdot 10^{-3}$	2.30
8:4	$6.90 \cdot 10^{-3}$	0.91	$6.11 \cdot 10^{-4}$	1.22	$8.15 \cdot 10^{-1}$	3.16	$2.07 \cdot 10^{-4}$	2.29
9:5	$1.91 \cdot 10^{-3}$	1.85	$1.33 \cdot 10^{-4}$	2.20	$1.20 \cdot 10^{-1}$	2.76	$4.27 \cdot 10^{-5}$	2.28
10:6	$5.28 \cdot 10^{-4}$	1.86	$2.30 \cdot 10^{-5}$	2.53	$1.87 \cdot 10^{-2}$	2.69	$1.00 \cdot 10^{-5}$	2.09
11:7	$1.22 \cdot 10^{-4}$	2.12	$3.58 \cdot 10^{-6}$	2.68	$2.71 \cdot 10^{-3}$	2.79	$6.79 \cdot 10^{-6}$	0.56
12:8	$3.69 \cdot 10^{-5}$	1.72	$8.19 \cdot 10^{-7}$	2.13	$4.72 \cdot 10^{-4}$	2.52	$1.23 \cdot 10^{-5}$	-0.84
average		1.46		1.52		2.77		1.44
VP using $ S /2 = 5\Delta x_{\min}$								
Levels	$e_{L^\infty}(\phi)$	Order	$e_{L^1}(\phi)$	Order	Mass loss (%)	Order	$d_{VP}(\xi)$	Order
6:2	$2.44 \cdot 10^{-2}$	-	$1.01 \cdot 10^{-3}$	-	$5.01 \cdot 10^1$	-	$5.99 \cdot 10^{-4}$	-
7:3	$1.35 \cdot 10^{-2}$	0.86	$1.75 \cdot 10^{-3}$	-0.78	$5.34 \cdot 10^0$	3.23	$1.36 \cdot 10^{-4}$	2.14
8:4	$6.90 \cdot 10^{-3}$	0.96	$6.22 \cdot 10^{-4}$	1.49	$9.73 \cdot 10^{-1}$	2.46	$2.56 \cdot 10^{-5}$	2.41
9:5	$1.92 \cdot 10^{-3}$	1.85	$1.34 \cdot 10^{-4}$	2.21	$1.29 \cdot 10^{-1}$	2.92	$5.61 \cdot 10^{-6}$	2.19
10:6	$5.29 \cdot 10^{-4}$	1.86	$2.36 \cdot 10^{-5}$	2.51	$1.91 \cdot 10^{-2}$	2.75	$1.60 \cdot 10^{-6}$	1.81
11:7	$1.22 \cdot 10^{-4}$	2.12	$4.29 \cdot 10^{-6}$	2.46	$2.65 \cdot 10^{-3}$	2.84	$1.65 \cdot 10^{-6}$	-0.43
12:8	$3.68 \cdot 10^{-5}$	1.73	$1.85 \cdot 10^{-6}$	1.21	$2.74 \cdot 10^{-4}$	3.28	$2.78 \cdot 10^{-6}$	-0.75
average		1.56		1.21		2.91		1.23
VP using $ S /2 = R$								
Levels	$e_{L^\infty}(\phi)$	Order	$e_{L^1}(\phi)$	Order	Mass loss (%)	Order	$d_{VP}(\xi)$	Order
6:2	$2.13 \cdot 10^{-2}$	-	$1.73 \cdot 10^{-3}$	-	$5.04 \cdot 10^1$	-	$2.38 \cdot 10^{-4}$	-
7:3	$1.23 \cdot 10^{-2}$	0.79	$1.35 \cdot 10^{-3}$	0.36	$8.34 \cdot 10^0$	2.60	$5.25 \cdot 10^{-5}$	2.18
8:4	$6.51 \cdot 10^{-3}$	0.92	$9.15 \cdot 10^{-4}$	0.56	$1.18 \cdot 10^0$	2.82	$9.34 \cdot 10^{-6}$	2.49
9:5	$3.56 \cdot 10^{-3}$	0.87	$3.56 \cdot 10^{-4}$	1.36	$2.34 \cdot 10^{-1}$	2.33	$1.89 \cdot 10^{-6}$	2.30
10:6	$1.80 \cdot 10^{-3}$	0.98	$1.66 \cdot 10^{-4}$	1.10	$1.84 \cdot 10^{-2}$	3.66	$6.65 \cdot 10^{-7}$	1.51
11:7	$8.75 \cdot 10^{-4}$	1.04	$7.76 \cdot 10^{-5}$	1.10	$5.63 \cdot 10^{-2}$	-1.61	$4.03 \cdot 10^{-7}$	0.72
12:8	$4.74 \cdot 10^{-4}$	0.88	$3.45 \cdot 10^{-5}$	1.17	$3.97 \cdot 10^{-2}$	0.51	$2.84 \cdot 10^{-7}$	0.51
average		0.92		0.94		1.72		1.38
VP using $ S /2 = 2R$								
Levels	$e_{L^\infty}(\phi)$	Order	$e_{L^1}(\phi)$	Order	Mass loss (%)	Order	$d_{VP}(\xi)$	Order
6:2	$2.29 \cdot 10^{-2}$	-	$1.92 \cdot 10^{-3}$	-	$5.14 \cdot 10^1$	-	$2.38 \cdot 10^{-4}$	-
7:3	$1.48 \cdot 10^{-2}$	0.63	$1.03 \cdot 10^{-3}$	0.89	$8.10 \cdot 10^0$	2.67	$5.26 \cdot 10^{-5}$	2.18
8:4	$7.06 \cdot 10^{-3}$	1.07	$4.62 \cdot 10^{-4}$	1.16	$1.19 \cdot 10^0$	2.76	$9.50 \cdot 10^{-6}$	2.47
9:5	$3.32 \cdot 10^{-3}$	1.09	$1.99 \cdot 10^{-4}$	1.21	$2.31 \cdot 10^{-1}$	2.37	$1.87 \cdot 10^{-6}$	2.34
10:6	$1.69 \cdot 10^{-3}$	0.97	$7.61 \cdot 10^{-5}$	1.39	$6.59 \cdot 10^{-2}$	1.81	$6.58 \cdot 10^{-7}$	1.51
11:7	$8.24 \cdot 10^{-4}$	1.04	$4.15 \cdot 10^{-5}$	0.88	$5.54 \cdot 10^{-2}$	0.25	$4.01 \cdot 10^{-7}$	-0.75
12:8	$4.62 \cdot 10^{-4}$	0.83	$2.49 \cdot 10^{-5}$	0.74	$3.85 \cdot 10^{-2}$	0.52	$2.84 \cdot 10^{-7}$	0.50
average		0.94		1.04		1.73		1.37
VP using $ S /2 = \infty$								
Levels	$e_{L^\infty}(\phi)$	Order	$e_{L^1}(\phi)$	Order	Mass loss (%)	Order	$d_{VP}(\xi)$	Order
6:2	$2.44 \cdot 10^{-2}$	-	$2.07 \cdot 10^{-3}$	-	$5.03 \cdot 10^1$	-	$2.39 \cdot 10^{-4}$	-
7:3	$1.31 \cdot 10^{-2}$	0.89	$1.12 \cdot 10^{-3}$	0.85	$7.97 \cdot 10^0$	2.66	$5.27 \cdot 10^{-5}$	2.18
8:4	$6.15 \cdot 10^{-3}$	1.09	$4.04 \cdot 10^{-4}$	1.47	$1.21 \cdot 10^0$	2.71	$9.53 \cdot 10^{-6}$	2.47
9:5	$3.38 \cdot 10^{-3}$	0.86	$4.82 \cdot 10^{-5}$	3.07	$1.71 \cdot 10^{-1}$	2.83	$1.87 \cdot 10^{-6}$	2.35
10:6	$1.81 \cdot 10^{-3}$	0.89	$3.33 \cdot 10^{-5}$	0.53	$6.24 \cdot 10^{-2}$	1.45	$6.55 \cdot 10^{-7}$	1.51
11:7	$8.69 \cdot 10^{-4}$	1.06	$4.44 \cdot 10^{-5}$	-0.41	$5.60 \cdot 10^{-2}$	0.15	$4.00 \cdot 10^{-7}$	0.73
12:8	$4.25 \cdot 10^{-4}$	1.03	$2.33 \cdot 10^{-5}$	0.93	$4.08 \cdot 10^{-2}$	0.46	$2.83 \cdot 10^{-7}$	0.49
average		0.97		1.07		1.71		1.62

Table 3

Artificial expansion example: comparison between the VP and CLSRM methods, for various perturbation magnitudes ($\Delta x^{1.5}$, Δx^1 , $\Delta x^{0.6}$, $\Delta x^{0.25}$, $\Delta x^{0.1}$). Results are indexed by their maximum and minimum levels (\max_{level} : \min_{level}).

VP using $|S|/2 = R$

Levels	$\Delta x^{1.5}$		Δx^1		$\Delta x^{0.6}$		$\Delta x^{0.25}$		$\Delta x^{0.1}$	
	Mass loss (%)	Order								
6:2	$1.02 \cdot 10^{-2}$	-	$9.92 \cdot 10^{-2}$	-	$2.69 \cdot 10^{-2}$	-	$4.75 \cdot 10^1$	-	$1.45 \cdot 10^2$	-
7:3	$1.25 \cdot 10^{-3}$	3.16	$4.39 \cdot 10^{-3}$	2.46	$2.11 \cdot 10^{-3}$	3.67	$1.35 \cdot 10^1$	1.82	$1.48 \cdot 10^2$	-0.03
8:4	$1.87 \cdot 10^{-4}$	2.74	$1.79 \cdot 10^{-4}$	1.80	$9.83 \cdot 10^{-4}$	1.10	$1.68 \cdot 10^0$	3.01	$1.15 \cdot 10^2$	0.36
9:5	$3.83 \cdot 10^{-5}$	2.29	$1.27 \cdot 10^{-4}$	0.53	$1.63 \cdot 10^{-3}$	-0.73	$1.68 \cdot 10^{-1}$	3.31	$5.41 \cdot 10^{-1}$	4.77
10:6	$8.17 \cdot 10^{-5}$	2.23	$6.58 \cdot 10^{-4}$	1.12	$7.93 \cdot 10^{-4}$	1.04	$2.18 \cdot 10^{-2}$	2.95	$5.31 \cdot 10^{-1}$	0.02
11:7	$4.23 \cdot 10^{-5}$	4.27	$1.37 \cdot 10^{-7}$	9.32	$4.43 \cdot 10^{-5}$	4.16	$2.76 \cdot 10^{-1}$	-3.67	$1.12 \cdot 10^0$	-1.07
12:8	$6.68 \cdot 10^{-6}$	2.66	$1.02 \cdot 10^{-6}$	-1.96	$8.00 \cdot 10^{-6}$	2.47	$2.98 \cdot 10^{-2}$	3.21	$2.37 \cdot 10^{-1}$	2.24
average		2.89		2.22		1.95		1.77		1.04

CLSRM

Levels	$\Delta x^{1.5}$		Δx^1		$\Delta x^{0.6}$		$\Delta x^{0.25}$		$\Delta x^{0.1}$	
	Mass loss (%)	Order	Mass loss (%)	Order	Mass loss (%)	Order	Mass loss (%)	Order	Mass loss (%)	Order
6:2	$8.10 \cdot 10^{-1}$	-	$6.65 \cdot 10^1$	-	$3.76 \cdot 10^1$	-	$2.03 \cdot 10^2$	-	$4.68 \cdot 10^2$	-
7:3	$2.89 \cdot 10^{-1}$	1.50	$3.30 \cdot 10^1$	1.01	$2.41 \cdot 10^1$	0.64	$1.63 \cdot 10^2$	0.31	$4.24 \cdot 10^2$	0.14
8:4	$1.02 \cdot 10^{-1}$	1.50	$1.64 \cdot 10^1$	1.01	$1.56 \cdot 10^1$	0.63	$1.32 \cdot 10^2$	0.30	$3.85 \cdot 10^2$	0.13
9:5	$3.61 \cdot 10^{-2}$	1.50	$8.20 \cdot 10^{-1}$	1.00	$1.02 \cdot 10^1$	0.62	$1.07 \cdot 10^2$	0.30	$3.50 \cdot 10^2$	0.13
10:6	$1.28 \cdot 10^{-2}$	1.50	$4.09 \cdot 10^{-1}$	1.00	$6.65 \cdot 10^0$	0.61	$8.78 \cdot 10^1$	0.29	$3.19 \cdot 10^2$	0.13
11:7	$4.52 \cdot 10^{-3}$	1.50	$2.05 \cdot 10^{-1}$	1.00	$4.36 \cdot 10^0$	0.61	$7.20 \cdot 10^1$	0.28	$2.91 \cdot 10^2$	0.13
12:8	$1.60 \cdot 10^{-3}$	1.50	$1.02 \cdot 10^{-1}$	1.00	$2.87 \cdot 10^0$	0.61	$5.92 \cdot 10^1$	0.28	$2.65 \cdot 10^2$	0.13
average		1.50		1.00		0.62		0.29		0.13

Table 4

Split in two example: error table and computational times using either the CLSRM or VP method. As for previous examples, results are indexed by their maximum and minimum levels.

CLSRM

Levels	$e_{L\infty}(\phi)$	Order	$e_{L^1}(\phi)$	Order	Mass loss (%)	Order	Time (s)
6:2	$1.78 \cdot 10^{-2}$	-	$1.01 \cdot 10^{-3}$	-	$4.59 \cdot 10^0$	-	$1.72 \cdot 10^{-1}$
7:3	$7.54 \cdot 10^{-3}$	1.24	$3.26 \cdot 10^{-4}$	1.64	$1.93 \cdot 10^0$	1.25	$7.95 \cdot 10^{-1}$
8:4	$3.27 \cdot 10^{-3}$	1.21	$6.53 \cdot 10^{-5}$	2.32	$8.51 \cdot 10^{-1}$	1.18	$2.95 \cdot 10^0$
9:5	$1.95 \cdot 10^{-3}$	0.74	$2.19 \cdot 10^{-5}$	1.58	$4.12 \cdot 10^{-1}$	1.05	$1.21 \cdot 10^1$
10:6	$1.19 \cdot 10^{-3}$	0.72	$5.68 \cdot 10^{-6}$	1.95	$2.05 \cdot 10^{-1}$	1.01	$4.87 \cdot 10^1$
11:7	$4.33 \cdot 10^{-4}$	1.46	$1.70 \cdot 10^{-6}$	1.74	$1.01 \cdot 10^{-1}$	1.02	$2.12 \cdot 10^2$
12:8	$2.52 \cdot 10^{-4}$	0.78	$4.05 \cdot 10^{-7}$	2.07	$5.00 \cdot 10^{-2}$	1.02	$1.00 \cdot 10^3$
average		1.02		1.88		1.09	

VP using $|S|/2 = R$

Levels	$e_{L\infty}(\phi)$	Order	$e_{L^1}(\phi)$	Order	Mass loss (%)	Order	Time (s)
6:2	$1.78 \cdot 10^{-2}$	-	$1.02 \cdot 10^{-3}$	-	$4.59 \cdot 10^0$	-	$2.48 \cdot 10^{-1}$
7:3	$1.25 \cdot 10^{-2}$	0.51	$3.95 \cdot 10^{-4}$	1.36	$1.90 \cdot 10^0$	1.28	$1.25 \cdot 10^0$
8:4	$4.99 \cdot 10^{-3}$	1.33	$2.30 \cdot 10^{-4}$	0.78	$9.92 \cdot 10^{-1}$	0.94	$7.01 \cdot 10^0$
9:5	$2.59 \cdot 10^{-3}$	0.94	$2.53 \cdot 10^{-4}$	-0.14	$4.01 \cdot 10^{-1}$	1.31	$3.43 \cdot 10^1$
10:6	$1.70 \cdot 10^{-3}$	0.61	$1.18 \cdot 10^{-4}$	1.11	$2.25 \cdot 10^{-1}$	0.83	$1.50 \cdot 10^2$
11:7	$9.47 \cdot 10^{-4}$	0.84	$3.88 \cdot 10^{-5}$	1.60	$1.19 \cdot 10^{-1}$	0.91	$6.15 \cdot 10^2$
12:8	$4.33 \cdot 10^{-4}$	1.13	$3.07 \cdot 10^{-5}$	0.34	$3.29 \cdot 10^{-2}$	1.85	$2.55 \cdot 10^3$
average		0.89		0.84		1.18	

$$e_f^n = 2 \cdot \frac{\int_{\Gamma} (||\mathbf{u}_d^n - \mathbf{u}_w^n||)}{\int_{\Gamma} (||\mathbf{u}_d^n|| + ||\mathbf{u}_w^n||)}, \tag{95}$$

and the relative error on the tip velocity (measured as the relative absolute difference between the tip velocity and the rising velocity obtained by numerically differentiating the tip position as a function of time). The first metric will be compared to experimental measurements for validation, while the other three will guide our convergence study.

6.2.1. Oil droplet

We begin our analysis by considering the experimental setup described in [27], defining

$$\mu_d = 0.098 \text{ g} \cdot \text{cm}^{-1} \cdot \text{s}^{-1}, \quad \mu_w = 0.01 \text{ g} \cdot \text{cm}^{-1} \cdot \text{s}^{-1}, \quad \rho_w = 1.117 \text{ g} \cdot \text{cm}^{-3}, \tag{96}$$

and setting the surface tension to $\sigma = 30 \text{ g} \cdot \text{s}^{-2}$. The whole computational domain Ω is taken to a cube of side 10 cm. The droplet is initially centered in the xy -plane at a height of twice its diameter. The gravity is acting in the negative z direction. We consider the lower fluid, in case B on page 10 of [27], for which the droplet density, diameter and asymptotic rising

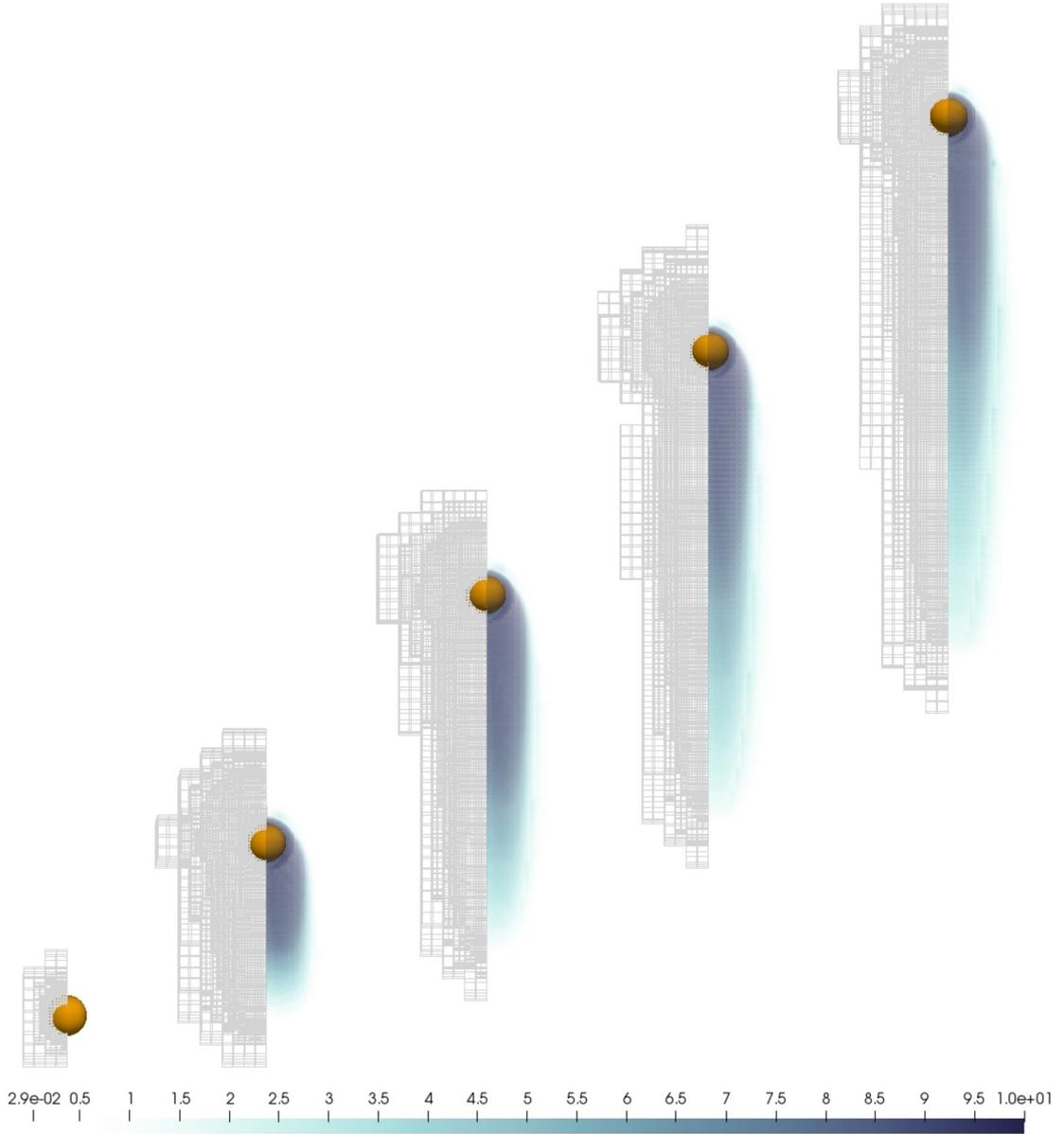


Fig. 8. Oil droplet: snapshots of the vorticity magnitude profile and adaptive grid. Calculations were performed on an Octree with $\text{max}_{\text{level}} = 9$, using the volume-preserving method and a shell of size $|S|/2 = 2R$.

velocity are

$$\rho_d = 0.9927 \text{ g} \cdot \text{cm}^{-3}, \quad d = 0.28 \text{ cm}, \quad U = 6.25 \text{ cm} \cdot \text{s}^{-1}, \quad g = 9.81 \text{ cm} \cdot \text{s}^{-2}. \tag{97}$$

The time step Δt is chosen such that

$$\Delta t < \Delta x \cdot \min \left(\frac{1}{\|\mathbf{u}\|_{L^\infty}}, 10 \cdot \frac{\min(\mu^-, \mu^+)}{\sigma} \left(1 + \sqrt{1 + \frac{(\rho^+ + \rho^-)\sigma \Delta x}{4\pi \min(\mu^-, \mu^+)^2}} \right) \right), \tag{98}$$

which the stability condition proposed in [59] with a larger multiplicative coefficient (10 instead of 1) in front of the second term between parenthesis. The minimum Octree level is set to 3, and the maximum level varies from 6 to 9. The final time is set to $t_f = 1$ s.

The characteristic flow evolution is depicted in Fig. 8. As we observe, the droplet retains its spherical symmetry, and the wake remains stable, which is in good qualitative agreement with the shadowgraph observations of Mandel et al. [27]. The temporal evolution of all four metrics and for each method is depicted in Fig. 9. The case $|S|/2 = \infty$ is the only one

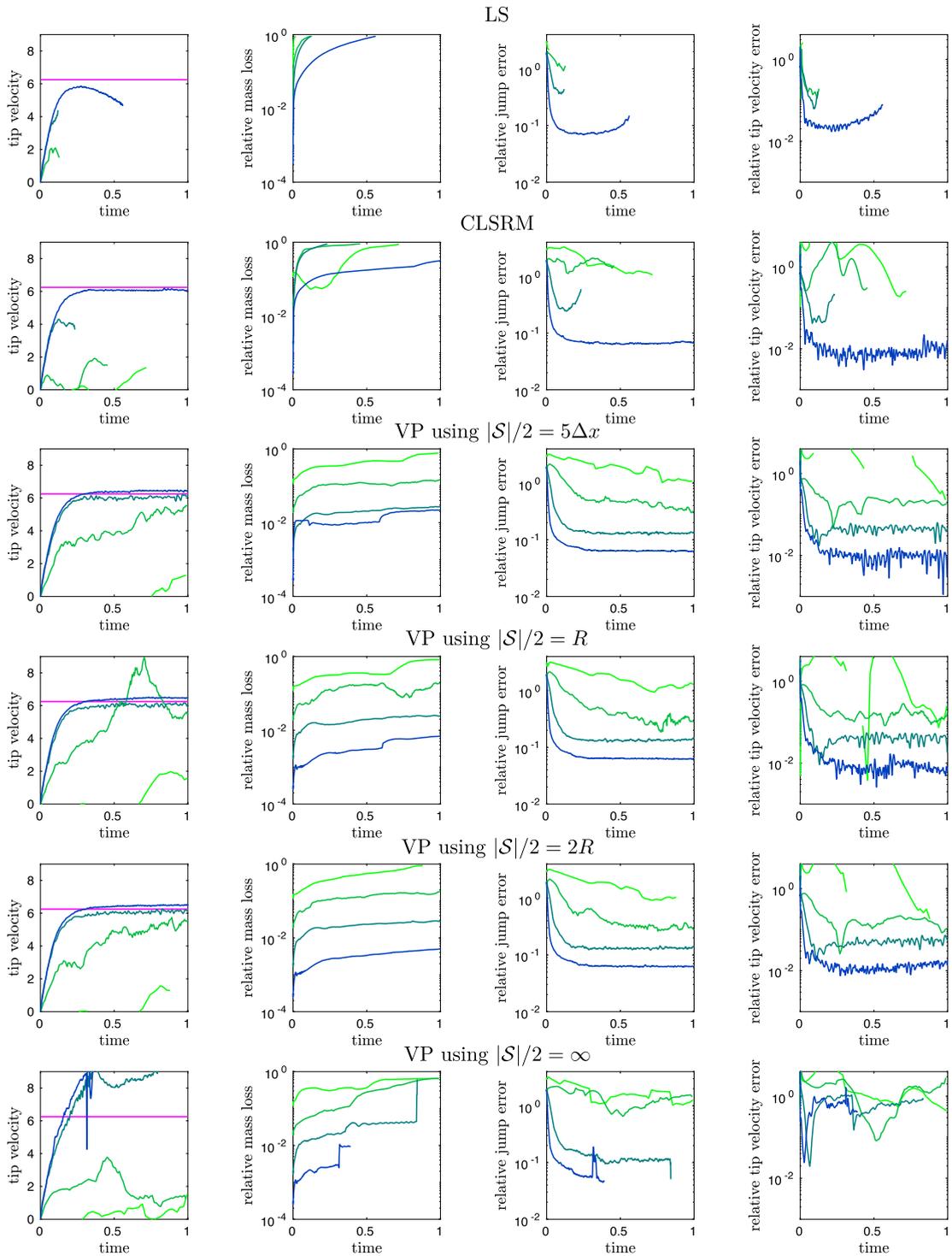


Fig. 9. Oil droplet example: comparison of the standard level set (LS), coupled level set and reference map (CLSRM), and new volume-preserving (VP) methods for increasing maximum grid resolution. Lines are colored by the maximum tree level: the coarsest simulations ($\max_{\text{level}} = 6$) correspond to the bright green curves, while the finest ones ($\max_{\text{level}} = 9$) are drawn in dark blue. The tip velocity is compared to the experimental measurements of Mandel et al. [27] (pink solid line).

where the simulation is not converging as the maximum resolution increases. In this case, because the correction is done on the entire domain, including large areas with coarse resolution, the correction is non-converging. In all other cases (on which the remainder of this discussion will concentrate), all four metrics are converging. In particular, the limit asymptotic tip velocity is in excellent agreement with the experimental observation ($U = 6.25 \text{ cm} \cdot \text{s}^{-1}$).

With the standard level set method, the droplet systematically loses its entire mass before the final time. With the coupled method, the mass losses are reduced; nonetheless, for the lowest resolutions ($\max_{\text{level}} = 6, 7, 8$) the droplet vanishes before the end of the simulation. With the volume-preserving method the droplet persists until the final time (except for $\max_{\text{level}} = 6$ and $|\mathcal{S}|/2 = R$ where it disappears around $t = 0.9$). On the finest grid ($\max_{\text{level}} = 9$), there are about two orders of magnitude between the final relative mass loss with the CLSRM ($3.2 \cdot 10^{-1}$) and the most efficient volume-preserving method ($6.8 \cdot 10^{-3}$ and $4.7 \cdot 10^{-3}$ for the shells of size R and $2R$ respectively). In these last cases, the final relative mass loss is slightly better than second-order accurate (2.26 and 2.52 respectively). For the smallest shell ($|\mathcal{S}|/2 = 5\Delta x$), coarse grids ($\max_{\text{level}} = 6, 7, 8$) provide similar results but further refinement does not improve the mass loss. We interpret this convergence stagnation as the evidence that the shell is too thin to adequately resolve the correction. The interface representation method has no remarkable impact on the convergence of the relative jump and tip velocity errors, which are both systematically converging.

The ideal shell appears to be of width R or $2R$, in other words just large enough that it contains the entire domain $\Omega(t)$ enclosed by the interface $\Gamma(t)$. It echoes back to the initial remark that the correction is only guaranteed to conserve mass if $\Omega(t) \subset \mathcal{S}$. Using unnecessary large shells may only pollute the results with non-local and potentially under resolved information.

6.2.2. Non-spherical droplets: Bhaga and Weber's experiment

For this last example, we reproduce some of the experimental observations of Bhaga and Weber [6]. The general setup is the same as for the oil droplet example, only the fluid parameters differ. As it was done in [59], from the Eotvos (EO), Morton (Mo), and Reynolds (Re) numbers

$$EO = \frac{gd^2\rho_w}{\sigma} \quad Mo = \frac{g\mu_w^4}{\rho_w\sigma^3} \quad Re = \frac{\rho_w dU}{\mu_w}, \quad (99)$$

we define the fluids parameters by setting the rising velocity U and undeformed diameter d to 1:

$$\rho_w = 1 \quad \frac{\rho_w}{\rho_d} = 10^3 \quad \mu_w = \frac{\rho_w}{Re} \quad \frac{\mu_w}{\mu_d} = 10^2 \quad \sigma = \frac{\mu_w^2}{\rho_w} \sqrt{\frac{EO}{Mo}} \quad g = \frac{Mo\rho_w\sigma^3}{\mu_w^4}. \quad (100)$$

Doing so, we expect the asymptotic rising velocity to be 1. The whole computational domain is taken as the cube of length $L = 32$, and the initial spherical droplet is placed at a height $H = 4$. For each example, the final time is chosen larger than the characteristic period T_σ and decay time τ_σ of the capillary oscillations. These two quantities are estimated from the first term of Lamb's harmonic expansion [23] of the linearized solution for an oscillating droplet as

$$T_\sigma = \sqrt{\frac{d^3\pi^2(3\rho_w + 2\rho_d)}{48\sigma}}, \quad \tau_\sigma = \frac{d^2\rho_w}{20\mu_w}. \quad (101)$$

The grid maximum and minimum levels are set to 11 and 3 respectively, and we focus on the examples (b-d) presented in Figure 2 in [6]. From the conclusions of the above example, the shell half width is set to be equal to the droplet initial radius ($|\mathcal{S}|/2 = d/2$).

The final simulated droplet shapes (see Fig. 10) are in very good qualitative agreement with the original experimental observations and previous numerical studies [56,59]. In all three cases, the final rising velocity (see Table 5) is close to the expected value ($U = 1$). The mass loss is smaller than what was reported in [59], despite the time step and final times being both two to five times larger in the current study. They range from 0.5% for the less deformed droplet (case b), to 6.3% for the most elongated example (case d). These observations illustrate the ability of the volume-preserving method to improve mass conservation without sacrificing the interface's accuracy, making it a method of choice for the simulation of incompressible multiphase flow.

7. Conclusions

The volume-preserving reference-map method is an extension of the coupled reference map method, designed to reduce the artificial mass loss. It introduces an additional step to project the reference map on the space of volume-preserving diffeomorphisms. In the limit of small numerical error, this projection can be shown to be uniquely defined from the solution of a Poisson problem with Dirichlet boundary conditions. This projection can be performed on the entire domain or only on a small shell containing the interface, which is particularly advantageous for problems where the mesh is only adapted to the interface location. Independently of the shell selection, the method converges as the local spatial resolution goes to zero.

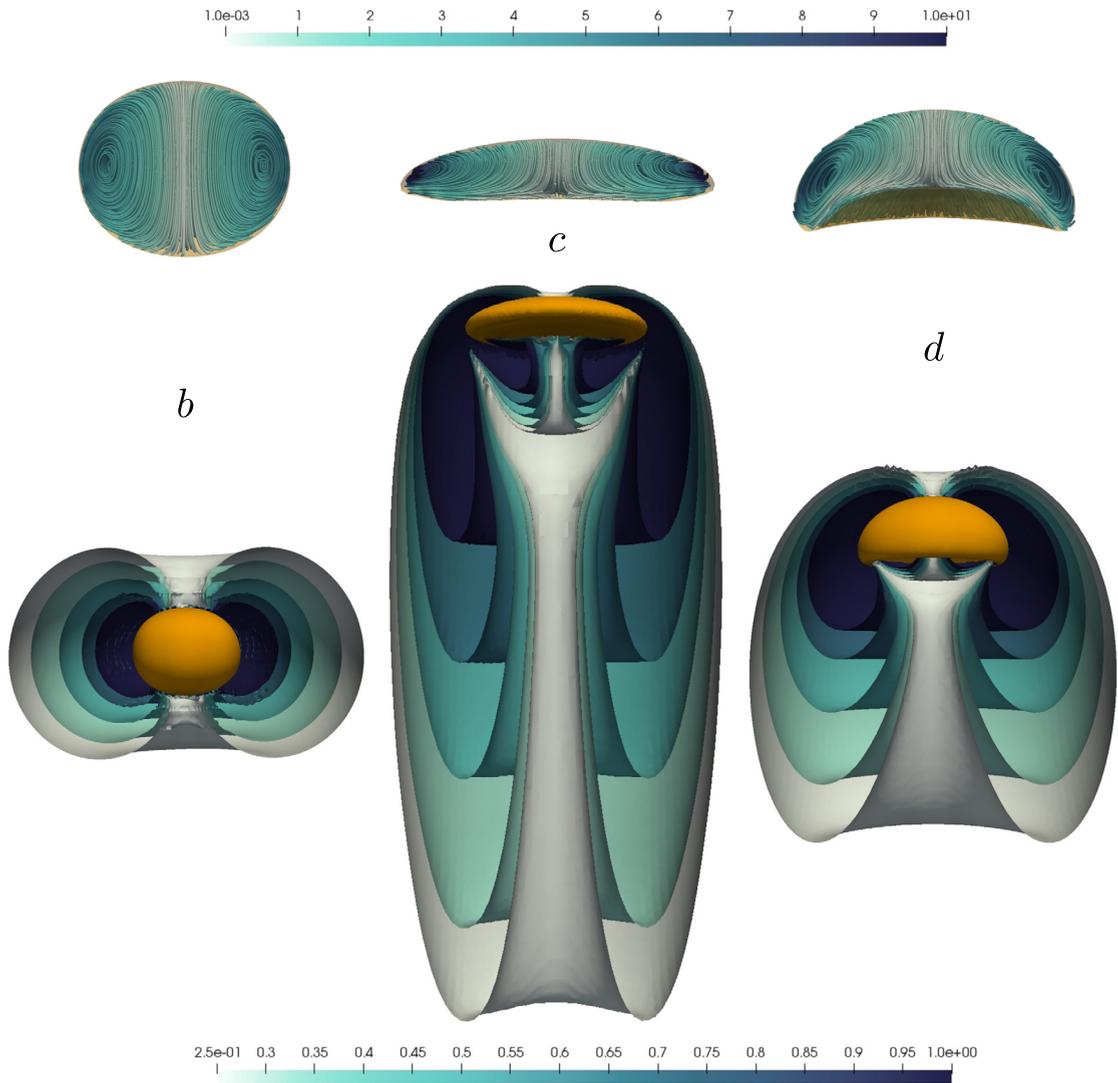


Fig. 10. Non-spherical rising droplets: final droplet shape (orange), inner apparent velocity profile (top), and outer vorticity isocontours (bottom). Both the streamlines and isocontours are colored by the magnitude of the vorticity.

Table 5
 Non-spherical rising droplets: dimensionless numbers, characteristic times, asymptotic rising velocity, and mass loss for all three examples.

	b	c	d
Eotvos number	17.7	711	0.232
Morton number	32.2	8.2×10^{-4}	55.3
Reynolds number	243	266	7.7
Oscillation period	0.45	3.08	6.24
Decay time	0.012	2.76	0.38
Final time	5	20	10
Final Rising velocity	1.21	1.10	0.95
Mass loss (%)	0.5	6.3	1.6

This extra step introduces additional errors on the interface position and computational costs, which are counterbalanced by possible large improvements on the mass conservation. The method should, therefore, be used only when pertinent. For simple problems, such as the analytic vortex presented here, where the velocity is analytically divergence-free, the volume-preserving method does not grant any advantage and only deteriorates the interface accuracy. Additionally, the projection step, which requires solving a Poisson system, becomes the most time-consuming part of the overall algorithm. For more practical applications, such as the rising oil droplet example, where the interface velocity is inexact and might present

spurious compressible components, the volume-preserving method surpasses the CLSRM method, reducing the mass loss by two orders of magnitude while having an imperceptible influence on the interface location. In this case, the cost of the projection step is derisory in comparison to the overall computational cost.

Though the construction of the method allows the projection to be performed on any arbitrary shell, in practice, we observed that the shell thickness has an important impact on the numerical results. An excessively wide shell might include under-resolved areas and therefore produce an inaccurate or non-converging correction. On the other hand, a too thin shell might not resolve the projection correctly, making it ineffective. From the presented computational exploration, a shell just large enough to contain the entire enclosed domain appears to be the optimal strategy.

While the present study focuses on the motion of interfaces under area-preserving velocity fields, this new framework can be extended to a wealth of constrained advection problem for which a reference map can be defined, such as the advection of a passive quantity under an incompressible velocity field, the simulation of phase change flows or constrained shape optimization problems.

CRedit authorship contribution statement

Maxime Theillard created the method, implemented it, ran all examples and prepared the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The author acknowledges computing time on the Multi-Environment Computer for Exploration and Discovery (MERCED) cluster at the University of California, Merced, which was funded by National Science Foundation Grant No. ACI-1429783.

References

- [1] D. Adalsteinsson, J.A. Sethian, Transport and diffusion of material quantities on propagating interfaces via level set methods, *J. Comput. Phys.* 185 (1) (2003) 271–288.
- [2] L. Anumolu, M.F. Trujillo, Gradient augmented level set method for phase change simulations, *J. Comput. Phys.* 353 (2018) 377–406.
- [3] A. Banyaga, Formes-volume sur les variétés à bord, *Enseign. Math.* (2) 20 (1974) 127–131.
- [4] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 85 (2) (1989) 257–283.
- [5] T. Bellotti, M. Theillard, A coupled level-set and reference map method for interface representation with applications to two-phase flows simulation, *J. Comput. Phys.* 392 (2019) 266–290.
- [6] D. Bhaga, M. Weber, Bubbles in viscous liquids: shapes, wakes and velocities, *J. Fluid Mech.* 105 (1981) 61–85.
- [7] F. Boudon, J. Chopard, O. Ali, B. Gilles, O. Hamant, A. Boudaoud, J. Traas, C. Godin, A computational framework for 3D mechanical modeling of plant morphogenesis with cellular resolution, *PLoS Comput. Biol.* 11 (1) (01 2015) 1–16.
- [8] E. Brun, A. Guittet, F. Gibou, A local level-set method using a hash table data structure, *J. Comput. Phys.* 231 (6) (Mar. 2012) 2528–2536.
- [9] H. Chen, C. Min, F. Gibou, A supra-convergent finite difference scheme for the Poisson and heat equations on irregular domains and non-graded adaptive Cartesian grids, *J. Sci. Comput.* 31 (1) (May 2007) 19–60.
- [10] G.-H. Cottet, E. Maitre, T. Milcent, Eulerian formulation and level set models for incompressible fluid-structure interaction, *ESAIM: Math. Model. Numer. Anal.* 42 (3) (2008) 471–492.
- [11] B. Dacorogna, J. Moser, On a partial differential equation involving the Jacobian determinant, *Ann. Inst. Henri Poincaré, Anal. Non Linéaire* 7 (1) (1990) 1–26.
- [12] C.C. de Langavant, A. Guittet, M. Theillard, F. Temprano-Coleto, F. Gibou, Level-set simulations of soluble surfactant driven flows, *J. Comput. Phys.* 348 (2017) 271–297.
- [13] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, *J. Comput. Phys.* 227 (18) (2008) 8395–8416.
- [14] Z. Fu, S. Yakovlev, R.M. Kirby, R.T. Whitaker, Fast parallel solver for the levelset equations on unstructured meshes, *Concurr. Comput., Pract. Exp.* 27 (7) (2015) 1639–1657.
- [15] F. Gibou, R. Fedkiw, S. Osher, A review of level-set methods and some recent applications, *J. Comput. Phys.* 353 (2018) 82–109.
- [16] J. Glimm, J. Grove, X. Li, K. Shyue, Y. Zeng, Q. Zhang, Three-dimensional front tracking, *SIAM J. Sci. Comput.* 19 (3) (1998) 703–727.
- [17] J. Glimm, E. Isaacson, D. Marchesin, O. McBryan, Front tracking for hyperbolic systems, *Adv. Appl. Math.* 2 (1) (1981) 91–119.
- [18] 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.
- [19] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, *J. Comput. Phys.* 227 (4) (2008) 2674–2706.
- [20] S.S. Jain, K. Kamrin, A. Mani, A conservative and non-dissipative Eulerian formulation for the simulation of soft solids in fluids, *J. Comput. Phys.* 399 (2019) 108922.
- [21] K. Kamrin, C.H. Rycroft, J.-C. Nave, Reference map technique for finite-strain elasticity and fluid–solid interaction, *J. Mech. Phys. Solids* 60 (11) (2012) 1952–1969.
- [22] A. Karakus, T. Warburton, M. Aksel, C. Sert, A GPU-accelerated adaptive discontinuous Galerkin method for level set equation, *Int. J. Comput. Fluid Dyn.* 30 (1) (2016) 56–68.
- [23] H. Lamb, *Hydrodynamics*, Cambridge University Press, 1932.
- [24] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, *J. Comput. Phys.* 115 (1) (1994) 200–212.
- [25] F. Losasso, R. Fedkiw, S. Osher, Spatially adaptive techniques for level set methods and incompressible flow, *Comput. Fluids* 35 (10) (2006) 995–1010.

- [26] E. Maitre, T. Milcent, G.-H. Cottet, A. Raoult, Y. Usson, Applications of level set methods in computational biophysics, *Math. Comput. Model.* 49 (11–12) (2009) 2161–2169.
- [27] T. Mandel, D.Z. Zhou, L. Waldrop, M. Theillard, D. Kleckner, S. Khatri, Retention of rising droplets in density stratification, *Phys. Rev. Fluids* (2020).
- [28] O. Mercier, X.-Y. Yin, J.-C. Nave, The characteristic mapping method for the linear advection of arbitrary sets, *SIAM J. Sci. Comput.* 42 (3) (2020) A1663–A1685.
- [29] C. Min, F. Gibou, Geometric integration over irregular domains with application to level-set methods, *J. Comput. Phys.* 226 (2007) 1432–1443.
- [30] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive Cartesian grids, *J. Comput. Phys.* 225 (1) (2007) 300–321.
- [31] C. Min, F. Gibou, H.D. Ceniceros, A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids, *J. Comput. Phys.* 218 (1) (2006) 123–140.
- [32] M. Mirzadeh, A. Guittet, C. Burstedde, F. Gibou, Parallel level-set methods on adaptive tree-based grids, *J. Comput. Phys.* 322 (2016) 345–364.
- [33] M. Mirzadeh, M. Theillard, F. Gibou, A second-order discretization of the nonlinear Poisson–Boltzmann equation over irregular geometries using non-graded adaptive Cartesian grids, *J. Comput. Phys.* 230 (5) (2011) 2125–2140.
- [34] N.R. Morgan, J.I. Waltz, 3D level set methods for evolving fronts on tetrahedral meshes with adaptive mesh refinement, *J. Comput. Phys.* 336 (2017) 492–512.
- [35] J. Moser, On the volume elements on a manifold, *Trans. Am. Math. Soc.* 120 (2) (1965) 286–294.
- [36] J.-C. Nave, R.R. Rosales, B. Seibold, A gradient-augmented level set method with an optimally local, coherent advection scheme, *J. Comput. Phys.* 229 (10) (2010) 3802–3827.
- [37] L.C. Ngo, H.G. Choi, A multi-level adaptive mesh refinement method for level set simulations of multiphase flow on unstructured meshes, *Int. J. Numer. Methods Eng.* 110 (10) (2017) 947–971.
- [38] J. Nocedal, S. Wright, *Numerical Optimization*, Springer, 2000.
- [39] T. Nonomura, K. Kitamura, K. Fujii, A simple interface sharpening technique with a hyperbolic tangent function applied to compressible two-fluid modeling, *J. Comput. Phys.* 258 (2014) 95–117.
- [40] S.A. Ocko, A. Heyde, L. Mahadevan, Morphogenesis of termite mounds, *Proc. Natl. Acad. Sci.* 116 (9) (2019) 3379–3384.
- [41] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, *J. Comput. Phys.* 210 (1) (2005) 225–246.
- [42] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow II, *J. Comput. Phys.* 225 (1) (2007) 785–807.
- [43] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* 79 (1) (1988) 12–49.
- [44] M. Owkes, O. Desjardins, A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows, *J. Comput. Phys.* 249 (C) (Sept. 2013) 275–302.
- [45] J.-P. Pons, G. Hermosillo, R. Keriven, O. Faugeras, Maintaining the point correspondence in the level set framework, *J. Comput. Phys.* 220 (1) (2006) 339–354.
- [46] P. Rasetarinera, M. Hussaini, An efficient implicit discontinuous spectral Galerkin method, *J. Comput. Phys.* 172 (2) (2001) 718–738.
- [47] J.-F. Remacle, N. Chevaugnon, E. Marchandise, C. Geuzaine, Efficient visualization of high-order finite elements, *Int. J. Numer. Methods Eng.* 69 (4) (2007) 750–771.
- [48] R.D. Richtmyer, K.W. Morton, Difference methods for initial-value problems, *SIAM Rev.* 10 (3) (1968) 381–383.
- [49] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, *J. Comput. Phys.* 141 (2) (1998) 112–152.
- [50] W. Rudin, *Functional Analysis*, International Series in Pure and Applied Mathematics, 1991.
- [51] G. Russo, P. Smereka, A remark on computing distance functions, *J. Comput. Phys.* 163 (1) (2000) 51–67.
- [52] R. Saye, Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction, and free surface flow: part I, *J. Comput. Phys.* 344 (2017) 647–682.
- [53] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* 77 (2) (1988) 439–471.
- [54] M. Sussman, A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles, *J. Comput. Phys.* 187 (1) (2003) 110–136.
- [55] 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 (2) (2000) 301–337.
- [56] M. Sussman, K. Smith, M. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, *J. Comput. Phys.* 221 (2) (2007) 469–505.
- [57] M. Theillard, L.F. Djodjom, J.-L. Vié, F. Gibou, A second-order sharp numerical method for solving the linear elasticity equations on irregular domains and adaptive grids—application to shape optimization, *J. Comput. Phys.* 233 (2013) 430–448.
- [58] M. Theillard, F. Gibou, T. Pollock, A sharp computational method for the simulation of the solidification of binary alloys, *J. Sci. Comput.* 63 (2) (2015) 330–354.
- [59] M. Theillard, F. Gibou, D. Saintillan, Sharp numerical simulation of incompressible two-phase flows, *J. Comput. Phys.* 391 (2019) 91–118.
- [60] M. Theillard, C.H. Rycroft, F. Gibou, A multigrid method on non-graded adaptive octree and quadtree Cartesian grids, *J. Sci. Comput.* 55 (1) (Apr 2013) 1–15.
- [61] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, *J. Comput. Phys.* 169 (2) (2001) 708–759.
- [62] B. Valkov, C.H. Rycroft, K. Kamrin, Eulerian method for fluid–structure interaction and submerged solid–solid contact problems, *J. Appl. Mech.* (2015).
- [63] Z. Wang, Z. Wang, The level set method on adaptive Cartesian grid for interface capturing, in: *42nd AIAA Aerospace Sciences Meeting and Exhibit*, 2004.
- [64] F. Xiao, Y. Honma, T. Kono, A simple algebraic interface capturing scheme using hyperbolic tangent function, *Int. J. Numer. Methods Fluids* 48 (9) (2005) 1023–1040.
- [65] J.-J. Xu, H.-K. Zhao, An Eulerian formulation for solving partial differential equations along a moving interface, *J. Sci. Comput.* 19 (1) (Dec 2003) 573–594.
- [66] D. Ye, Prescribing the Jacobian determinant in Sobolev spaces, *Ann. Inst. Henri Poincaré, Anal. Non Linéaire* 11 (3) (1994) 275–296.
- [67] X.-Y. Yin, O. Mercier, B. Yadav, K. Schneider, J.-C. Nave, A characteristic mapping method for the two-dimensional incompressible Euler equations, *J. Comput. Phys.* 424 (2021) 109781.
- [68] Q. Zhang, Fourth- and higher-order interface tracking via mapping and adjusting regular semianalytic sets represented by cubic splines, *SIAM J. Sci. Comput.* 40 (6) (2018) A3755–A3788.