Geometrical level set reinitialization using closest points method and kink detection for thin filaments, topology changes and two-phase flows

Félix Henri, Mathieu Coquerelle, Pierre Lubin

Geometrical level set reinitialization using closest point method and kink detection for thin filaments, topology changes and two-phase flows

Félix Henri*, Mathieu Coquerelle*, Pierre Lubin*

Bordeaux INP, I2M, UMR 5295, F-33400 Talence, France.

Abstract

We introduce a robust and high order strategy to perform the reinitialization in a level set framework. The reinitialization by closest points (RCP) method is based on geometric considerations. It relies on a gradient descent to find the closest points at the interface in order to solve the Eikonal equation and thus reinitializing the level set field. Furthermore, a new algorithm, also based on a similar geometric approach, is introduced to detect precisely all the ill-defined points of the level set. These points, also referred to as kinks, can mislead the gradient descent and more widely impact the accuracy of level set methods. This algorithm, coupled with the precise computation of the closest points of the interface, permits the novel method to be robust and accurate when performing the reinitialization every time step after solving the advection equation. Furthermore, they both require very few given parameters with the advantage of being based on a geometrical approach and independent of the application. The proposed method was tested on various benchmarks, and demonstrated equivalent or even better results compared to solving the Hamilton-Jacobi equation.

Keywords: geometrical approach, level set, reinitialization, closest points, kink detection, medial axis

*Corresponding Author

Email addresses: felix.henri@u-bordeaux.fr (Félix Henri), mathieu.coquerelle@bordeaux-inp.fr (Mathieu Coquerelle), pierre.lubin@bordeaux-inp.fr (Pierre Lubin)

Highlights

- A geometric approach to reinitialize the level set function based on a gradient descent.
- Applicable every time step after transporting the level set field.
- Accurate detection of *kink points* of the level set field.
- Both algorithms rely on very few given constant geometrical criteria, independent of the application.
- Robust and accurate in 2D and 3D, from simple advection to two-phase flow subjected to surface tension.

1. Introduction

The accurate representation of a geometrical surface and its motion is essential in many applications. Simulation of two-phase flows requires adequate numerical methods for localizing precisely each phase and their interactions at the interface such as buoyancy forces and surface tension. The dynamics of such interfaces can be relatively complex as they are subjected to high velocity gradients, shear and consequently topology changes.

For this purpose, the Level Set Methods [1] (LSM) are largely used to capture evolving interfaces which are implicitly represented through a scalar field ϕ, usually defined as a signed distance function, where the surface is defined as its zero level set. Its simplicity of implementation and its robustness are the key advantages of this method. Furthermore, LSM captures naturally topological changes without necessitating explicit treatment of connection or disconnection of the surface. Transporting ϕ, regardless of the chosen numerical method, will lead to significant distortion of this field. To ensure accurate computation of quantities linked to the interface such as the curvature and the normal, or the induced volume fraction of each fluid, it is essential to perform a reinitialization process of the level set field. Several approaches exist, among them the fast marching method [2] and fast sweeping method [3] are based on an iterative process for finding values of ϕ starting from the interface position, however they suffer from a lack of precision. Another common approach consists in solving iteratively the Hamilton-Jacobi partial differential equation [4–6] whose stationary solution is a signed distance function. Combined with well suited numerical schemes, this method can give high order results.

On the other hand, applying this process too frequently will introduce undesirable displacement on the interface position, a problem which can be reduced with modifications such as [6–8]. However, as we will see in Sec. 6.8, this methodology can also introduce large errors on the interface dynamic when topological changes occur. Hence, to reduce the displacement of the interface, it is common to perform this reinitialization procedure after a few interface advection steps. Furthermore, this approach is sensitive to the choice of how frequently it is applied. Too often it can lead to important deformation and diffusion of the surface, too rarely the level set function will eventually be too heavily distorted. The choice of this reinitialization frequency has seen no consensus in the literature and usually depends on the underlying application.

The present work introduces a new robust strategy following a geometric approach to perform the reinitialization procedure that can be applied to various applications. This method is primarily based on the minimal distance, also named *closest point*, to the surface which can be computed thanks to a gradient descent. Chopp [9] used this principle to improve the fast marching method of Sethian [2], where the closest points are computed through a modified Newton's method. Later, Anumolu [10] applied a hybrid reinitialization process with the closest point approach for the cells cutting the surface and solved the Hamilton-Jacobi equation further. However, when topology changes, large distortions or under-resolved zones eventually arise after the transport of the level set. Those ill-defined regions will eventually disturb the accuracy of the gradient descent. Those regions, also referred to as *kinks* [11], need to be detected and treated adequately for the robustness of the method.

We extend these closest point reinitialization approaches in an algorithm that fully takes advantage of the gradient descent and is applied as well for cells close and far from the surface, thus granting a better accuracy for the overall level set methods. One of the main advantages of this method is that it can be safely applied every time after transporting the level set without compromising the underlying interface dynamics. Conjointly, a new algorithm is also introduced to detect the kinks.

The performance of the method will be tested on a series of benchmarks. First, we study the cases of pure advection of various surfaces and assess the capabilities of the method on simple and complex geometries with thin layers, under-resolved regions or discontinuities. Then, we will consider test cases coupled with Navier-Stokes equations with surface tension to demonstrate the capacity of the method to capture accurately the behaviour of inviscid two-phase flows. Finally, to test robustness and applicability, we apply the proposed method to the bubble rise and the dam break problem in 2D and 3D.

2. The Level Set Method coupled with Navier-Stokes equations

2.1. Level set definition

Consider a spatial domain Ω, composed by two subdomains Ω^- and Ω^+ separated by an interface Γ. In a level set framework, Γ is represented implicitly by a scalar function $\phi : \Omega \to \mathbf{R}$ which is commonly defined as a signed distance function:

$$\phi(\mathbf{x}) = \begin{cases} -\text{dist}(\mathbf{x}, \Gamma) \text{ if } \mathbf{x} \in \Omega^- \\ +\text{dist}(\mathbf{x}, \Gamma) \text{ if } \mathbf{x} \in \Omega^+ \end{cases} \quad \text{with} \quad \Gamma = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x}) = 0\}.$$

where $\text{dist}(\mathbf{x}, \Gamma)$ is the Euclidean distance of $\mathbf{x}$ to the interface, defined by the zero level set of ϕ. Furthermore, in that particular case, ϕ is solution of the eikonal equation :

$$|\nabla \phi| = 1. \tag{1}$$

2.2. Navier-Stokes equations for incompressible two-phase flows

We considered the incompressible form of the Navier-Stokes equation where the momentum equation can be written in a conservative form as:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot (2\mu \mathbf{D}) + \mathbf{f} \tag{2}$$

where $\mathbf{u}$ is the fluid velocity, ρ its density, μ its dynamic viscosity, p the pressure, $\mathbf{D} = (\nabla \mathbf{u} + \nabla^{\mathrm{T}} \mathbf{u})/2$ is the deformation tensor and $\mathbf{f}$ encompasses external body forces. Under the assumption of incompressibility, the continuity equation reduces to a divergence-free constraint on the velocity field:

$$\nabla \cdot \mathbf{u} = 0. \tag{3}$$

In the case of an immiscible two-phase flow simulation, a discontinuity in density and viscosity stands at the interface Γ. This discontinuity is numerically treated following the one fluid model, first introduced in [12]. In the case where the density and viscosity are constant within each phase, ρ and μ are expressed as:

$$\begin{aligned} \rho(\mathbf{x}) &= \rho_2 + (\rho_1 - \rho_2) c(\mathbf{x}) \\ \mu(\mathbf{x}) &= \mu_2 + (\mu_1 - \mu_2) c(\mathbf{x}) \end{aligned} \tag{4}$$

where ρ_1 (resp. ρ_2) and μ_1 (resp. μ_2) are the values of the first (resp. second) phase and c a characteristic function.

Level set coupling. In a level set framework, c is expressed as a function of ϕ. For the designated two-phase flows with high density ratio in a one-fluid model, the regularity of this transition between phases is important for stability reasons. Hence, in order to obtain a smooth representation of the interfacial region, a regularized form of the Heaviside function is used herein:

$$c(\mathbf{x}) = H_\epsilon(\phi(\mathbf{x})) \tag{5}$$

with

$$H_\epsilon(\phi) = \begin{cases} 0 & \text{if} \quad \phi < \epsilon \\ \frac{1}{2}(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin(\pi\frac{\phi}{\epsilon})) & \text{if} \quad |\phi| \leq \epsilon. \\ 1 & \text{if} \quad \phi > \epsilon \end{cases} \tag{6}$$

Therefore, the density and viscosity vary smoothly within an interfacial region of thickness 2ϵ, where usually ϵ is proportional to the cell size h, i.e. $\epsilon = O(h)$. For different applications, other approximations could indeed be used, such as sharper approaches as proposed in [**?**].

The interface thickness problem. In the case where ϕ is a signed distance function and $\epsilon = kh$, with k a real number, the interfacial region is of thickness $2kh$. Ensuring this thickness remains constant is a crucial criterion for the accuracy of the one fluid model and makes the reinitialization of the level set function an essential matter. Hence, performing the reinitialization step systematically after transporting the level set field is essential to maintain a constant interface thickness.

Surface tension model

Surface tension can be modeled as a pressure jump across the interface, from a volume point of view, as $\mathbf{f}_\sigma = \sigma\kappa_\Gamma\mathbf{n}_\Gamma\delta_\Gamma$, where σ is the surface tension coefficient, κ_Γ the curvature of the surface and $\mathbf{n}_\Gamma$ its normal and δ_Γ is the Dirac function associated with the surface. Brackbill et al. [13] introduced the Continuum Surface Force (CSF) which approximates this body force as:

$$\sigma\kappa\mathbf{n}\delta_\Gamma \simeq \sigma\kappa\nabla c. \tag{7}$$

Using the level set formulation, the normal $\mathbf{n}$ to the interface and the associated mean curvature κ are defined as:

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

The normal and curvature approximation problem. Tthe normal and the curvature are defined by extension in the whole domain through the level set function ϕ. Consequently, a good regularity of ϕ in the vicinity of the interface is necessary to accurately discretize surface tension and thus reduce spurious currents [14–16]. A frequent reinitialization of the level set act towards ensuring this property.

2.3. Level set transport

In this framework, the interface and the quantities associated with it are directly linked to the level set function ϕ. Following the interface position over time is obtained by solving the advection equation, with an underlying velocity field $\mathbf{u}$, applied to the level set:

$$\frac{\partial\phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = 0 \tag{8}$$

Yet, in a general case, resolving the advection equation will induce deviation of ϕ to be a signed distance function and consequently cease to be a solution of the eikonal equation 1. The reasons are twofold. First, distortions come from numerical errors when solving the advection Eq. (8). But most importantly, as demonstrated by Trujillo et al. [17], these distortions are directly connected to the nature of the flow field,

regardless of the method used to evolve the level set. In fact, any transported function will have its gradient and derivatives of higher degree stretched under the presence of a moving fluid with a non-zero strain rate tensor. In a two-phase fluid simulation, the numerical thickness of the interface will not remain constant and the curvature will be miscalculated. This will lead to large errors in the continuity and momentum equation, as well as surface tension, and consequently in the pressure and velocity fields. [16, 18]. Hence, after the advection of ϕ, it is essential, for accuracy and stability, to use a reinitialization algorithm and preserve the signed distance function property (see Eq. (1)).

3. Level set reinitialization

3.1. Existing methods and their drawbacks

Level set reinitialization has been the subject of numerous researches and various strategies have been proposed. Among them, the fast marching method [2] or fast sweeping methods [3] consist on an iterative process for extrapolating the distance function starting from the cells closest to the interface. They have the main advantage of being relatively fast to compute but suffer from a lack of precision that is essential for two-phase flow applications.

Another approach has been introduced by Sussman et al. [4], where a front propagates in the normal direction from the interface by solving the PDE over a fictitious time τ:

$$\frac{\partial \psi}{\partial \tau} + \text{sgn}(\psi_0)(\|\nabla \psi\| - 1) = 0 \tag{9}$$

with the initial condition:

$$\psi_0 \equiv \psi(\tau = 0) = \phi.$$

After integration of ψ over the pseudo time τ, the result is transferred back to ϕ. The complete reinitialization of the level set function is the stationary solution of Eq. (9) which derives from the Hamilton-Jacobi equation. Indeed, the solution is obtained when $\tau \to \infty$. Eq. (9) will be referred to as H-J equation from here on.

Even if solving the H-J equation is expensive compared to the fast marching or fast sweeping method, it gives accurate solutions thanks to the use of high order schemes. The principal drawback of this method is the important number of parameters that will impact the solution. Hence, it requires adapted methods and is subjected to a CFL condition on the pseudo time step $d\tau$. Also, the number of iterations over which the H-J equation needs to be solved is up to a convergence criterion that is, or should be, based on the deviation of the level set field to the eikonal equation. This criterion is not trivial to evaluate explicitly, locally or globally in the whole domain, leaving an important uncertainty when setting up a simulation. Luddens et al. [19] proposed a criterion for easing the automation of the choice of H-J frequency and the number of iterations, based on the L_1 norm of $\|\nabla\phi\| - 1$. However, as pointed out by Solomenko et al. [20], the choice of the proposed threshold is yet not trivial and, moreover, this error measure is subject to the presence of kinks where, by nature, $\|\nabla\phi\| \neq 1$, as further explained in Sec. 5.4.1.

Finally, and most importantly is the reinitialization frequency parameter, i.e. how frequently the level set is reinitialized with the H-J equation. Ideally, one would expect to apply it after every advection equation. However, in practice, as pointed by the authors in [6, 7], this method introduces displacements of the interface position because of numerical errors, leading to mass loss/gain. It also affects the geometrical properties of the interface which, as we will see in Sec. 6.8, may impact significantly on topological changes. Even if methods exist to reduce the displacement of the surface [5–7], it is usual to reinitialize the level set field after solving the advection equation a few times, to reduce the numerical errors introduced during the reinitialization procedure. Hence, a question arises on the impact of the deviation of ϕ from a signed distance function between two reinitializations.

To summarize, finding the right parameters is still bleary and depends largely on the underlying application. In a recent work, Solomenko et al. [20] did a comparative study on some of these parameters and showed how it may affect the level set when coupled with the Navier-Stokes equations in presence of surface tension.

3.2. Objectives of the proposed approach

We believe that a geometry-based approach leads to a more natural construction of a level set function solution of the eikonal equation. The main ambition of this article is to introduce a robust and accurate high order reinitialization method with a straightforward numerical parameterization and free of the cumbersome frequency parameter. Thus, it is applicable at every time step after solving the advection equation without affecting negatively the dynamic of the underlying two-phase flow.

4. Leading idea of the method

The precursor works of [9, 10, 21] introduced an alternative and original strategy based on a geometrical approach, consisting in performing a gradient descent algorithm to find the closest point to the interface, used in a reinitialization procedure.

4.1. Our main contributions

Contrary to [9] and [10], wherein the authors enhanced the Fast Marching and HJ reinitialization with closest points near the interface, i.e. restricting themselves to the cut-cells, we extend the closest point computation to the whole region of interest where the level set has to be accurately computed. While [21] have shown high order results on static cases, we propose a method that is robust and accurate with moving interfaces and two-phase flow with topology changes, thanks to the use of a newly developed kink detection algorithm and adequate treatment. Furthermore, the method is safely applied after every advection of the interface. In the next sections, the proposed method will be referred to as the Reinitialization using the Closest Point algorithm (RCP).

4.2. The closest point

In the general case of an arbitrary surface representation, finding the closest point to the interface requires an optimization algorithm which seeks to minimize an objective function. Herein, for any point $\mathbf{x}$ of the domain, finding a closest point of $\mathbf{x}$ to the surface Γ consists in finding a point $\mathbf{y}$ on Γ which minimizes the value $\|\overrightarrow{\mathbf{xy}}\|$:

$$\forall \mathbf{x} \in \Omega, \mathbf{y} = \mathrm{CP}(\mathbf{x}), \|\overrightarrow{\mathbf{xy}}\| = \min_{\mathbf{y}\in\Gamma}(\|\mathbf{x}-\mathbf{y}\|)$$

It is important to note that all points $\mathbf{x}$ which have more than one closest point define the medial axis of a surface. This particular topic is subjected to a detailed discussion in section Eq. (12).

The gradient descent. Within the level set framework, in practice, the closest point algorithm is implemented to satisfy a twofold condition:

$$\phi(\mathbf{y}) = 0 \quad \text{and} \quad \overrightarrow{\mathbf{xy}}.\mathbf{t}_{\mathbf{y}} = 0,$$

where $t_{\mathbf{y}}$ is a vector part of the tangent plane of Γ at $\mathbf{y}$. The first equation guarantees that $\mathbf{y}$ is on Γ and the second ensures that the local tangent plane of the interface is orthogonal to the vector $\overrightarrow{\mathbf{xy}}$. Given a sufficiently well-defined level set field, if both conditions are satisfied, $\mathbf{y}$ is defined as a closest point of $\mathbf{x}$ to the interface.

A simple strategy consists in a variant of the Newton method that seeks to minimize the absolute value of ϕ by descending along the normal direction and then looking for the orthogonality in the tangent plane. At each step of the algorithm, interpolations are needed in order to compute the value of ϕ and the normal. Implementation details are given in [9, 16].

4.3. Reinitialization with closest points

Let $\psi_{dist}(\mathbf{x})$ be the Eulerian distance between $\mathbf{x}$ and its closest point:

$$\forall \mathbf{x} \in \Omega : \psi_{dist}(\mathbf{x}) = \|\overrightarrow{\mathbf{x}\mathrm{CP}(\mathbf{x})}\|$$

and the sign function sgn as:

$$\mathrm{sgn}(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x} \in \Omega^- \\ 0 & \text{if } \mathbf{x} \in \Gamma \\ +1 & \text{if } \mathbf{x} \in \Omega^+ \end{cases}$$

which is equivalent to the sign of the level set function. Multiplying ψ_{dist} by the sign function leads to:

$$\psi(\mathbf{x}) = \mathrm{sgn}(\mathbf{x}) \cdot \psi_{dist}(\mathbf{x}) = \begin{cases} -\|\overrightarrow{\mathbf{x}\mathrm{CP}(\mathbf{x})}\| & \text{if } \mathbf{x} \in \Omega^- \\ 0 & \text{if } \mathbf{x} \in \Gamma \\ +\|\overrightarrow{\mathbf{x}\mathrm{CP}(\mathbf{x})}\| & \text{if } \mathbf{x} \in \Omega^+. \end{cases} \tag{10}$$

which is the definition of the signed distance function given in Sec. 2.1. Conversely, this solution ensures that:

$$\forall \mathbf{x} \in \Omega : \|\nabla(\mathrm{sgn}(\mathbf{x}) \cdot \psi_{dist}(\mathbf{x}))\| = 1$$

and proves that it is possible to successfully convert any function into a signed distance function capturing the same surface, as long as one knows how to compute the closest point.

4.4. Towards numerical discretization

In practice, ϕ will remain close to a signed distance function after the advection step, especially if the reinitialization procedure is achieved systematically since maximal distortion will be limited by the CFL restriction. Consequently, if the scalar field ϕ is sufficiently smooth and regular, particularly near the surface, then the gradient descent algorithm can be safely employed. This approach leads to an efficient strategy that gives a solution of the closest point at the precision of the objective function derivative evaluation. In practice, this evaluation will be done thanks to high order interpolations as detailed further.

Problems arise when, from a numerical point of view, ϕ is under-resolved (as the narrow tail of the interface illustrated in Fig. 1) or close to a non-smooth region (for example, during a topological change). Such particular configurations eventually arise with the dynamic motion of the surface and represent a challenging problem as they can perturb the gradient descent. Indeed, when the first derivative of ϕ is non-differentiable, numerical interpolations will lead to large errors and hence the key ingredient of the method, the closest point, is miscalculated. It is essential for the robustness of the method, as illustrated in Fig. 1, to treat adequately such regions, named kinks, as will be detailed in the next sections.

5. Proposed method and implementation

For the sake of clarity, until the end of the article, it will be assumed that the level set field is discretized on a uniform Cartesian mesh. Yet, as we will see below, the method is principally based on interpolations and derivatives calculation of ϕ. Thus, it can be naturally extended on an arbitrary mesh. Furthermore, all algorithms are presented as if they were processed on a single processor, nevertheless they can be adapted for a parallel application. In the result section 6, all test cases were executed with more than one processor. We will not detail further these two particular implementation details.

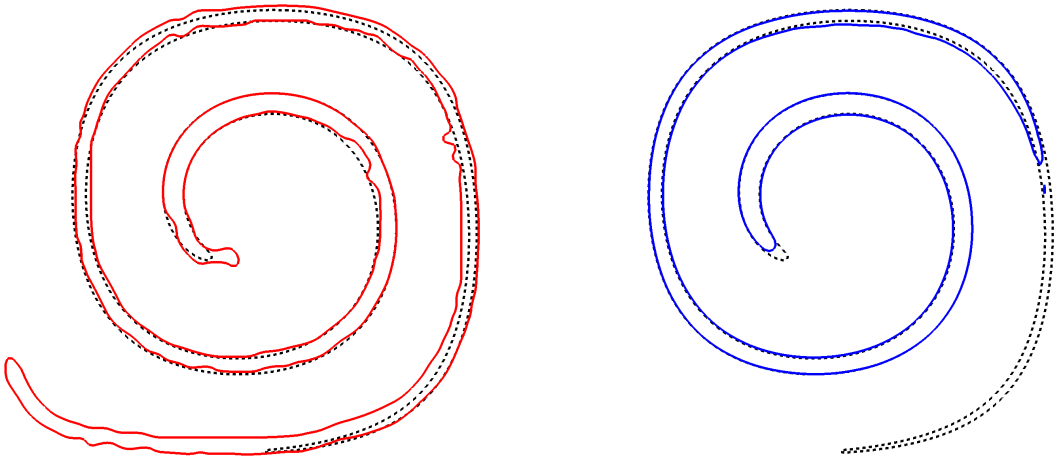


Figure 1: Illustration of the RCP algorithm result on the single vortex 2D case, without kinks treatment (left) and with kinks treatment (right). The reference solution is drawn with black dashed lines. The absence of treatment of kinks leads to heavy distortion of the interface and exhibits anti-diffusive behaviour. The interface is given at maximal deformation for a grid resolution of 128^2 cells.

5.1. Locality of the algorithm

As the surface is unambiguously defined by the level set thanks to a finite number of cells surrounding it, the locality of the proposed algorithm is crucial to its efficient application. Conversely, the overall accuracy of the method will not be increased by considering cells far from the interface and can then be ignored, resulting in a significant gain in computational efficiency.

Similarly, to the concept of *level set band*, we will thus define an ensemble of sets of cells, at an increasing distance of the surface, that will be used to locate different processes executed in the algorithm. For the interface to be accurately captured, one key criterion is the necessity to preserve a certain number of cells around the interface, i.e. within the employed interpolation/derivatives stencils. The higher the desired accuracy, the larger the stencil. We define S_{c_l} to be the interpolation stencil surrounding a cell c_l. For example, in 2D, a fourth order interpolation requires 4×4 cells. In that case, S_{c_l} encompasses a zone of 2 cells to the left/bottom of c_l and 2 cells to the right/top of it.

We distinguish 4 nested sets of cells, as illustrated in Fig. 2:

- Ω_Γ: all cells crossed by the surface Γ;
- $\Omega_{Stencil}$: the union of all stencils S_{c_l} associated with all cells c_l of Ω_Γ;
- Ω_{CP}: all cells where the closest point is computed;
- Ω_{Band}: the largest band of cells where the level set function is required.

Where Ω_{CP} and Ω_{Band} are constructed iteratively by growing the band starting from the cells containing the surface, i.e. Ω_Γ, towards the desired distance, in a similar manner as described in [15].
Hence, $\Omega_\Gamma \subset \Omega_{Stencil} \subset \Omega_{CP} \subset \Omega_{Band} \subset \Omega$.

We then define two particular sets of cells that will be thoroughly detailed further:

– Ω_{Kink}: all cells containing a kink (see Sec. 5.4.1);

– $\Omega_{PseudoLS}$: the region of cells where kinks need to be replaced by a tailor-made smooth function (see Sec. 5.5.1).

The concept of *kink*, which is a key point of the proposed method, will be thoroughly defined and discussed in the dedicated section 5.4.

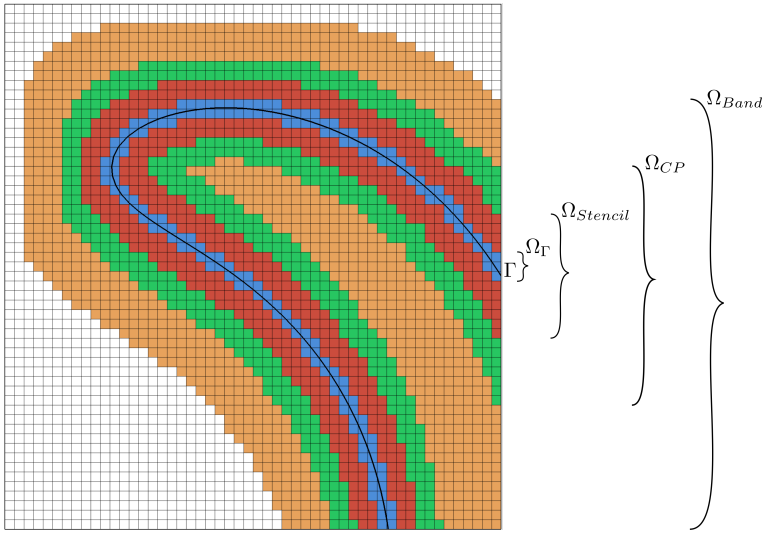


Figure 2: Illustration of the four nested sets of cells. The underlying interface is drawn as a blue line, Ω_Γ as blue colored cells, $\Omega_{Stencil}$ in red, Ω_{CP} in green and Ω_{Band} in orange.

5.2. Proposed algorithm

The RCP procedure is applied after the advection of ϕ (see Eq. (8)) and preceded by the construction of the sets described above. Algorithm 1 outlines the prime steps of the proposed method. First, a pre-processing is necessary in order to detect all cells (i.e. build Ω_{Kink} and $\Omega_{PseudoLS}$) that will require a particular treatment. Then, regularization through a pseudo level set is applied (step 2) for all ill-defined cells that are sufficiently far from the surface (cells part of $\Omega_{PseudoLS}$). Thereafter, the closest points are computed (step 3a) for all cells in Ω_{CP}, except for particular cells at the interface for which the associated interpolation stencil contains at least one kink i.e. $(\Omega_{Kink} \cap S_{c_l}) \neq \emptyset$. The reinitialized level set is then computed (step 3b) thanks to Eq. (11). Finally, a low-order HJ reinitialization procedure is applied (step 4) to all remaining cells. We present in the following sections these successive steps in more detail, particularly the treatment of kink cells. Moreover, the reader can find in appendix Appendix B a detailed pseudo code of the RCP algorithm to ease the implementation.

5.3. Research of the closest points

One of the key points of the method is the accuracy of the gradient descent to find the closest points. Therefore, we follow the algorithms proposed in [15, 16] where a fourth-order convergence accuracy is achieved thanks to fourth-order interpolation schemes and an orthogonality criterion. For a better understanding and reproductibility, we recall in algorithm 2 the base gradient descent method of [15] that we have complemented with the orthogonality criterion of [16] for increasing the accuracy, particularly when curvature is large or when the level set is more distorted. The latter algorithm is not being detailed herein for clarity reasons. For our use, the threshold ζ is set to $10^{-6} \times h$ and the descent is done for maximum 50 iterations. In most cases, when coupled to the complementary orthogonal criterion, around 10 steps are required to converge

Algorithm 1 Outline of the RCP algorithm

1. Detection of kinks: build Ω_{Kink} (see algorithm 3);
2. $\forall$cell $\in \Omega_{PseudoLS}$ (see Sec. 5.5.1);
   (a) Compute a pseudo distance function;
   (b) Apply a low-order HJ reinitialization;
3. $\forall$cell $\in \Omega_{CP} \setminus \{c_l \in \Omega_\Gamma \mid (\Omega_{Kink} \cap S_{c_l}) \neq \emptyset\}$:
   (a) Compute the closest point with gradient descent (see Sec. 5.3);
   (b) Update the level set value with Eq. (11) (see Sec. 5.3 and Sec. 5.5.2);
4. $\forall$cell $\in \Omega \setminus \Omega_{CP}$: apply a low-order HJ reinitialization (see Sec. 5.6).

satisfactorily. These parameters lead to a good balance between accuracy and efficiency. Also, in practice, the solution increment is cropped so as not to exceed the cell size. As presented by [21], it is worth noting that sixth-order accuracy can be obtained for the gradient descent. Nevertheless, we found that a fourth-order accuracy is enough in all our test cases section 6. Consequently, using higher-order numerical schemes was not considered for efficiency reasons. Also, it is worth to note that other closest point approximations or algorithms such as proposed in [9] could be used in RCP, as long as they are sufficiently precise to maintain an accurate level set field.

Algorithm 2 Base closest point algorithm in pseudo code.

$\mathbf{CP}^0 \leftarrow$ `Coordinates`$(c_{i,j,k})$; $n \leftarrow 0$
while | `Intp`$(\phi, \mathbf{CP}^n)$ | $> \zeta$ and $n < nitmax$ **do**
  $\mathbf{grad} \leftarrow$ `Intp`$(\nabla\phi, \mathbf{CP}^n)$)
  $dist \leftarrow$ `Intp`$(\phi, \mathbf{CP}^n) / |\mathbf{grad}|$
  $\mathbf{normal} \leftarrow \mathbf{grad} / |\mathbf{grad}|$
  $\mathbf{CP}^{n+1} \leftarrow \mathbf{CP}^n - dist \times \mathbf{normal}$
  $n \leftarrow n + 1$
end
return $\mathbf{CP}^n$

It should be noted that performing the algorithm for every cells in the domain would be time-consuming and not relevant. Indeed, as explained in the previous section, all the physical quantities which are directly linked to the level set such as the curvature or the volume fraction are only needed close to the surface. Moreover, the precise position of the surface is captured by the cells surrounding it: for instance, a n^{th} order precision is expected with a stencil of n^d cells, with d the dimension. However, cells far from the surface should not interfere with it as long as they are treated adequately. Consequently, for computational efficiency the closest points are only computed where accuracy on the level set is crucial, and thus in a narrow band around the interface which is noted by Ω_{CP}. This band is constructed iteratively by growing the band starting from the cells containing the surface towards the desired distance, as described by [15]. We fix the width of the band to be $10h$, i.e. 5 cells on each side of the surface, permitting to use accurately a fifth order advection scheme.

Associated signed distance function. Once the closest point has been computed, equation 10 can be evaluated from a discrete point of view through the equation:

$$\psi(\mathbf{x}) = \text{sgn}(\phi(\mathbf{x}))\|\overrightarrow{\mathbf{x}\text{CP}(\mathbf{x})}\|. \tag{11}$$

The associated resulting reinitialized level set is thus set accordingly.

5.4. Level set kinks

5.4.1. Definition

The reliability of the method is inherently linked to the accuracy of the closest point computation. It may suffer from cells where ϕ is ill-defined or its derivatives undefined, namely at *kinks*, which can mislead the gradient descent towards surfaces which could not be the closest one. We distinguish two types of kinks:

- inherent kinks: points that are part of the medial axis, as part of the level set representation;
- numerical kinks: points that are not part of the medial axis and that appear because of numerical errors, after topological changes or near under-resolved regions.

Inherent kinks. The inherent kinks, as illustrated in Fig. 3, are points which are equidistant to at least two surfaces, i.e. when they have two or more closest points. They define the medial axis of the interface and, as also noted by [10, 11, 22], on those kinks the derivatives of ϕ are not defined. Hence, those kinks intrinsically arise from the definition of the level set as a signed distance function and cannot be suppressed, even when refining the mesh. From a discrete point of view, significant errors will be made if a kink lies inside the stencil used to approximate derivatives or perform an interpolation. Consequently, the closest point accuracy will be impacted, as also noted by [10, 21].

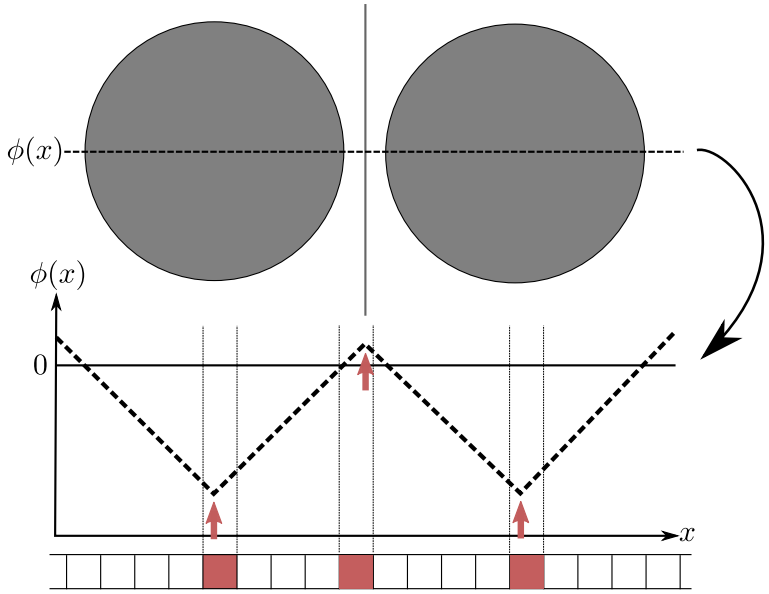


Figure 3: Illustration of the inherent kinks. On the top, a 2D visualization of two drops. The middle plot represents the level set function over a 1D cut, the associated kinks are highlighted by the red arrows. The bottom part is the associated discrete 1D mesh, the kink cells are filled in red.

Numerical kinks. The second type of kinks, referred to as numerical kinks, can lead the descent algorithm towards an undesired local minimum. We distinguish three different origins.

- First, such kinks arise when transporting under-resolved structures. For example, as illustrated in Fig. 4, when a small structure such as a bubble vanishes due to numerical diffusion when solving the advection, the underlying medial axis (in that case, reduced to a point), depicted as an extremum in the level set field, is still present. However, the surface has indeed disappeared as ϕ is not crossing the zero value anymore. In such a case, a local minimum emerges in the level set field that can perturb the interpolations.

– Secondly, numerical kinks also appear after topological changes, e.g. when two bubbles merge. In a similar way as the first category, the medial axis which resides at mid-distance to the bubbles before merging will also be transformed into a local minimum after the topology change.

– The last category comes from the stretching and compression due to the underlying velocity field which makes ϕ depart severely from a distance function, inducing local steep and flat variations of level sets that will mislead numerical schemes.

Consequently, near kinks of the first or second type, the local derivatives and interpolations will be miscalculated. More severely, local minima can act as well potentials, where the gradient descent algorithm will plunge and stop without reaching the surface. As a result, to fulfill accuracy and robustness, it is crucial to detect carefully those points.

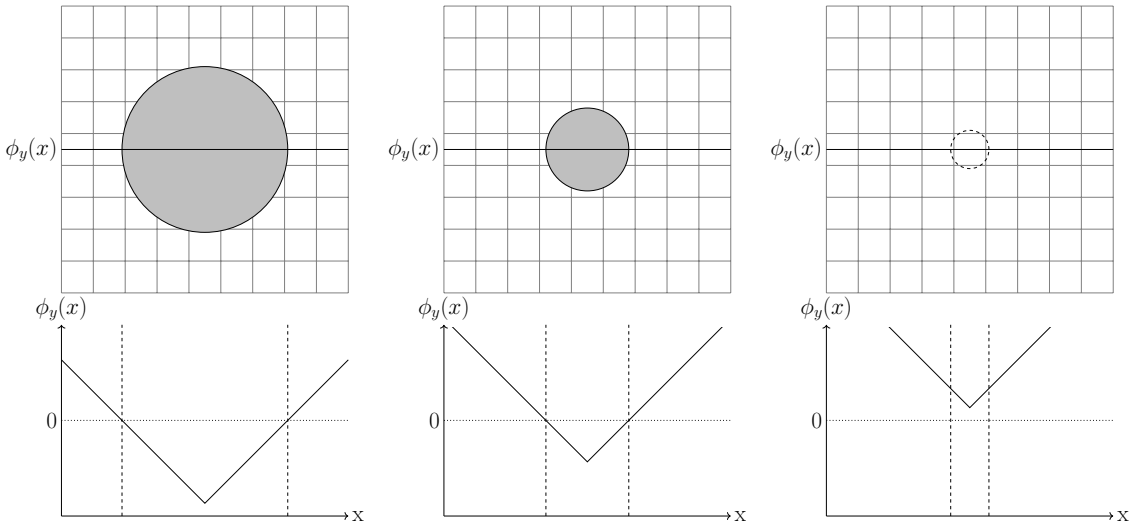


Figure 4: Creation of a numerical kink when a small structure, such as a bubble, vanishes due to numerical diffusion when solving the advection equation Eq. (8). On the top a 2D visualization of the drop at different times and on the bottom a 1D cut of the associated level set.

5.4.2. Numerical detection of kinks

From a discrete point of view, we do not require the kink detection process to be fully accurate, but to be reliable while not being too costly. Hence, a low order method can suffice to detect non-smooth level set regions. For detecting cells containing or close to a kink, we propose a new criterion that relates the local ambiguity in the closest points around a point $\mathbf{x}$: $\mathbf{x}$ is said to be close to a kink if, for two different points $\mathbf{x}_{\xi_1}$ and $\mathbf{x}_{\xi_2}$ at a very small distance from $\mathbf{x}$ in directions $\overrightarrow{\xi_1}$ and $\overrightarrow{\xi_2}$, the Euclidean distance between their respective closest point is above a certain threshold ϵ. A geometric representation of the concept is given in Fig. 5. In a compact form:

$$\mathbf{x} \text{ is a kink if } \exists(\mathbf{x}_{\xi_1}, \mathbf{x}_{\xi_2}), \|\mathrm{CP}(\mathbf{x}_{\xi_1}) - \mathrm{CP}(\mathbf{x}_{\xi_2})\| > \epsilon. \tag{12}$$

The choice of the threshold ϵ relates to the sensitivity of the detector and is discussed in details further in this section. We can already note that it is of the order of the distance to the interface. Thus, it should manage to differentiate between a kink and a small interface structure, i.e., from a discrete point of view, a sphere of radius of the order of mesh cells.

First order closest point approximation. Since there is no analytical method for computing the closest points and we cannot enumerate all points $\mathbf{x}_\xi$ surrounding $\mathbf{x}$, the criterion has to be approximated. For the first matter, we propose to use a first order local approximation of the closest point:

$$\mathrm{CP}(\mathbf{x}) \simeq \mathbf{x} - d(\mathbf{x})\,\mathbf{n}(\mathbf{x}) \tag{13}$$

where $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ is the normal and d the signed Euclidean distance to the surface. Still, within first-order approximation, one could use the approximation $d \simeq \frac{\phi}{|\nabla\phi|}$, and thus Eq. (13) can be rewritten as:

$$\mathrm{CP}(\mathbf{x}) \simeq \mathbf{x} - \frac{\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}\frac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}. \tag{14}$$

which can be related to the first step of the general closest point algorithm based on the gradient descent described in Sec. 5.3. Once appropriately discretized, this equation will be used to compare closest points in the surrounding of a point $\mathbf{x}$ and determine if it is close to a kink.

For efficiency reasons, we restrict the number of surrounding points $\mathbf{x}_\xi$ to an acceptable number. Furthermore, instead of using interpolation to evaluate the $\mathbf{x}_\xi$ in several directions, we simply approximate Eq. (14) at a mesh cell center with off centered derivatives using the neighbouring cells. This is justified since, at a kink, different off centered schemes (e.g. east or west biased for the x direction) used to approximate the normal will vary significantly depending on the direction used to compute it, as if we had set a different starting point to the gradient descent. This fact will lead the surrounding closest point approximations to spread into disperse positions, attaining the desired property. Conversely, in smooth regions, all biased gradient approximations will be numerically close and so will be their associated closest points.

Biased approximation of the closest point. Classical off centered schemes can thus be used to approximate the normal. For the sake of simplicity, we have only shown here the example for the south west direction. The reader can easily build the remaining schemes based on the following formulation; the detailed derivation has been reported in appendix Appendix A.

In 2D, if we consider a point $\mathbf{x}_{\xi_{SW}} = \mathbf{x} + dl\,(-1,-1)^T$, with $dl \ll h$ a very small fictitious length, at the *south-west* of $\mathbf{x}_{i,j}$, the center of the cell $\Omega_{i,j}$: on a uniform grid, the gradient operator can be approximated with a first order *upwind-biased* scheme in both x and y directions by:

$$\nabla\phi(\mathbf{x}_{\xi_{SW}}) \simeq \left(\frac{\phi_{i,j} - \phi_{i-1,j}}{\delta x}, \frac{\phi_{i,j} - \phi_{i,j-1}}{\delta y}\right) \tag{15}$$

Hence, the closest point biased in the south-west direction can be approximated by the formula:

$$\mathrm{CP}(\mathbf{x}_{\xi_{SW}}) \simeq \mathbf{x}_{i,j} - \frac{\phi_{i,j}}{|\nabla\phi_{i,j}|}\frac{\nabla\phi(\mathbf{x}_{\xi_{SW}})}{|\nabla\phi(\mathbf{x}_{\xi_{SW}})|}. \tag{16}$$

Similar schemes are designed for the three other neighbours (*south-east*, *north-west* and *north-east*). These schemes are local and fast to compute and the 3D case is straightforward to obtain as presented in the algorithm description.

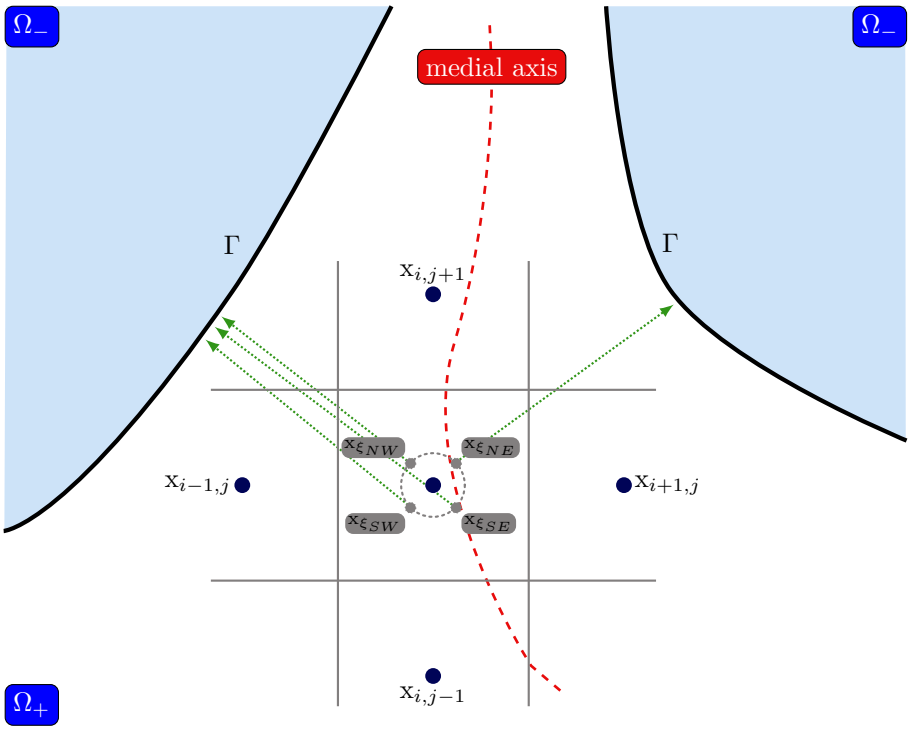


Figure 5: Illustration of the kink detection algorithm on a 2D mesh in the vicinity of a medial axis. Three of the four closest point approximations (SW, NW and SE) surrounding $\mathbf{x}_{i,j}$ point towards a narrow region on the left most surface whilst the last one (NE) is located on the right most surface. The distance between those CP being sufficiently large, the cell is detected as a kink.

The relative Euclidean distance between closest points. The Euclidean distance is used in Eq. (12) to determine if two closest point approximations are far enough (i.e. $> \epsilon$) from each other to consider the relative cell to contain a kink. Algebraically, this formula can be simplified by the definition of the closest point biased approximations (as in Eq. (16)) for two different directions ξ_1 and ξ_2:

$$\begin{aligned} \mathrm{CP}\left(\mathbf{x}_{\xi_1}\right) - \mathrm{CP}\left(\mathbf{x}_{\xi_2}\right) &\simeq \frac{\phi_{i,j}}{|\nabla\phi(\mathbf{x}_{i,j})|}\left(\frac{\nabla\phi(\mathbf{x}_{\xi_1})}{|\nabla\phi(\mathbf{x}_{\xi_1})|} - \frac{\nabla\phi(\mathbf{x}_{\xi_2})}{|\nabla\phi(\mathbf{x}_{\xi_2})|}\right) \\ &= d(\mathbf{x}_{i,j})\left(\mathbf{n}_{\xi_1} - \mathbf{n}_{\xi_2}\right). \end{aligned}$$

Thus, the kink criterion can be equivalently written in a normalized form as:

$$\left\|\mathrm{CP}\left(\mathbf{x}_{\xi_1}\right) - \mathrm{CP}\left(\mathbf{x}_{\xi_2}\right)\right\| > \epsilon \iff \left\|\mathbf{n}_{\xi_1} - \mathbf{n}_{\xi_2}\right\| > \frac{\epsilon}{d} \tag{17}$$

where $d \equiv |d(\mathbf{x}_{i,j})|$ for clarity and the choice for ϵ is still to be determined, as detailed below. As a result, the numerical evaluation of the criterion reduces to computing the biased closest points' normal.

Kink detection algorithm. Based on these biased schemes, we can now compute the associated closest point approximations' normal in the four directions: $\mathbf{n}_{\xi_{SW}}$, $\mathbf{n}_{\xi_{SE}}$, $\mathbf{n}_{\xi_{NW}}$, and $\mathbf{n}_{\xi_{NE}}$. Finally, we evaluate the maximum normalized Euclidean distance as in Eq. (17) to detect if a pair is sufficiently distant to consider the region to be near a kink. We have summarized the method in algorithm 3, extended in the general 3D

case. It is worth noting that this algorithm can be optimized by stopping it whenever the criterion has been met once and thus avoiding unnecessary computations. Illustrations of the resulting detected kink cells are shown in Fig. 6 for the single vortex 2D test case (see Sec. 6.5) and in Fig. 7 for the Zalesak disk test case (see Sec. 6.4).

Algorithm 3 Kink detection algorithm in 3D. In practice, the threshold is chosen as $\epsilon = d/2$.

Compute all $\mathbf{n}_{\alpha_x,\alpha_y,\alpha_z}$ using algorithm 4 ▷ `the 8 biased normal approximations`
MNED $\leftarrow 0$ ▷ `the maximum normalized Euclidean distance`
for $\overrightarrow{\mathbf{d}_1} \in \{-1,+1\}^3$ **do**
  ▷ `directions towards cell's vertices:` $\overrightarrow{\mathbf{d}_1} = \{(-1,-1,-1),(-1,-1,+1),\ldots\}$
  for $\overrightarrow{\mathbf{d}_2} \in \{-1,+1\}^3$ **do**
    if $\overrightarrow{\mathbf{d}_1} \neq \overrightarrow{\mathbf{d}_2}$ **then**
      MNED $\leftarrow \max\left(\text{MNED}, \left\|\mathbf{n}_{\overrightarrow{\mathbf{d}_1}} - \mathbf{n}_{\overrightarrow{\mathbf{d}_2}}\right\|\right)$ ▷ $\overrightarrow{\mathbf{d}_1} = (-1,-1,-1) \Leftrightarrow \mathbf{n}_{\overrightarrow{\mathbf{d}_1}} = \mathbf{n}_{-1,-1,-1}$
    end
  end
end
if MNED $> 1/2$ **then** ▷ `since` $\epsilon = d/2$ `(see Eq. (18))`
  return true
else
  return false
end

Algorithm 4 Biased closest point normal approximations at a cell $\Omega_{i,j,k}$ in 3D, using variations of Eq. (15), as fully detailed in appendix Appendix A

for $\alpha_x \in \{-1,+1\}$ **do**
  for $\alpha_y \in \{-1,+1\}$ **do**
    for $\alpha_z \in \{-1,+1\}$ **do**
      grad $\leftarrow \left(\alpha_x \frac{\phi_{i+\alpha_x,j,k}-\phi_{i,j,k}}{\delta x}, \alpha_y \frac{\phi_{i,j+\alpha_y,k}-\phi_{i,j,k}}{\delta y}, \alpha_z \frac{\phi_{i,j,k+\alpha_z}-\phi_{i,j,k}}{\delta z}\right)$
      $\mathbf{n}_{\alpha_x,\alpha_y,\alpha_z} \leftarrow$ **grad**$/|$**grad**$|$
    end
  end
end

Detector threshold. The threshold ϵ used in Eq. (12) and algorithm 3 is a key in the kink detection process. Ideally, it has to be infinitesimally small. However, due to numerical approximations, even in a smooth region, the distance between closest points computed with biased approximations of Eq. (14) can be of the order of h, where h is the cell width. This is particularly true for points far from the interface where $d \simeq |\phi| \,/\, |\nabla\phi|$, the approximated distance to the surface, is large, or for points near a surface of high curvature where the normal vector varies fast. Starting from the assumption that the smallest smooth structure captured by the mesh is a sphere of radius $h/2$, using criterion of Eq. (17), this translates to the fact that the normalized distance to the interface should exceed half a unit cell. In other terms: $h\,\|\mathbf{n}_{\xi_1} - \mathbf{n}_{\xi_2}\| > h/2 \iff \epsilon/d = 1/2$, from which we deduce $\epsilon = d/2$. As a result, the normalized criterion of Eq. (17) finally reduces to:

$$\|\mathbf{n}_{\xi_1} - \mathbf{n}_{\xi_2}\| > \frac{1}{2} \tag{18}$$

We have found this value to detect kinks with good reliability while not introducing too many false positives. Indeed, we restrict the closest point algorithm to points sufficiently close to the interface. Moreover,

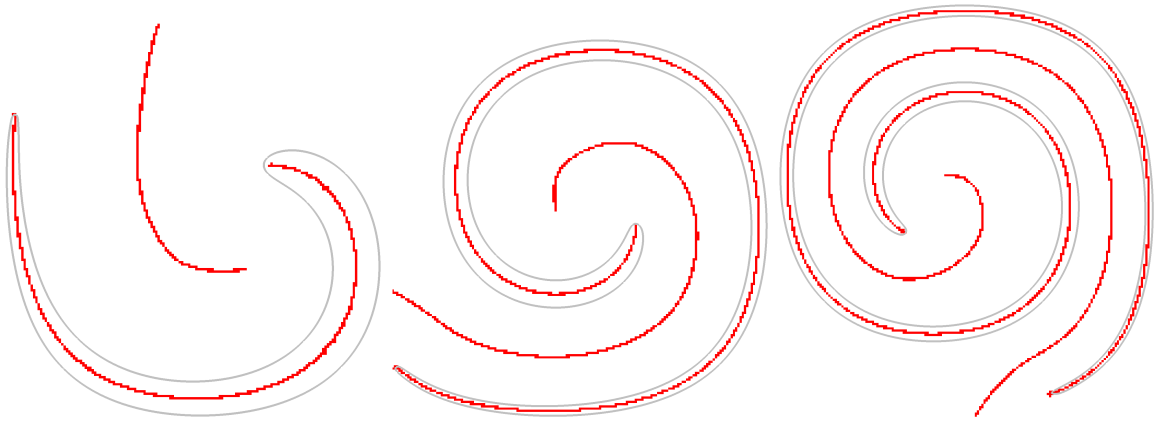


Figure 6: Illustrations of the kink detection algorithm 3 on the 2D single vortex test case (see Sec. 6.5) at different instants, for a grid resolution of 256^2 cells. The surface is drawn in gray and the detected kink cells are marked in red. Over time, the kink detector successfully captures all inherent kinks that appear as the tail is stretched and locates the medial axis.

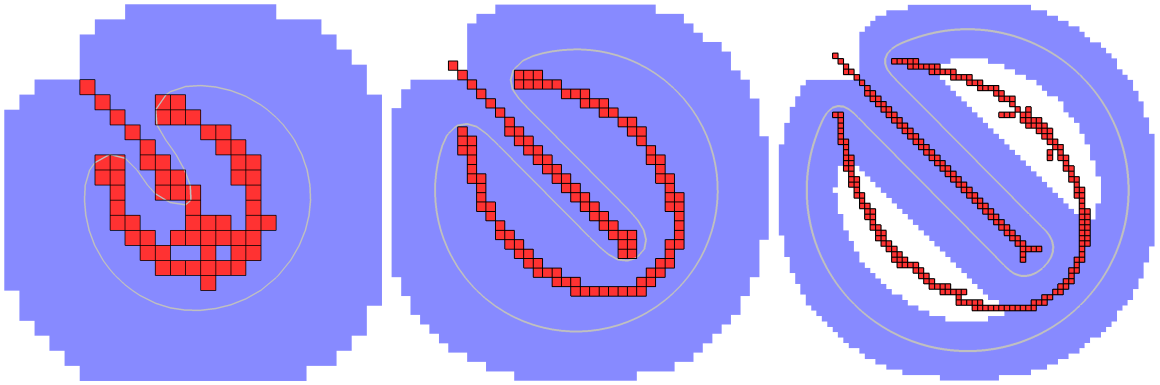


Figure 7: Illustration of the kink detection algorithm 3 on the Zalesak test case (see Sec. 6.4), for a grid resolution of 50^2 (left), 100^2 (center) and 200^2 (right) cells, after approximately 5/8 of rotation. The surface is drawn in gray, the detected kink cells are marked in red and Ω_{CP} in blue. As the cell size diminishes, the kink detector successfully captures all cells part of the medial axis and hence containing an inherent kink. For clarity, figures are not at the same scale in order to better observe the detected kinks. For the finest mesh (100^2), some kinks are not part of Ω_{CP} and, consequently, they are not reinitialized with the RCP method and hence do not disturb the gradient descent.

in regions of high curvature, i.e. $\kappa \sim O\left(h^{-1}\right)$, more kinks will indeed be detected. This is not a problem as those sensitive regions are usually under resolved and the use of high order interpolation with a large stencil would anyway suffer from including kink cells. A sensitivity study on the threshold ϵ is presented in Sec. 6.4.2.

Discussion on another approach: the quality function $Q(\mathbf{x})$. To detect kinks, another approach was introduced by Macklin and Lowengrub in [22] wherein the authors defined a *normal quality function* $Q(\nabla\phi(\mathbf{x}))$ as:

$$Q(\nabla\phi(\mathbf{x})) = |1 - |\nabla\phi||, \tag{19}$$

which measures the deviation of ϕ to a distance function. If $Q(\nabla\phi(\mathbf{x})) > \eta$ for a relatively small positive value of η, then the point $\mathbf{x}$ is considered to be near a kink. In their work, Macklin and Lowengrub fixed η at 0.1 and found this value reliable enough to detect kinks with few false positives. In the same way, Ervik et al. [11] used $\eta = 0.005$.

However, we found that this criterion could include more false positives than the one we proposed, based on a geometrical approach. Indeed, the quality function highlights areas where ϕ deviates from a distance

function, which means that if the gradient of ϕ is solely stretched or compressed in those areas, i.e. $\nabla\phi = \alpha$ with $\alpha \in \mathbb{R}$, then it will indicate the presence of a kink (depending on the relative values of η and α), while there might be none. Actually, these areas may neither be inherent nor numerical kink.

On the other hand, our geometrical approach is quite different since a kink is detected when the variation of the closest point is more than a given threshold ϵ. This means the sensitivity of our detector, i.e. the choice of ϵ value, will directly depend on the maximum acceptable threshold to detect under-resolved structures. Herein, we have set $\epsilon = h/2$: thus, we consider that a structure with a radius of curvature less than $h/2$ (in 2D) cannot be well detected with our closest point method and thus represents an under-resolved structure that would lead to unwanted numerical errors, as discussed in Sec. 5.5.2.

5.5. Kinks treatment

5.5.1. Cells far from the interface

Kinks can deteriorate the level set regularity and lead gradient descent towards local minima. Hence, they need to be smoothed adequately before applying the closest point algorithm. However, as the accurate position of the interface are captured by the cells in $\Omega_{Stencil}$, this treatment is only applied to ill-defined cells sufficiently far from Γ, i.e. for the subset $\Omega_{PseudoLS}$ defined as:

$$\Omega_{PseudoLS} = (\Omega_{Band} \cap \Omega_{Kink+}) \setminus \Omega_{Stencil} \tag{20}$$

where Ω_{Kink+} is the set of cells of Ω_{Kink} augmented by their neighbours. We made this choice to increase numerical stability in the vicinity of kinks. The smoothing procedure consists in two consecutive steps applied on those cells: first the creation of a pseudo level set, followed by a fast low-order smoothing procedure.

The pseudo level set. The first step aims to give an acceptable initial guess for the corrected level set function that will be further smoothed. We defined the pseudo level set $\psi_{PseudoLS}$ as a rough first order approximation of a signed distance function, pointing towards the interface. It is created by iteratively growing a band starting from the interface. A cell part of the n^{th} layer of that band is located at an approximate distance nh to the interface, where h is the uniform cell size. Thus, to all cells c_l part of $\Omega_{PseudoLS}$, we set:

$$\psi_{c_l} = nh \times \mathrm{sgn}(\phi_{c_l}) \tag{21}$$

Fast low-order smoothing. The second step consists in smoothing the pseudo level set with a fast reinitialization algorithm that will guarantee a smooth solution regarding the neighbouring cells concerned. Though several strategies may be considered, we have found that the use of a low-order solving of the HJ equation 9 suffices to obtain satisfactory results. As standing relatively far from the interface permits to loosen the accuracy criterion, we have used a first order upwind scheme for the gradient evaluation and integrate the equation over 20 iterations. This procedure leads to a very efficient strategy to clear kink perturbations while obtaining a regularized function that points smoothly towards the interface, as required by the descent algorithm.

5.5.2. Particular case near the interface

We chose not to alter cells that are close to the interface and which are in the vicinity of kinks, as prescribed in the third step of algorithm 3, and formally defined as the subset $\Omega_{CP} \setminus \{c_l \in \Omega_\Gamma \mid (\Omega_{Kink} \cap S_{c_l}) \neq \emptyset\}$. This is motivated by the fact that the presence of such local extrema has a great influence on the interface capture, as they lie within the interpolation stencil. Hence, any alteration made to those cells will definitely modify the surface position with no guarantee in giving a more accurate approximation. This concurs with the remarks of Trujillo [17] stating that reinitializing under-resolved areas will degrade more the level set function than diminishing the degree of error. Moreover, these cells may continue to represent sub-mesh interfaces that would be lost without a precise and specific treatment, a work that is beyond the scope of this article. Consequently, all cells part of $\Omega_{CP} \setminus \{c_l \in \Omega_\Gamma \mid (\Omega_{Kink} \cap S_{c_l}) \neq \emptyset\}$ are not modified by the algorithm.

Nevertheless, we believe that developing and adapted closest point computation method near level set kinks would surely benefit to the proposed algorithm. This would permit applying the closest point

reinitialization (step 3a of algorithm 1) to all cells in Ω_{CP}. Some authors tried to resolve the issue of calculating accurately the interpolations/derivatives near a kink [11, 22]. Yet, the integration of such methods within the RCP algorithm could be the object of future works.

5.6. Smoothing outside the closest point band

Also, for the sake of numerical stability, we have applied the same fast low order smoothing procedure as a post-processing of the algorithm (see step 3 of algorithm 1) in cells part of $\Omega \setminus \Omega_{CP}$. In practice, a low-order HJ equation is solved over 5 iterations, which we have found to be sufficient to maintain a smooth field far from the interface.

6. Results

In this section, we detail the numerical framework used and propose a series of test cases which demonstrate the accuracy and robustness of the method. For the latter, we will first test the capability of the method on state-of-the-art benchmarks using analytical velocity fields, from the simple test of a sphere deformation to more complex ones involving thin layers or sharp corners. Then the coupling with the Navier-Stokes equations will be considered to assess the capability of the method to accurately and robustly capture the behavior of inviscid two-phase flows.

In order to gauge the results of the new method, all test cases will be compared to two other approaches to reinitialize the level set. The first is the H-J equation Eq. (9) solved after each advection step (noted HJ-1), as it is done for RCP and, for the second, the same H-J equation is solved every 10 advection steps (noted HJ-10).

As the overall strategy of RCP an H-J are quite different, the comparisons are only made as a guideline to position the method amongst the family of numerical methods for level set reinitialization. Numerical details of the implementation of the reinitialization with the H-J equation will be given in the following section.

6.1. Numerical methods

6.1.1. Flow solver

The method was implemented and tested using the massively parallel incompressible open-source CFD code Notus [23], for which the following test cases are available or easily reproducible. The Navier-Stokes equations are solved on a staggered grid within a finite volume framework and a time splitting correction method for the velocity-pressure coupling [24]. A first order semi-implicit backward difference (SBDF-1) scheme is used for the momentum equation. The inertial term is computed with a second order Runge-Kutta integration and the associated spatial discretization scheme will be specified for each particular test case. Phase's density and viscosity are expressed as a function of the level set and the Heaviside H_ϵ from Eqs. (4) to (6), where the regularization parameters is set to $\epsilon = 2h$.

For the sake of understanding and maximum reproducibility, we detail below the principle steps of the algorithm for solving the Navier-Stokes equations. We refer the reader to [25] for more details about the associated numerical methods.

1. Solve the level set advection and reinitialization: $\phi^n \rightarrow \phi^{n+1}$;
2. Compute volume fraction c^{n+1} thanks to Eq. (5);
3. Compute density ρ^{n+1} and viscosity μ^{n+1} thanks to Eq. (4);
4. Solve for the predicted velocity $\mathbf{u}^*$ of the momentum equation 2:

$$\rho^{n+1}\frac{\mathbf{u}^* - \mathbf{u}^n}{\delta t} + \rho^{n+1}\int_{t^n}^{t^{n+1}} \nabla\cdot(\mathbf{u}\otimes\mathbf{u})dt - \nabla\cdot(\mu^{n+1}(\nabla\mathbf{u}^* + \nabla^{\mathrm{T}}\mathbf{u}^*)) = \mathbf{f}$$

where $\mathbf{f} = \rho^{n+1}\mathbf{g} + \sigma\kappa^{n+1}\nabla c^{n+1}$ accounts for body forces such as buoyancy and surface tension (as in Eq. (7)). The integral of the advective term is approximated with a classical explicit Runge-Kutta scheme coupled to upwind WENO for spatial approximation.

5. Solve for pressure p^{n+1}: $\nabla \cdot (\frac{\delta t}{\rho^{n+1}} \nabla p^{n+1}) = \nabla \cdot \mathbf{u}^*$
6. Apply the velocity correction to obtain $\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\delta t}{\rho^{n+1}} \nabla p^{n+1}$.

6.1.2. Level set

In order to solve precisely the advection equation 8 for the level set, second order Runge-Kutta NSSP 3,2 integration coupled with a fifth order finite difference WENO scheme [26, 27], are used. When considered, the H-J equation is implemented based on the algorithm of [4], where the numerical parameters are fixed based on the comparative study of [20] and adapted for this work to obtain an accurate reinitialization of the level set. Hence, a second order Runge-Kutta integration coupled with a fifth order WENO scheme are also used. The pseudo time step is fixed at $\delta\tau = 0.3\delta x$ and the maximum number of iterations for the temporal integration is set to 16.

The overall algorithm can be summarized as follows:

1. Solve the advection equation 8 with RK-NSSP $3, 2$:

$$\begin{aligned} k^{*,1} &= (\mathbf{u}^n \cdot \nabla)\phi^n; \quad \phi^{*,1} = \phi^n - \frac{\delta t}{3} k^{*,1}; \\ k^{*,2} &= (\mathbf{u}^{n+1/3} \cdot \nabla)\phi^{*,1}; \quad \phi^{*,2} = \phi^n - \delta t k^{*,2}; \\ k^{*,3} &= (\mathbf{u}^{n+1} \cdot \nabla)\phi^{*,2}; \\ & \phi^{n+1} = \phi^n - \frac{\delta t}{2}(k^{*,1} + k^{*,3}) \end{aligned}$$

   where $\mathbf{u}^{n+1/3}$ and $\mathbf{u}^{n+1}$ are velocity approximations obtained by extrapolation of $\mathbf{u}^{n-1}$ and $\mathbf{u}^n$, and the gradient of ϕ by upwind WENO $5, 3$ scheme.
2. Reinitialize the level set with RCP or H-J.

It should be stressed that other high order methods exist to reinitialize the level set when considering the H-J equation, as well as other strategies to set the numerical parameters. Nevertheless, as stated previously, the comparisons are made as a guideline to assess the accuracy of the introduced method.

6.2. Error measures

Several errors measurements are defined to assess the performance of the proposed method. Those measures will either be computed on the whole domain Ω or only in the vicinity of the interface defined as Ω_{E_Γ} of cardinal N_{E_Γ}. In practice, Ω_{E_Γ} encompasses all the cells inside a two cells band width centered on the interface.

Shape errors. For known analytical solutions, we define the shape error, in a L_2 or L_∞ norm, as the variation of ϕ to its exact value in the set of cells in Ω_{E_Γ} by:

$$E^{L_2}_{shape} = \sqrt{\frac{1}{N_{E_\Gamma}} \sum_{\mathbf{x}_l \in \Omega_{E_\Gamma}} |\phi_{ex}(\mathbf{x}_l) - \widetilde{\phi}(\mathbf{x}_l)|^2} \quad \text{and} \quad E^{L_\infty}_{shape} = \max_{\mathbf{x}_l \in \Omega_{E_\Gamma}} (|\phi_{ex}(\mathbf{x}_l) - \widetilde{\phi}(\mathbf{x}_l)|)$$

where $\phi_{ex}(\mathbf{x}_l)$ is the expected value of the signed distance field at $\mathbf{x}_l$ and $\widetilde{\phi}(\mathbf{x}_l)$ is the computed numerical value.

Volume conservation. The volume error E_V can be defined as:

$$E_V = \frac{|V(t) - V(t=0)|}{V(t=0)}$$

where the total volume of a phase is computed through the associated volume fractions c_i (see Eqs. (5) and (6)) associated to the cells of volume V_{CV_i} as: $V = \sum_i^{N_\Omega} c_i V_{\mathrm{CV}_i}$.

Criterion for the deviation to a signed distance function. In the literature, a common manner to evaluate the property of ϕ to be a signed distance function is to compute the L_1 norm of the quality function from Eq. (19) as:

$$E_{|\nabla\phi|} = \frac{1}{N_\Gamma} \sum_{\mathbf{x}_l \in \Omega_{E_\Gamma}} Q(\nabla\phi(\mathbf{x}_l)).$$

Yet, while if gives good knowledge about the variations of the level set field, we have found this approach to be insufficient to study the accuracy of the reinitialization process. An accurate criterion for the variation of ϕ to a signed distance function has to be the same if ϕ has a slope of 2 (i.e. $|\nabla\phi| = 2$) or if its slope is 1/2 (i.e. $|\nabla\phi| = \frac{1}{2}$).

In the general case, for $\widetilde{\phi}$ a field which deviates from the signed distance function ϕ likes $\widetilde{\phi}(\mathbf{x}) = \alpha(\mathbf{x})\phi(\mathbf{x})$, with $\alpha(\mathbf{x}) : \Omega \to \mathbf{R}$, the local error measure $e_{\nabla\phi}(\mathbf{x})$ needs to be the same for a factor $\alpha(\mathbf{x})$ and $\alpha(\mathbf{x})^{-1}$, i.e. :

$$e_{\nabla\phi}(\alpha\nabla\phi) = e_{\nabla\phi}(\alpha^{-1}\nabla\phi)) \tag{22}$$

In consequence, we define $e_{\nabla\phi}$ as:

$$e_{\nabla\phi}(\nabla\widetilde{\phi}) = |\ln(|\nabla\widetilde{\phi}|)|$$

which satisfies Eq. (22). Following this definition, we propose the L_2 norm of the global variation of $\widetilde{\phi}$ to the signed distance function:

$$E^{L_2}_{\nabla\phi} = \sqrt{\frac{1}{N_\Gamma} \sum_{i}^{N_\Gamma} e_{\nabla\phi}(\nabla\widetilde{\phi})^2}.$$

The L_∞ norm of the error, noted $E^{L_\infty}_{\nabla\phi}$, is constructed similarly.

6.3. Disks rotation

First, a new benchmark is introduced to obtain a qualitative overview of the capacity of each method to carry and conserve small to large structures from a mesh point of view i.e. depending on the number of cells which represent this one. We consider multiple circles of different radius which are transported by rigid body rotation. As demonstrated by [17], with this particular flow, the gradient of the level set should remain unaltered and numerical errors are solely induced from the resolution of the advection equation and the reinitialization procedure. Thus, the interface should remain unchanged and an analytical solutions for the expected level set field can be found.

Hence, six circles denoted as $\{C_0, C_1, C_2, C_3, C_4, C_5\}$, with respective diameter of $\{D_0, D_0 + h, D_0 + 2h, D_0 + 3h, D_0 + 4h, D_0 + 5h\}$ are initialized in a $[0, 1]^2$ domain at a distance of 0.2 of the center, each other separated by an angle of $\pi/6$. In order to study small structures $D_0 = 3h$ i.e. the smallest and reference diameter D_0 is 3 cells large. The mesh is made of 64^2 cells. The velocity field is given by:

$$(u, v) = \frac{2\pi}{T}(0.5 - y, x - 0.5) \tag{23}$$

The time step is fixed and set to $\Delta t = 1.25\text{e-}3$. The simulation is stopped after one full rotation and hence, when $t = 1$.

As represented by Fig. 8, after a full rotation, all the methods fail to preserve the circles C_0 and C_1. In fact, even without re-initialization, C_0 disappears due to the advection but C_1 still remains with some volume loses. For C_2 only RCP retains the most volume and shape while HJ-1 fails totally to do it and HJ-10 just partially. For C_3, C_4 and C_5, RCP and HJ-10 have similar results, contrarily to HJ-1 which fails to preserve C_3 and induce large errors in the shape on C_4 and C_5. Hence, only the RCP methods seems to produce correct result when re-initializing the level set at each time step, compared to the HJ equations

which introduce more errors and thus need to be used less frequently. Qualitatively, this test case shows that the RCP method exhibits better results for preserving structure with low resolution.

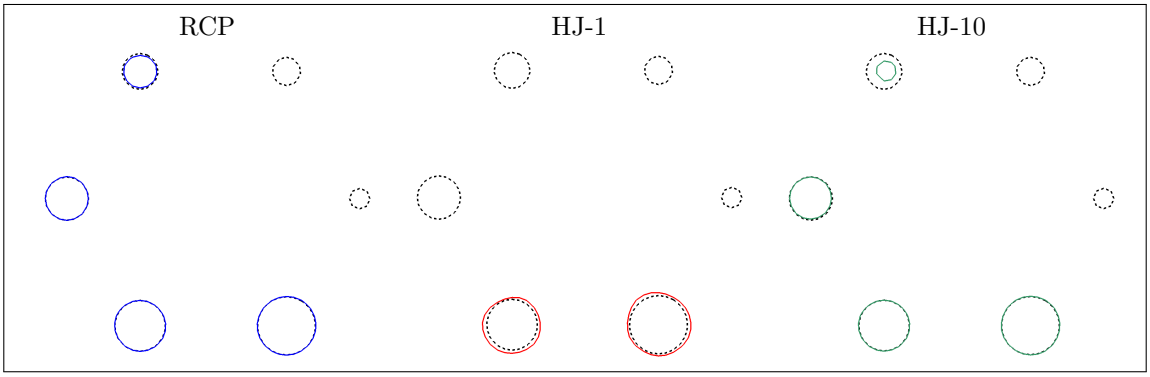


Figure 8: Disks rotation test case: interface shapes after a full rotation for a grid resolution of 64^2 cells. The reference solution is plotted in gray dashed lines.

6.3.1. Simple advection - One circle rotation

We now propose to study more quantitatively the impact of each method on a unique rotating circle in the same setup as above. A mesh convergence is performed focusing on the capacity to conserve shape, volume and on the capacity to reinitialize the level set to a signed distance function.

The lowest resolution is equivalent to describing the circle with a diameter of $8h$, with a mesh size of 64^2 cells.For this resolution, the time step is fixed and set to $\Delta t = 1.25\text{e-}3$ and it is reduced proportionally for each finer meshes in order to keep a constant CFL number. The simulation is stopped after one full rotation and hence, when $t = 1$.

As illustrated on Fig. 9, all methods exhibit similar results for the $E^{L_\infty}_{\nabla\phi}$. However, it is clear that the RCP method exhibits a convergence rate that is one order higher than both HJ-1 and HJ-10 for the shape errors and the enclosed volume conservation.

6.4. Zalesak Disk Advection

6.4.1. Quantitative and qualitative study

This test case follows the one proposed in [28] to appraise the capacity of the reinitialization method to preserve sharp corners on the interface. In a $[0, 1]^2$ domain, a slotted disk is initially centered at $(0.5, 0.75)$ with a diameter of $D = 0.3$ and a slot of 0.05 of width and 0.25 of length. The velocity field is set to transport the interface in a counterclockwise rotation around the point $(0.5, 0.5)$ and is defined as:

$$(u, v) = (0.5 - y, x - 0.5) \tag{24}$$

As in the literature, the time step is fixed to $\Delta t = 2\pi/628$ for the corresponding mesh of 100^2 cells, and it is adapted proportionally for other meshes. The simulation is stopped after one full rotation of the slotted disk which corresponds at a time $t = 2\pi$.

In this test case, as in the previous ones, the numerical errors principally originate from the reinitialization.

For the coarsest mesh, as illustrated in Fig. 10, we observe that both RCP and HJ-10 successfully conserve the global shape and it seems that HJ-10 manages to conserve a better representation of the slot than RCP, which are both acceptable, regarding the mesh resolution. On the other side, the frequent reinitialization of HJ-1 introduces more smoothing/errors that leads to a round shape that hardly captures the expected solution. For the intermediate resolution, all methods succeed in preserving the global shape of the interface even if the RCP and HJ-10 methods show better conservation of the slot depth and the corners. For the finest resolution, all methods accurately preserve the shape.

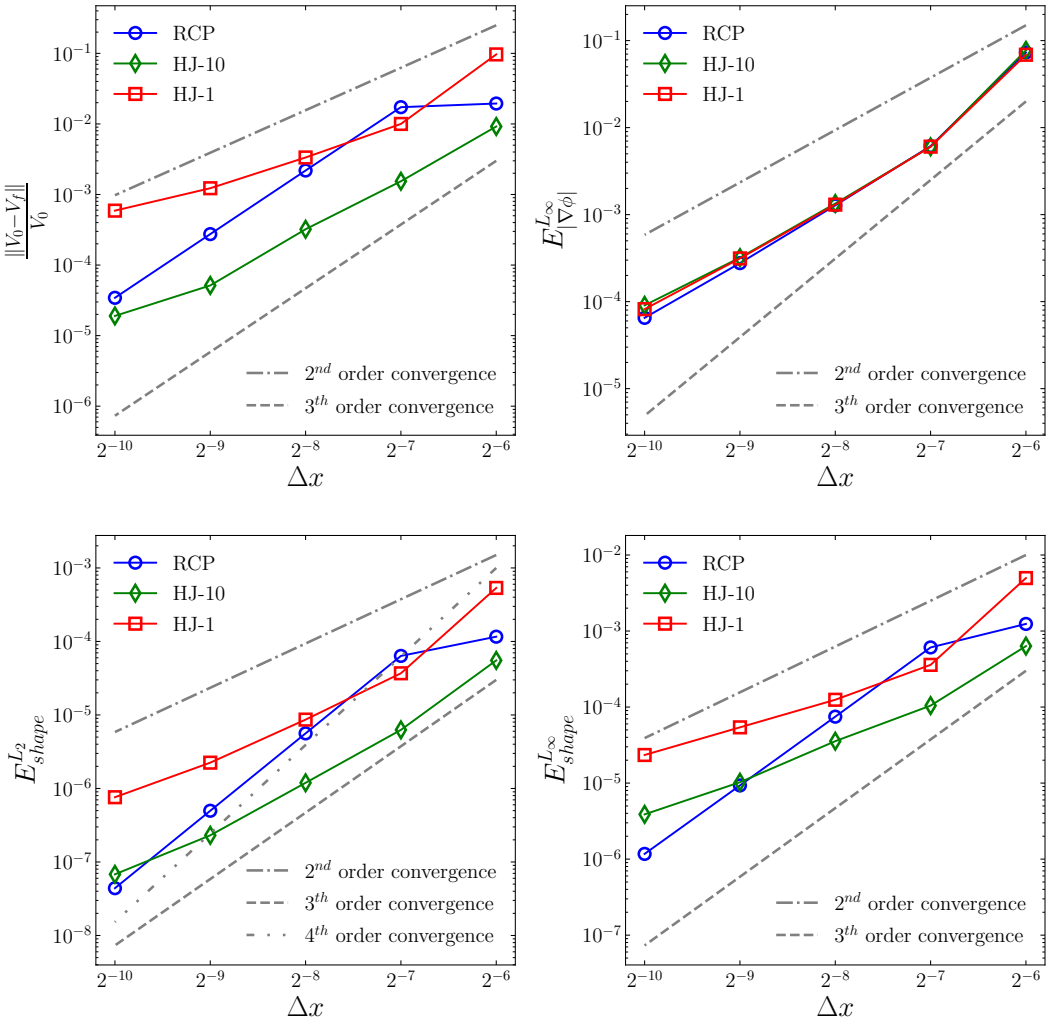


Figure 9: One circle rotation test case: error measures and convergence. From left to right and top to bottom: volume conservation, L_∞ norm of the error on the deviation to a signed distance function ($E^{L_\infty}_{\nabla\phi}$), L_2 and L_∞ norm of shape errors ($E^{L_2}_{shape}$ and $E^{L_\infty}_{shape}$).

For analyzing numerical convergence, only shape error has been reported in Fig. 11. Indeed, precautions must be taken since the four sharp corners can lead to a simultaneous volume loss and gain that can cancel and thus bias the interpretations. Furthermore, the deviation of ϕ to be a signed distance function (i.e. $E_{\nabla\phi}$) is also biased due to the presence of the four sharp corners which introduces error on that measure because of the inherent kinks of the level set at those points. All methods show a similar convergence rate around second order on the L_2. However, for the L_∞ norm, the convergence is highly impacted because of the discussed sharp corners.

6.4.2. Kink detection sensitivity using the RCP method

As described in Sec. 5.4.2, the threshold ϵ used in the kink detection algorithm (algorithm 3) is set to $\epsilon = d/2$, from geometric considerations. Here, in order to gauge the sensitivity of this criterion on detected kinks and on the RCP, a study is performed on the Zalesak disk test cases with a mesh resolution of 100^2 cells for $\epsilon/d = \{0.25, 0.5, 0.75\}$.

As excepted, as observable in Fig. 12, the higher the threshold ϵ, the lower the number of detected kinks.

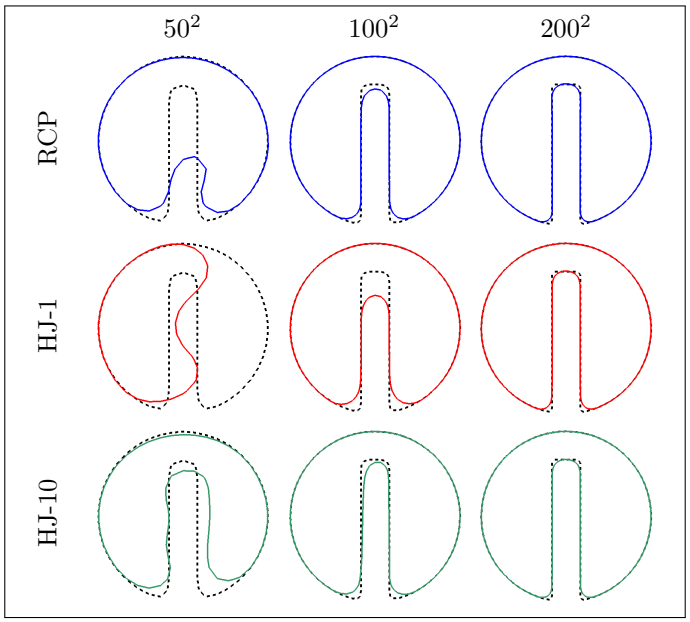


Figure 10: Zalesak disk test case: interface shape after one rotation ($t = T$). The dashed line represents the initial interface for the associated mesh.

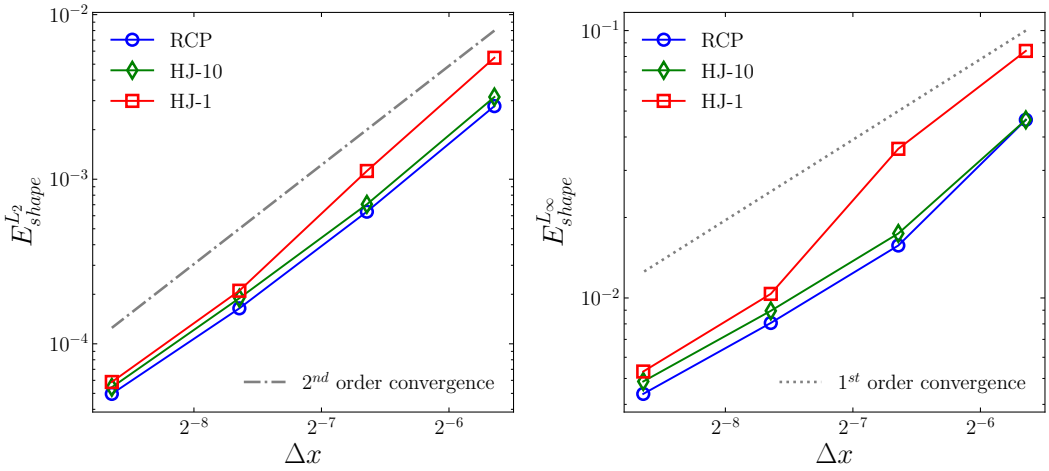
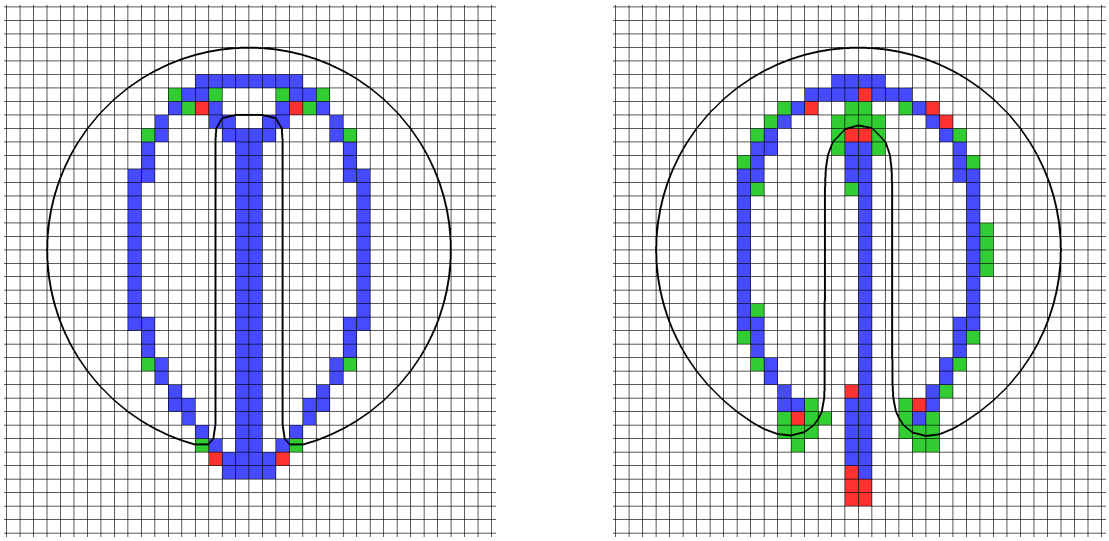


Figure 11: Zalesak disk test case: error measures and convergence. L_2 (left) and L_∞ (right) norm of shape error ($E^{L_2}_{shape}$ and $E^{L_\infty}_{shape}$).

This variation is relatively small as it can be seen in the figures, even after a full rotation of the disk. They are all present in the vicinity of the medial axis. Due to numerical errors, inducing smoothing of the interface, we can observe in the second picture less kinks near the initial corners. This is expected as the detector is based on a geometric approach, related to the interface normal variation, i.e. the curvature. Furthermore, as presented in table 1, these small variations on the detected kinks have almost no impact on the numerical results of the RCP method.



(a) At initialization (b) After a full rotation (at $t = T$)

Figure 12: Interface shape and kinks at initialization (left) and after a full rotation (right) for different values of the threshold ϵ (see Sec. 5.4.2). The smaller the value of ϵ the more kinks are detected. For comprehension, only kinks part of Ω_{CP} are shown. Cells are colored for $\epsilon/d = 0.75$: in blue; for $\epsilon/d = 0.5$: in blue and red; for $\epsilon/d = 0.25$: in blue, red and green. The interface for all three values are indistinguishable and drawn in black.

ϵ/d	$\lvert\frac{V_0 - V_{final}}{V_0}\rvert$	$E^{L_2}_{\lvert\nabla\phi\rvert}$	$E^{L_2}_{shape}$	$E^{L_\infty}_{shape}$
0.25	5.34e-4	1.29e-03	6.38e-04	1.57e-02
0.5	4.39e-4	1.15e-03	6.35e-04	1.57e-02
0.75	3.30e-4	1.11e-03	6.36e-04	1.58e-02

Table 1: Numerical results of sensitivity study of the threshold ϵ on the Zalesak disk: the enclosed volume error, the L_2 norm of the deviation to a signed distance function ($E^{L_2}_{\lvert\nabla\phi\rvert}$), the L_2 norm of the shape error ($E^{L_2}_{shape}$) and L_∞ norm of the shape error ($E^{L_\infty}_{shape}$).

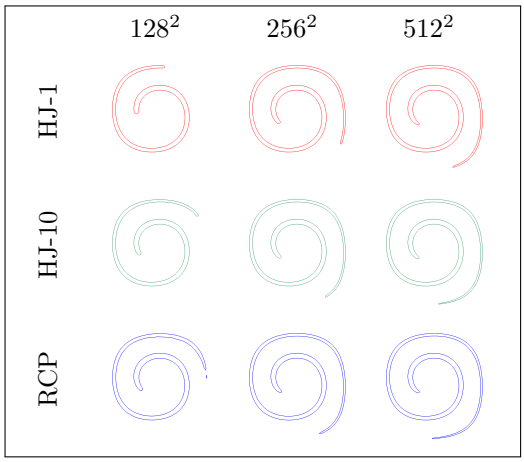
6.5. Single vortex 2D

Following the widely studied test case [29–31] to test the ability of the level set method to resolve and maintain thin filaments. A $[0, 1]^2$ domain is considered, with a circle of a diameter $D = 0.3$ is initialized at the coordinates (0.5, 0.75). The velocity field $(u, v) = (\frac{\partial \Psi}{\partial y}, -\frac{\partial \Psi}{\partial x})$ is derived from the stream function:

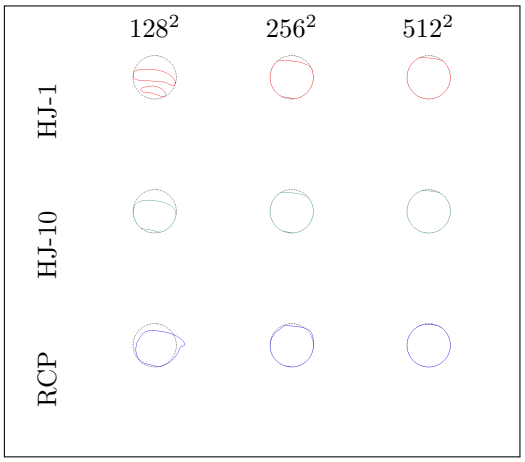
$$\Psi = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos(\frac{\pi t}{T}).$$

The term $\cos(\frac{\pi t}{T})$ which appears in the velocity field definition ensures that the flow returns to its initial state at the time T and that the maximal deformation appears at T/2. We have fixed $T = 8$ and the time step to $\Delta t = 1.6\text{e-}3$ for the coarsest grid 128^2. For the other meshes, the time step is reduced in order to keep a constant CFL number.

Fig. 13 illustrates the ability of the advection equation coupled with the three reinitialization procedures to conserve thin filaments at maximal deformation (Fig. 13a) and to correctly reverse to the initial circular shape (Fig. 13b). Globally, all methods suffer from large errors on the coarse grid, particularly on the trailing and the heading of the deformed interface, errors that will diminish with mesh refinement. As emphasized by Herrmann [15], the underlying reason is twofold. First, errors arise from the advection of the level set and from the displacement of the interface introduced by the reinitialization procedure. This can result in



(a) Interface at $t = T/2$.



(b) Interface $t = T$, the dashed line is the initial interface.

Figure 13: Single vortex 2D test case: interface shapes at $t = T/2$ (left) and $t = T$ (right).

the dismantlement of thin filament structures. Secondly, errors are prominent because the trailing filament thickness falls below grid resolution and thus cannot be captured by standard advection methods.

All methods have comparable results for maximal deformation and when the interface returns to its initial shape. Nevertheless, we can notice that for the coarsest mesh, the RCP method seems to produce better results for capturing the trailing interface.

When looking closely at the convergence rate when $t = T$, as reported in Fig. 14, all three methods produce comparable results. Also, for both the enclose volume and the L_2 norm for the shape error, they all exhibit 2^{nd} order convergence rate and between 1^{st} and 2^{nd} order for the L_∞ norm. However, for the L_∞ norm error on the deviation to be a signed distance function, the RCP method gives a higher order converge rate of order between 4 and 5 while HJ-1 and HJ-10 are 2^{nd} order.

6.6. Single vortex 3D

A 3D adaptation of the test case was proposed by LeVeque [30] and applied by Enright et al. [31] to tests the ability of the level set method to resolve and maintain thin filaments with deformation in both x-y and x-z planes.

The same configuration as in 2D is adapted here for a 3D test case. A $[0, 1]^3$ domain is considered where a sphere of radius 0.15 is initialized at the coordinates $(0.35, 0.35, 0.35)$. The resulting velocity field is given by:

$$u = 2\sin^2(\pi x)\sin(\pi y)\sin(\pi z)\cos(\frac{\pi t}{T})$$
$$v = 2\sin(\pi x)\sin^2(\pi y)\sin(\pi z)\cos(\frac{\pi t}{T})$$
$$z = 2\sin(\pi x)\sin(\pi y)\sin^2(\pi z)\cos(\frac{\pi t}{T}).$$

Where the same term $\cos(\frac{\pi t}{T})$ ensures that the interface will go back to its initial state at $t = T$ and that maximal deformation is obtained at $t = T/2$. Here this term is fixed at $T = 3$.

Fig. 15 and table 2 shows that the new method is robust also in 3D and gives similar results than the ones obtained in the 2D case. Both HJ-10 and RCP present comparable results, while HJ-1 suffers from volume loss leading to errors on the final shape.

6.7. 2D column at equilibrium - spurious currents

In order to validate that the proposed reinitialization method captures correctly surface tension forces computation, which are dominant at small scales, we propose to study the parasitic currents arising from

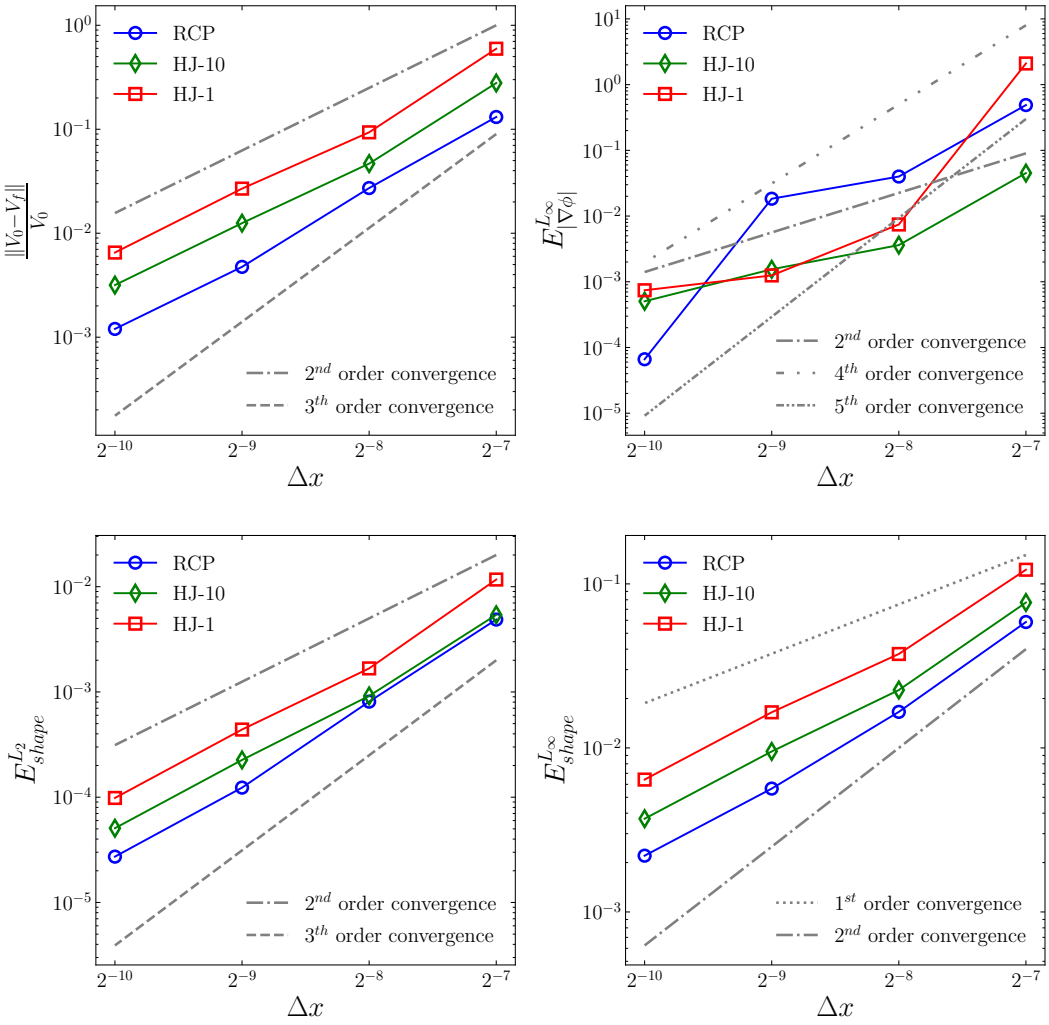


Figure 14: Single vortex 2D test case: error measures and convergence. From left to right and top to bottom: volume conservation, L_∞ norm of the error on the deviation to a signed distance function ($E^{L_\infty}_{\nabla\phi}$), L_2 and L_∞ norm of shape error ($E^{L_2}_{shape}$ and $E^{L_\infty}_{shape}$).

	$\lvert\frac{V_0 - V_{final}}{V_0}\rvert$		$E^{L_2}_{\lvert\nabla\phi\rvert}$		$E^{L_2}_{shape}$		$E^{L_\infty}_{shape}$	
Method	128^3	256^3	128^3	256^3	128^3	256^3	128^3	256^3
RCP	8.80e-02	1.85e-03	2.46e-06	1.75e-08	3.30e-03	1.97e-04	1.55e-01	1.70e-02
HJ-1	3.89e-01	6.22e-02	4.33e-07	2.35e-07	4.15e-03	1.17e-03	1.59e-01	9.97e-02
HJ-10	5.15e-02	1.03e-02	1.57e-06	3.95e-08	1.98e-03	1.45e-04	8.68e-02	1.54e-02

Table 2: Single vortex 3D test case: numerical results for the enclosed volume error, L_2 norm of the deviation to a signed distance function ($E^{L_2}_{|\nabla\phi|}$), L_2 ($E^{L_2}_{shape}$) and L_∞ ($E^{L_\infty}_{shape}$) norms of the shape error. Results are given for the RCP, HJ-1 and HJ-10 methods, for 128^3 and 256^3 grids, at a time $t = T$.

discretization errors in the static column test case. The equilibrium of a drop or bubble, at rest in the absence of gravity, implies the absence of momentum and thus a null velocity field. In practice, because of numerical errors emanating from the interface position, normal and curvature computation appearing in

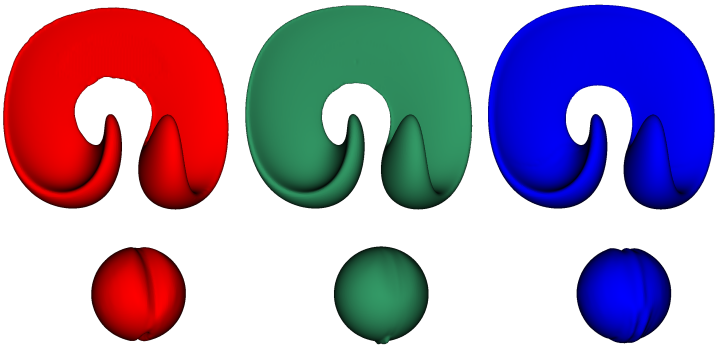


Figure 15: Single vortex 3D test case: interface shape for 256^3 grid, at $t = T/2$, using the reinitialization method: HJ-1 (red), HJ-10 (green), RCP (blue).

Eq. (7), so called *parasitic currents* [14, 16, 18] indeed arise. Particularly, as explained by Francois et .al [14] using a balanced-force approach within the CSF model while imposing an exact curvature should reduce parasitic currents up to machine precision.

A 2D column at rest is considered, with a diameter $D = 2R = 0.4$ at the center of $[0, 1]^2$ square. In order to solely focus on surface tension, both density and viscosity are constant for all the simulations and equal to 1 in each phase. The Laplace number $La = \sigma\rho L/\mu^2$ is obtained by varying the surface tension coefficient σ, the reference length is chosen as $L = D$. No-slip conditions are applied to all boundaries. A centered second order implicit scheme is used for the inertial term of Eq. (2). Simulations have been conducted until a numerical steady state has been attained, i.e. when spurious currents appear to have reached a minimum. The maximum capillary number $\mathrm{Ca}^*_{max} = \mu|\mathbf{u}|_{max}/\sigma$ is compared for various Laplace numbers for various meshes. In practice, we study the scaled capillary number $\mathrm{Ca}_{max} = \mathrm{Ca}^*_{max}/U_\sigma$ where the characteristic velocity U_σ is defined as $U_\sigma = \sqrt{\sigma/(\rho D)}$. Also, time is adimenzionalized as $t_\sigma = t/T_\sigma$, with $T_\sigma = \sqrt{\rho D^3/\sigma}$. The numerical time step is ensuring the revised capillary time step constraint from the work of Denner et al. [18] regarding the stability of flows subjected to surface tension. For a static case, it leads to the following condition:

$$\Delta t_\sigma^{static} \leq \sqrt{\frac{(\rho_1 + \rho_2)h^3}{2\pi\sigma}}. \tag{25}$$

6.7.1. Fixed Laplace number

The first study focuses on the spatial converge of parasitic currents for a given Laplace number ($La = 120$), by considering a mesh refined from 16^2 to 256^2 cells. The time step is kept constant for all resolutions, i.e. $\Delta t =$ 3e-6, thus always respecting the constraint given by Eq. (25) for the finest mesh.

The convergence rates of the error on the enclosed volume and Capillary number are shown in Fig. 16. As the surface is far from any kink, the RCP method gives a high order converge rate of 4 on the level set volume error while the HJ-1 and HJ-10 exhibit 2^{nd} order accuracy. The Ca_{max} diminishes at 2^{nd} order rate for all methods. Thus, the RCP method converges accurately for this surface tension test case.

6.7.2. Varying Laplace number

Within the same set up, we study the dynamic response of our method with an increasing Laplace number, i.e. increasing surface tension. For high Laplace numbers, small numerical errors will induce important spurious currents that can eventually make simulations unstable.

The mesh size is fixed at 64^2. The time step is scaled along with the Laplace number to match with the time step restriction: $\Delta t = \{$6e-5, 2e-5, 0.6e-5, 0.5e-5$\}$ for corresponding La $= \{120, 1200, 12000, 120000\}$.

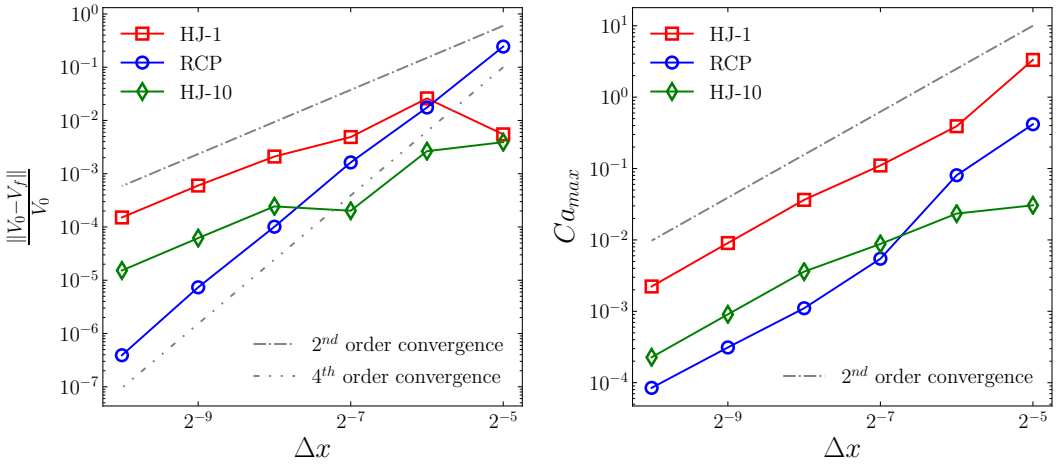


Figure 16: 2D column at equilibrium test case: convergence study for the enclosed volume variation (left) and Ca_{max} (right) at $La = 120$ for HJ-1, HJ-10 and RCP when a steady state has been reached.

	Ca_{max}			
Method	120	1200	12000	120000
RCP	9.63e-07	3.02e-06	9.58e-06	3.00e-05
HJ-1	1.78e-06	3.63e-06	1.02e-05	3.07e-05
HJ-10	9.82e-07	2.97e-06	8.97e-06	2.53e-05

Table 3: 2D column at equilibrium test case: Ca_{max} for varying Laplace number with 32^2 cells. Numerical results of the static column case of the Ca_{max} values for $La = \{120, 1200, 12000, 120000\}$, with a mesh size of 64^2.

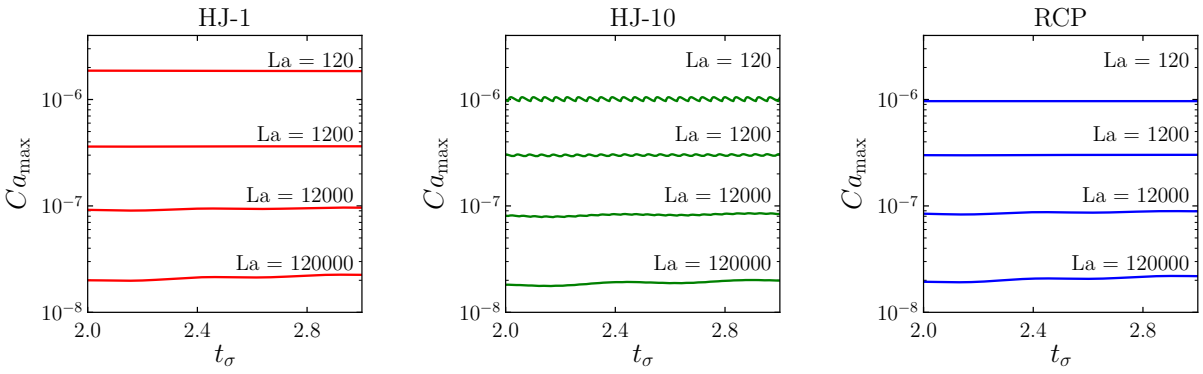


Figure 17: 2D column at equilibrium test case: convergence of Ca_{max} when varying the Laplace number for HJ-1 (left), HJ-10 (center) and RCP (right) method in the interval $t_\sigma \in [2.0, 3.0]$ for 64^2 cells.

Fig. 17 and table 3 show very similar results for all three approaches. It should be noted, as clearly visible on the central graph of Fig. 17 concerning HJ-10, that performing the reinitialization process every 10 time steps brings oscillations on Ca_{max}. Nevertheless, a question may arise on the possible impact of these small oscillations on complex two-phase flow simulations. Conversely, performing the reinitialization at every time step produces more consistent results as shown in the RCP and HJ-1 respective figures.

6.8. Droplet impact

We have noticed that, in some configurations, the HJ method may be unable to correctly capture topological changes. This occurs, for example, when small droplets impact a liquid pool at low velocity. Hence, we tested the capacity of the proposed method to adequately handle topological changes without restriction by considering the simple test case of a falling small droplet positioned above an initially steady volume of water surrounded by quiescent air. The purpose of this validation test is to capture the merging of the drop with the underlying surface, and not to study the behaviour of the impact that occurs later.

For relating to a realistic configuration, the drop is initialized at terminal velocity (see [32] for more details). In our case, the drop diameter is $D = 0.06mm$ and the associated terminal velocity is $U_0 = 0.27m.s^{-1}$. For the sake of simplicity, the initial velocity field is set uniformly to U_0 inside the drop and null elsewhere. No-slip conditions are applied on the boundaries. An explicit WENO $5,3$ scheme is used for the inertial term of Eq. (2). Herein, surface tension and gravity have been taken into consideration. The physical properties of the two phases, are ~~the one of water and air used in the next section,~~ detailed in table 4.

Phase	$\rho\ (kg.m^{-3})$	$\mu\ (Pa.s)$	$\sigma\ (N.m^{-1})$
Water	$\rho_l = 998.2$	$\mu_l =$ 1.00e-3	$72.8e-3$
Air	$\rho_g = 1.2$	$\mu_g =$ 18.2e-6	

Table 4: Droplet impact test case: physical properties of water and air phases.

The 2D domain is of size $[4D, 5D]$ and the pool depth is $2D$ above the bottom boundary. The mesh size is defined by the value of $N_c = D/h$ which represents the number of cells per drop diameter. The time step is set to $\Delta t = \{$9.604e-05, 3.395e-05, 1.200e-05$\}$ for meshes associated to $N_c = \{4, 8, 16\}$.

For the coarsest mesh as shown in Fig. 18, both methods resolving the H-J equation (HJ-1 and HJ-10) are unable to capture the merging of the drop with the surface of water. It appears that the H-J reinitialization introduces relatively important displacement of the interface position. In these cases, the drop acts as if it was *levitating* above the surface of water and, moreover, loses volume over time and vanishes without producing the expected cavity. On the other hand, the RCP method is able to correctly handle the merging of water, as explained below, even though the reinitialization is performed at every time step. The results for the medial mesh in Fig. 19 show that the HJ-1 still misses to capture the coalescence while both HJ-10 and RCP therein succeeded. For the finest mesh, as presented in Fig. 20, all methods produce satisfactorily comparable results.

We deem that one of the possible reasons for the inability of the HJ equation to handle topological changes for coarser meshes, in this configuration, comes from the presence of a kink between the two surfaces before merging. It introduces too much perturbations in the reinitialization using the HJ equation. The use of the kink detection conjointly with a HJ approach is beyond the scope of the article but could be explored in future works. The finer mesh is less impacted by the existence of the kink as the impacted region is relatively smaller. Moreover, the less frequent the HJ equation is applied, the *easier* the two surfaces can merge. As discussed in previous sections, this strongly questions the reinitialization frequency parameter that appears to be dependent on the underlying interface dynamic and hence not trivial to determine. On the other hand, the proposed approach overcomes these issues thanks to the accurate kink detection algorithm coupled to the adapted geometrical reinitialization procedure.

We believe that this issue may arise in various situations - and sometimes unnoticed - particularly when topology changes happen at small velocity for small structures such as droplets, bubbles and thin films, for example, during atomization, breaking waves or film instabilities. Consequently, when the HJ equation is considered, it becomes difficult for such applications, to correctly set the reinitialization frequency since many structures coexist at different sizes and velocities, relative to mesh refinement. The RCP brings a robust method to correctly reinitialize the level set without compromising topological changes.

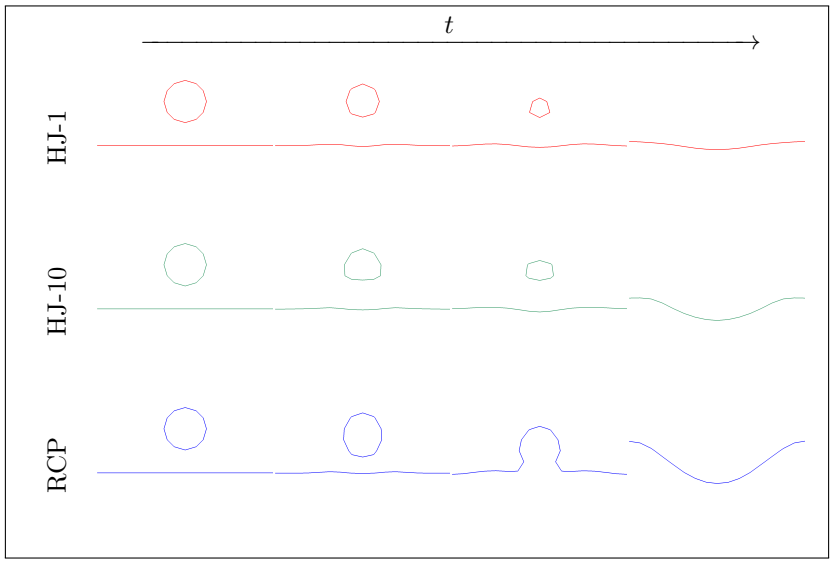


Figure 18: Droplet impact test case: interface at time $t = \{0, 2.4\text{e-}3, 4.8\text{e-}3, 9.6\text{e-}3\}$ for the coarse mesh ($N_c = 4$).

7. Applications

7.1. Bubble rise

We apply the proposed algorithm to the bubble rise problem, based on Hnat et Buckmaster experiments [33]. Among all cases, we followed numerical simulation studies as in [34, 35] for spherical-cap case A at $Re = 19.6$. The bubble diameter $D = 12.2mm$ is initially at rest at position $[0, 2.5D]$ in an axisymmetric domain $\Omega = [0, 4D] \times [0, 16D]$. Slip condition is applied to the right boundary and wall condition at the top and bottom boundaries. Fluids' physical parameters are given in table 5. Gravity magnitude is $g = -9.81m.s^{-1}$.

Phase	$\rho\ (kg.m^{-3})$	$\mu\ (Pa.s)$	$\sigma\ (N.m^{-1})$
Liquid	$\rho_l = 875.5$	$\mu_l = 118.0\text{e-}3$	32.2e-3
Gas	$\rho_g = 1.0$	$\mu_g = 1.0\text{e-}3$	

Table 5: Bubble rise test case: physical properties of liquid and gas phases.

When accelerating, the bubble deforms, becoming concave at the bottom and flatter on top. It reaches a steady-state with associated terminal velocity $V_\infty = 0.215\, m.s^{-1}$. We measured the mean upward velocity inside the bubble by computing:

$$V_b = \frac{\int_{\Omega_b} v(\mathbf{x}) d\mathbf{x}}{|\Omega_b|} = \frac{\int_\Omega \chi(\mathbf{x}) v(\mathbf{x}) d\mathbf{x}}{\int_\Omega \chi(\mathbf{x}) d\mathbf{x}}$$

where we recall that χ is the characteristic function associated to the bubble. In order to obtain a more accurate measure of the mean velocity, in that post-processing step, χ has been computed with the sharp method from [36]. We have used two meshes, the coarsest (resp. the finest) consists of 128×512 (resp. 256×1024) cells, equivalent to 32 (resp. 64) cells per diameter. The time step is kept constant at $\delta t = 4\text{e-}4\, s$ (resp. $\delta t = 2\text{e-}4\, s$) for the coarsest (resp. finest) mesh so as to attain a CFL number of around 0.3 at steady-state.

As reported in Fig. 21, after $0.3\, s$ of simulation, the numerical simulation using finest mesh converges to a terminal velocity $V_b = 0.212\, m.s^{-1}$, i.e. less than 2% error, which concurs well with experimental and

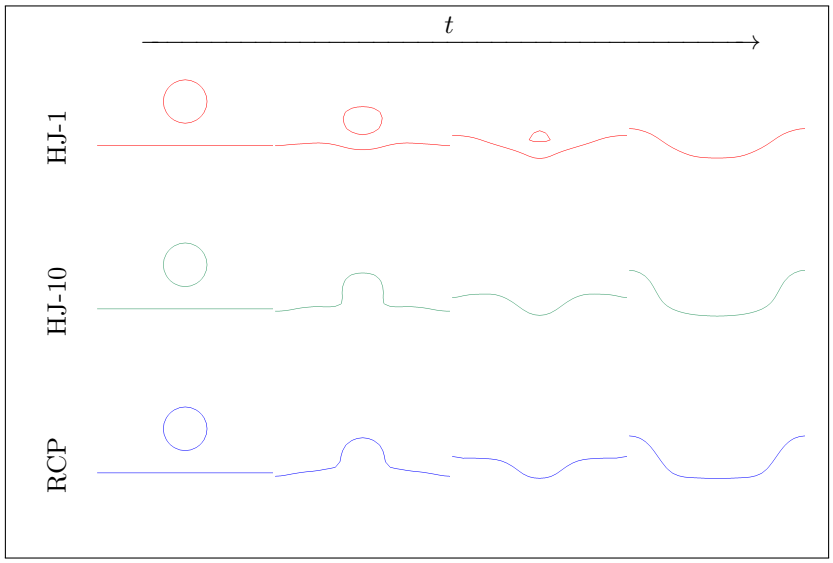


Figure 19: Droplet impact test case: interface at time $t = \{0, 6.8\text{e-}3, 1.0\text{e-}2, 1.4\text{e-}2\}$ for the medial mesh ($N_c = 8$).

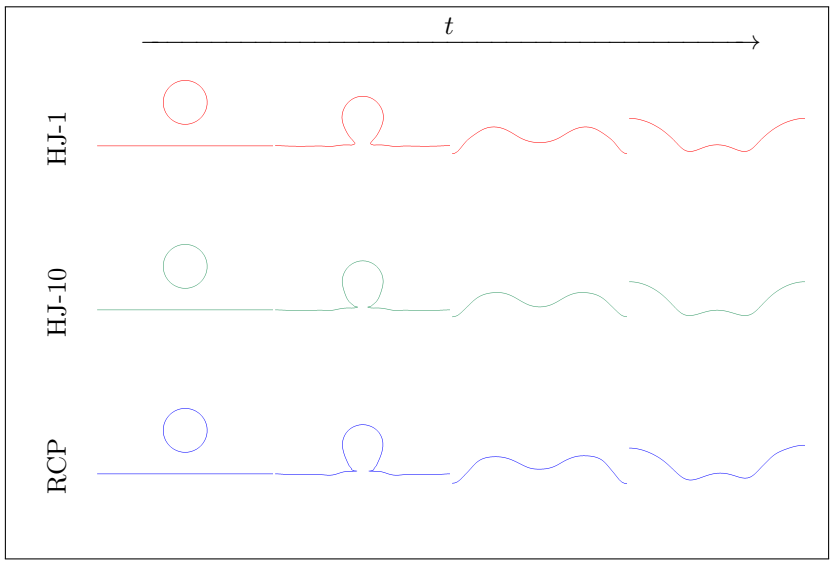


Figure 20: Droplet impact test case: interface at time $t = \{0, 7.2\text{e-}3, 1.1\text{e-}2, 1.6\text{e-}2\}$ for the fine mesh ($N_c = 16$).

numerical results [33, 34]. Also, as illustrated in Fig. 21, the simulated shape of the bubble is satisfactory as well as the streamlines showing internal recirculation and vortices in the wake. On the other hand, simulating using the coarsest mesh shows an under estimation of the terminal velocity with around 8% error, in the same order of magnitude as [35] for the same spatial discretization. This noticeable error is correlated to a mass loss of the level set, induced by diffusion at the *corner* of the bubble. Indeed, as illustrated in Fig. 22, the omnipresence of inherent kinks in the vicinity of the interface prevents from using the high-order closest point computation, as discussed in Sec. 5.5.2, thus inducing a slow but sensible retraction of the interface towards the center of the bubble. Even if the interface becomes smoother at that particular location, the bubble dynamic imposes a persistent kink at its base. However, with the finest mesh, kinks are farther from the interface, i.e. not in the closest point interpolation stencil, and numerical errors are much reduced, leading to accurate results.

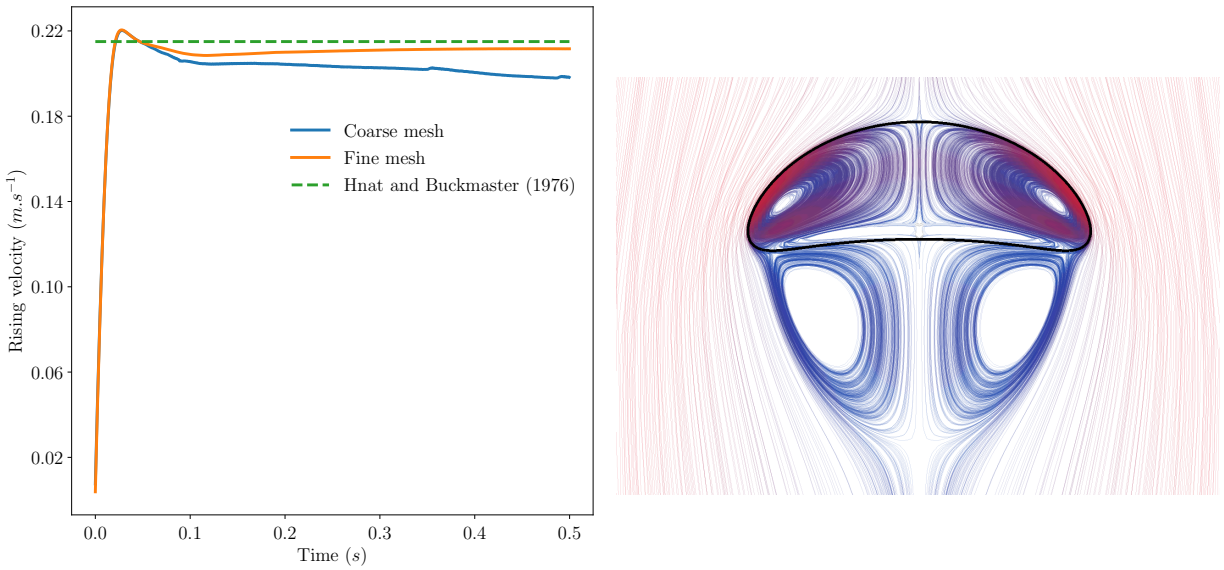


Figure 21: Bubble rise test case: bubble velocity over time (left) and streamlines of relative flow velocity (right) at $t = 0.4\,s$. For the later, color gradient goes from blue (null velocity) to red ($0.2\,m.s^{-1}$).

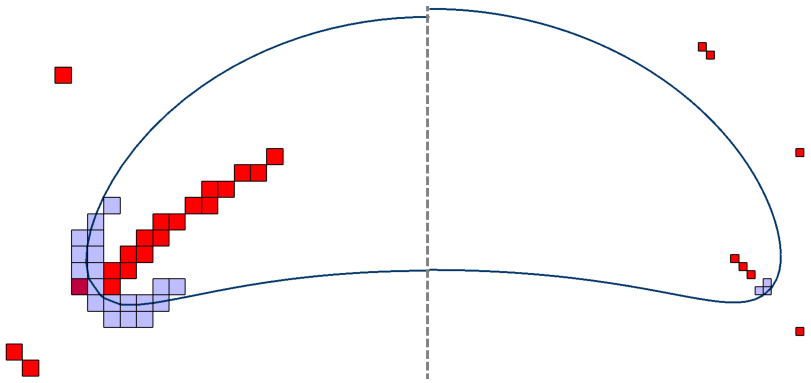


Figure 22: Bubble rise test case: comparison of detected kinks (colored in red) for coarse (left) and fine (right) meshes, at $t = 0.14\,s$. Interface cells close to kinks (colored in light blue) are not reinitialized with the closest point distance (see Sec. 5.5.2).

This demonstrates that, for capturing such particular cases, one has to use a sufficiently fine mesh in order to accurately solve interface regions of high curvature. Also, as discussed in Sec. 5.5.2, these numerical errors are induced by the closest point computation that is sensitive to kinks because of interpolation, a problem that will be solved by adapting the closest point algorithm to such regions. This point will be the object of future works.

Finally, for such simulation where the interface consists of a single bubble or droplet, a global mass redistribution method (as in [34] for example) should be applied on the level set at every time step in order to ensure exact volume conservation. Without changing the proposed strategy, such methods could be used as a post-processing to the proposed algorithm.

7.2. Dam break

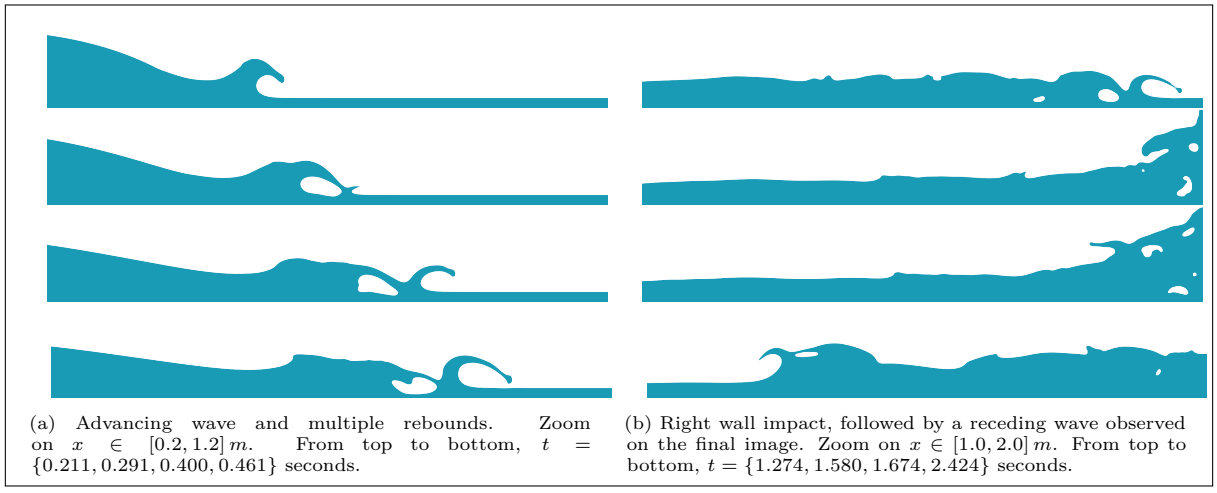
We finally propose to stress the RCP algorithm and test its robustness on the complex two-phase problem referred to as the dam break. The experimental works of Jánosi et al. [37] have been taken as a reference for

comparing numerical simulation. Even though the case is sensitive to initial conditions, the multiple bounces of the advancing wave, the captured air tubes and the eventual ejection of small drops represent challenges for all interface methods.

We based our simulations on Jánosi wet bed case B. The water level is initialized up to a height $d_0 = 150\,mm$ for $x \in [0, 380]\,mm$ and down to $d = 18\,mm$ for the rest of the domain. Physical parameters are the same as the one given in the droplet impact test case (see table 4). Free-slip conditions are used everywhere but on the bottom where a wall condition has been imposed.

First, a simulation is conducted in 2D in a domain of size $[0, 2]\,m \times [0, 1.2]\,m$. We chose to reduce the length of the canal because the most important events occur around $1\,m$ and in order to save computational time. Also, this permits observing earlier the splash-up on the boundary and augment the complexity of the geometry. The mesh is made of 800×480 cells. The time step is controlled so as the CFL number never exceeds 0.25, leading to a time step varying between $0.1\,ms$ to $0.3\,ms$. In Fig. 23, we reported water phase position over time. The result concurs satisfactorily with photographs available in [37]: very early, a mushroom shape front erects from the corner of the virtual gate which afterwards breaks over the wet bed resulting in the capture of an ellipsoidal air tube. Later on, splash-ups are observed similar patterns with air entrapment while air tubes in the wake break near the surface. Finally, the front hits the right wall and a receding wave arise.

While we observe several small structures in the simulation, the mesh cannot captures the very fine bubbles as visible in the experiment. However, we see that RCP is able to adequately capture topology changes, thin jets, as well as trapped bubbles. In order to help visualize the underlying RCP algorithm, we show in Fig. 24 the position of detected kinks and the closest points, when the second wave is about to merge with the wet bed. We can see that most kinks are located on the medial axis, i.e. inherent kinks, and few other capture non-smooth regions of the level set. Thanks to this detection, closest points can be used to reinitialize the level set.



(a) Advancing wave and multiple rebounds. Zoom on $x \in [0.2, 1.2]\,m$. From top to bottom, $t = \{0.211, 0.291, 0.400, 0.461\}$ seconds.

(b) Right wall impact, followed by a receding wave observed on the final image. Zoom on $x \in [1.0, 2.0]\,m$. From top to bottom, $t = \{1.274, 1.580, 1.674, 2.424\}$ seconds.

Figure 23: Dam break test case, 2D simulation: water phase.

Finally, we extend the simulation in 3D. The third dimension is $1\,m$ profound, while, for computational savings, we have imposed a symmetry condition at $z = 0.5m$. The mesh is made of $400 \times 240 \times 100$ cells. Fig. 25 shows the surface of the advancing front at various time. After some time, triggered by numerical errors, capillary instabilities arise and eventually break the 2D flow, as observed in the experiment. After the second break-up, several *fingers* appear, splitting the surface into smaller structures of the order of the cell size inducing more complex interactions. A finer mesh would be necessary to capture more accurately small bubbles and droplets. Nevertheless, the overall dynamic of the bore front is satisfactorily reproduced.

Over time, the RCP algorithm manages to preserve the consistency of the surface permitting capturing

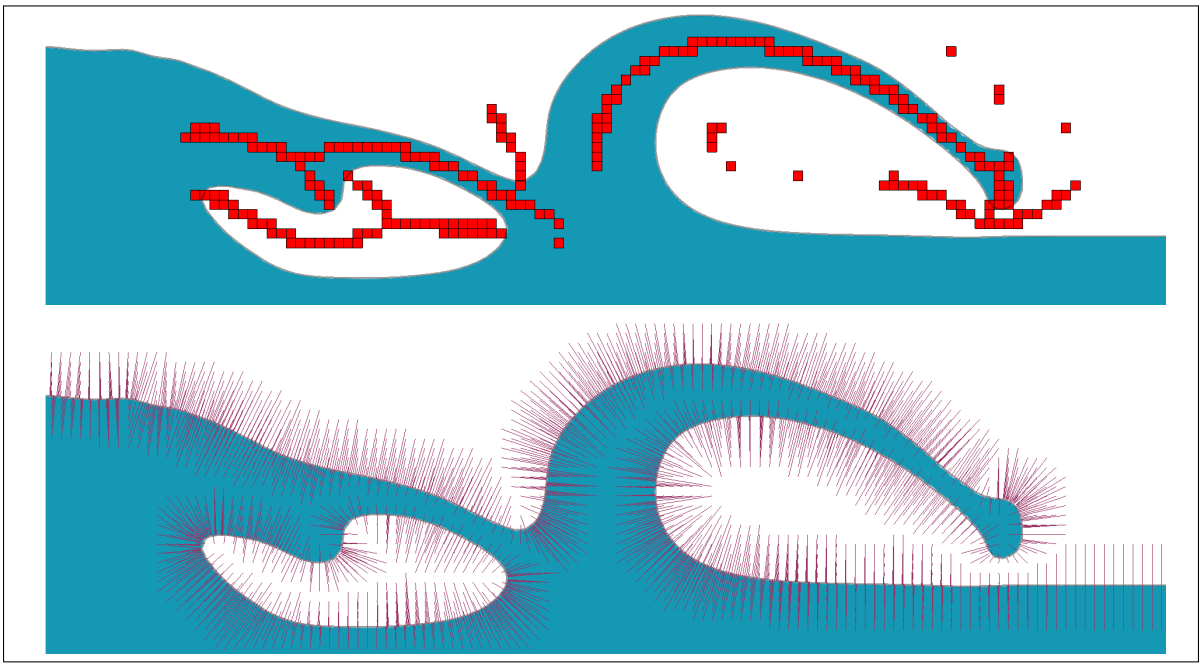


Figure 24: Dam break test case, 2D simulation: illustration of detected kinks (top) and closest points (bottom) near water surface at time $t = 0.461\,s$.

the expect dynamics. This test cases demonstrates that the proposed method is robust and can be effectively used on complex two-phase flows for applications.

8. Conclusion

In this paper, we have presented a robust and high order strategy to perform reinitialization in a level set framework. The proposed RCP method differs from the widely used Hamilton-Jacobi PDE approach by following a geometric approach. We use a gradient descent to find the closest points at the interface, in order to solve the eikonal equation, i.e. reinitializing the level set field. Furthermore, a new algorithm is introduced to reliably detect inherent and numerical kinks, also based on a geometric strategy. The RCP method is robust and accurate, even when performing the reinitialization systematically after solving the advection equation. This ensures a precise signed distance function at every time step. These conjoint methods both require very few given parameters, which are based on geometrical considerations. The method is tested on various benchmarks, from simple advection to two-phase flow simulation with surface tension and coalescence. They demonstrate better or at least equivalent results compared to the classical H-J approach. Finally, we show that the RCP method can successfully be used to simulate complex applications, in 2D and 3D, with various topology changes.

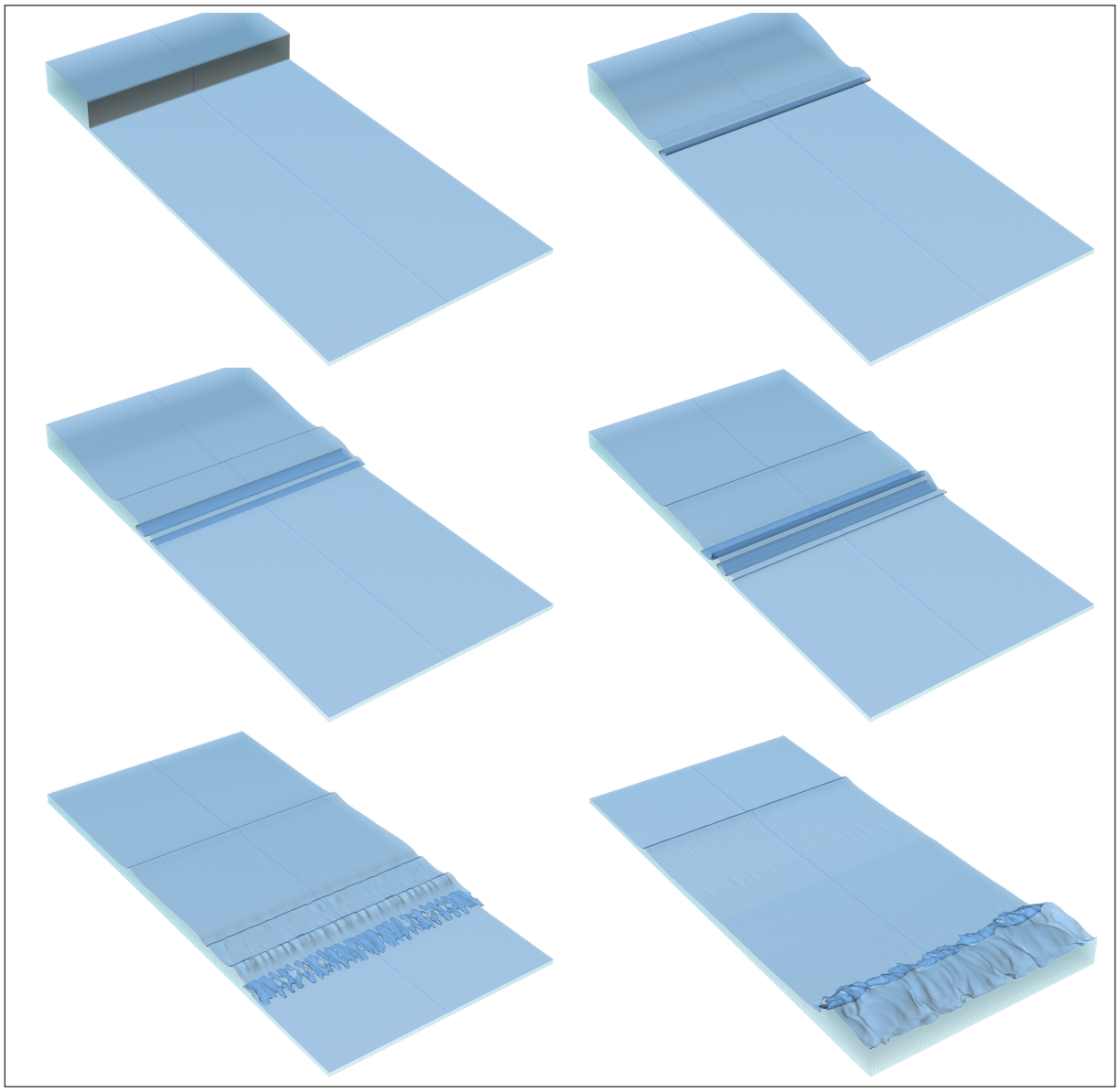


Figure 25: Dam break test case, 3D simulation: water phase. From left to right and top to bottom, $t = \{0.143, 0.211, 0.291, 0.400, 0.517\}$ seconds. We observe the formation of the wave, followed by the breaking, capturing the tube. Later on, multiple rebounds appear and tubes eventually break near the surface. In the last picture, we observe the receding wave after the front has hit the right wall.

Appendix A. Level set kinks detection and approximation

Equation 14 gives a first order approximation of the closest point at any point $\mathbf{x}$. In order to detect kinks, we need to evaluate that function for points in the vicinity of $\mathbf{x}$. For that purpose, we first consider Eq. (14) evaluated a point $\mathbf{x}_\xi$:

$$\mathrm{CP}\left(\mathbf{x}_\xi\right) \simeq \mathbf{x}_\xi - \frac{\phi\left(\mathbf{x}_\xi\right)}{\left|\nabla\phi\left(\mathbf{x}_\xi\right)\right|^2}\nabla\phi\left(\mathbf{x}_\xi\right)$$

and an approximation of it, being at an infinitesimally small distance to $\mathbf{x}$ in the direction $\overrightarrow{\xi}$, as:

$$\mathrm{CP}\left(\mathbf{x}_\xi\right) \simeq \mathbf{x} - \frac{\phi\left(\mathbf{x}\right)}{\left|\widetilde{\nabla}\phi(\mathbf{x}, \overrightarrow{\xi})\right|^2}\widetilde{\nabla}\phi(\mathbf{x}, \overrightarrow{\xi})$$

where $\widetilde{\nabla}\phi(\mathbf{x}, \overrightarrow{\xi})$ is a discrete off-centered approximation of $\nabla\phi\left(\mathbf{x}_\xi\right)$ in the bias direction $\overrightarrow{\xi}$, not to be confound with $\nabla\phi \cdot \overrightarrow{\xi}$, the variation of ϕ in that direction.

Generalization and implementation. When searching for a limited number of closest points in the vicinity of $\mathbf{x}$, one can derive several strategies. We have considered to use diagonal directions instead of direct mesh cells neighbours. This has the advantage to maximize the use of desired biases by considering diagonal cells and augment their number to eight instead of six in 3D for increased accuracy.

Let $\mathbf{x}_{i,j,k}$ be the center of the cell $\Omega_{i,j,k}$ of a 3D Cartesian mesh. We consider the eight vertices of that cell and associated directions, noted by variations of their index: $\overrightarrow{\xi}_{\alpha_x,\alpha_y,\alpha_z} = (\alpha_x, \alpha_y, \alpha_z)^t$, where $\alpha_x = \{-1, +1\}$ (resp. α_y and α_z) represents the left or right bias for the x (resp. y and z) direction. Hence, we can write a general formula for a biased first order scheme in the $\overrightarrow{\xi}_{\alpha_x,\alpha_y,\alpha_z}$ direction for the gradient computation:

$$\widetilde{\nabla}\phi(\mathbf{x}_{\overrightarrow{\xi}_{\alpha_x,\alpha_y,\alpha_z}}) \simeq \left(\alpha_x \frac{\phi_{i+\alpha_x,j,k} - \phi_{i,j,k}}{\delta x}, \alpha_y \frac{\phi_{i,j+\alpha_y,k} - \phi_{i,j,k}}{\delta y}, \alpha_z \frac{\phi_{i,j,k+\alpha_z} - \phi_{i,j,k}}{\delta z}\right)^t$$

which can be used similarly as Eq. (16) in algorithm 3 for computing the associated closest point approximations and hence detecting kinks.

Appendix B. Pseudo code for RCP algorithm

Algorithm 5 Pseudo code for RCP algorithm 1 - Main algorithm

$\phi_{RCP} \leftarrow \phi$

◇ `Pre-processing (step 1)`

Build Ω_{Kink} (see algorithm 3) ▷ $\Omega_{Kink}[c_{i,j,k}] = 1$ `for cells containing kinks`

◇ `Pseudo level set and fast low order smoothing (step 2)`

$\psi_{PseudoLS} \leftarrow$ `ComputePseudoLS(`$\phi, width$`)` ▷ `same width than` Ω_{Band}

$\Omega_{Kink+} \leftarrow \Omega_{Kink}$

$\Omega_{Kink+} \leftarrow$ `MarkNeighbours(`$\Omega_{Kink+}, 1$`)` ▷ `Augment the size of the kink mask`

$\Omega_{PseudoLS} \leftarrow \{(\Omega_{Band} \cap \Omega_{Kink+}) \setminus \Omega_{Stencil})\}$

for $c_{i,j,k} \in \Omega_{PseudoLS}$ **do**
  $\phi_{RCP}[c_{i,j,k}] \leftarrow \psi_{PseudoLS}[c_{i,j,k}]$
end

$\phi_{RCP} \leftarrow$ `ApplyLowOrderHJ(`$\Omega_{PseudoLS}, \phi_{RCP}, 20$`)`

$\phi'_{RCP} \leftarrow \phi_{RCP}$

◇ `Reinitialization with closest points (step 3)`

for $c_{i,j,k} \in \Omega_{CP}$ **do**
  $do\_reinit\_CP \leftarrow$ **true**
  if $c_{i,j,k} \in \Omega_\Gamma$ **then** ▷ `Sensitive cells near the interface`
    if $\sum \Omega_{Kink}[S_{c_{i,j,k}}] \neq 0$ **then** ▷ `At least one kink in the CP interpolation stencil`
      $do\_reinit\_CP \leftarrow$ **false**
    end
  end

  if do_reinit_CP **then**
    $\mathbf{x} \leftarrow$ `Coordinates(`$c_{i,j,k}$`)`
    $\mathbf{CP} \leftarrow$ `ComputeClosestPoint(`ϕ'_{RCP}, $c_{i,j,k}$`)`
    $\phi_{RCP}[c_{i,j,k}] \leftarrow \text{sgn}(\phi[c_{i,j,k}]) \times \|\mathbf{CP} - \mathbf{x}\|$
  end
end

◇ `Post-processing: low-order smoothing outside` Ω_{CP} `(step 4)`

$\phi_{RCP} \leftarrow$ `ApplyLowOrderHJ(`$\{\Omega \setminus \Omega_{CP}\}, \phi_{RCP}, 5$`)`

return ϕ_{RCP}

Algorithm 6 Pseudo code for RCP algorithm 1 - Utility functions for algorithm 5

▷ Mark with n all cells neighbouring at least one cell previously marked

function MarkNeighbours($pseudo\_LS, n$):
    $result \leftarrow mask$
    for $c_{i,j,k} \in \Omega_{Band}$ **do**
        if $c_{i,j,k} = 0$ and $\sum pseudo\_LS[c_{i-1:i+1,j-1:j+1,k-1:k+1}] \neq 0$ **then**
            $result[c_{i,j,k}] \leftarrow n$
        end
    end
    return $result$
end

function ComputePseudoLS($\phi, bandwidth$):
    $result \leftarrow 0$
    for $c_{i,j,k} \in \Omega_{\Gamma}$ **do**
        $result \leftarrow 1$
    end
    for $n \in [2 : bandwidth]$ **do**
        $result \leftarrow$ MarkNeighbours($result, n$)
    end
    for $c_{i,j,k} \in \Omega_{Band}$ **do**
        $result[c_{i,j,k}] \leftarrow result[c_{i,j,k}] \times \mathrm{sgn}(\phi[c_{i,j,k}])$
    end
    return $result$
end

function ApplyLowOrderHJ($mask, \phi, NbIt$):
    $\psi^0 \leftarrow \phi$
    for $n \in [1 : NbIt]$ **do**
        for $c_{i,j,k} \in mask$ **do**
            $\psi^n[c_{i,j,k}] \leftarrow$ LowOrderHJ($\psi^{n-1}, c_{i,j,k}$) ▷ First order in time and space
        end
    end
    return ψ^n
end

References

[1] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, Journal of Computational Physics 79 (1) (1988) 12–49. doi:10.1016/0021-9991(88)90002-2.
URL http://www.sciencedirect.com/science/article/pii/0021999188900022

[2] J. A. Sethian, Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts, Journal of Computational Physics 169 (2) (2001) 503–555. doi:10.1006/jcph.2000.6657.
URL http://www.sciencedirect.com/science/article/pii/S0021999100966579

[3] H. Zhao, A fast sweeping method for Eikonal equations, Mathematics of Computation 74 (250) (2004) 603–628. doi:10.1090/S0025-5718-04-01678-3.
URL http://www.ams.org/journal-getitem?pii=S0025-5718-04-01678-3

[4] M. Sussman, P. Smereka, S. Osher, A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow, Journal of Computational Physics 114 (1) (1994) 146–159. doi:10.1006/jcph.1994.1155.
URL http://www.sciencedirect.com/science/article/pii/S0021999184711557

[5] M. Sussman, E. G. Puckett, A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows, Journal of Computational Physics 162 (2) (2000) 301–337. doi:10.1006/jcph.2000.6537.
URL http://www.sciencedirect.com/science/article/pii/S0021999100965379

[6] G. Russo, P. Smereka, A Remark on Computing Distance Functions, Journal of Computational Physics 163 (1) (2000) 51–67. doi:10.1006/jcph.2000.6553.
URL http://www.sciencedirect.com/science/article/pii/S0021999100965537

[7] D. Hartmann, M. Meinke, W. Schröder, The constrained reinitialization equation for level set methods, Journal of Computational Physics 229 (5) (2010) 1514–1535. doi:10.1016/j.jcp.2009.10.042.
URL https://linkinghub.elsevier.com/retrieve/pii/S0021999109006032

[8] A. du Chéné, C. Min, F. Gibou, Second-Order Accurate Computation of Curvatures in a Level Set Framework Using Novel High-Order Reinitialization Schemes, Journal of Scientific Computing 35 (2) (2008) 114–131. doi:10.1007/s10915-007-9177-1.
URL https://doi.org/10.1007/s10915-007-9177-1

[9] D. L. Chopp, Some Improvements of the Fast Marching Method, SIAM Journal on Scientific Computing 23 (1) (2001) 230–244, publisher: Society for Industrial and Applied Mathematics. doi:10.1137/S106482750037617X.
URL https://epubs.siam.org/doi/abs/10.1137/S106482750037617X

[10] L. Anumolu, M. F. Trujillo, Gradient augmented reinitialization scheme for the level set method, International Journal for Numerical Methods in Fluids 73 (12) (2013) 1011–1041, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.3834. doi:10.1002/fld.3834.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3834

[11] A. Ervik, K. Y. Lervag, S. T. Munkejord, A robust method for calculating interface curvature and normal vectors using an extracted local level set, Journal of Computational Physics 257 (2014) 259–277. doi:10.1016/j.jcp.2013.09.053.
URL http://www.sciencedirect.com/science/article/pii/S0021999113006712

[12] I. Kataoka, Local instant formulation of two-phase flow, International Journal of Multiphase Flow 12 (5) (1986) 745–758. doi:10.1016/0301-9322(86)90049-2.
URL http://www.sciencedirect.com/science/article/pii/0301932286900492

[13] J. U. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, Journal of Computational Physics 100 (2) (1992) 335–354. doi:10.1016/0021-9991(92)90240-Y.
URL http://www.sciencedirect.com/science/article/pii/002199919290240Y

[14] M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, M. W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, Journal of Computational Physics 213 (1) (2006) 141–173. doi:10.1016/j.jcp.2005.08.004.
URL http://www.sciencedirect.com/science/article/pii/S0021999105003748

[15] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, Journal of Computational Physics 227 (4) (2008) 2674–2706. doi:10.1016/j.jcp.2007.11.002.
URL http://www.sciencedirect.com/science/article/pii/S0021999107004998

[16] M. Coquerelle, S. Glockner, A fourth-order accurate curvature computation in a level set framework for two-phase flows subjected to surface tension forces, Journal of Computational Physics 305 (2016) 838–876. doi:10.1016/j.jcp.2015.11.014.
URL http://www.sciencedirect.com/science/article/pii/S0021999115007548

[17] M. F. Trujillo, L. Anumolu, D. Ryddner, The distortion of the level set gradient under advection, Journal of Computational Physics 334 (2017) 81–101. doi:10.1016/j.jcp.2016.11.050.
URL http://www.sciencedirect.com/science/article/pii/S0021999116307045

[18] F. Denner, B. G. M. van Wachem, Numerical time-step restrictions as a result of capillary waves, Journal of Computational Physics 285 (2015) 24–40. doi:10.1016/j.jcp.2015.01.021.
URL http://www.sciencedirect.com/science/article/pii/S002199911500025X

[19] F. Luddens, M. Bergmann, L. Weynans, Enablers for high-order level set methods in fluid mechanics: Enablers for High Order Level Set Methods in Fluid Mechanics, International Journal for Numerical Methods in Fluids 79 (12) (2015) 654–675. doi:10.1002/fld.4070.
URL http://doi.wiley.com/10.1002/fld.4070

[20] Z. Solomenko, P. D. M. Spelt, L. Ó Náraigh, P. Alix, Mass conservation and reduction of parasitic interfacial waves in level-set methods for the numerical simulation of two-phase flows: A comparative study, International Journal of Multiphase

Flow 95 (2017) 235–256. doi:10.1016/j.ijmultiphaseflow.2017.06.004.
URL http://www.sciencedirect.com/science/article/pii/S0301932216304669
[21] R. Saye, High-order methods for computing distances to implicitly defined surfaces, Communications in Applied Mathematics and Computational Science 9 (1) (2014) 107–141, publisher: Mathematical Sciences Publishers. doi:10.2140/camcos.2014.9.107.
URL https://projecteuclid.org/euclid.camcos/1513732106
[22] P. Macklin, J. Lowengrub, Evolving interfaces via gradients of geometry-dependent interior Poisson problems: application to tumor growth, Journal of Computational Physics 203 (1) (2005) 191–220. doi:10.1016/j.jcp.2004.08.010.
URL http://www.sciencedirect.com/science/article/pii/S0021999104003249
[23] Notus – Computational Fluid Dynamics.
URL https://notus-cfd.org/
[24] K. Goda, A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows, Journal of Computational Physics 30 (1) (1979) 76–95. doi:10.1016/0021-9991(79)90088-3.
URL http://www.sciencedirect.com/science/article/pii/0021999179900883
[25] F. Desmons, M. Coquerelle, A generalized high-order momentum preserving (homp) method in the one-fluid model for incompressible two phase flows with high density ratio, Journal of Computational Physics 437 (2021) 110322.
[26] G.-S. Jiang, C.-W. Shu, Efficient Implementation of Weighted ENO Schemes, Journal of Computational Physics 126 (1) (1996) 202–228. doi:10.1006/jcph.1996.0130.
URL http://www.sciencedirect.com/science/article/pii/S0021999196901308
[27] R. Wang, R. J. Spiteri, Linear Instability of the Fifth-Order WENO Method, SIAM Journal on Numerical Analysis 45 (5) (2007) 1871–1901, publisher: Society for Industrial and Applied Mathematics. doi:10.1137/050637868.
URL https://epubs.siam.org/doi/abs/10.1137/050637868
[28] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, Journal of Computational Physics 31 (3) (1979) 335–362. doi:10.1016/0021-9991(79)90051-2.
URL http://www.sciencedirect.com/science/article/pii/0021999179900512
[29] J. B. Bell, P. Colella, H. M. Glaz, A second-order projection method for the incompressible navier-stokes equations, Journal of Computational Physics 85 (2) (1989) 257–283. doi:10.1016/0021-9991(89)90151-4.
URL http://www.sciencedirect.com/science/article/pii/0021999189901514
[30] R. J. LeVeque, High-Resolution Conservative Algorithms for Advection in Incompressible Flow, SIAM Journal on Numerical Analysis 33 (2) (1996) 627–665, publisher: Society for Industrial and Applied Mathematics. doi:10.1137/0733033.
URL https://epubs.siam.org/doi/abs/10.1137/0733033
[31] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A Hybrid Particle Level Set Method for Improved Interface Capturing, Journal of Computational Physics 183 (1) (2002) 83–116. doi:10.1006/jcph.2002.7166.
URL http://www.sciencedirect.com/science/article/pii/S0021999102971664
[32] R. Gunn, G. D. Kinzer, THE TERMINAL VELOCITY OF FALL FOR WATER DROPLETS IN STAGNANT AIR, Journal of the Atmospheric Sciences 6 (4) (1949) 243–248, publisher: American Meteorological Society Section: Journal of the Atmospheric Sciences. doi:10.1175/1520-0469(1949)006<0243:TTVOFF>2.0.CO;2.
URL https://journals.ametsoc.org/view/journals/atsc/6/4/1520-0469_1949_006_0243_ttvoff_2_0_co_2.xml
[33] J. Hnat, J. Buckmaster, Spherical cap bubbles and skirt formation, Physics of Fluids 19 (2) (1976) 182–194, cited By 127. doi:10.1063/1.861445.
URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-0001413156&doi=10.1063%2f1.861445&partnerID=40&md5=34890044f95df009c858413e9956ad6f
[34] Y. Wang, S. Simakhina, M. Sussman, A hybrid level set-volume constraint method for incompressible two-phase flow, Journal of Computational Physics 231 (19) (2012) 6438–6471. doi:https://doi.org/10.1016/j.jcp.2012.06.014.
URL https://www.sciencedirect.com/science/article/pii/S002199911200321X
[35] B. Lalanne, L. R. Villegas, S. Tanguy, F. Risso, On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method, Journal of Computational Physics 301 (2015) 289–307. doi:https://doi.org/10.1016/j.jcp.2015.08.036.
URL https://www.sciencedirect.com/science/article/pii/S002199911500563X
[36] C. Min, F. Gibou, Robust second-order accurate discretizations of the multi-dimensional Heaviside and Dirac delta functions, Journal of Computational Physics 227 (22) (2008) 9686–9695. doi:10.1016/j.jcp.2008.07.021.
URL https://www.sciencedirect.com/science/article/pii/S0021999108003999
[37] I. M. Jánosi, D. Jan, K. G. Szabó, T. Tél, Turbulent drag reduction in dam-break flows, Experiments in Fluids 37 (2) (2004) 219–229. doi:10.1007/s00348-004-0804-4.
URL https://doi.org/10.1007/s00348-004-0804-4