



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

► To cite this version:

Félix Henri, Mathieu Coquerelle, Pierre Lubin. Geometrical level set reinitialization using closest points method and kink detection for thin filaments, topology changes and two-phase flows. 2021. hal-03159919v1

HAL Id: hal-03159919

<https://hal.science/hal-03159919v1>

Preprint submitted on 4 Mar 2021 (v1), last revised 18 Oct 2021 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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*

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 even 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 during 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 connections / disconnections of the surface. Nevertheless, the transport of 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 methods [3] are based on an iterative process for finding values of ϕ starting from the interface position, they however suffer of 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]. Even though, 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 principally 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 containing the surface and solved the Hamilton-Jacobi equation farther. 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 that 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 dynamic. 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.

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

2.1. Level set definition

Let us 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 $\phi : \Omega \rightarrow \mathbf{R}$ 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 :

$$|\nabla \phi| = 1. \quad (1)$$

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:

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

where \mathbf{u} is the fluid velocity, ρ its density, μ its dynamic viscosity, p the pressure, $\mathbf{D} = (\nabla \mathbf{u} + \nabla^T \mathbf{u})/2$ 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:

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

In the case of an immiscible two-phase flow simulation where a discontinuity in density and viscosity appears 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:

$$\begin{aligned} \rho(\mathbf{x}) &= \rho_2 + (\rho_1 - \rho_2)c(\mathbf{x}) \\ \mu(\mathbf{x}) &= \mu_2 + (\mu_1 - \mu_2)c(\mathbf{x}) \end{aligned} \quad (3)$$

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, also referred as the volume fraction.

Level set coupling. In a level set framework, c is expressed as a function of ϕ . Where, in order to obtain a smooth representation of the interfacial region, a regularized form of the Heaviside function is usually used:

$$c(\mathbf{x}) = H_\epsilon(\phi(\mathbf{x})). \quad (4)$$

with

$$H_\epsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \frac{1}{2}(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi} \sin(\pi \frac{\phi}{\epsilon})) & \text{if } |\phi| \leq \epsilon \\ 1 & \text{if } \phi > \epsilon \end{cases} \quad (5)$$

Hence, the density and viscosity vary smoothly within a interfacial region of thickness 2ϵ , where usually ϵ is proportional to the cell size h , i.e. $\epsilon = O(h)$.

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 that 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}_Γ its normal and δ_Γ is the Dirac function associated to the surface. Brackbill et al. [13] introduced the Continuum Surface Force (CSF) which approximates this body force as:

$$\sigma \kappa_\Gamma \mathbf{n}_\Gamma \delta_\Gamma \simeq \sigma \kappa \nabla c. \quad (6)$$

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. Hence the 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 the curvature and so reduce spurious currents [14–16]. Hence, 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:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \quad (7)$$

Yet, in a general case, resolving the advection equation will induce deviation of ϕ to be a signed distance function and consequently will cease to be a solution of the eikonal equation 1. The reasons are twofold. First, distortions come from numerical errors when resolving the advection Eq. (7). 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 (see Sec. 2.2) 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 to 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 a lot of research and a variety of strategies has 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 :

$$\frac{\partial \psi}{\partial \tau} + \text{sgn}(\psi_0)(\|\nabla \psi\| - 1) = 0 \quad (8)$$

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. (8) which derives from the Hamilton-Jacobi equation. Indeed, the solution is obtained when $\tau \rightarrow \infty$. Until the end, Eq. (8) will be referred to as H-J equation.

Even if solving the H-J equation is expensive comparatively 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.

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 want 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 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. [19] did a comparative study on some of these parameters and showed how it may impact 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, 20] 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 [20] 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 $\|\overrightarrow{\mathbf{x}\mathbf{y}}\|$:

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

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. (11).

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

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

where $\mathbf{t}_{\mathbf{y}}$ 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 $\overrightarrow{\mathbf{x}\mathbf{y}}$. 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 a Newton method that searches 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:

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

and the sign function sgn as:

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

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

$$\psi(\mathbf{x}) = \text{sgn}(\mathbf{x}) \cdot \psi_{dist}(\mathbf{x}) = \begin{cases} -\|\overrightarrow{\mathbf{x}CP(\mathbf{x})}\| & \text{if } \mathbf{x} \in \Omega^- \\ 0 & \text{if } \mathbf{x} \in \Gamma \\ +\|\overrightarrow{\mathbf{x}CP(\mathbf{x})}\| & \text{if } \mathbf{x} \in \Omega^+. \end{cases} \quad (9)$$

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

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

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 the 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 gradient descent algorithm can be safely exploited. 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 farther.

Problems arise where, 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 the 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 the test cases were executed with more than one processor. We will not detail further these two particular implementation details in this article.

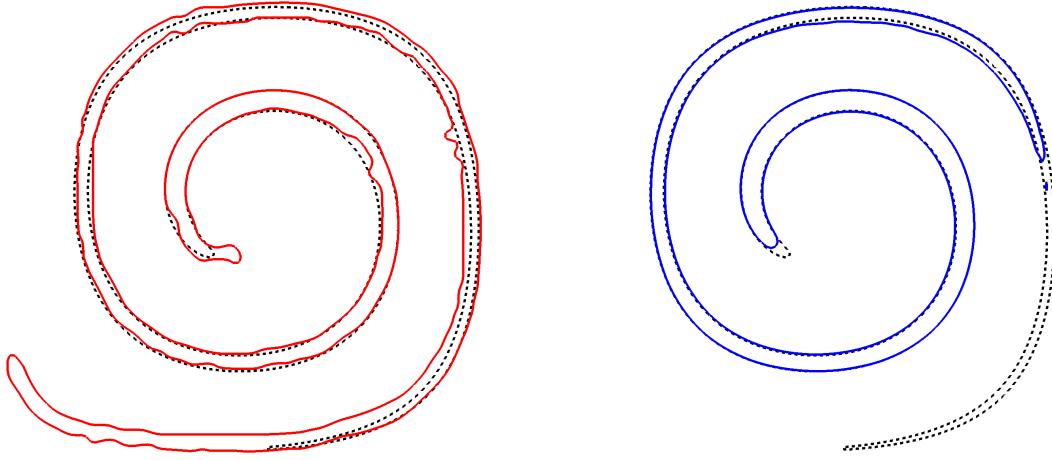


Figure 1: Illustration of the RCP algorithm result on the single vortex 2D case, without kinks treatment (left) and with kinks treatment (right). The reference solution is drawn with black dashed lines. The absence of treatment of kinks leads to heavy distortion of the interface and exhibits anti-diffusive 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, representing 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 to 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.

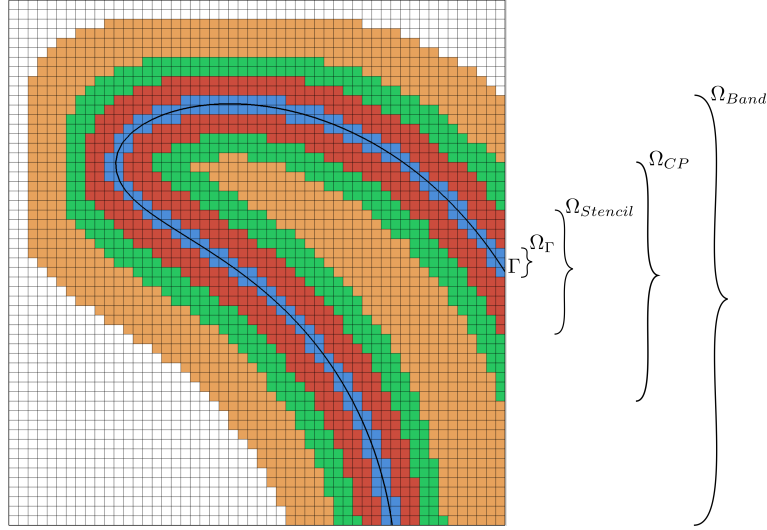


Figure 2: Illustration of the four nested sets of cells. In black the underlying interface, blue Ω_Γ , red $\Omega_{Stencil}$, green Ω_{CP} and in orange Ω_{Band} .

5.2. Proposed algorithm

The RCP procedure is applied after the advection of ϕ (see Eq. (7)) and preceded by the construction of the sets described above. Algorithm 1 outlines the prime steps of the proposed method. First, a pre-process is necessary in order to detect all cells (i.e. build Ω_{Kink} and $\Omega_{PseudoLS}$) that will require a particular treatment. Then, a regularization through a pseudo level set is applied (1) for all ill-defined cells that are sufficiently far from the surface (cells part of $\Omega_{PseudoLS}$). Thereafter, the closest points is computed (2a) 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 (2b) thanks to Eq. (10). Finally, a low-order HJ reinitialization procedure is applied (3) to all remaining cells. We present in next sections the successive steps in more details, particularly the treatment of kink cells.

Algorithm 1 Outline of the RCP algorithm

1. Detection of kinks: build Ω_{Kink} (see algorithm 2);
 2. $\forall \text{cell} \in \Omega_{PseudoLS}$ (see Sec. 5.5.1);
 - (a) Compute a pseudo distance function;
 - (b) Apply a low-order HJ reinitialization;
 3. $\forall \text{cell} \in \Omega_{CP} \setminus \{c_l \in \Omega_\Gamma \mid (\Omega_{Kink} \cap S_{c_l}) \neq \emptyset\}$:
 - (a) Compute the closest point with gradient descent (see Sec. 5.3);
 - (b) Update the level set value with Eq. (10) (see Sec. 5.3 and Sec. 5.5.2);
 4. $\forall \text{cell} \in \Omega \setminus \Omega_{CP}$: apply a low-order HJ reinitialization (see Sec. 5.6).
-

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 and for all the following sections, the algorithm used is the one described by Coquerelle et al. [16] where a fourth-order convergence accuracy is achieved thanks to high order interpolation schemes and an orthogonality criterion. As presented by [20], 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.

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]. Concretely, 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 9 can be evaluated from a discrete point of view through the equation:

$$\psi(\mathbf{x}) = \text{sgn}(\phi(\mathbf{x})) \overrightarrow{\|\mathbf{xCP}(\mathbf{x})\|}. \quad (10)$$

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, 21], 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 interpolation. Consequently, the closest point accuracy will be impacted, as also noted by [10, 20].

Numerical kinks. The second type of kinks, referred 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 figure 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 case, a local minimum emerges in the level set field that can perturb the interpolations.

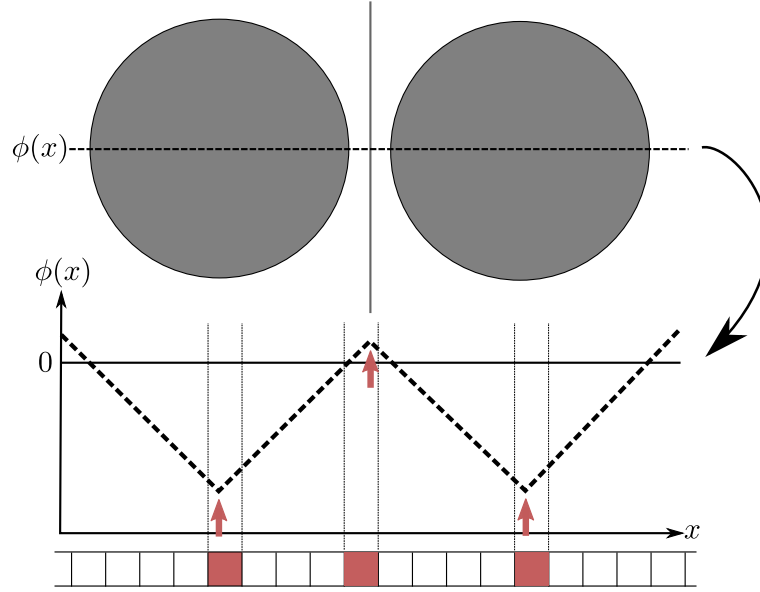


Figure 3: Illustration of the inherent kinks. On the top, a 2D visualization of two drops, the medial axis is drawn in red. The middle plot represents the level set function over a 1D cut, the associated kinks are highlighted by the red arrows. The bottom part is the associated discrete 1D mesh, the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

- 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 make 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 dreadfully, local minima can act as well potentials, where the gradient descent algorithm will plunge and stop without reaching the surface. In consequence, to fulfil accuracy and robustness, it is crucial to detect carefully those points.

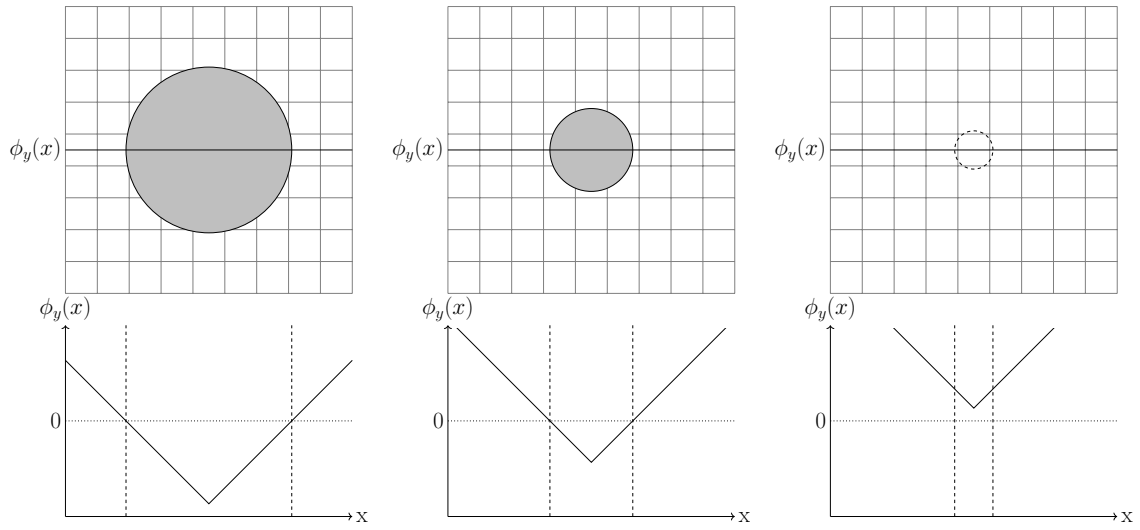


Figure 4: Creation of a numerical kink when a small structure, such as a bubble, vanishes due to numerical diffusion when solving the advection equation Eq. (7). On the top a 2D visualization of the drop at different times and on the bottom a 1D cut of the associated level set.

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}_{ξ_1} and \mathbf{x}_{ξ_2} at a very small distance of \mathbf{x} in directions $\vec{\xi}_1$ and $\vec{\xi}_2$, 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:

$$\mathbf{x} \text{ is a kink if } \exists(\mathbf{x}_{\xi_i}, \mathbf{x}_{\xi_j}), \|\text{CP}(\mathbf{x}_{\xi_i}) - \text{CP}(\mathbf{x}_{\xi_j})\| > \epsilon. \quad (11)$$

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 cell size, i.e. of the maximum radius of curvature of the surface at the discrete level.

First order closest point approximation. Since there is no analytical method for computing the closest points and we cannot enumerate all points \mathbf{x}_{ξ} 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:

$$\text{CP}(\mathbf{x}) \simeq \mathbf{x} - \mathbf{n}(\mathbf{x}) d(\mathbf{x}) \quad (12)$$

where $\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}$ is the normal and d the Euclidian distance to the surface. Again, within a first order approximation, by using $d \simeq \frac{\phi}{|\nabla \phi|}$, Eq. (12) can be rewritten as:

$$\text{CP}(\mathbf{x}) \simeq \mathbf{x} - \frac{\phi(\mathbf{x})}{|\nabla \phi(\mathbf{x})|^2} \nabla \phi(\mathbf{x}). \quad (13)$$

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

For efficiency reasons, we restrict the number of surrounding points \mathbf{x}_{ξ} to an acceptable number. Furthermore, instead of using interpolation to evaluate the \mathbf{x}_{ξ} in several directions, we simply approximate Eq. (13) 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 gradient $\nabla \phi$ will vary significantly depending of 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 gradient. For the sake of simplicity, we have only showed 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 $dl \ll h$, at the *south-west* of \mathbf{x} , where \mathbf{x} is 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:

$$\nabla \phi(\mathbf{x}_{\xi_{sw}}) \simeq \left(\frac{\phi_{i,j} - \phi_{i-1,j}}{\delta x}, \frac{\phi_{i,j} - \phi_{i,j-1}}{\delta y} \right) \quad (14)$$

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. In that case, the closest

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

$$\text{CP}(\mathbf{x}_{\xi_{SW}}) \simeq \mathbf{x}_{i,j} - \frac{\phi_{i,j}}{|\nabla\phi(\mathbf{x}_{\xi_{SW}})|^2} \nabla\phi(\mathbf{x}_{\xi_{SW}}). \quad (15)$$

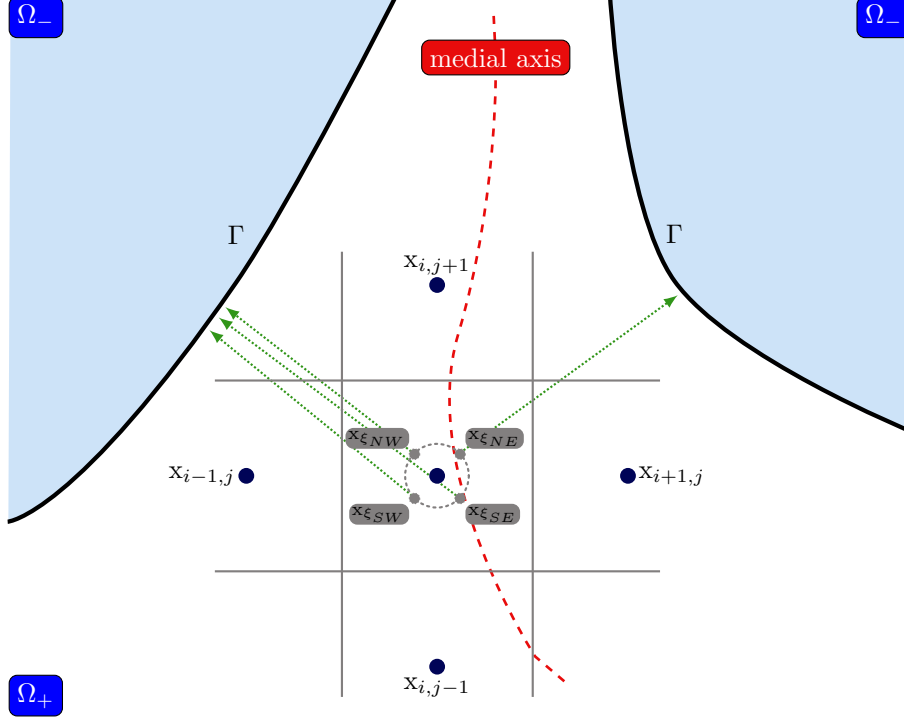


Figure 5: Illustration of the kink detection algorithm on a 2D mesh in the vicinity of a medial axis. Three of the four closest point approximations (SW, NW and SE) surrounding $\mathbf{x}_{i,j}$ point towards a narrow region on the left most surface whilst the last one (NE) is located on the right most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

Kink detection algorithm. Based on these biased schemes, we can now compute the associated closest point approximations in the four directions: $\text{CP}(\mathbf{x}_{\xi_{SW}})$, $\text{CP}(\mathbf{x}_{\xi_{SE}})$, $\text{CP}(\mathbf{x}_{\xi_{NW}})$ and $\text{CP}(\mathbf{x}_{\xi_{NE}})$, as illustrated on Fig. 5. Finally, we evaluate the maximum relative Euclidean distance to detect if a pair is sufficiently distant to consider the region to be near a kink. We have summarized the method in algorithm 2. It is worth to note that this algorithm can be optimized by stopping it whenever the maximum Euclidean distance criterion has been exceeded and thus avoiding the computation of unnecessary closest points. An illustration of the resulting detected kink cells on the single vortex 2D test case (see Sec. 6.5) is shown in Fig. 6 and on the Zalesak disk test case (see Sec. 6.4) in Fig. 7.

Detector threshold. The threshold ϵ used in algorithm 2 is a key point in the kink detection process. Ideally, it has to be infinitely small. However, due to numerical approximations, even in a smooth region, the distance between closest points computed with biased approximations of Eq. (13) can be of the order of h , where h is the cell width. This is particularly true for points far from the interface where $d \simeq \phi/|\nabla\phi|$, i.e. the approximated distance to the surface is large, or for points near a surface of high curvature where the normal vector varies fast. In consequence, we set $\epsilon = h/2$. 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, in regions of high curvature, i.e. $\kappa \sim O(h^{-1})$, more kinks will indeed be detected. This is not a problem as those sensitive regions are usually under resolved and the

Algorithm 2 Kink detection algorithm in 2D.

1. For all directions $d \in \{SW, SE, NW, NE\}$:
 - (a) Compute the biased closest point approximation CP_d using eq. 13 with variations of formula 14
 2. Initialize the maximum Euclidean distance: $MED \leftarrow 0$
 3. For all directions $d_1 \in \{SW, SE, NW, NE\}$:
 - (a) For all directions $d_2 \in \{SW, SE, NW, NE\}$ with $d_2 \neq d_1$:
 - i. $MED \leftarrow \max(MED, \|CP_{d_1} - CP_{d_2}\|)$
 4. If $MED > \epsilon$ return true; else return false
-

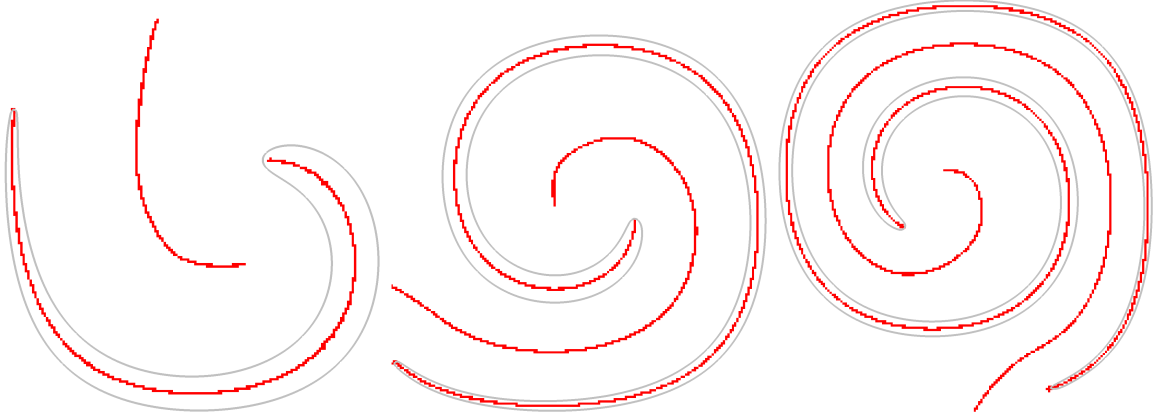


Figure 6: Illustrations of the kink detection algorithm 2 on the 2D single vortex test case (see Sec. 6.5) at different instants, for a grid resolution of 256^2 cells. The surface is drawn in gray and the detected kink cells are marked in red. Over time, the kink detector successfully captures all inherent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

use of high order interpolation with a large stencil would anyway suffer from including kink cells.

Discussion on another approach: the quality function $Q(\mathbf{x})$. To detect kinks, another approach was introduced by Macklin and Lowengrub in [21] wherein the authors defined a *normal quality function* $Q(\nabla\phi(\mathbf{x}))$ as:

$$Q(\nabla\phi(\mathbf{x})) = |1 - |\nabla\phi||, \quad (16)$$

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, those areas may neither be inherent nor numerical kinks.

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 that the sensitivity of our detector, i.e. the choice of ϵ value, will straightly depend on the maximum acceptable threshold to detect under-resolved structures. Herein, we have set $\epsilon = h/2$: thus, we consider that a structure which has 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 will be discussed in Sec. 5.5.2.

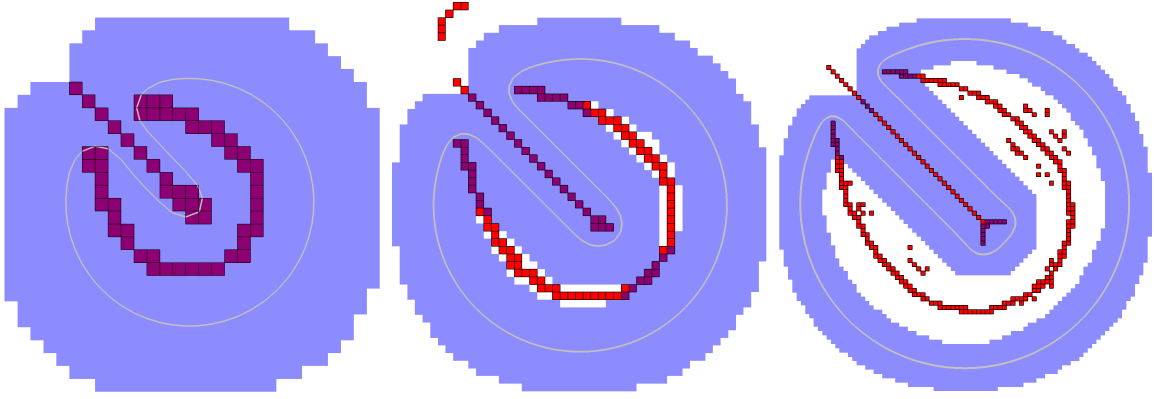


Figure 7: Illustration of the kink detection algorithm 2 on the Zalesak test case (see Sec. 6.4), for a grid resolution of 64^2 (left), 128^2 (center) and 256^2 (right) cells, after $5/8$ of rotation. The surface is drawn in gray, the detected kink cells are marked in red and Ω_{CP} in blue. As the cell size diminishes, the kink detector successfully captures all the cells part of the medial axis and hence containing an inherent kink. Moreover, it can be seen that for the grid resolution of 128^2 (center) and 256^2 (left), few other numerical kinks are detected. As represented in blue, those points are not part of Ω_{CP} and consequently, they are not reinitialized with the RCP method but are solely smoothed (see Sec. 5.6) and do not disturb the gradient descent.

5.5. Kinks treatment

5.5.1. Cells far from the interface

Kinks can deteriorate the level set regularity and can lead the 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 of $\Omega_{Stencil}$, this treatment is only applied for ill-defined cells sufficiently far from Γ , i.e. for the subset $\Omega_{PseudoLS}$ defined as:

$$\Omega_{PseudoLS} = (\Omega_{Band} \cap \Omega_{Kink+}) \setminus \Omega_{Stencil} \quad (17)$$

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 a sufficiently 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. This one 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:

$$\psi_{c_l} = nh \times \text{sign}(\phi_{c_l}) \quad (18)$$

Fast low-order smoothing. The second step consists in smoothing the pseudo level set with a fast reinitialization algorithm that will guaranty a smooth solution regarding the concerned cells neighbours. Though several strategies may be considered, we have found that the use of a low-order solving of the HJ equation 8 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 2, 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 guaranty to give 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.

Some authors tried to resolve the issue of calculating accurate interpolations/derivatives near a kink [11, 21]. 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 numerical stability, we have applied the same fast low order smoothing procedure as a post-process of the algorithm (see step 3 of algorithm 1) in cells part of $\Omega \setminus \Omega_{CP}$. Concretely, 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 two other approaches to reinitialize the level set. The first is the H-J equation Eq. (8) 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 and 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 [22], 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 [23]. 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. (3) to (5), where the regularization parameters is set to $\epsilon = 2h$.

6.1.2. Level set

In order to solve precisely the advection equation 7 for the level set, second order Runge-Kutta NSSP 3,2 integration coupled with a fifth order finite difference WENO scheme [24, 25], are used. When it's 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 [19] 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.

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:

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

where $\phi_{ex}(\mathbf{x}_l)$ is the expected value of the signed distance field at \mathbf{x}_l and $\tilde{\phi}(\mathbf{x}_l)$ 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. (4) and (5)) associated to the cells of volume V_{CV_i} as: $V = \sum_i^{N_\Omega} c_i V_{CV_i}$.

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. (16) as:

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

Yet, while it 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 $\tilde{\phi}$ a field which deviates from the signed distance function ϕ likes $\tilde{\phi}(\mathbf{x}) = \alpha(\mathbf{x})\phi(\mathbf{x})$, with $\alpha(\mathbf{x}) : \Omega \rightarrow \mathbf{R}$, 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. :

$$e_{\nabla\phi}(\alpha\nabla\phi) = e_{\nabla\phi}(\alpha^{-1}\nabla\phi) \quad (19)$$

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

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

which satisfies Eq. (19). Following this definition, we propose the L_2 norm of the global variation of $\tilde{\phi}$ to the signed distance function:

$$E_{\nabla\phi}^{L_2} = \sqrt{\frac{1}{N_\Gamma} \sum_i^{N_\Gamma} e_{\nabla\phi}(\nabla\tilde{\phi})^2}.$$

The L_∞ norm of the error, noted $E_{\nabla\phi}^{L_\infty}$, 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 :

$$(u, v) = \frac{2\pi}{T}(0.5 - y, x - 0.5) \quad (20)$$

where a full rotation of the circles is done at a time $t = T$. Here T is chosen to be equal to 500.

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 disappear 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 ton 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.

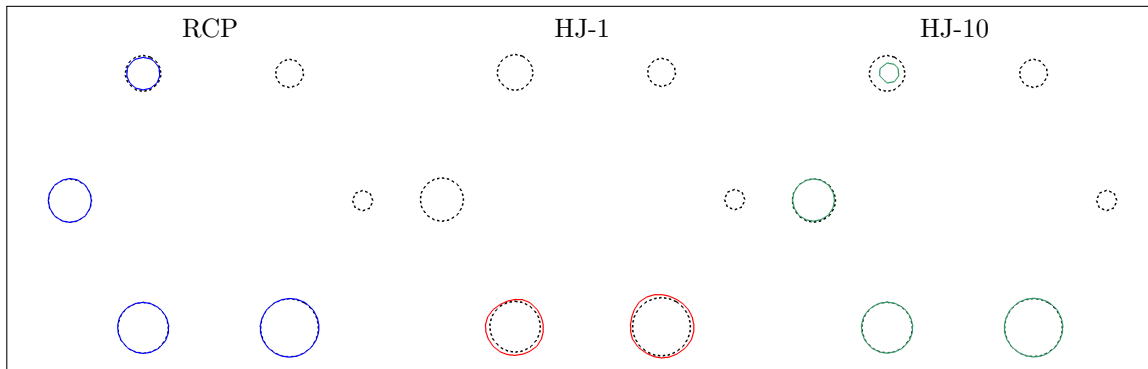


Figure 8: Interface shapes after a full rotation for the disk's rotation test case for a grid resolution of 64^2 cells, the reference solution is plotted in gray dashed lines.

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 $N = 64$. Time step is fixed set to $\Delta t = 0.625$ for coarsest mesh of 64^2 cells. For the other meshes, the time step is reduced in order to keep a constant CFL number.

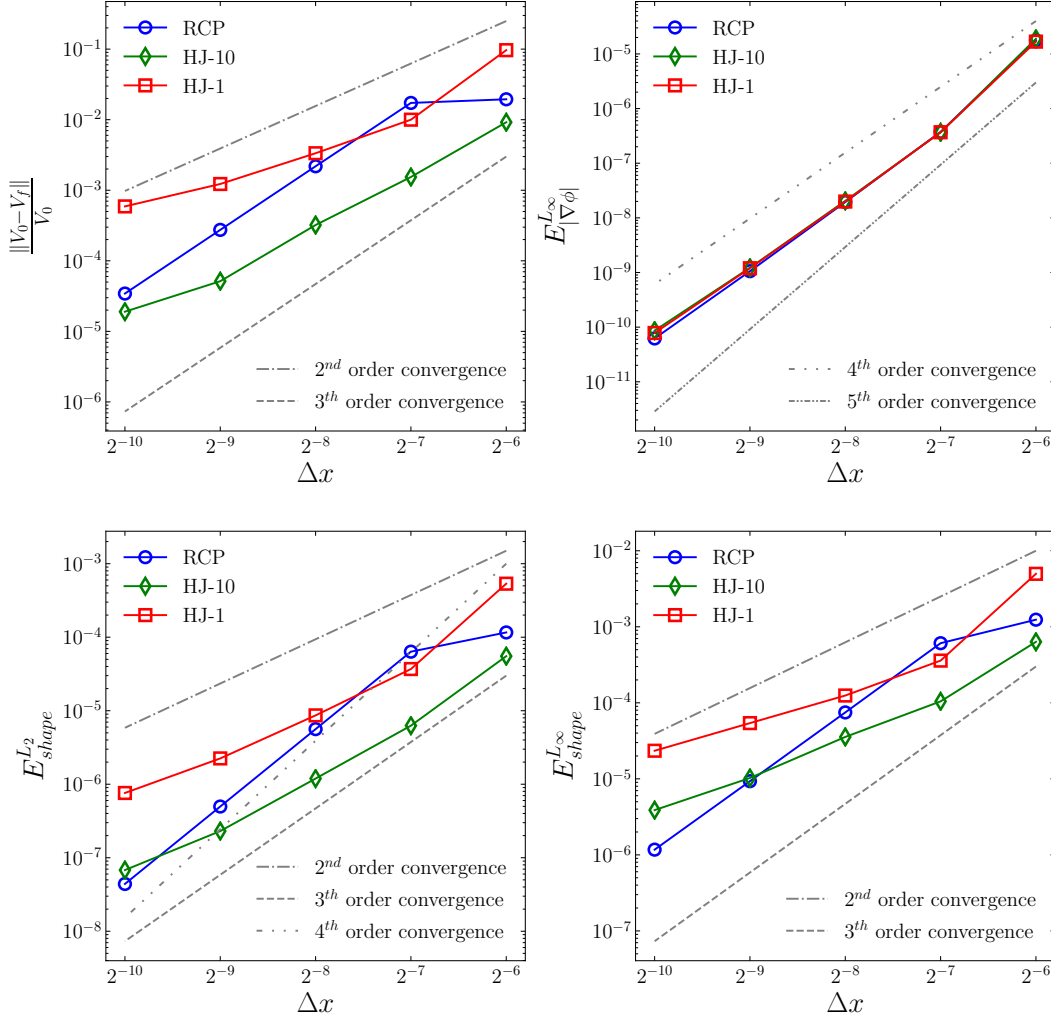


Figure 9: From left to right and top to bottom. Convergence rate, for the one circle rotation test case, of the volume conservation, L_∞ norm for the error on the deviation to be a signed distance function ($E_{\nabla\phi}^{L_\infty}$) and the L_∞ and L_2 norm on for the shape errors ($E_{shape}^{L_\infty}$ and $E_{shape}^{L_2}$)

As illustrated on Fig. 9, all methods exhibit similar results for the $E_{\nabla\phi}^{L_\infty}$. 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

This test case follows the one proposed in [26] 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 initialized centred at the grid

point $(0.5, 0.75)$ with a radius of 0.15 and a slot of 0.5 of width and 0.20 of length. The velocity field, as described above, is set to carry the slotted disk in a counterclockwise rotation around the point $(0.5, 0.5)$ and to perform of full rotation after $T=500$ units of time using Eq. (20).

The time step is fixed, set to $\Delta t = 1.25$ for the coarsest mesh of 32^2 cells, and is reduced in order to keep a constant CFL number for finer meshes. The simulation is stopped after one full rotation of the slotted disk. In this test case, as in the previous ones, the numerical errors principally originate from the reinitialization.

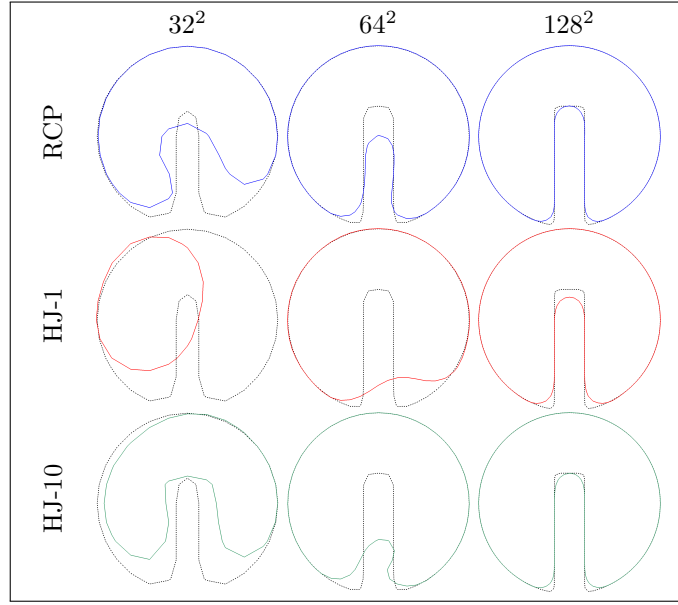


Figure 10: Interfaces shape after one rotation for the Zalesak disk case. The dashed line is the initial interface.

For the coarsest mesh, as illustrated in Fig. 10, we observe that both RCP and HJ-10 manage to conserve the slot and a shape that is globally acceptable regarding the mesh resolution. On the other side, the frequent reinitialization of HJ-1 introduces more smoothing/errors that lead(s) to a round shape that hardly captures the expected solution. For the intermediate resolution, all methods succeed in preserving the round part of the shape. Nevertheless, only the RCP method achieves to conserve a deep slot with relatively sharp corners compared to HJ-1 and HJ-10. For the finest resolution, all methods accurately preserve the global shape, while HJ-10 and RCP seem to converge better towards the capture of sharp corners. This test case confirms that care has to be taken on the reinitialization frequency using the H-J method. The RCP method, even when applied at every time step, avoid undesired smoothing of the surface.

For analysing the 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.

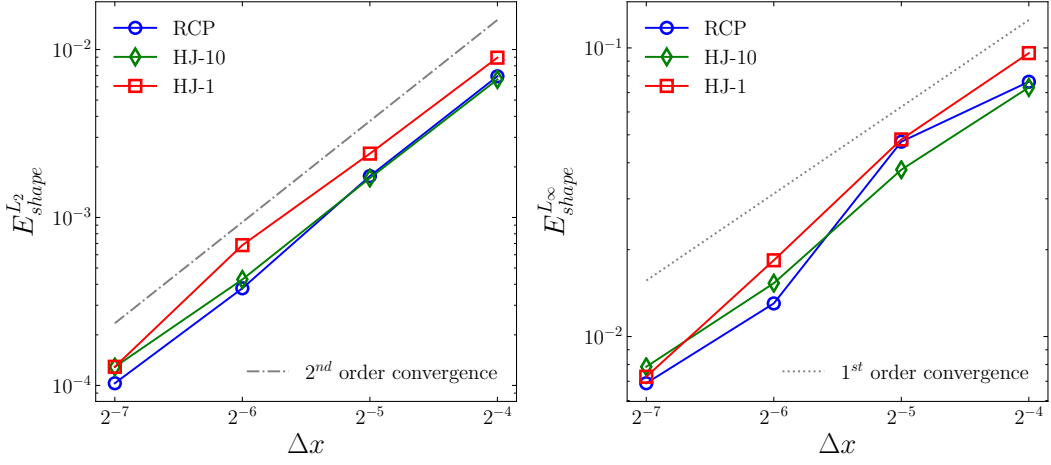


Figure 11: From left to right . Convergence rate for the Zalesak disk case of the L_∞ and L_2 norm for the shape errors ($E_{shape}^{L_\infty}$ and $E_{shape}^{L_2}$)

6.5. Single vortex 2D

Following the widely studied test case [27–29] 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:

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

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 $\Delta t = 1.6e-3$ for the coarsest grid 128^2 . For the other meshes, the time step is reduced in order to keep a constant CFL number.

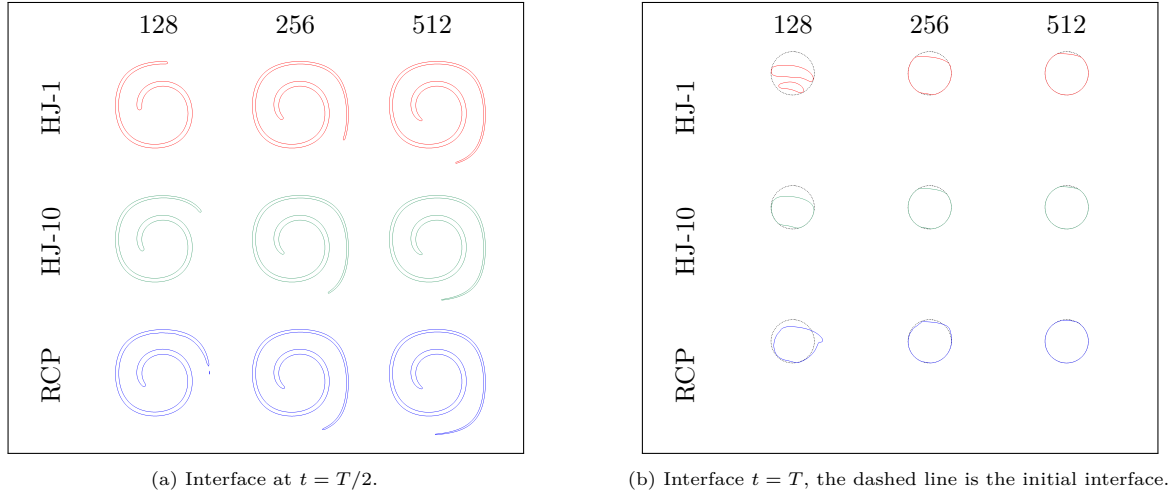


Figure 12: Interface shapes at $t = T/2$ (left) and $t = T$ (right) for the single vortex 2D case.

Fig. 12 illustrates the aptitude of the advection equation coupled with the three reinitialization procedures to conserve thin filaments at maximal deformation (Fig. 12a) and to correctly reverse to the initial circular

shape (Fig. 12b). 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, the 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 the annihilation of thin filament structures. Secondly, because the trailing filament thickness falls below the grid resolution and thus cannot be captured by standard advection methods.

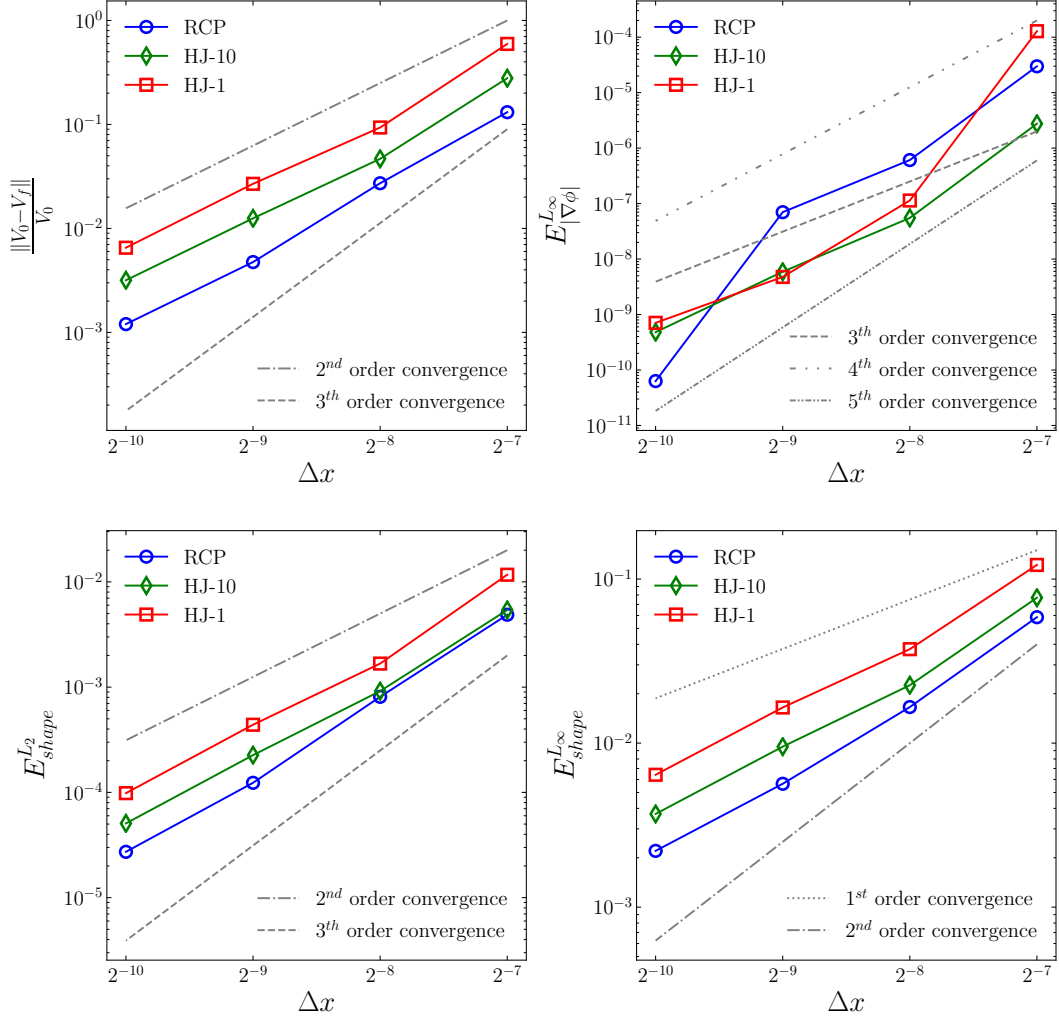


Figure 13: From left to right and top to bottom. Convergence rate of the single vortex 2D test cases, at $t = T$, for the volume conservation, L_∞ norm for the error on the deviation to be a signed distance function ($E_{\nabla\phi}^{L_\infty}$) and the L_∞ and L_2 norm on for the shape errors ($E_{shape}^{L_\infty}$ and $E_{shape}^{L_2}$)

All the 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. 13, all the tree methods produce comparable results. For both the enclosed volume and the L_2 norm for the shape error, all the tree methods exhibit 2nd order convergence rate and between 1st and 2nd 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 5th when the HJ-1 and HJ-10 are 3th order.

6.6. Single vortex 3D

A 3D adaptation of the test case was proposed by LeVeque [28] and applied by Enright et al. [29] 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:

$$\begin{aligned} u &= 2 \sin^2(\pi x) \sin(\pi y) \sin(\pi z) \cos\left(\frac{\pi t}{T}\right) \\ v &= 2 \sin(\pi x) \sin^2(\pi y) \sin(\pi z) \cos\left(\frac{\pi t}{T}\right) \\ z &= 2 \sin(\pi x) \sin(\pi y) \sin^2(\pi z) \cos\left(\frac{\pi t}{T}\right). \end{aligned}$$

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 $T = 3$.

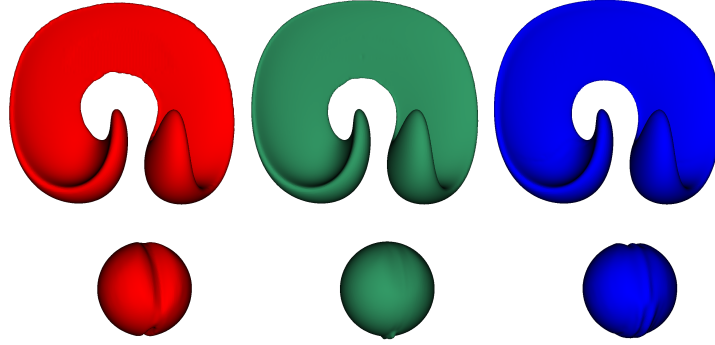


Figure 14: Interface shape of the single vortex 3D for a grid resolution of 256^2 cells, at $t = T/2$, using the reinitialization method in red: HJ-1, green: HJ-10, blue: RCP

Fig. 14 and table 1 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 resulting with errors on the final shape.

Method	$\left \frac{V_0 - V_{final}}{V_0} \right $		$E_{ \nabla \phi }^{L_2}$		$E_{shape}^{L_2}$		$E_{shape}^{L_\infty}$	
	128^2	256^2	128^2	256^2	128^2	256^2	128^2	256^2
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 1: Numerical results of the single vortex 3D case on the enclosed volume error, the L_2 norm on the deviation to be a signed distance function ($E_{|\nabla \phi|}^{L_2}$), the L_2 norm on the shape error ($E_{shape}^{L_2}$) and L_∞ norm on the shape error ($E_{shape}^{L_\infty}$). The results are given for the RCP, HJ-1 and HJ-10 methods, for a grid resolution of 128^2 and 256^2 cells, at a time $t = T$.

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 discretization errors 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 Eq. (6), 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 simulation 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 $Ca_{max}^* = \mu |\mathbf{u}|_{max} / \sigma$ is compared for various Laplace numbers for various meshes. In practice, we study the scaled capillary number $Ca_{max} = Ca_{max}^* / U_\sigma$ where the characteristic velocity U_σ is defined as $U_\sigma = \sqrt{\sigma / (\rho D)}$. Also, time is adimensionalized 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:

$$\Delta t_\sigma^{static} \leq \sqrt{\frac{(\rho_1 + \rho_2) h^3}{2\pi\sigma}}. \quad (21)$$

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. (21) for the finest mesh.

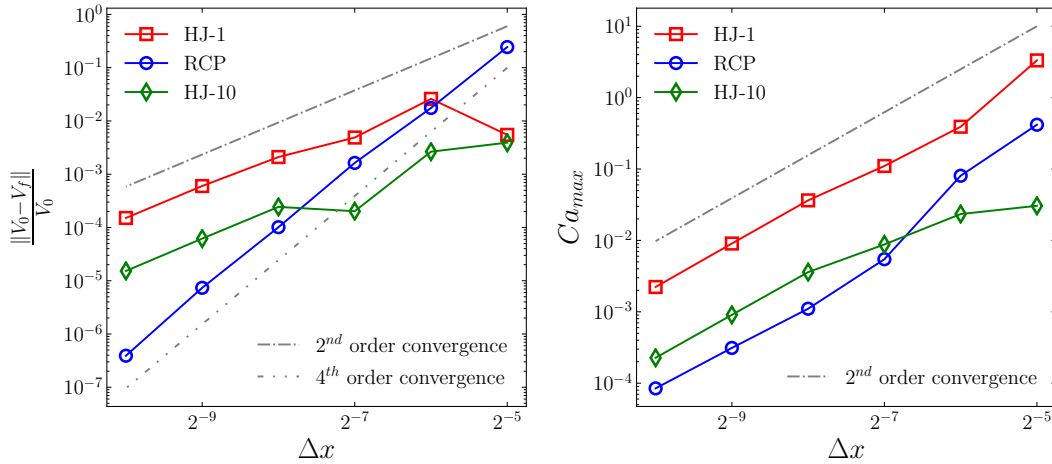


Figure 15: Convergence study of the 2D column at equilibrium for the enclosed volume variation (left) and Ca_{max} (right) at $La = 120$ for HJ-1, HJ-10 and RCP when a steady state has been reached.

The convergence rates of the error on the enclosed volume and Capillary number are shown in Fig. 15. 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 $La = \{120, 1200, 12000, 120000\}$.

Method	Ca_{max}			
	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 2: 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 .

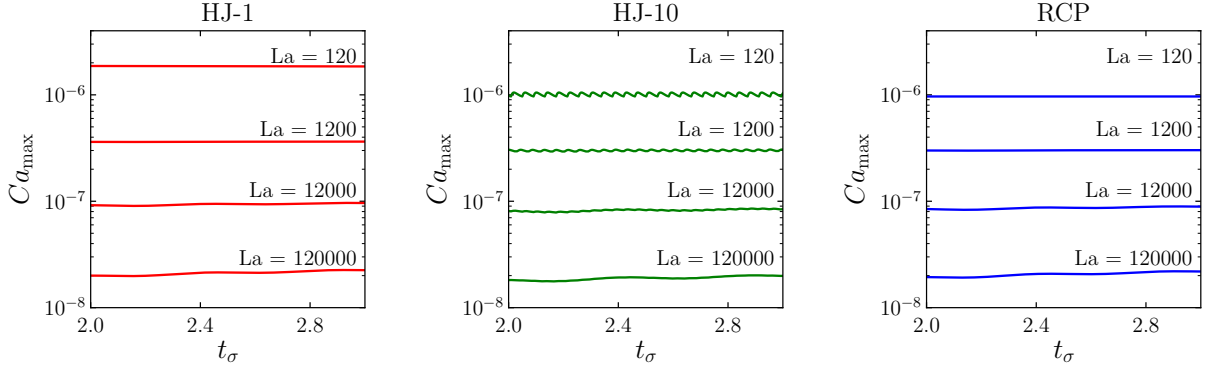


Figure 16: Ca_{max} when varying the Laplace number for HJ-1 (left), HJ-10 (center) and RCP (right) method in the interval $t_\sigma \in [2.0, 3.0]$ for 64^2 cells.

Fig. 16 and table 2 show very similar results for all three approaches. It should be noted, as clearly visible on the central graph of Fig. 16 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 continuous results as shown in the RCP and HJ-1 respective figures.

6.8. Impact of a droplet

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 will test 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 a 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 [30] 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, as detailed in ???. 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. 17, 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. 18 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. 19, 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 therein relatively smaller. Moreover, the less frequent the HJ equation is applied, the *easier* the two surfaces can merge. As discussed in previous sections, this questions strongly 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 permits to overcome these issues thanks to the accurate kink detection algorithm coupled to the adapted geometrical reinitialization procedure.

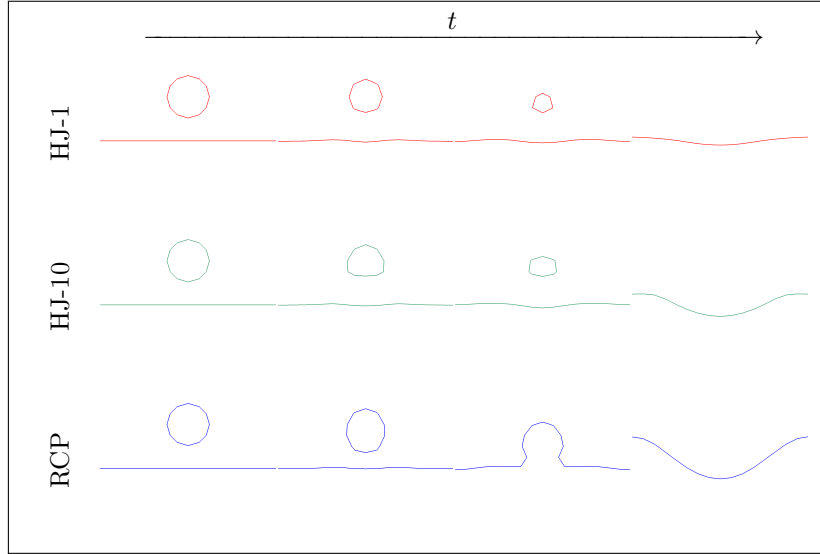


Figure 17: Interface shape of the impacting drop case at time $t = \{0, 2.4e-3, 4.8e-3, 9.6e-3\}$ for the coarse mesh with $N_c = 4$.

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, relatively to the mesh refinement. The RCP brings a robust method to correctly reinitialize the level set without compromising topological changes.

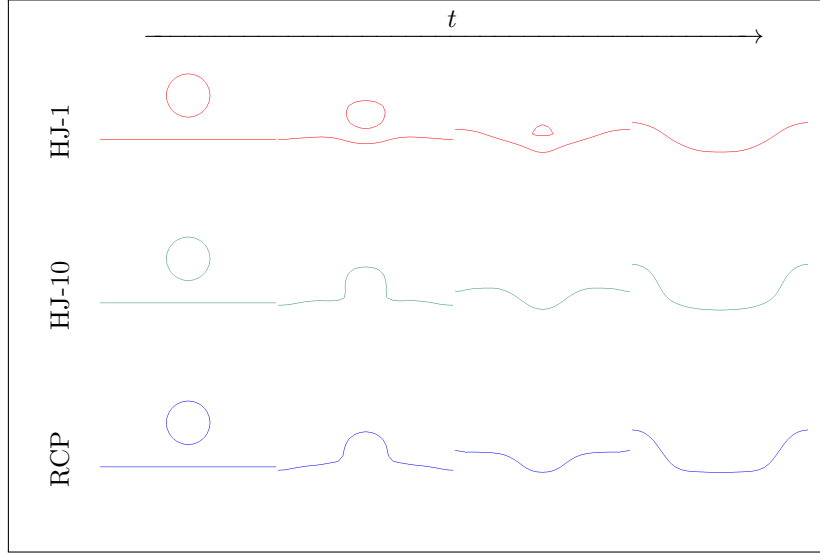


Figure 18: Interface shape of the impacting drop case at time $t = \{0, 6.8e-3, 1.0e-2, 1.4e-2\}$ for the medial mesh with $N_c = 8$.

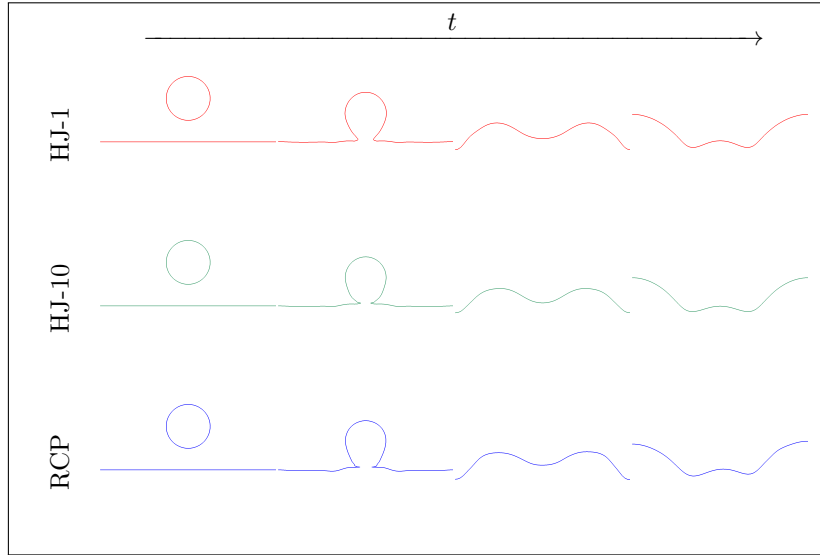


Figure 19: Interface shape of the impacting drop case at time $t = \{0, 7.2e-3, 1.1e-2, 1.6e-2\}$ for the fine mesh with $N_c = 16$.

7. 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 to obtain 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 such as the impacting droplet on a liquid pool, capturing various topology changes at different scales.

Appendix A. Level set kinks detection and approximation

Equation 13 gives a first order approximation of the closest point at any point \mathbf{x} . In order to detect for kinks, we need to evaluate that function for points in the vicinity of \mathbf{x} . For that purpose, we first consider Eq. (13) evaluated at a point \mathbf{x}_ξ :

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

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

$$\text{CP}(\mathbf{x}_\xi) \simeq \mathbf{x} - \frac{\phi(\mathbf{x})}{|\tilde{\nabla}\phi(\mathbf{x}, \vec{\xi})|^2} \tilde{\nabla}\phi(\mathbf{x}, \vec{\xi})$$

where $\tilde{\nabla}\phi(\mathbf{x}, \vec{\xi})$ is a discrete off-centered approximation of $\nabla\phi(\mathbf{x}_\xi)$ in the bias direction $\vec{\xi}$, not to be confound with $\nabla\phi \cdot \vec{\xi}$, 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: $\vec{\xi}_{\alpha_x, \alpha_y, \alpha_z} = (\alpha_x, \alpha_y, \alpha_z)^t$, 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 $\vec{\xi}_{\alpha_x, \alpha_y, \alpha_z}$ direction for the gradient computation:

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

which can be used similarly as Eq. (15) in algorithm 2 for computing the associated closest point approximations and hence detecting kinks.

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/S0021999100965379>
- [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] Z. Solomenko, P. D. M. Spelt, L. Ó Náirigh, 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>
- [20] 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>
- [21] 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>
- [22] Notus – Computational Fluid Dynamics.
 URL <https://notus-cfd.org/>
- [23] 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>
- [24] 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>
- [25] 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>
- [26] 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>
- [27] 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>
- [28] 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>
- [29] 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>
- [30] 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_ttvoft_2_0_co_2.xml
-